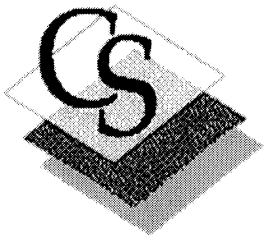
CrimeStat® *III*

VERSION 3.0

A Spatial Statistics Program for the Analysis of Crime Incident Locations



Ned Levine & Associates
Houston, TX

The National Institute of Justice
Washington, DC
November 2004

Table of Contents

Table of Contents (continued)

Table of Contents (continued)

Table of Contents (continued)

Table of Contents (continued)

Table of Contents (continued)

Acknowledgments

CrimeStat III was developed under the direction of Dr. Ned Levine of *Ned Levine & Associates*, Houston, TX, from Grant No. 2002-IJ-CX-0007, awarded by the National Institute of Justice (NIJ), Office of Justice Programs, US Department of Justice. The developer would like to thank the many individuals who contributed to this program:

1. Mr. Long Doan of Doan Consulting, Falls Church, VA who was the original programmer for the project. Mr. Doan's brilliance in programming was essential to the development of the initial program. For this version, he had the role of ensuring system integration.

2. Ms. Haiyan Teng of Houston, TX, who was the primary programmer for version 3.0. Her high level of programming competence and mathematical expertise was essential for the successful completion of the crime travel demand routines.

3. Professor Richard Block of Loyola University and Mr. Daniel Helms of the Law Enforcement Corrections and Technology Center at the University of Denver who served as criminal justice advisors to the project. They played critical roles in testing the crime travel demand routines with data from Chicago and Las Vegas. They are co-authors of one chapter.

4. Professor Luc Anselin of the University of Illinois at Urbana-Champaign who provided technical advice and documentation on the regression models used in the crime travel demand model.

5. Professor Peter Stopher of the University of Sidney in Australia who provided technical advice on the crime travel demand model.

6. Mr. Phil Canter of the Baltimore County Police Department, Towson, MD who has been with the project since its inception. For this round, he provided support and data for analysis.

7. Ms. Sandra Wortham of Wortham Design, Wilmington, DE who designed the graphical icons used in the program.

8. Mr. Ian Cahill of Cahill Software, Ottawa, Ontario for providing Poisson and OLS regression maximum likelihood code, based on his MLE++ software package. http://www.magma.ca/~cahill.

9. The GNU project library for providing F-test and t-test code. http://www.gnu.org.

10. Dr. Carolyn Rebecca Block of the Illinois Criminal Justice Information Authority for providing the STAC routine.

11. The dedicated project managers at the Mapping and Analysis for Public Safety Program (MAPS) at NIJ: Ms. Debra Stoe and Mr. Ron Wilson. They both supported the project through this development and provided valuable feedback on the new routines and their utility.

12. All the other individuals from the MAPS unit who have supported the project in earlier stages: for the second version, Ms. Elizabeth Groff of the Institute of Law and Justice, Mr. Eric Jefferis of the University of Akron, and Professor Robert Langworthy of the University of Alaska; and, for the first version, Ms. Cindy Mamalian and Dr. Nancy LaVigne of the Urban Institute.

13. To the individuals of the Baltimore Metropolitan Council who provided network and other data on both Baltimore County and the City of Baltimore, in particular Jacqueline Zee, Matt de Rouville, and Gene Bandy. Thanks also to Alan Clark of the Houston-Galveston Area Council for making available data on Houston motor vehicle crashes.

14. To individuals who have provided feedback and information for this and previous versions of CrimeStat: Professor Eric Renshaw of the University of Strathclyde in Glasgow, Mr. John DeVoe of Siebel Systems, Professor Jim LeBeau of Southern Illinois University, Mr. Bryan Hill of the Glendale (Arizona) Police Department, Professor Karl Kim of the University of Hawaii, Mr. Luben Dimov of Louisiana State University, Mr. Weijie Zhou of the Houston-Galveston Area Council, and Mr. Martin Hittleman of Valley Community College in Los Angeles.

15. To the individuals who provided a examples for the manual: Renato Assunção, Cláudio Beato, Bráulio Silva of the Federal University of Minas Gerais in Belo Horizonte, Brazil; Daniel Bibel of the Massachusetts State Police; Gilberto Câmara, Silvana Amaral, Antônio Miguel V. Monteiro, and José A. Quintanilha of the Instituto Nacional de Pesquisas Espaciais in Brazil; Spencer Chainey of InfoTech Enterprises Europe in London, England; Richard Crepeau of Appalachian State University; Jaishankar Karuppannan of the University of Madras in Chepauk, India; Yongmei Lu of Southwest Texas State University; David McGrath of the Johnstown Castle Research Centre in Wexford, Ireland; Dietrich Oberwittler and Marc Wiesenhütter of the Max Planck Institute for Foreign and International Criminal Law in Freiburg, Germany; Derek Paulsen of Appalachian State University; Gaston Pezzuchi of the Buenos Aires Province Police Force; Mike Saweda of the University of Ottawa; Takahito Shimada of the National Police Agency in Chiba, Japan; Brent Snook, Paul Taylor & Craig Bennell of the University of Liverpool, England; Matthew Stone of the California Department of Health Services, Chaosheng Zhang of the National University of Ireland in Galway, Ireland; Marta A. Guerra of the Centers for Disease Control and Prevention; Richard Hoskins of the State of Washington Department of Health; Tom Reynolds of the University of Texas School of Public Health along with Luc

Anselin, Richard Block, Carolyn Block, Phil Canter, Long Doan, Daniel Helms, Jim LeBeau, Ron Wilson and Bryan Hill mentioned above.

16. To the dozens of individuals who provided feedback and suggestions for improving the program. They are, unfortunately, too numerous to mention by name.

17. Finally, this program is dedicated to my wife, Dr. C. Elizabeth Castro, for being so patient and supportive throughout this long process. She is the inspiration for this whole effort.

License Agreement and Disclaimer

This project was supported by Grant No. 2002-IJ-CX-0007 awarded by the National Institute of Justice, Office of Justice Programs, US Department of Justice. Points of view in this document are those of the author and do not necessarily represent the official position or policies of the US Department of Justice.

Any questions about the use of this program should be directed to either:

Dr. Ned Levine
Ned Levine & Associates
Houston, TX
ned@nedlevine.com

Mr. Ron Wilson
Mapping and Analysis for Public Safety Program
National Institute of Justice
U. S. Department of Justice
810 7th St, NW
Washington, DC 20531
Ronald.Wilson@usdoj.gov

CrimeStat III

Part I: Program Overview

Chapter 1
Introduction to *CrimeStat*

CrimeStat® is a spatial statistics package that can analyze crime incident location data. Its purpose is to provide a variety of tools for the spatial analysis of crime incidents or other point locations. It is a stand-alone *Windows® XP Professional®* program that can interface with most desktop geographic information systems (GIS). It is designed to operate with large crime incident data sets collected by metropolitan police departments. However, it can be used for other types of applications involving point locations, such as the location of arrests, motor vehicle crashes, emergency medical service pickups, or facilities (e.g., police stations).

Uses of Spatial Statistics in Crime Analysis

Most GIS packages, such as *MapInfo®*, *ArcView®*, *ArcGIS®*, *ARC/INFO®*, *Atlas*GIS™*, and *Maptitude®*, have very sophisticated data base operations. They do not, however, have statistical methods other than means and standard deviations of variables. For most purposes, GIS can provide great utility for crime analysis, allowing the plotting of different incident locations and the ability to select subsets of the data (e.g., incidents by precinct, incidents by time of day). Most crime analysts visually inspect incident maps and, based on their experience, draw conclusions about shifts over time, 'hot spots' and other patterns suggested by the data.

There are times, however, when a more quantitative approach is needed. For example, an analyst wishing to examine patterns of streets robberies over time will need indices which document how the robberies may have shifted. For a neighborhood showing an apparent sudden increase in auto thefts, there needs to be a quantitative standard to define the 'typical' level of auto thefts. In assigning police cars to patrol particular major arteries, the center of minimum travel needs to be identified in order to maximize response time to calls for service. For research, as well, quantification is important. In examining correlates of burglaries, for example, a researcher needs to determine the exposure level, namely how many residences or commercial buildings exist in a community in order to establish a level of burglary risk. Or a precinct may want to target areas for which there is a high concentration of incidents occurring within a short time ('hot spots'). While some of these analyses can be conducted with GIS queries, quantification can allow a more precise identification and the ability to compare different types of incidents. In short, there are many uses for quantitative analysis for which a statistical program becomes important.

The *CrimeStat III* Spatial Statistics Program

CrimeStat is a tool designed to provide statistical summaries and models of crime incident data. The tool kit provides crime analysts and researchers with a wide range of spatial statistical procedures that can be linked to a GIS. The procedures vary from the simple to some very sophisticated 'cutting edge' routines. The reasoning is that different audiences vary in their needs and requirements. The program should be of benefit to different organizations. For many crime analysts, simple descriptions of the spatial

distribution will be sufficient with the aim being practical intervention over a short time period. For these persons, many of the techniques provided in *CrimeStat* will be unnecessary.

For other analysts, statistical tools can supplement a much larger GIS effort, such as the Regional Crime Analysis System (RCAGIS) that was developed by the U.S. Department of Justice in cooperation with a number of police departments in the Baltimore-Washington metropolitan area (USDOJ, 2000). For other researchers, even more demanding techniques may be needed to detect the underlying spatial structure as a means for formulating a temporal-spatial theory. A pattern in and of itself has little meaning unless it is linked to some framework. The ability to quantify relationships with a large amount of data can address problems that previously were avoided and can be a first step in developing an explanatory framework or interventionist strategy. *CrimeStat* attempts to address both types of needs by providing statistics in a 'toolbox' framework. We recognize that today's exotic statistical techniques may become tomorrow's practical diagnostics and want the program to be useful for many years.

Input and Output

CrimeStat is a full-featured *Windows®XP Professional®* program using a graphical interface with database and expanded statistical functions. It can read files in various formats - *dBase®* (III, IV, or V), which is a common file format in desktop GIS programs, *ArcView* Shape (shp) files, *MapInfo* data (dat) files, and files conforming to the ODBC standard, such as Excel, Lotus 1-2-3, Microsoft Access, and Paradox (Borland.Com, 1998; ESRI, 1998a; Microsoft, 1999). In addition, many other GIS packages, such as *Maptitude®* can read 'dbf', 'shp', 'bna' or 'mif' files.

Output includes both displayed tables, which can be printed as text or copied to a word processing program, and graphical output. *CrimeStat* can write graphical objects to the *ArcView®*, *ArcGis®* , *MapInfo®*, and *Atlas*GIS™* GIS programs and can write interpolation files to these programs, to programs that read Ascii grid files (e.g., *Vertical Mapper®*), and to the *Surfer® for Windows* and *ArcView Spatial Analyst©* programs (Golden Software, 1994; ESRI, 1998a; 1998b; 1998c; 1997; MapInfo, 1998).

Statistical Routines

CrimeStat III includes routines for:

Type of distance measurement
Direct distance
Indirect distance
Network distance

Spatial distribution
Mean center
Standard distance deviation

Standard deviational ellipse
Median center
Center of minimum distance
Directional mean and variance
Convex Hull
Moran's I spatial autocorrelation index
Geary's C spatial autocorrelation index
Moran Correlogram

Distance analysis

Nearest neighbor analysis
Ripley's K statistic
Assign primary points to secondary points
Within primary file distance matrix
Between primary file and secondary file distance matrix
Between primary file and grid distance matrix
Between secondary file and grid distance matrix

Hot spot analysis

Mode
Fuzzy mode
Nearest neighbor hierarchical clustering
Risk-adjusted nearest neighbor hierarchical clustering
Spatial and temporal analysis of crime routine (STAC)
K-mean clustering
Anselin's local Moran test

Interpolation

Single variable variable kernel density interpolation
Duel variable variable kernel density interpolation

Space-time analysis

Knox index
Mantel index
Correlated walk model

Journey-to-Crime analysis

Calibrate Journey-to-crime function
Journey-to-crime estimation
Draw crime trips

Crime Travel Demand: Trip Generation

Skewness diagnostics
Calibrate model
Make prediction
Balance predicted origins & destinations

Crime Travel Demand: Trip Distribution
Calculate observed origin-destination trips
Calibrate impedance function
Calibrate origin-destination model
Apply predicted origin-destination model
Compare observed and predicted origin-destination trip lengths

Crime Travel Demand: Mode Split
Calculate mode split

Crime Travel Demand: Network Assignment
Check for one-way streets
Create a transit network from primary file
Network assignment

Many of these routines allow variations yielding an even larger number of statistics to be calculated. Also, *CrimeStat* has Dynamic Data Exchange (DDE) capabilities so that it can be accessed from within another program.

CrimeStat is a program that specializes in the analysis of point locations. Over the years, many statistical tools have been developed for analyzing point locations. Many of these have either not been implemented as computer programs or were collected together as part of a specialized statistical system. They have been typically unavailable to crime analysts and the major statistical packages (e.g., *SAS®*, *SPSS™*, *Systat®*) do not include these routines. Consequently, we have collected those that are most appropriate for crime analysis and detection and organized them into a single package with a common graphical interface. They represent a wide variety of tools that can be used for crime analysis. *CrimeStat* can also analyze zonal data by treating them as 'pseudo' points. For example, the centroid of a census tract can be treated as a point and a value associated with the tract (e.g., its population) can be treated as an Intensity value (see chapter 3).

Program Requirements

Required Hardware and Operating System

CrimeStat III was developed for the *Windows®XP Professional®* operating system, though it will also work with the *Windows®2000®*, or *Windows®NT®* operating system; it is not hardware dependent so that any processor that can run *Windows XP Professional/ 2000/ NT* will suffice. Some of the routines can also run on the *Windows®95®* (Microsoft, 1995) or *Windows® 98®* (Microsoft, 1998c) operating systems. However, the program was not designed around nor fully tested for those operating systems. It is highly recommended that the program be run on a more current version of *Windows*.

While it can run on a relatively slow computer (e.g., 250 MHz clock speed) with limited RAM (e.g., 64 MB), it will run much better on a 1.6 GHz computer (or faster) with more than 256 MB of RAM. The faster the processor used, the quicker the program will

run. The more RAM the computer has, the quicker the program will run. The program is very intensive with respect to calculations. Some of the statistics produce very large matrices (e.g., the trip distribution routines in the Crime Travel Demand module). Depending on the size of the data files that will be processed, there may be hundreds of millions of calculations on any one run. It is critical, therefore, that the computer be fast and have sufficient amounts of RAM. The program was designed on an *Windows® XP Professional®* system with 1 GB of RAM running a single processor 1.6 GHz computer.

For most of the simple statistics, a reasonably fast computer will be adequate. However, several of the trip distribution routines will push the limits of most computer systems. The current 32 bit Windows operating system has a maximum limit of 4 Gb of RAM (actual and virtual). With a trip distribution matrix, there are M x N cells where M is the number of rows (origins) and N is the number of columns (destinations). With 8 bits being assigned to a number, practically a square matrix of about 10,000 x 10,000 would be close to the theoretical maximum allowed. Aside from taking a very long time to be calculated (days, if not weeks), the storage space required to save such a matrix will be very large. In short, the size of the files that can be processed will depend on the particular routines being run.

CrimeStat is a multi-threaded application written to take advantage of multiple processors if the hardware and operating system support multiple processors. The program is designed to be multi-threading which means that it will take advantage of multiple processors using *Windows® XP Professional®, Windows® 2000®*, or *Windows® NT®*. These operating systems support two processors. *Windows2003 Server®* supports up to four processors. Thus, if there are two processors and *Windows® XP Professional®* or *Windows® 2000®* is the operating system, *CrimeStat* will calculate routines in about half the time. If there are four processors and *Windows® 2003 Server®* is the operating system, *CrimeStat* will calculate routines in about a quarter of the time. The multiples are not exact since processing time must be allocated for input of data and output of tables.

For small data sets, this feature is not important as most runs will be very quick. However, for large data sets (e.g., 3000 cases or larger), the speed of calculations become important. For example, on a 1.6 GHZ single-processor *Pentium M®* computer with 1 GB of RAM running *Windows® XP Professional®*, it takes about 4 minutes to complete a nearest neighbor analysis on 14,853 cases involving the calculating of distance from every point to every other point multiple times (for different neighbors). On a similar 1 GHz dual-processor *Pentium®* computer with 1 GB of RAM running *Windows® XP Professional®*, it takes about 2 minutes to complete the same task. Slower systems will produce correspondingly slower times. The larger the file that is being processed, the more critical becomes the calculating efficiency of the computer.

If a police department is expecting to run large data sets, it would benefit them to purchase fast multiple-processor computers with lots of RAM and fast hard disks to speed calculating times. The evolution of new processors is moving in this direction anyway so that a multi-processor computer will become the norm in the next couple of years.

Required Software

CrimeStat needs a Windows environment to operate. The program was designed for a *Windows® XP Professional®/ 2000®/ NT®* operating system so it is better optimized for that system. In particular, *Windows® XP Professional®/ 2000®/ NT®* has two features that allows *CrimeStat* to run more efficiently. First, it is a multi-threading operating system and can utilize multiple processors, as mentioned above. Neither *Windows® XP Home®*, *Windows® 95®* nor *Windows® 98®* can utilize multiple processors. Second, it addresses memory in a more efficient way, as a large flat block. *Windows® 95®* cannot handle cache memory above 64 MB. *Windows 98* can handle RAM above 64 MB, but still has poorer memory management than *NT*. Consequently, for the same machine, *CrimeStat* will run more efficiently (i.e., more quickly) in *XP Professional®/ 2000®/ NT®* than in older or more limited operating systems.

CrimeStat is a stand-alone program. Hence, it does not require any other program other than a Windows operating system. However, to be maximally useful, there should be an accompanying GIS program. While point data can be obtained from a non-GIS system (e.g., census files include lat/lon coordinates for the centroid of census units), the use of the GIS to assign the coordinates is almost necessary. Further, many of the outputs of *CrimeStat* are for GIS programs. Thus, to view an ellipse or to view a three dimensional interpolation produced by *CrimeStat* will require an appropriate GIS package.

Installing the Program

CrimeStat comes compressed in a zipped file called *CrimeStat.zip*. To install the program, it is necessary to have a compression program that recognizes the 'zip' format:

1. Create a directory using *Windows Explorer* and copy the file to that directory.

2. Double click on the file name in *Explorer*. When the name *CrimeStat.zip* is visible in the dialog box name field, double click the name with the left mouse button. *CrimeStat* will be installed in that directory.

3. The program help menu can also access the manual. For this feature to work, however, it is important the chapters of the manual be kept in the same directory as the program.

Adding an Item to the Start Menu

To add *CrimeStat* to the start menu:

1. Click on the *Start* button in Windows followed by *Settings* then *Taskbar*. Click on *Start Menu Programs* followed by *Add*.

2. In the dialog box, click on *Browse,* point to the directory where *CrimeStat*

resides, and click on its name followed by *Open*. When the name *CrimeStat* is in the dialog box name field, click on the *Next* button.

3. Double-click on the folder to which *CrimeStat* is to be assigned.

4. Finally, type a name for *CrimeStat* (e.g., CrimeStat) followed by *Finish*.

Adding an Icon to the Desktop

To add *CrimeStat* to the desktop:

1. Double-click on *My Computer*.

2. Double-click on the drive in which *CrimeStat* resides followed by the directory that it is in (it may be several levels down).

3. Click once on the name *CrimeStat* with the left button and then hold down the right mouse button.

4. While holding the right mouse button, scroll to *Create Shortcut*.

5. The name *Shortcut to CrimeStat* will be placed at the end of the list of files.

6. Highlight the name by clicking on it once. Hold the left mouse button down and drag this name on to the desktop.

7. You can rename it *CrimeStat* by clicking on its icon with the right mouse button followed by *Rename*.

8. Alternatively, you can use *Windows Explorer* to create a shortcut and then drag the shortcut to the desktop.

Installing the Sample Data Sets

There are four sample data sets that can be used to run the program, also in 'zip' format. Since the data are simulated, they should not be used for real applications:

1. **SampleData.zip**. The data are simulated incident points from Baltimore City and Baltimore County in Maryland.[1] They are provided to allow a user to become familiar with the program quickly. However, ultimately, the value of the program must be tested on real data, rather than simulated data.

2. **JtcSampleData.zip**. There are three files of simulated data for use with the Journey-to-crime routine (chapter 8):

   A. *JtcTest1.dbf* - A simulated data set of 2000 robberies in Baltimore

County that can be used for calibrating a travel demand function. Each record has a crime location and a residence location of the offender.

B. *JtcTest2.dbf* - A simulated data set of 2500 burglaries in Baltimore County that can be used for calibrating a travel demand function. Each record has a crime location and a residence location of the offender.

C. *Serial1.dbf* - A simulated data set of the location of seven incidents committed by a single serial offender. To become familiar with the journey to crime routine, they can be treated as either robberies or burglaries.

3. **CorrelatedWalk.zip**. These are three files of simulated data for use with the Correlated Walk Analysis routine (chapter 9):

   A. *TestSerial1.dbf* - A simulated data set for an algorithmic offender who committed 13 incidents.

   B. *TestSerial2.dbf* -A simulated data set for an algorithmic offender who committed 12 incidents.

   C. *TSerl13.dbf* - A simulated data set for a realistic offender who committed 13 incidents.

4. **BaltCountyZones.zip**. There are two files of data on crime incidents by zone for Baltimore County, Md. They are examples used in the crime travel demand module (chapter 11-17):

   A. *BaltOrigins.dbf* - a data set on 532 origin zones in both Baltimore County and the City of Baltimore from the late 1990s. There are data on crimes originating from each zone and demographic, economic and land use variables associated with those zones.

   B. *BaltDest.dbf* - a data set of 325 destination zones in Baltimore County only. There are data on crimes occurring in each zone and demographic, economic and land use variables associated with those zones.

To install any of these sample data files, it is necessary to have a compression program that recognizes the 'zip' format:

1. Create a data directory using *Windows Explorer* and copy the files to that directory.
2. In *Windows Explorer*, double-click on its name and then follow the instructions.

Step-by-Step Instructions

This manual will go through the program step-by-step to address how it can be used by a crime mapping/analysis unit within a police department. Chapter 2 provides a quick guide for all the data definition and program routines and chapter 3 provides detailed instructions on setting up data to run with *CrimeStat*. The statistical routines are described in parts II, III, and IV. Part II presents a number of statistics for spatial description, part III presents a number of statistics for spatial modeling, while part IV presents a crime travel demand module. The different statistics are presented and detailed examples of each technique are shown.

Options

There is an option tab that allows the saving and loading of program parameters and the setting of colors for each of main headings: Data setup, Spatial description, and Spatial modeling. One can also output simulated data during the simulation runs; this will be explained in the appropriate section.

Short Applications

The manual also includes a number of applications conducted by other researchers and analysts. These are presented as one page sidebars in the various chapters. Most of these are from criminal justice. But, applications from other fields have also been included. The aim is to show the diversity of applications that researchers and analysts have used with the various routines in *CrimeStat*.

On-line Help

In addition, there is on-line help for the program. There is a *Help* button that can be pushed to access all the help items. In addition, the program has context-sensitive help. On any page or routine, typing *F1* will pop up an appropriate help item. The on-line help can also access the program manual. For this to be available, be sure to store the chapters of the manual in the *same directory* as the program.

Endnotes for Chapter 1

1. The data were simulated by a random number generator following the distribution of several types of crime incidents. Because the data were selected by a random generator, the points do not necessarily fall on streets or even stay within the boundaries of Baltimore City and Baltimore County; some even fall into the Chesapeake Bay! Their purpose is to provide a simple data set so users can become familiar with the program.

Chapter 2
Quickguide to *CrimeStat*

The following are quick instructions for the use of *CrimeStat®III* , paralleling the online help menus in the program. Detailed instructions should be obtained from chapters 3-17 in the documentation. *CrimeStat* has five basic groupings in seventeen program tabs and one option tab. Each tab lists routines, options and parameters:

Data setup

1. Primary file
2. Secondary file
3. Reference file
4. Measurement parameters

Spatial description

5. Spatial distribution
6. Distance analysis I
7. Distance analysis II
8. 'Hot Spot' analysis I
9. 'Hot Spot' analysis II

Spatial modeling

10. Interpolation
11. Space-time analysis
12. Journey to crime estimation

Crime Travel Demand

13. Trip generation
14. Trip distribution
15. Mode split
16. Network assignment
17. File worksheet

Options

18. Saving parameters, colors and options

Figure 2.1-2.18 show the ten operational tab screens with examples of data input and routine selection.

I. Data Setup

Primary File

A primary file is required for *CrimeStat*. It is a point file with X and Y coordinates. For example, a primary file could be the location of street robberies, each of which have an associated X and Y coordinate. There can be associated weights or intensities, though these are optional. There may be time references, though these are optional. For example, if the points are the locations of police stations, then the intensity variable could be the number of calls for service at each police station while the weighting variable could be service zones. More than one file can be selected. The time references are used in the space-time analysis routines and are by hours, days, weeks, months, or years.

Select Files

Select the primary file. *CrimeStat* can read ASCII, dBase®III/IV/V 'dbf', ArcView® 'shp', and MapInfo® 'dat' files, Microsoft Access 'mdb' files and files formats that correspond to the ODBC standard interface. Select the type of file to be selected. Use the browse button to search for a particular file name. If the file type is ASCII, select the type of data separator (comma, semicolon, space, tab) and the number of columns. ODBC files have to be defined for the particular computer on which it runs. See chapter 3 for instructions on defining ODBC files. Use the browse button to search for the file name.

Variables

Define the file that contains the X and Y coordinates. *CrimeStat* can accept values associated with the X and Y coordinates. These are called *Intensities* or *Weights*. Essentially, these are two different types of weights that could be used. If weights or intensities are being used, define the file that contains these variables. Certain statistics (e.g., spatial autocorrelation, local Moran) require intensity or weight values and many other statistics can use intensity or weight values. It is possible to have both an intensity variable and a weighting variable, though the user should be cautious in doing this to avoid 'double weighting'. If a time variable is used, it must be an integer or real number (e.g., 1, 36892). Do not use formatted dates (e.g., 01/01/2001, October 1, 2001). Convert these to real numbers before using the space-time analysis routines.

Column

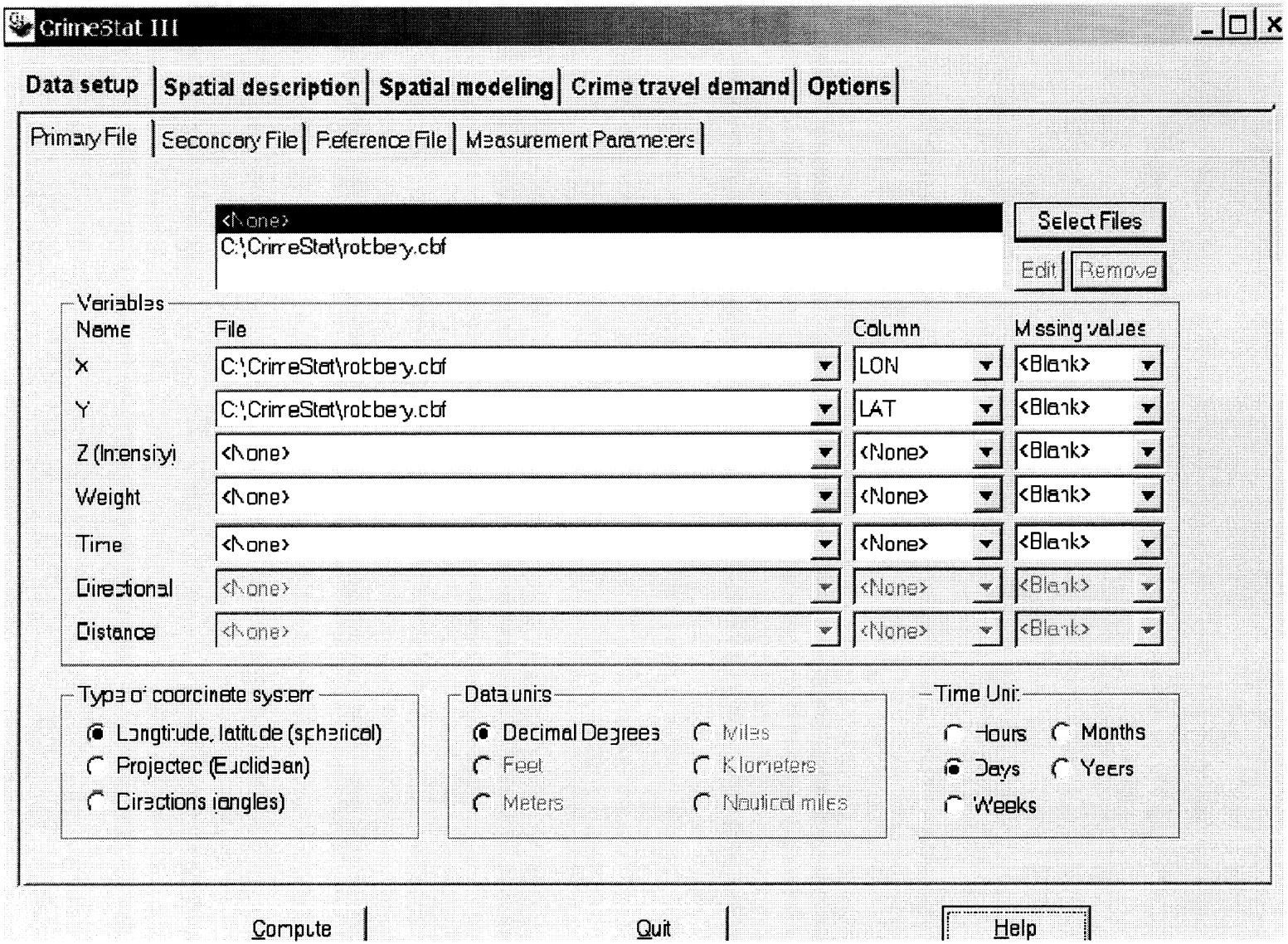
Select the variables for the X and Y coordinates respectively (e.g., Lon, Lat, Xcoord, Ycoord). If weights or intensities are being used, select the appropriate variable names. If a time variable is used, select the appropriate variable name.

Missing values

Identify whether there are any missing values. By default, *CrimeStat* will ignore records with blank values in any of the eligible fields or records with non-numeric values

Figure 2.1

Primary File Screen



(e.g., alphanumeric characters, #, *). Blanks will always be excluded unless the user selects ***<none>***. There are 8 possible options:

1. **<blank>** fields are automatically excluded. This is the default
2. **<none>** indicates that no records will be excluded. If there is a blank field, CrimeStat will treat it as a 0
3. **0** is excluded
4. **–1** is excluded
5. **0** and **–1** indicates that both 0 and -1 will be excluded
6. **0, -1 and 9999** indicates that all three values (0, -1, 9999) will be excluded
7. **Any** other numerical value can be treated as a missing value by typing it (e.g., 99)
8. **Multiple** numerical values can be treated as missing values by typing them, separating each by commas (e.g., 0, -1, 99, 9999, -99)

Directional

If the file contains directional coordinates (angles), define the file name and variable name (column) that contains the directional measurements. If directional coordinates are used, there can be an optional distance variable for the measurement. Define the file name and variable name (column) that contains the distance variable.

Type of Coordinate System and Data Units

Select the type of coordinate system. If the coordinates are in longitudes and latitudes, then a spherical system is being used and data units will automatically be decimal degrees. If the coordinate system is projected (e.g., State Plane, Universal Transverse Mercator – UTM), then data units can be in feet (e.g., State Plane), meters (e.g., UTM), miles, kilometers, or nautical miles. If the coordinate system is directional, then the coordinates are angles and the data units box will be blanked out. For directions, an additional distance variable can be used. This measures the distance of the incident from an origin location; the units are undefined.

Time units

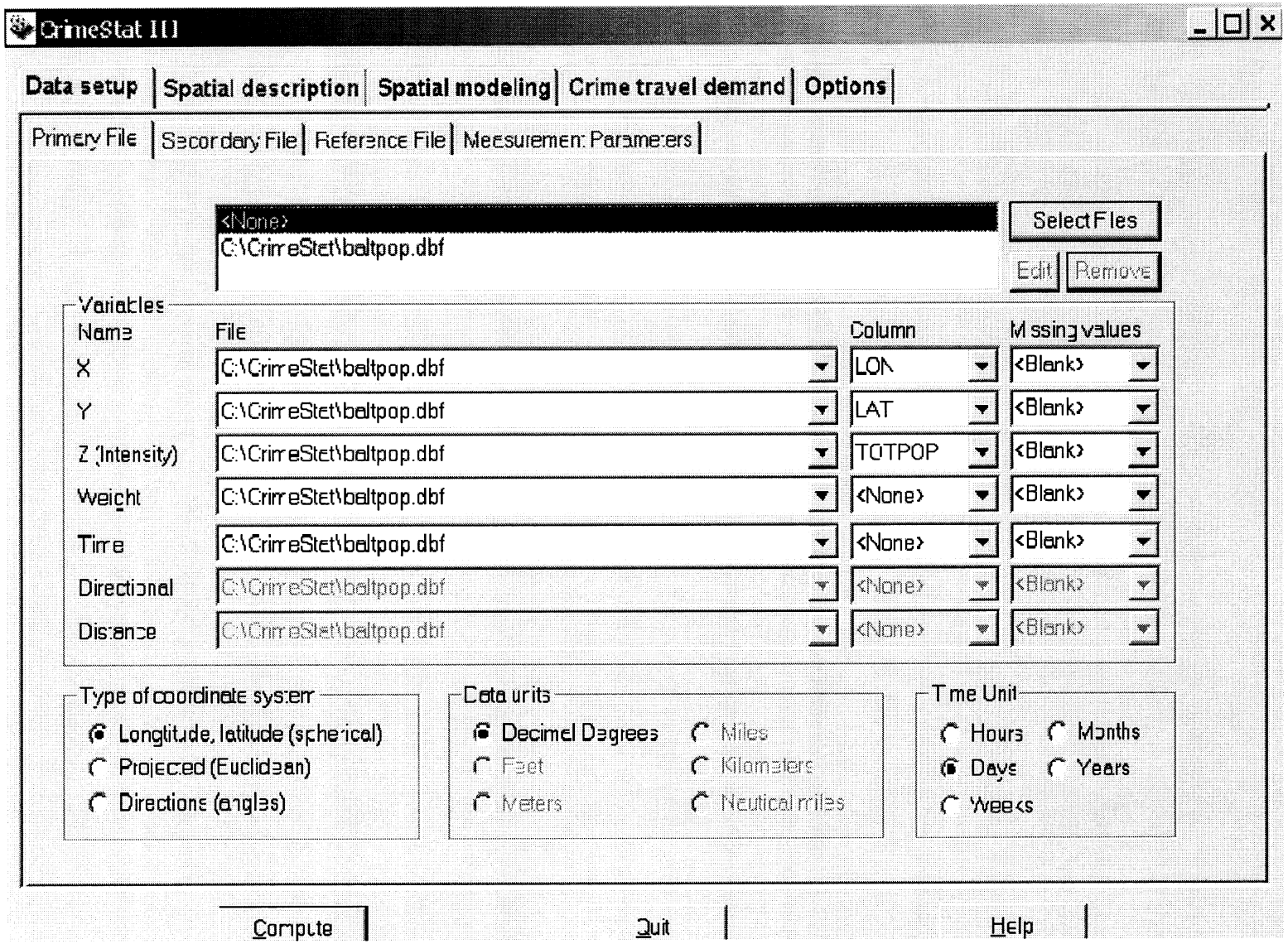
Define the units for the time variable. Time is defined in terms of hours, days, weeks, months, or years. The default value is days. Note, only integer or real numbers can be used (e.g., 1, 36892). Do not use formatted dates (e.g., 01/01/2001, October 1, 2001). Convert these to integer or real numbers before using the space-time analysis routines.

Secondary File

A secondary data file is optional. It is also a point file with X and Y coordinates. It is usually used in comparison with the primary file. There can be weights or intensities variables associated, though these are optional. For example, if the primary file is the location of motor vehicle thefts, the secondary file could be the centroid of census block

Figure 2.2:

Secondary File Screen



groups that have the population of the block group as the intensity (or weight) variable. In this case, one could compare the distribution of motor vehicle thefts with the distribution of population in, for example, the Ripley's "K" routine or the dual kernel density estimation routine. More than one file can be selected. Time units are not used in the secondary file.

Select Files

Select the secondary file. *CrimeStat* can read ASCII, dbase '.dbf', ArcView '.shp' MapInfo 'dat' files, Microsoft Access 'mdb' files and files formats that correspond to the ODBC standard interface. Select the type of file to be selected. Use the browse button to search for a particular file name. If the file type is ASCII, select the type of data separator (comma, semicolon, space, tab) and the number of columns. ODBC files have to be defined for the particular computer on which it runs. See chapter 3 for instructions on defining ODBC files. Use the browse button to search for the file name.

Variables

Define the file that contains the X and Y coordinates. If weights or intensities are being used, define the file that contains these variables. Certain statistics (e.g., spatial autocorrelation, local Moran) require intensity values and most other statistics can use intensity values. Most other statistics can use weights. It is possible to have both an intensity variable and a weighting variable, though the user should be cautious in doing this to avoid 'double weighting'. Time units are not used in the secondary file.

Column

Select the variables for the X and Y coordinates respectively (e.g., Lon, Lat, Xcoord, Ycoord). If there are weights or intensities being used, select the appropriate variable names. Time units are not used in the secondary file.

Missing values

Identify whether there are any missing values. By default, *CrimeStat* will ignore records with blank values in any of the eligible fields or records with non-numeric values (e.g., alphanumeric characters, #, *). Blanks will always be excluded unless the user selects ***<none>***. There are 8 possible options:

1. **<blank>** fields are automatically excluded. This is the default
2. **<none>** indicates that no records will be excluded. If there is a blank field, CrimeStat will treat it as a 0
3. **0** is excluded
4. **–1** is excluded
5. **0** and **–1** indicates that both 0 and -1 will be excluded
6. **0, -1 and 9999** indicates that all three values (0, -1, 9999) will be excluded
7. **Any** other numerical value can be treated as a missing value by typing it (e.g., 99)

8. **Multiple** numerical values can be treated as missing values by typing them, separating each by commas (e.g., 0, -1, 99, 9999, -99)

Type of Coordinate System and Data Units

The secondary file must have the same coordinate system and data units as the primary file. This selection will be blanked out, indicating that the secondary file carries the same definition as the primary file. Directional coordinates (angles) are not allowed for the secondary file.

Reference File

For referencing the study area, there is a reference grid and a reference origin. The reference file is used in the risk-adjusted nearest neighbor hierarchical clustering routine, journey-to-crime estimation and in the single and dual variable kernel density estimation routines. The file can be an external file that is input or can be created by *CrimeStat*. It is usually, though not always, a grid which is overlaid on the study area. The reference origin is used in the directional mean routine. The file can be an external file that is input or can be created by *CrimeStat*.

Create reference grid

If allowing *CrimeStat* to generate a true grid, click on 'Create Grid' and then input the lower left and upper right X and Y coordinates of a rectangle placed over the study area. Cells can be defined either by cell size, in the same coordinates and data units as the primary file, or by the number of columns in the grid (the default). In addition, a reference origin can be defined for the directional mean routine. The reference grid can be saved and re-used. Click on 'Save' and enter a file name. To use an already saved file, click on 'Load' and the file name. The coordinates are saved in the registry, but can be re-saved in any directory. To save to a particular directory, with the Load screen open, click on 'Save to file' and then enter a directory and a file name. The default file extension is 'ref.

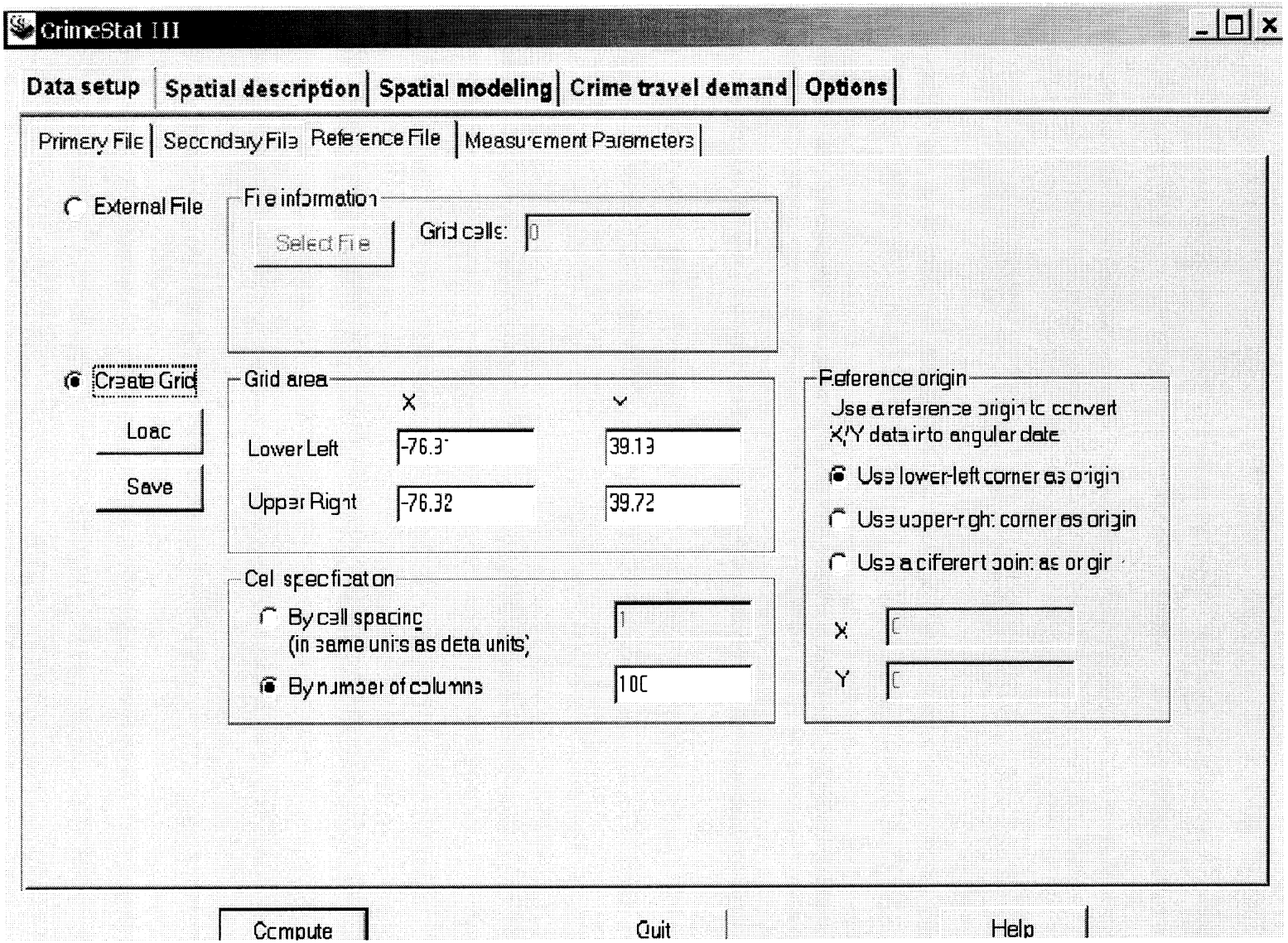
Input external file

If an external file that stores the coordinates of each grid cell is to be used, select the name of the reference file. *CrimeStat* can read ASCII, dBase '.dbf', ArcView '.shp', MapInfo 'dat' files, Microsoft Access 'mdb' files and files formats that correspond to the ODBC standard interface. Select the type of file to be selected. Use the browse button to search for the file. If the file type is ASCII, select the type of data separator (comma, semicolon, space, tab) and the number of columns. ODBC files have to be defined for the particular computer on which it runs. See chapter 3 for instructions on defining ODBC files. Use the browse button to search for the file name.

A reference file that is read into *CrimeStat* need not be a true grid (a matrix with k columns and l rows). However, an external reference file that is read in can only be output

Figure 2.8:

Reference File Screen



to *Surfer for Windows* since the other output formats – *ArcView*, *MapInfo*, *Atlas*GIS*, *ArcView Spatial Analyst*, and ASCII grid require the reference file to be a true grid.

Reference origin

A reference origin can be defined for the directional mean routine. The reference origin can be assigned to:

1. Use the lower-left corner defined by the minimum X and Y values. This is the default
2. Use the upper-right corner defined by the maximum X and Y values
3. Use a different origin point. With the latter, the user must define the origin

Measurement Parameters

The measurement parameters page defines the measurement units of the coverage and the type of distance measurement to be used. There are three components that are defined:

Area

First, define the geographical area of the study area in area units (square miles, square nautical miles, square feet, square kilometers, square meters.) Irrespective of the data units that are defined for the primary file, CrimeStat can convert to various area measurement units. These units are used in the nearest neighbor, Ripley's "K", nearest neighbor hierarchical clustering, risk-adjusted nearest neighbor hierarchical clustering, Stac, and K-means clustering routines. If no area units are defined, then CrimeStat will define a rectangle by the minimum and maximum X and Y coordinates.

Length of street network

Second, define the total length of the street network within the study area or an appropriate comparison network (e.g., freeway system) in distance units (miles, nautical miles, feet, kilometers, meters.) The length of the street network is used in the linear nearest neighbor routine. Irrespective of the data units that are defined for the primary file, CrimeStat can convert to distance measurement units. The distance units should be in the same metric as the area units (e.g., miles and square miles/meters and square meters.)

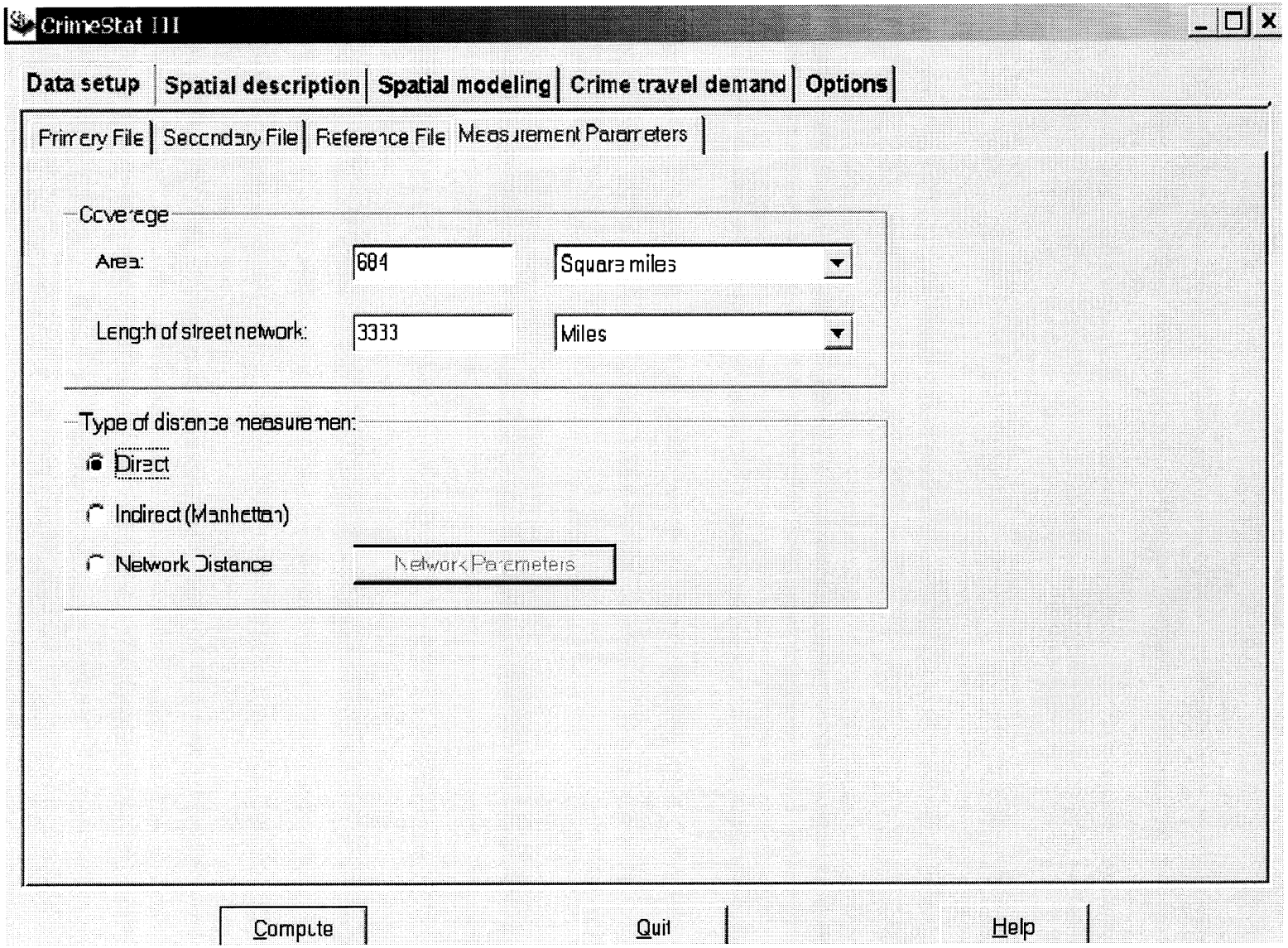
Type of distance measurement

Third, define how distances are to be calculated. There are three choices:

1. Direct distance
2. Indirect (Manhattan) distance
3. Network distance

Figure 2.4:

Measurement Parameters Screen



Direct

If direct distances are used, each distance is calculated as the shortest distance between two points. If the coordinates are spherical (i.e., latitude, longitude), then the shortest direct distance is a 'Great Circle' arc on a sphere. If the coordinates are projected, then the shortest direct distance is a straight line on a Euclidean plane.

Indirect

If indirect distances are used, each distance is calculated as the shortest distance between two points on a grid, that is with distance being constrained to the horizontal or vertical directions (i.e., not diagonal.) This is sometimes called 'Manhattan' metric. If the coordinates are spherical (i.e., latitude, longitude), then the shortest indirect distance is a modified right angle on a spherical right triangle; see the documentation for more details. If the coordinates are projected, then the shortest indirect distance is the right angle of a right triangle on a two-dimensional plane

Network distance

If network distances are used, each distance is calculated as the shortest path between two points using the network. Alternatives to distance can be used including speed, travel time, or travel cost. Click on 'Network parameters' and identify a network file.

Type of network

Network files can bi-directional (e.g., a TIGER file) or single directional (e.g., a transportation modeling file). In a bi-directional file, travel can be in either direction. In a single directional file, travel is only in one direction. Specify the type of network to be used.

Network input file

The network file can either be a shape file (line, polyline, or polylineZ file) or another file, either dBase IV 'dbf', Microsoft Access 'mdb', Ascii 'dat', or an ODBC-compliant file. The default is a shape file. If the file is a shape file, the routine will know the locations of the nodes. For a dBase IV or other file, the X and Y coordinate variables of the end nodes must be defined. These are called the "From" node and the "End" node. An optional weight variable is allowed for all types of file0073. The routine identifies nodes and segments and finds the shortest path. If there are one-way streets in a bi-directional file, the flag fields for the "From" and "To" nodes should be defined.

Network weight field

Normally, each segment in the network is not weighted. In this case, the routine calculates the shortest distance between two points using the distance of each segment. However, each segment can be weighted by travel time, speed or travel costs. If travel time is used for weighting the segment, the routine calculates the shortest time for any route

between two points. If speed is used for weighting the segment, the routine converts this into travel time by dividing the distance by the speed. Finally, if travel cost is used for weighting the segment, the routine calculates the route with the smallest total travel cost. Specify the weighting field to be used and be sure to indicate the measurement units (distance, speed, travel time, or travel cost) at the bottom of the page. If there is no weighting field assigned, then the routine will calculate using distance.

From one-way flag and To one-way flag

One-way segments can be identified in a bi-directional file by a 'flag' field (it is not necessary in a single directional file). The 'flag' is a field for the end nodes of the segment with values of '0' and '1'. A '0' indicates that travel can pass through that node in either direction whereas a '1' indicates that travel can only pass from the other node of the same segment (i.e., travel cannot occur from another segment that is connected to the node). The default assumption is for travel to be allowed through each node (i.e., there is a '0' assumed for each node). For each one-way street, specify the flags for each end node. A '0' allows travel from any connecting segments whereas a '1' only allows travel from the other node of the same segment. Flag fields that are blank are assumed to allow travel to pass in either direction.

FromNode ID and ToNode ID

If the network is single directional, there are individual segments for each direction. Typically, two-way streets have two segments, one for each direction. On the other hand, one-way streets have only one segment. The FromNode ID and the ToNode ID identify from which end of the segment travel should occur. If no FromNode ID and ToNode ID is defined, the routine will chose the first segment of a pair that it finds, whether travel is in the right or wrong direction. To identify correctly travel direction, define the FromNode and ToNode ID fields.

Type of coordinate system

The type of coordinate system for the network file is the same as for the primary file.

Measurement unit

By default, the shortest path is in terms of distance. However, each segment can be weighted by travel time, travel speed, or travel cost.

1. For travel time, the units are minutes, hours, or unspecified cost units.

2. For speed, the units are miles per hour and kilometers per hour. In the case of speed as a weighting variable, it is automatically converted into travel time by dividing the distance of the segment by the speed, keeping units constant.

3. For travel cost, the units are undefined and the routine identifies routes by those with the smallest total cost.

Network graph limit

Finally, the number of graph segments to be calculated is defined as the network limit. The default is 50,000 segments. Be sure that this number is slightly greater than the number of segments in your network. Note: using network distance for distance calculations can be a slow process, for example taking up to several hours for calculating an entire matrix. Use only if more precision is needed or for the network assignment routine in the crime travel demand module.

Saving Parameters

All the input parameters can be saved. In the options section, there is a 'Save parameters' button. A parameters file must have a 'param' extension. A saved parameters file can be re-loaded with the 'Load parameters' button.

II. Spatial Description

The spatial description section calculates spatial description, distance analysis, and 'Hot Spot' statistics. The 'Hot Spot' statistics are on two separate tabs.

Spatial Distribution

Spatial distribution provides statistics that describe the overall spatial distribution. These are sometimes called centrographic, global, or first-order spatial statistics. There are four routines for describing the spatial distribution and two routines for describing spatial autocorrelation. An intensity variable and a weighting variable can be used for the first three routines. An intensity variable is required for the two spatial autocorrelation routines; a weighting variable can also be used for the spatial autocorrelation indices. All outputs can be saved as text files. Some outputs can be saved as graphical objects for import into desktop GIS programs.

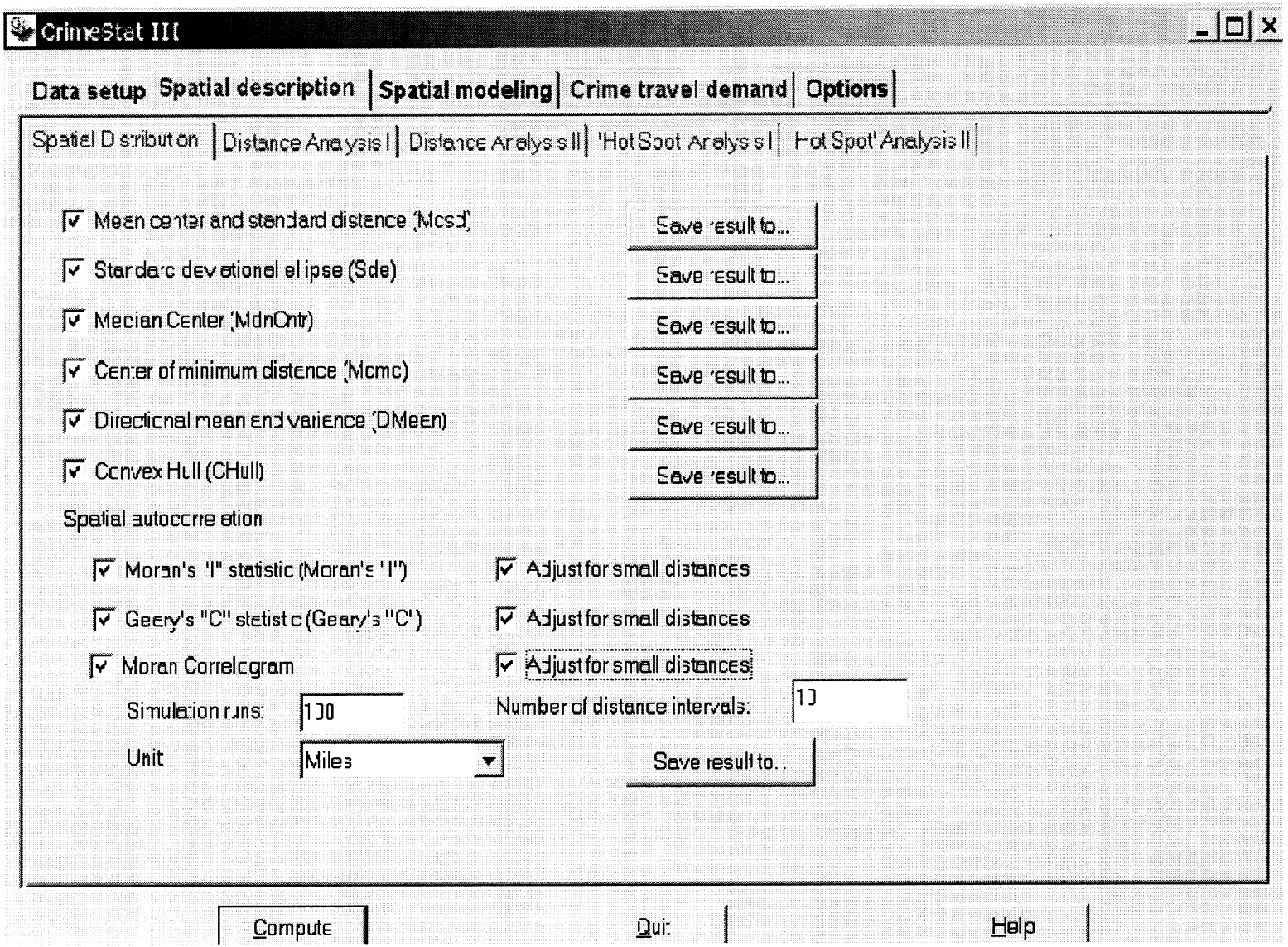
Mean Center and Standard Distance (Mcsd)

The mean center and standard distance define the arithmetic mean location and the degree of dispersion of the distribution. The Mcsd routine calculates 9 statistics:

1. The sample size
2. The minimum X and Y values
3. The maximum X and Y values
4. The X and Y coordinates of the mean center
5. The standard deviation of the X and Y coordinates
6. The X and Y coordinates of the geometric mean

Figure 2.5:

Spatial Distribution Screen



7. The X and Y coordinates of the harmonic mean
8. The standard distance deviation, in meters, feet and miles. This is the standard deviation of the distance of each point from the mean center.
9. The circle area defined by the standard distance deviation, in square meters, square feet and square miles.

The tabular output can be printed and the mean center (mean X, mean Y), the geometric mean, the harmonic mean, the standard deviations of the X and Y coordinates, and the standard distance deviation can be output as graphical objects to ArcView '.shp', MapInfo '.mif' and Atlas*GIS '.bna' formats. A root name should be provided. The mean center is output as a point (MC<*root name*>). The geometric mean is output as a point (GM<*root name*>). The harmonic mean is output as a point (HM<*root name*>). The standard deviation of both the X and Y coordinates is output as a rectangle (XYD<*root name*>). The standard distance deviation is output as a circle (SDD<*root name*>).

Standard Deviational Ellipse (Sde)

The standard deviational ellipse defines both the dispersion and the direction (orientation) of that dispersion. The Sde routine calculates 9 statistics:

1. The sample size
2. The clockwise angle of Y-axis rotation in degrees
3. The ratio of the long to the short axis after rotation
4. The standard deviation along the new X and Y axes
5. The X and Y axes length
6. The area of the ellipse defined by these axes
7. The standard deviation along the X and Y axes
8. The X and Y axes length for a 2X standard deviational ellipse
9. The area of the 2X ellipse defined by these axes

The tabular output can be printed and the 1X and 2X standard deviational ellipses can be output as graphical objects to ArcView 'shp', MapInfo 'mif' and Atlas*GIS 'bna' formats. A root name should be provided. The 1X standard deviational ellipse is output as an ellipse (SDE<root name>). The 2X standard deviational ellipse is output as an ellipse with axes that are twice as large as the 1X standard deviational ellipse (2SDE<root name>). If data are normally distributed, then the 1X standard deviational ellipse will capture approximately 68% of the cases and the 2X standard deviational ellipse will capture approximate 95% of the cases; however, any particular distribution may deviate considerably from normal and the actual percentages may vary.

Median Center (MdnCntr)

The median center is the intersection of the median of the X coordinate and the median of the Y coordinate. This is the approximate middle of the distribution. However, the median center is dependent on the axis of orientation, so it should be used with caution. The MdnCntr routine outputs 3 statistics:

1. The sample size
2. The median of X
3. The median of Y

The tabular output can be printed and the median center can be output as a graphical object to ArcView 'shp', MapInfo 'mif' or Atlas*GIS 'bna' files. A root name should be provided. The median center is output as a point (MdnCntr<root name>).

Center of Minimum Distance (Mcmd)

The center of minimum distance defines the point at which the distance to all other points is at a minimum. The Mcmd routine outputs 5 statistics:

1. The sample size
2. The mean of the X and Y coordinates
3. The number of iterations required to identify a center
4. The degree of error (tolerance) for stopping the iterations
5. The X and Y coordinates defining the center of minimum distance.

The tabular output can be printed and the center of minimum distance can be output as a graphical object to ArcView 'shp', MapInfo 'mif' or Atlas*GIS 'bna' files. A root name should be provided. The center of minimum distance is output as a point (Mdn<root name>).

Directional Mean and Variance (DMean)

The angular mean and variance are properties of angular measurements. The angular mean is an angle defined as a bearing from true North: 0 degrees. The directional variance is a relative indicator varying from 0 (no variance) to 1 (maximal variance). Both the angular mean and the directional variance can be calculated either through angular (directional) coordinates or through X and Y coordinates.

If the primary file cases are directional coordinates (bearings/angles from 0 to 360 degrees), the angular mean is calculated directly from the angles. An optional distance variable can be included. In this case, the directional mean routine will output five statistics:

1. The sample size
2. The unweighted mean angle
3. The weighted mean angle
4. The unweighted circular variance
5. The weighted circular variance.

On the other hand, if the primary file incidents are defined in X and Y coordinates, the angles are defined relative to the reference origin (see Reference file) and the angular

mean is converted into an equation. In this case, the directional mean routine will output nine statistics:

1. The sample size;
2. The unweighted mean angle
3. The weighted mean angle
4. The unweighted circular variance
5. The weighted circular variance
6. The mean distance
7. The intersection of the mean angle and the mean distance (directional mean)
8. The X and Y coordinates for the triangulated mean
9. The X and Y coordinates for the weighted triangulated mean

The directional mean and triangulated mean can be saved as an *ArcView* 'shp', *MapInfo* 'mif', or *Atlas*GIS* 'bna' file. The unweighted directional mean - the intersection of the mean angle and the mean distance is output with the prefix 'Dm' while the unweighted triangulated mean location is output with a 'Tm' prefix. The weighted triangulated mean is output with a 'TmWt' prefix. The tabular output can be printed.

Convex hull (Chull)

The convex hull draws a polygon around the outer points of the distribution. It is useful for viewing the shape of the distribution. The routine outputs three statistics:

1. The sample size;
2. The number of points in the convex hull
3. The X and Y coordinates for each of the points in the convex hull

The convex hull can be saved as an ArcView 'shp', MapInfo 'mif', or Atlas*GIS 'bna' file with a 'Chull' prefix.

Spatial Autocorrelation Indices

Spatial autocorrelation indices identify whether point locations are spatially related, either clustered or dispersed. Two spatial autocorrelation indices are calculated. Both require an intensity variable in the primary file.

Moran's "I" (MoranI)

Moran's "I" statistic is the classic indicator of spatial autocorrelation. It is an index of covariation between different point locations and is similar to a product moment correlation coefficient, varying from –1 to +1. The Moran's I routine calculates 6 statistics:

1. The sample size
2. Moran's "I"
3. The spatially random (expected) "I"

4. The standard deviation of "I"
5. A significance test of "I" under the assumption of normality (Z-test)
6. A significance test of "I" under the assumption of randomization (Z-test)

Values of *I* greater than the expected *I* indicate clustering while values of *I* less than the expected *I* indicate dispersion. The significance test indicates whether these differences are greater than what would be expected by chance. The tabular output can be printed.

Adjust for small distances

If checked, small distances are adjusted so that the maximum distance weighting is 1 (see documentation for details). This ensures that *I* will not become excessively large for points that are close together. The default setting is no adjustment.

Geary's "C" (GearyC)

Geary's "C" statistic is an alternative indicator of spatial autocorrelation. It is an index of paired comparison between different point locations and varies from 0 (similar values) to 2 (dissimilar values). The Geary's C routine calculates 5 statistics:

1. The sample size
2. Geary's "C""
3. The spatial random (expected) "C"
4. The standard deviation of "C"
5. A significance test of "I" under the assumption of normality (Z-test)

Values of *C* less than the expected *C* indicate clustering while values of *C* greater than the expected *C* indicate dispersion. The significance test indicates whether these differences are greater than what would be expected by chance. The tabular output can be printed.

Adjust for small distances

If checked, small distances are adjusted so that the maximum distance weighting is 1 (see documentation for details). This ensures that *C* will not become excessively large or excessively small for points that are close together. The default setting is no adjustment.

Moran Correlogram

The Moran Correlogram calculates the Moran's "I" index (not adjusted for small distances) for different distance intervals/bins. Like The user can select any number of distance intervals. The default is 10 distance intervals.

Adjust for small distances

If checked, small distances are adjusted so that the maximum weighting is 1 (see

documentation for details.) This ensures that the I values for individual distances won't become excessively large or excessively small for points that are close together. The default value is no adjustment.

Simulation of confidence intervals

A Monte Carlo simulation can be run to estimate approximate confidence intervals around the "I" value. Specify the number of simulations to be run (e.g., 100, 1000, 10000).

Output

The output includes:

1. The sample size
2. The maximum distance
3. The bin (interval) number
4. The midpoint of the distance bin
5. The "I" value for the distance bin (I[B])

and if a simulation is run:

6. The minimum "I" value for the distance bin
7. The maximum "I" value for the distance bin
8. The 0.5 percentile for the distance bin
9. The 2.5 percentile for the distance bin
10. The 97.5 percentile for the distance bin
11. The 99.5 percentile for the distance bin.

The two pairs of percentiles (2.5 and 97.5; 0.5 and 99.5) create an approximate 5% and 1% confidence interval. The minimum and maximum "I" values create an envelope.

The tabular results can be printed, saved to a text file or saved as a '.dbf' file. For the latter, specify a file name in the "Save result to" in the dialogue box.

Graphing the 'I: values by distance

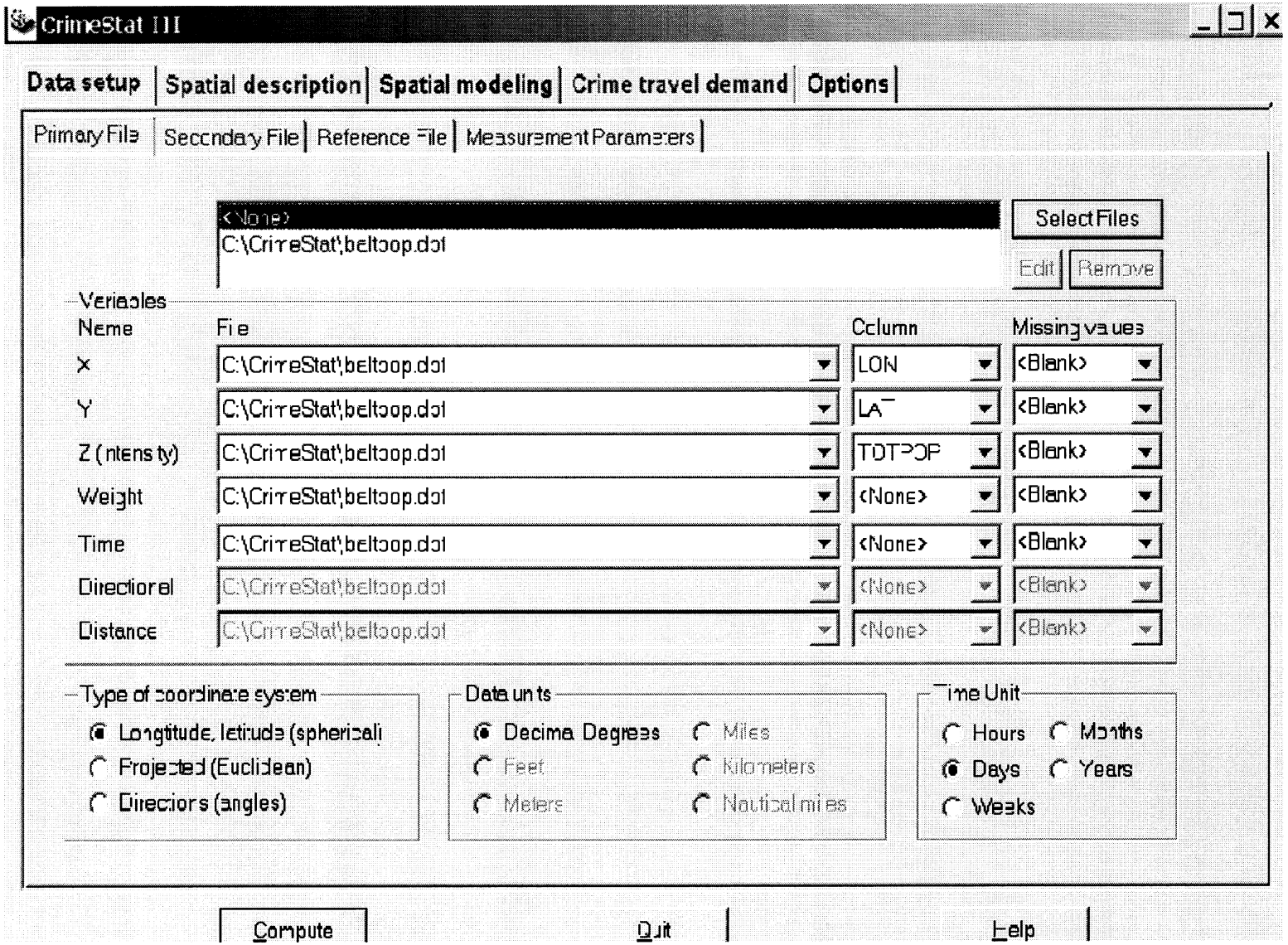
A graph is produced that shows the "I" value on the Y-axis by the distance bin on the X-axis. Click on the "Graph" button. The graph displays the reduction in spatial autocorrelation with distance. The graph is useful for selecting the type of kernel in the Single- and Duel-kernel interpolation routines when the primary variable is weighted (see Interpolation).

Distance Analysis I

Distance analysis provides statistics about the distances between point locations. It is useful for identifying the degree of clustering of points. It is sometimes called

Figure 2.6:

Distance Analysis I Screen



second-order analysis. The distance routines are divided into two pages. On the first page, there are four routines for describing properties of the distances.

Nearest Neighbor Analysis (Nna)

The nearest neighbor index provides an approximation about whether points are more clustered or dispersed than would be expected on the basis of chance. It compares the average distance of the nearest other point (nearest neighbor) with a spatially random expected distance by dividing the empirical average nearest neighbor distance by the expected random distance (the nearest neighbor index). The nearest neighbor routine requires that the geographical area be entered on the Measurement Parameters page and that direct distances be used. The *Nna* routine calculates 10 statistics:

1. The sample size
2. The mean nearest neighbor distance
3. The standard deviation of the nearest neighbor distance
4. The minimum distance
5. The maximum distance
6. The mean random distance (for both the bounding rectangle and the user input area, if provided
7. The mean dispersed distance (for both the bounding rectangle and the user input area, if provided)
8. The nearest neighbor index (for both the bounding rectangle and the user input area, if provided)
9. The standard error of the nearest neighbor index (for both the bounding rectangle and the user input area, if provided)
10. A significance test of the nearest neighbor index (Z-test)

The tabular results can be printed, saved to a text file, or saved as a 'dbf' file.

Number of nearest neighbors

The K-nearest neighbor index compares the average distance to the K^{th} nearest other point with a spatially random expected distance. The user can indicate the number of K-nearest neighbors to be calculated, if more than one are to be calculated. *CrimeStat* will calculate 3 statistics for each order specified:

1. The mean nearest neighbor distance in meters for the order
2. The expected nearest neighbor distance in meters for the order
3. The nearest neighbor index for the order

The *Nna* routine will use the user-defined area unless none is provided in which case it will use the bounding rectangle. The tabular results can be printed, saved to a text file or output as a 'dbf' file.

Linear Nearest Neighbor Analysis

The linear nearest neighbor index provides an approximation as to whether points are more clustered or dispersed along road segments than would be expected on the basis of chance. It is used with indirect (Manhattan) distances and requires the input of the total length of a road network on the measurement parameters page (see Measurement Parameters). If indirect distances are checked on the measurement parameters page, then the linear nearest neighbor will be calculated when the Nna box is checked. The linear nearest neighbor index is the ratio of the empirical average linear nearest neighbor distance to the expected linear random distance. The *Nna* routine calculates 9 statistics for the linear nearest neighbor index:

1. The sample size
2. The mean linear nearest neighbor distance in meters, feet and miles
3. The minimum distance between points along a grid network
4. The maximum distance between points along a grid network
5. The mean random linear distance
6. The linear nearest neighbor index
7. The standard deviation of the linear nearest neighbor distance in meters, feet and miles
8. The standard error of the linear nearest neighbor index
9. A t-test of the difference between the empirical and expected linear nearest neighbor distance

Number of linear nearest neighbors

Nna can calculate K-nearest linear neighbors and compare this distance the average linear distance to the K^{th} nearest other point with a spatially random expected distance. The user can indicate the number of K-nearest linear neighbors to be calculated, if more than one are to be calculated. *CrimeStat* will calculate 3 statistics for each order specified:

1. The mean linear nearest neighbor distance in meters for the order
2. The expected linear nearest neighbor distance in meters for the order
3. The linear nearest neighbor index for the order

Edge correction of nearest neighbors

The nearest neighbor analysis (either areal or linear) does not adjust for underestimation for incidents near the boundary of the study area. It is possible that there are nearest neighbors outside the boundary that are closer than the measured nearest neighbor. The nearest neighbor analysis has three edge correction options: 1) no adjustment – this is the default; 2) an adjustment that assumes the study area is a rectangle; and 3) an adjustment that assumes the study area is a circle. The rectangular and circular edge corrections adjust the nearest neighbor distances of points near the border. If a point is closer to the border (of either a rectangle or a circle) than to the measured nearest neighbor distance, then the distance to the border is taken as the

adjusted nearest neighbor distance.

Ripley's "K" Statistic (RipleyK)

Ripley's "K" statistic compares the number of points within any distance to an expected number for a spatially random distribution. The empirical count is transformed into a square root function, called L, and is adjusted for orientation (see documentation for more details). Values of L that are greater than the upper limit of the simulations indicate concentration while values of L less than the lower limit of the simulations indicate dispersion. L is calculated for each of 100 distance intervals (bins). The RipleyK routine calculates 6 statistics:

1. The sample size
2. The maximum distance
3. 100 distance bins
4. The distance for each bin
5. The transformed statistic, L(t), for each distance bin
6. The expected random L under complete spatial randomness, L(csr)

In addition, *CrimeStat* can estimate the sampling distribution by running spatially random Monte Carlo simulations over the study area. If one or more spatially random simulations are specified, there are 6 additional statistics:

7. The minimum L value for the spatially random simulations
8. The maximum L value for the spatially random simulations
9. The 0.5 percentile L value for the spatially random simulations
10. The 2.5 percentile L value for the spatially random simulations
11. The 97.5 percentile L value for the spatially random simulations
12. The 99.5 percentile L value for the spatially random simulations

The tabular results can be printed, saved to a text file, or saved as a 'dbf' file.

Edge correction of Ripley's K statistic

The default setting for the Ripley's "K" statistic does not adjust for underestimation for incidents near the boundary of the study area. However, it is possible that there are points outside the study area boundary that are closer than the search radius of the circle used to enumerate the "K" statistic. The Ripley's "K" statistic has three edge correction options: 1) no adjustment – this is the default; 2) an adjustment that assumes the study area is a rectangle; and 3) an adjustment that assumes the study area is a circle. The rectangular and circular edge corrections adjust the Ripley's "K" statistic for points near the border. If the distance of a point to the border (of either a rectangle or a circle) is smaller than to the radius of the circle used to enumerate the "K" statistics, then the point is weighted inversely proportional to the area of the search radius that is within the border.

Output intermediate results

There is a box labeled "Output intermediate results". If checked, a separate dbf file will be output that lists the intermediate calculations. The file will be called "RipleyTempOutput.dbf". There are five output fields:

1. The point number (POINT), starting at 0 (for the first point) and proceeding to N–1 (for the Nth point)
2. The search radius in meters (SEARCHRADI)
3. The count of the number of *other* points that are within the search radius (COUNT)
4. The weight assigned, calculated from equations 5.24 or 5.28 above (WEIGHT)
5. The count times the weight (CTIMESW)

Assign Primary Points to Secondary Points

This routine will assign each primary point to a secondary point and then will sum by the number of primary points assigned to each secondary point. It is useful for adding up the number of primary points that are close to each secondary point. For example, in the crime travel demand module, this routine can assign incidents to zones as the module uses zonal totals. The result is a count of primary points associated with each secondary point. It is also possible to sum different variables sequentially. For example, in the crime travel demand module, both the number of crimes originating in each zone and the number of crimes occurring in each zone are needed. This can be accomplished in two runs. First, sum the incidents defined by the origin coordinates to each zone (secondary file). Second, sum the incidents defined by the destination coordinates to each zone (also secondary file). The result would be two columns, one showing the number of origins in each secondary file zone and the second showing the number of destinations in each secondary file zone.

There are two methods for assigning the primary points to the secondary.

Nearest neighbor assignment

This routine assigns each primary point to the secondary point to which it is closest. If there are two or more secondary points that are exactly equal, the assignment goes to the first one on the list.

Point-in-polygon assignment

This routine assigns each primary point to the secondary point for which it falls within its polygon (zone). A zone (polygon) shape file must be provided and the routine checks which secondary zone each primary point falls within.

Zone file

A zonal file must be provided. This is a polygon file that defines the zones to which

the primary points are assigned. The zone file should be the same as the secondary file (see Secondary file). For each point in the primary file, the routine identifies which polygon (zone) it belongs to and then sums the number of points per polygon.

Name of assigned variable

Specify the name of the summed variable. The default name is FREQ.

Use weighting file

The primary file records can be weighted by another file. This would be useful for correcting the totals from the primary file. For example, if the primary file were robbery incidents from an arrest record, the sum of this variable (i.e. the total number of robberies) may produce a biased distribution over the secondary file zones because the primary file was not a random sample of all incidents (e.g., if it came from an arrest record where the distribution of robbery arrests is not the same as the distribution of all robbery incidents).

The secondary file or another file can be used to adjust the summed total. The weighting variable should have a field that identifies the ratio of the true to the measured count for each zone. A value of 1 indicates that the summed value for a zone is equal to the true value; hence no adjustment is needed. A value greater than 1 indicates that the summed value needs to be adjusted upward to equal the true value. A value less than 1 indicates that the summed value needs to be adjusted downward to equal the true value.

If another file is to be used for weighting, indicate whether it is the secondary file or, if another file, the name of the other file.

Name of assigned weighted variable

For a weighted sum, specify the name of the variable. The default will be ADJFREQ.

Save result to

For both routines, the output is a 'dbf' file. Define the file name. Note: be careful about using the same name as the secondary file as the saved file will have the new variable. It is best to give it a new name.

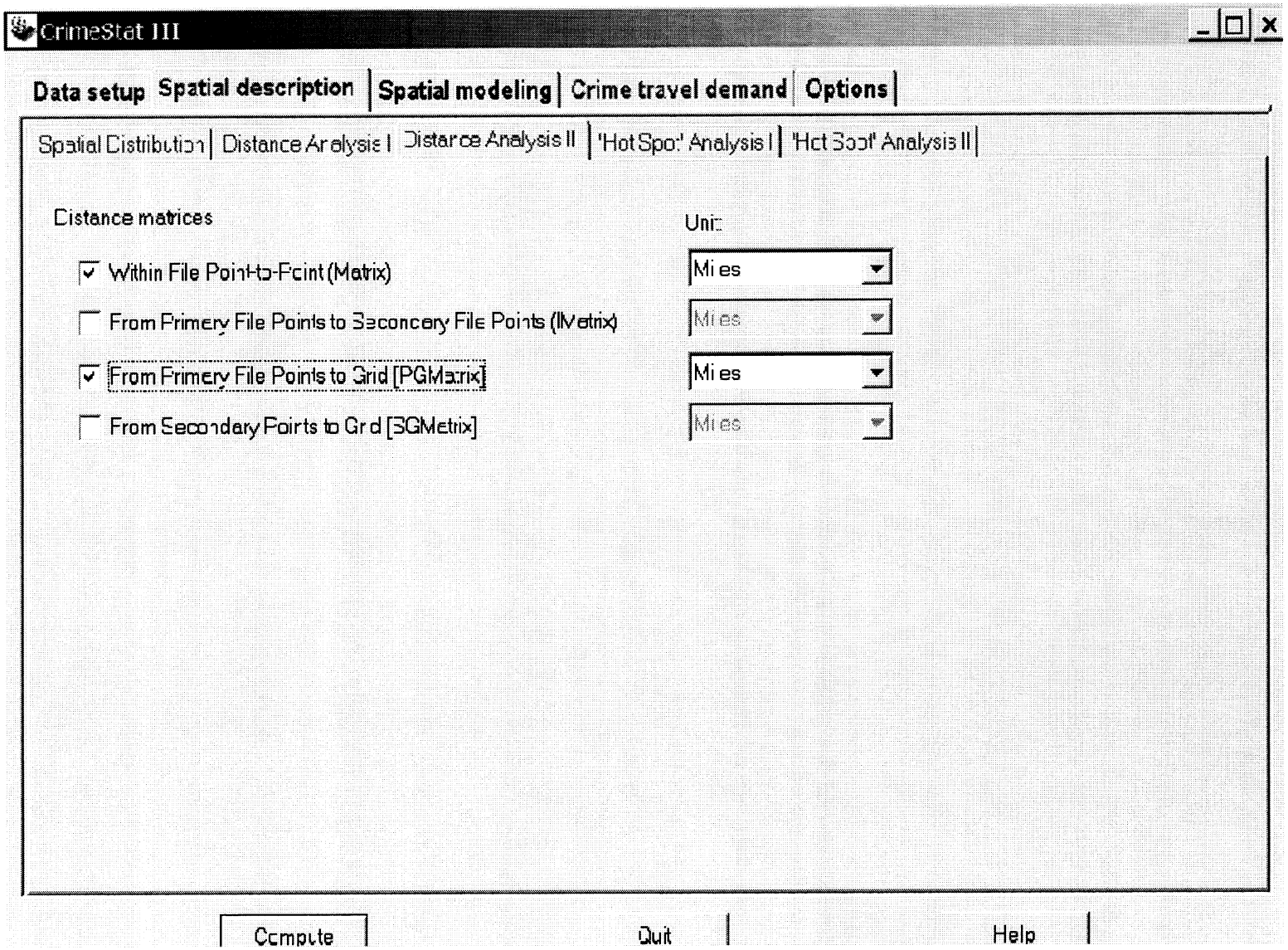
A new variable will be added to this file that gives the number of primary points in each secondary file zone and, if weighting is used, a secondary variable will be added which has the adjusted frequency.

Distance Analysis II

On the second Distance Analysis page, there are four routines that calculate distance matrices:

Figure 2-7:

Distance Analysis II Screen



Distance Matrices

1. From each primary point to every other primary point
2. From each primary point to each secondary point
3. From each primary point to the centroid of each reference file grid cell. This requires a reference file to be defined or used.
4. From each secondary point to the centroid of each reference file grid cell. This requires a reference file to be defined or used.

CrimeStat can calculate the distances between points for a single file or the distances between points for two different files. These matrices can be useful for examining the frequency of different distances or for providing distances for another program.

Within file point-to-point (Matrix)

This routine outputs the distance between each point in the primary file to every other point in a specified distance unit (miles, nautical miles, feet, kilometers, or meters.) The Matrix output can be saved as a CrimeStat distance file which can be used to speed up raw calculations (see Distance options under Data Setup). The Matrix output can also be saved to a text file.

From all primary file points to all secondary file points (IMatrix)

This routine outputs the distance between each point in the primary file to each point in the secondary file in a specified distance unit (miles, nautical miles, feet, kilometers, or meters). The IMatrix output can be saved as a CrimeStat distance file which can be used to speed up raw calculations (see Distance options under Data Setup). The IMatrix output can also be saved to a text file.

From primary points to grid (PGMatrix)

This routine outputs the distance between each point in the primary file to the centroid of each cell in the reference grid. A reference has to be defined or provided on the Reference file page. Again, the distance units must be specified (miles, nautical miles, feet, kilometers, or meters). The output can be saved as a CrimeStat distance file which can be used to speed up raw calculations (see Distance options under Data Setup). The output can be saved as a CrimeStat distance file which can be used to speed up raw calculations (see Distance options under Data Setup). The output can also be saved to a text file.

From secondary points to grid (SGMatrix)

This routine outputs the distance between each point in the secondary file to the centroid of each cell in the reference grid. A reference has to be defined or provided on the Reference file page. Again, the distance units must be specified (miles, nautical miles, feet, kilometers, or meters). The output can also be saved to a text file.

'Hot Spot' Analysis

'Hot spot' (or cluster) analysis identifies groups of incidents that are clustered together. It is a method of second-order analysis that identifies the cluster membership of points. There are a number of different 'hot spot' analysis routines in *CrimeStat*. They are organized on two program tabs: 'Hot Spot' analysis I and 'Hot Spot' analysis II.

'Hot Spot' Analysis I

The 'Hot Spot' Analysis I tab includes four different routines:

1. The mode (Mode)
2. The fuzzy mode (Fmode)
3. Nearest-neighbor hierarchical clustering (Nnh)
4. Risk-adjusted nearest-neighbor hierarchical clustering (Rnnh)

Mode

The mode calculates the frequency of incidents for each unique location, defined by an X and Y coordinate. It will output a list of all unique locations and their X and Y coordinates and the number of incidents occurring at each, ranked in decreasing order from most frequent to least frequent. It will also list their rank order from 1 to the last unique location. The data can be output to a 'dbf' file.

Fuzzy Mode

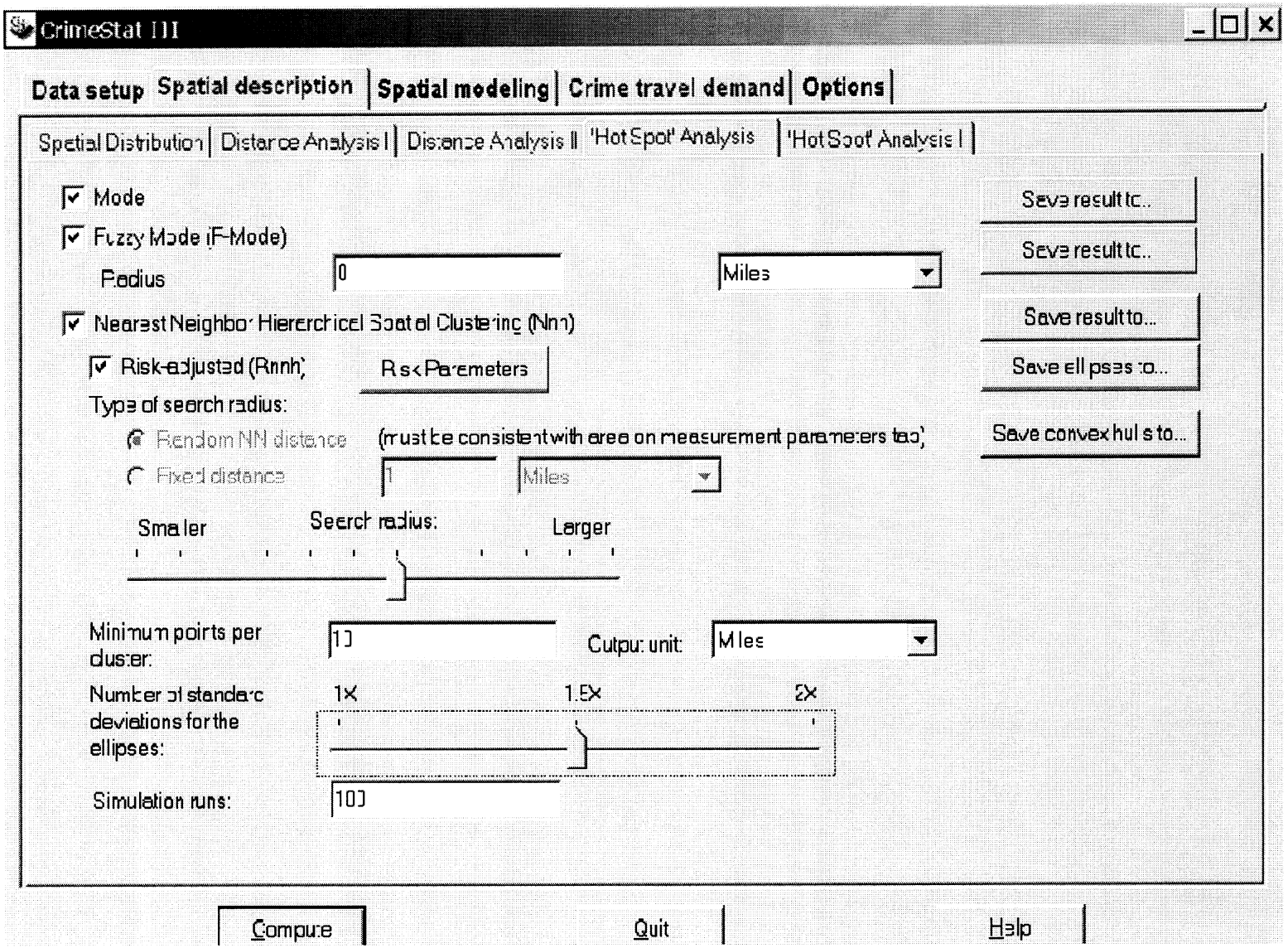
The fuzzy mode calculates the frequency of incidents for each unique location within a user-specified distance. The user must specify the search radius and the units for the radius (miles, nautical miles, feet, kilometers, meters). The routine will identify each unique location, defined by its X and Y coordinates, and will calculate the number of incidents that fall within the search radius. It will output a list of all unique locations and their X and Y coordinates and the number of incidents occurring at each within the search radius, ranked in decreasing order from most frequent to least frequent. It will also list their rank order from 1 to the last unique location. The data can be output to a 'dbf' file.

Nearest Neighbor Hierarchical Spatial Clustering (Nnh)

The nearest neighbor hierarchical spatial clustering routine is a constant-distance clustering routine that groups points together on the basis of spatial proximity. The user defines a threshold distance and the minimum number of points that are required for each cluster, and an output size for displaying the clusters with ellipses. The routine identifies first-order clusters, representing groups of points that are closer together than the threshold distance and in which there is at least the minimum number of points specified by the user. Clustering is hierarchical in that the first-order clusters are treated as separate points to be clustered into second-order clusters, and the second-order

Figure 2.8:

'Hot Spot' Analysis I Screen



clusters are treated as separate points to be clustered into third-order clusters, and so on. Higher-order clusters will be identified only if the distance between their centers are closer than the new threshold distance.

The tabular results can be printed, saved to a text file, or output as a '.dbf' file. The graphical results can be output as either ellipses or as convex hulls (or both) to ArcView '.shp', MapInfo '.mif' or Atlas*GIS '.bna' files. Separate file names must be selected for the ellipse output and for the convex hull output.

The routine outputs six results for each cluster that is calculated:

1. The hierarchical order and the cluster number
2. The mean center of the cluster (Mean X and Mean Y)
3. The standard deviational ellipse of the cluster (the rotation and the lengths of the X and Y axes
4. The number of points in the cluster
5. The area of the cluster
6. The density of the cluster (points divided by area)

Nnh threshold distance

The threshold distance is the *search radius* around a *pair* of points. For each pair of points, the routine determines whether they are closer together than the search radius. There are two ways to determine a threshold distance:

Random nearest neighbor distance

First, the search distance is chosen by the random nearest neighbor distance. The default value is 0.1 (i.e., fewer than 10% of the pairs could be expected to be as close or closer by chance.) Pairs of points that are closer together than the threshold distance are grouped together whereas pairs of points that are greater than the threshold distance are ignored. The smaller the significance level that is selected, the smaller the threshold distance and, usually, the fewer pairs will be selected. On the other hand, choosing a higher significance level, the larger the threshold distance and, usually, the more pairs will be selected. However, the higher the significance level chosen, the greater the likelihood that clusters could be chance groupings.

The slide bar is used to adjust the significance level. Move the slide bar to the left to choose a smaller threshold distance and to the right to choose a larger threshold distance.

Fixed distance

Second, a fixed distance can be selected. The default is 1 mile. In this case, the search radius uses the fixed distance and the slide bar is inoperative.

Nnh minimum number of points

The minimum number of points required for each cluster allows the user to specify a minimum number of points for each cluster. The default is 10 points. Restrictions on the number of clusters can be placed by defining a minimum number of points that are required. If there are too few points allowed, then there will be many very small clusters. By increasing the number of required points, the number of clusters will be reduced.

Ellipse output

The results can be output graphically as an ellipse to ArcView '.shp', MapInfo '.mif' or Atlas*GIS '.bna' files. The prefix will be 'NNH1' for the first-order ellipses, 'NNH2' for the second-order ellipses, and 'NNH3' for the third-order ellipses. Higher-order ellipses will only index the number.

Output size for Nnh ellipses

The cluster output size can be adjusted by the lower slide bar. This specifies the number of ellipse standard deviations to be calculated for each cluster: one standard deviation (1X - the default value), one and a half standard deviations (1.5X), or two standard deviations (2X). The default value is one standard deviation. Typically, one standard deviation will cover more than half the cases whereas two standard deviations will cover more than 99% of the cases, though the exact percentage will depend on the distribution. Slide the bar to select the number of standard deviations for the ellipses. The output file is saved as Nnh<number><root name>. The number is the order of the clustering (i.e., 1, 2...) while the root name is provided by the user.

Restrictions on the number of clusters can be placed by defining a minimum number of points that are required. The default is 10. If there are too few points allowed, then there will be many very small clusters. By increasing the number of required points, the number of clusters will be reduced.

Convex hull cluster output

The clusters can also be output as convex hulls to ArcView '.shp', MapInfo '.mif' or Atlas*GIS '.bna' files. Specify a file name. The name will be output with a 'CNNH1' prefix for the first-order clusters, a 'CNNH2' prefix for the second-order clusters, and a 'CNNH3' prefix for the third-order clusters. Higher-order clusters will index only the number.

Nnh simulation runs

A Monte Carlo simulation can be run to estimate the approximate confidence intervals around first-order Nnh clusters; second- and higher-order clusters are not simulated since their structure depends on first-order clusters. The user specifies the number of simulation runs and the Nnh clustering is calculated for randomly assigned data. The random output is sorted and percentiles are calculated. The output includes the

number of first-order clusters, the area, the number of points, and the density. Twelve percentiles are identified for these statistics:

1. The minimum for the spatially random simulations
2. The maximum for the spatially random simulations
3. The 0.5th percentile for the spatially random simulations
4. The 1st percentile for the spatially random simulations
5. The 2.5th percentile for the spatially random simulations
6. The 5th percentile for the spatially random simulations
7. The 10th percentile for the spatially random simulations
8. The 90th percentile for the spatially random simulations
9. The 95 percentile for the skpatially random simulations
10. The 97.5 percentile for the spatially random simulations
11. The 99 percentile for the spatially random simulations
12. The 99.5 percentile for the spatially random simulations

These can allow eight a one-tail or two-tail significance test. For example, for a 5% one-tail test, use the 95th percentile whereas for a 5% two-tail test, use the 2.5th and 97.5th percentiles. The simulated data that is used can be viewed by checking the 'Dump simulation data' box on the Options tab.

Risk-Adjusted Nearest Neighbor Hierarchical Spatial Clustering (Rnnh)

The risk-adjusted nearest neighbor hierarchical spatial clustering routine groups points together on the basis of spatial proximity, but the grouping is adjusted according to the distribution of a baseline variable. The routine requires both a primary file (e.g., robberies) and a secondary file (e.g., population). For the secondary variable, if an intensity or weight variable is to be used, it should be specified.

The user selects a threshold probability for grouping a pair of points together by chance and the minimum number of points that are required for each cluster, and an output size for displaying the clusters with ellipses. In addition, a kernel density model for the secondary variable must be specified. The threshold distance is determined by the threshold probability and the grid cell density produced by the kernel density estimate of the secondary variable. Thus, in areas with high density of the secondary variable, the threshold distance is smaller than in areas with low density of the secondary variable.

The routine identifies first-order clusters, representing groups of points that are closer together than the threshold distance and in which there is at least the minimum number of points specified by the user. Clustering is hierarchical in that the first-order clusters are treated as separate points to be clustered into second-order clusters, and the second-order clusters are treated as separate points to be clustered into third-order clusters, and so on. Higher-order clusters will be identified only if the distance between their centers are closer than the new threshold distance.

The tabular results can be printed, saved to a text file, or output as a '.dbf' file. The graphical results can be output as either ellipses or as convex hulls (or both) to ArcView

'.shp', MapInfo '.mif' or Atlas*GIS '.bna' files. Separate file names must be selected for the ellipse output and for the convex hull output.

The routine outputs six results for each cluster that is calculated:

1. The hierarchical order and the cluster number
2. The mean center of the cluster (Mean X and Mean Y)
3. The standard deviational ellipse of the cluster (the rotation and the lengths of the X and Y axes
4. The number of points in the cluster
5. The area of the cluster
6. The density of the cluster (points divided by area)

Rnnh threshold distance

The threshold distance is the confidence interval around a random expected distance for a pair of points. However, unlike the Nnh routine where the threshold distance is constant throughout the study area, the threshold distance for the Rnnh routine is adjusted inversely proportional to the distribution of the secondary (baseline) variable. In areas with a high density of the secondary variable, the threshold distance will be small whereas in areas with a low density of the secondary variable, the threshold distance will be large. The default threshold probability is 0.1 (i.e., fewer than 10% of the pairs could be expected to be as close or closer by chance.) Pairs of points that are closer together than the threshold distance are grouped together whereas pairs of points that are greater than the threshold distance are ignored. The smaller the significance level that is selected, the smaller the threshold distance and, usually, the fewer pairs will be selected. On the other hand, choosing a higher significance level, the larger the threshold distance and, usually, the more pairs will be selected. However, the higher the significance level chosen, the greater the likelihood that clusters could be chance groupings. Move the slide bar to the left to choose a smaller threshold distance and to the right to choose a larger threshold distance.

Rnnh risk parameters

A density estimate of the secondary variable must be calculated to adjust the threshold distance of the primary variable. This is done through kernel density estimation. The risk parameters tab defines this model. The secondary variable is automatically assumed to be the 'at risk' (baseline) variable. If an intensity or weight variable has been used in the secondary file, it should be checked. The user specifies a method of interpolation (normal, uniform, quartic, triangular, and negative exponential kernels) and the choice of bandwidth (fixed interval or adaptive interval). If an adaptive interval is used, the minimum sample size for the band width (search radius) must be specified. If a fixed interval is used, the size of the interval (radius) must be specified along with the measurement units (miles, nautical miles, feet, kilometers, meters). Finally, the units of the output density must be specified (squared miles, squared nautical miles, squared feet, squared kilometers, squared meters).

The routine overlays a 50 x 50 grid on the study area and calculates a kernel density estimate of the secondary variable. The density is then re-scaled to equal the sample size of the primary variable. For each grid cell, a cell-specific threshold distance is calculated for grouping a pair of points together by chance. The threshold probability selected by the user is applied to this cell-specific threshold distance to produce a threshold distance that corresponds to the cell-specific confidence interval. Pairs of points that are closer than the cell-specific threshold distance are selected for first-order clustering.

Rnnh minimum number of points

The minimum number of points required for each cluster allows the user to specify a minimum number of points for each cluster. The default is 10 points. Third, the output size for the clusters can be adjusted by the second slide bar. These are the number of standard deviations defined by the ellipse, from one standard deviation (the default value) to three standard deviations. Typically, one standard deviation will cover about 65% of the cases whereas three standard deviations will cover more than 99% of the cases.

Ellipse output

The results can be output graphically as an ellipse to ArcView '.shp', MapInfo '.mif' or Atlas*GIS '.bna' files. The prefix will be 'RNNH1' for the first-order ellipses, 'RNNH2' for the second-order ellipses, and 'RNNH3' for the third-order ellipses. Higher-order ellipses will only index the number.

Output size for Rnnh ellipses

The cluster output size can be adjusted by the lower slide bar. This specifies the number of ellipse standard deviations to be calculated for each cluster: one standard deviation (1X - the default value), one and a half standard deviations (1.5X), or two standard deviations (2X). The default value is one standard deviation. Typically, one standard deviation will cover more than half the cases whereas two standard deviations will cover more than 99% of the cases, though the exact percentage will depend on the distribution. Slide the bar to select the number of standard deviations for the ellipses. The output file is saved as Rnnh<number><root name>. The number is the order of the clustering (i.e., 1, 2...) while the root name is provided by the user.

Restrictions on the number of clusters can be placed by defining a minimum number of points that are required. The default is 10. If there are too few points allowed, then there will be many very small clusters. By increasing the number of required points, the number of clusters will be reduced.

Convex hull cluster output

The clusters can also be output as convex hulls to ArcView '.shp', MapInfo '.mif' or Atlas*GIS '.bna' files. Specify a file name. The name will be output with a 'CRNNH1' prefix for the first-order clusters, a 'CRNNH2' prefix for the second-order clusters, and a

'CRNNH3' prefix for the third-order clusters. Higher-order clusters will index only the number.

Rnnh simulation runs

A Monte Carlo simulation can be run to estimate the approximate confidence intervals around first-order Rnnh clusters; second- and higher-order clusters are not simulated since their structure depends on first-order clusters. The user specifies the number of simulation runs and the Rnnh clustering is calculated for randomly assigned data. The random output is sorted and percentiles are calculated. The output includes the number of first-order clusters, the area, the number of points, and the density. Twelve percentiles are identified for these statistics:

1. The minimum for the spatially random Rnnh simulations
2. The maximum for the spatially random Rnnh simulations
3. The 0.5 percentile for the spatially random Rnnh simulations
4. The 1 percentile for the spatially random Rnnh simulations
5. The 2.5 percentile for the spatially random Rnnh simulations
6. The 5 percentile for the spatially random Rnnh simulations
7. The 10 percentile for the spatially random Rnnh simulations
8. The 90 percentile for the spatially random Rnnh simulations
9. The 95 percentile for the spatially random Rnnh simulations
10. The 97.5 percentile for the spatially random Rnnh simulations
11. The 99 percentile for the spatially random Rnnh simulations
12. The 99.5 percentile for the spatially random Rnnh simulations

The simulated data that is used can be viewed by checking the 'Dump simulation data' box on the Options tab.

'Hot Spot' Analysis II

The 'Hot Spot' Analysis II tab includes three different routines:

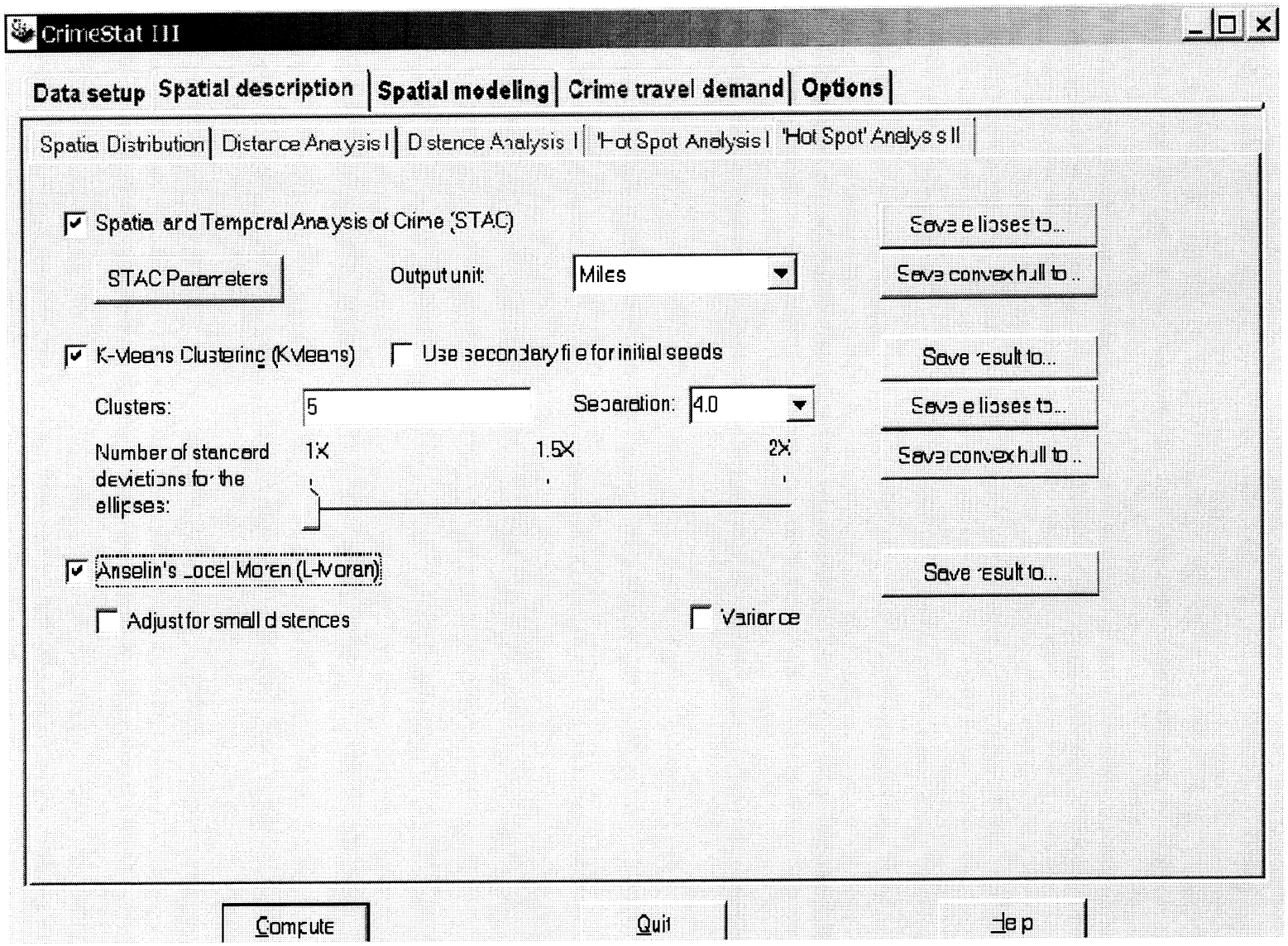
1. The Spatial and Temporal Analysis of Crime module (STAC)
2. K-Means clustering
3. Anselin's local Moran statistics

Spatial and Temporal Analysis of Crime (STAC)

The Spatial and Temporal Analysis of Crime (STAC) routine is a variable-distance clustering routine. It initially groups points together on the basis of a constant search radius, but then combines clusters that overlap. On the STAC Parameters tab, define a search radius, the minimum number of points that are required for each cluster, and an output size for displaying the clusters with ellipses. The results can be printed, saved to a text file, output as a '.dbf' file, or output as ellipses or as convex hulls (or both) to *ArcView* '.shp', *MapInfo* '.mif' or *Atlas*GIS* '.bna' files

Figure 2.9

'Hot Spot' Analysis II Screen



The routine outputs six results for each cluster that is calculated:

1. The hierarchical order and the cluster number
2. The mean center of the cluster (Mean X and Mean Y)
3. The standard deviational ellipse of the cluster (the rotation and the lengths of the X and Y axes
4. The number of points in the cluster
5. The area of the ellipse
6. The density of the ellipse (ellipse points divided by area)

STAC parameters

The STAC parameters tab allows the selection of a search radius, the minimum number of points, the scan type, the boundary definition, the number of simulation runs, and the output size of the STAC ellipses.

STAC search radius

The search radius is the distance within the STAC routine searches. The default is 0.5 miles. A 20 x 20 grid is overlaid on the study area. At each intersection of a row and a column, the routine counts all points that are closer than the search radius. Overlapping circles are combined to form variable-size clusters. The smaller the search radius that is selected, the fewer points will be selected. On the other hand, choosing a larger search area, the more points will be selected. However, the larger the search area, the greater the likelihood that clusters could be chance groupings. On the STAC Parameters tab, type the search radius into the box and indicate the measurement units (miles, nautical miles, feet, kilometers, meters).

STAC scan type

The scan type is the type of grid overlaid on the study area. There are two choices: rectangular (default) and triangular.

STAC boundary

The study area boundaries can be defined from the data set or the reference grid.

STAC minimum number of points

The minimum number of points required for each cluster allows the user to specify a minimum number of points for each cluster. The default is 5 points. If there are too few points allowed, then there will be many very small clusters. By increasing the number of required points, the number of clusters will be reduced. On the STAC Parameters tab, type the minimum number of points each cluster is required to have.

Output size for STAC ellipses

The cluster output size of the ellipses can be adjusted by the lower slide bar. The routine will output one standard deviation (1X), one and half standard deviations (1.5X), and two standard deviational (2X) ellipses. Typically, if the data are randomly distributed, one standard deviation will cover about 50% of the cases whereas two standard deviations will cover more than 99% of the cases; however, the actual percentages may differ.

On the STAC Parameters tab, Slide the bar to select the number of standard deviations for the ellipses. The output file is saved as ST<*root name*>. The root name is provided by the user.

Convex hull cluster output

The clusters can also be output as convex hulls to ArcView '.shp', MapInfo '.mif' or Atlas*GIS '.bna' files. Specify a file name. The name will be output with a 'CST' prefix.

STAC simulation runs

A Monte Carlo simulation can be run to estimate the approximate confidence intervals around the STAC clusters. The user specifies the number of simulation runs and the STAC clustering is calculated for randomly assigned data. The random output is sorted and percentiles are calculated. The output includes the number of clusters, the area, the number of points, and the density. Twelve percentiles are identified for these statistics:

1. The minimum for the spatially random simulations
2. The maximum for the spatially random simulations
3. The 0.5th percentile for the spatially random simulations
4. The 1st percentile for the spatially random simulations
5. The 2.5th percentile for the spatially random simulations
6. The 5th percentile for the spatially random simulations
7. The 10th percentile for the spatially random simulations
8. The 90th percentile for the spatially random simulations
9. The 95th percentile for the spatially random simulations
10. The 97.5th percentile for the spatially random simulations
11. The 99th percentile for the spatially random simulations
12. The 99.5th percentile for the spatially random simulations

These can allow eight a one-tail or two-tail significance test. For example, for a 5% one-tail test, use the 95th percentile whereas for a 5% two-tail test, use the 2.5th and 97.5th percentiles. The simulated data that is used can be viewed by checking the 'Dump simulation data' box on the Options tab.

K-means Clustering (KMeans)

The K-means clustering routine is a procedure for partitioning all the points into K groups in which K is a number assigned by the user. The routine finds K seed locations in which points are assigned to the nearest cluster. The default K is 5. If K is small, the clusters will typically cover larger areas.

The tabular results can be printed, saved to a text file, or output as a '.dbf' file. The graphical results can be output as either ellipses or as convex hulls (or both) to ArcView '.shp', MapInfo '.mif' or Atlas*GIS '.bna' files. Separate file names must be selected for the ellipse output and for the convex hull output.

Initial cluster locations

The routine starts with an initial guess (seed) for the K locations and then conducts local optimization. The user can modify the location of the initial clusters in two ways:

1. The separation between the initial clusters can be increased or decreased. There is a separation scale with pre-defined values from 1 to 10; the default is 4. The user can type in any number, however (e.g., 15). Increasing the number increases the separation between the initial cluster locations while decreasing the number decreases the separation.

2. The user can also define the initial locations and the number of clusters, K, with a secondary file. The routine takes K from the number of points in the secondary file and takes the X/Y coordinates of the points as the initial seed locations.

Output size for K-means ellipses

For both methods, the cluster output size of the ellipses can be adjusted by the lower slide bar. The routine will output one standard deviation (1X), one and half standard deviations (1.5X), and two standard deviational (2X) ellipses. Typically, if the data are randomly distributed, one standard deviation will cover about 50% of the cases whereas two standard deviations will cover more than 99% of the cases; however, the actual percentages may differ.

Slide the bar to select the number of standard deviations for the ellipses. The output file is saved as KM<*root name*>. The root name is provided by the user.

Convex hull cluster output

The clusters can also be output as convex hulls to ArcView '.shp', MapInfo '.mif' or Atlas*GIS '.bna' files. Specify a file name. The name will be output with a 'CKM' prefix.

Anselin's Local Moran (L-Moran)

Anselin's Local Moran statistic applies the Moran's "I" statistic to individual points (or zones) to assess whether particular points/zones are spatially related to the nearby points (or zones.) The statistic requires an intensity variable in the primary file. Unlike the global Moran's "I" statistic, the local Moran is applied to each individual point/zone. The index points to clustering or dispersion relative to the local neighborhood. Points (or zones) with high "I" values have an intensity value that is higher than their neighbors while points with low "I" values have intensity values lower than their neighbors. The output can be printed or output as a '.dbf' file.

Adjust for small distances

If checked, small distances are adjusted so that the maximum weighting is no higher than 1 (see documentation for details.) This ensures that the local "I" won't become excessively large for points that are grouped together. This is the default setting.

III. Spatial Modeling

The spatial modeling section conducts kernel density estimation, journey to crime calibration and estimation, and space-time analysis analysis.

Interpolation

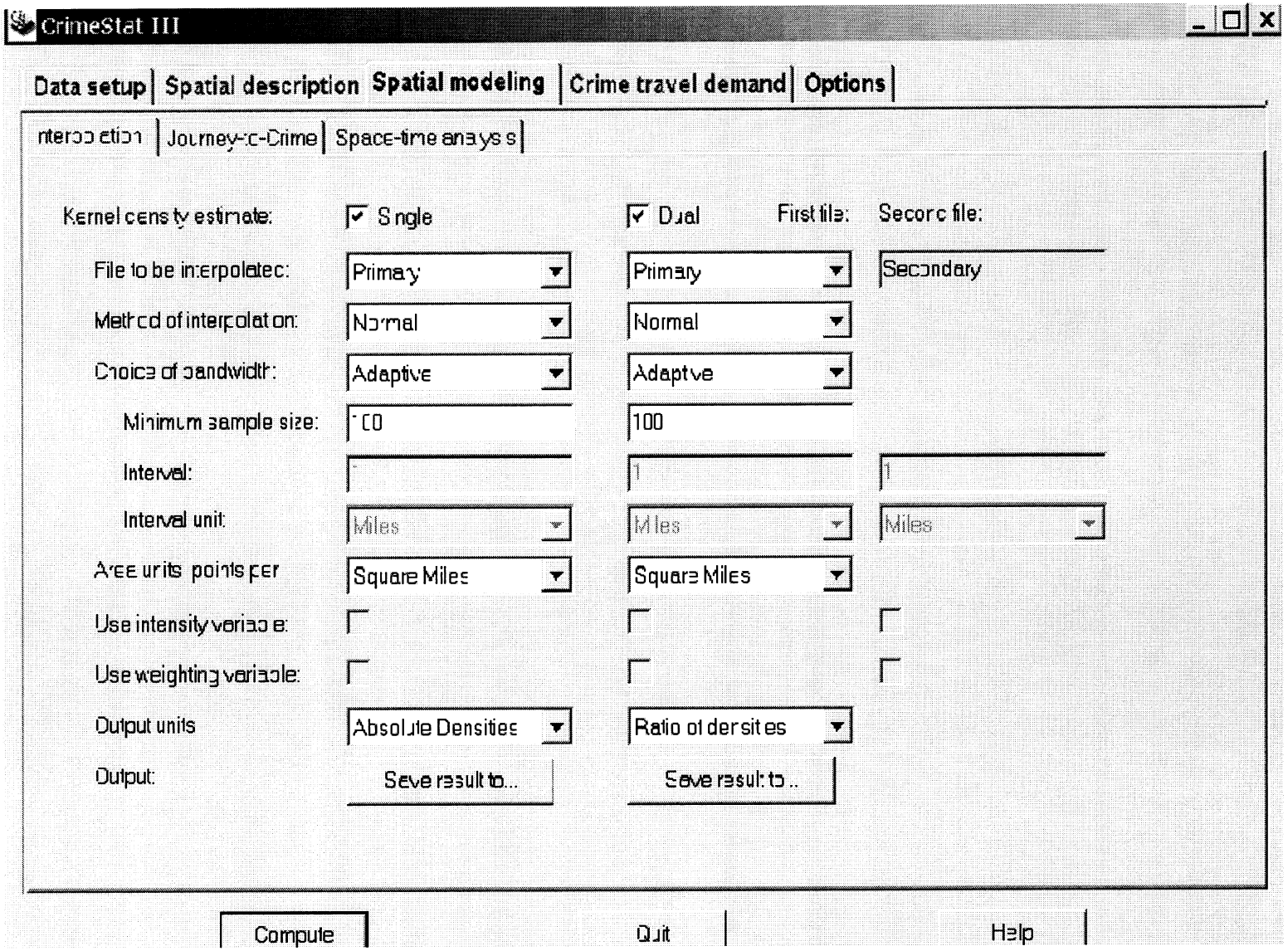
The interpolation tab allows estimates of point density using the kernel density smoothing method. There are two types of kernel density smoothing: one applied to a single distribution of points and the other applied to two different distributions. Each type has variations on the method that can be selected. Both types require a reference file that is overlaid on the study area (see Reference File). The kernels are placed over each point and the distance between each reference cell and each point is evaluated by the kernel function. The individual kernel estimates for each cell are summed to produce an overall estimate of density for that cell. The intensity and weighting variables can be used in the kernel estimate. The densities can be converted into probabilities.

Single Kernel Density Estimate (KernelDensity)

The single kernel density routine estimates the density of points for a single distribution by overlaying a symmetrical surface over each point, evaluating the distance from the point to each reference cell by the kernel function, and summing the evaluations at each reference cell.

Figure 2.10:

Interpolation Screen



File to be interpolated

The estimate can be applied to either the primary file (see Primary file) or a secondary file (see Secondary File). Select which file is to be interpolated. The default is the Primary.

Method of interpolation

There are five types of kernel distributions that can be used to estimate the density of the points. Four of the five distributions overlay a circle around each grid cell and assign weights to the points within the grid cell. The five types vary in the weights they assign to nearby points:

Kernel that assigns weights for entire study area

1. The **normal** kernel overlays a normal distribution over each point, which then extends over the entire study area defined by the reference file. This is the default kernel function. The distribution extends in all directions and is limited only by the study area.

Kernels that assign weights within a specific circle

2. The **uniform** kernel weights all points within the circle equally.

3. The **quartic** kernel overlays an inverted bell-shape surface that extends only for a limited distance from each point; the weights for points within the circle decline with distance, but gradually.

4. The **triangulated** (or conical) kernel overlays a cone over each grid cell; the weights for points within the circle decrease consistently with distance.

5. Finally, the **negative exponential** (or peaked) kernel overlays a sharply-decreasing function over the grid cell; the weights for points within the circle decrease very rapidly with distance. The five methods produce similar results although the normal is generally smoother for any given bandwidth.

Choice of bandwidth

The kernels are applied to a limited search distance, called 'bandwidth'. For the normal kernel, bandwidth is the standard deviation of the normal distribution. For the uniform, quartic, triangular and negative exponential kernels, bandwidth is the radius of a circle for the search distance. For all types, larger bandwidth will produce smoother density estimates and both adaptive and fixed bandwidth intervals can be selected.

Adaptive bandwidth

An adaptive bandwidth distance is identified by the minimum number of other points found within a circle drawn around a single point. A circle is placed around each point, in turn, and the radius is increased until the minimum sample size is reached. Thus, each point has a different bandwidth interval. This is the default bandwidth setting. The user can modify the minimum sample size. The default is 100 points.

Fixed bandwidth

A fixed bandwidth distance is a fixed interval for each point. The user must define the interval and the distance units by which it is calculated (miles, nautical miles, feet, kilometers, meters).

Output units

Specify the density units as points per square mile, per squared nautical miles, per square feet, per square kilometers, or per square meters. The default is points per square mile.

Use intensity variable

If an intensity variable is interpolated, then this box should be checked.

Use weighting variable

If a weighting variable is used in the interpolation, then this box should be checked.

Calculate densities or probabilities

The density estimate for each cell can be calculated in one of three ways:

1. **Absolute densities**. This is the number of points per grid cell and is scaled so that the sum of all grid cells equals the sample size. This is the default.

2. **Relative densities**. For each grid cell, this is the absolute density divided by the grid cell area and is expressed in the output units (e.g., points per square mile).
3. **Probabilities**. This is the proportion of all incidents that occur in the grid cell. The sum of all grid cells equals a probability of 1.

Select whether absolute densities, relative densities, or probabilities are to be output for each cell. The default is absolute densities.

Output

The results can be output as a *Surfer for Windows* file (for both an external or created reference file) or as an *ArcView* 'shp', *MapInfo* 'mif', *Atlas*GIS* 'bna', or *ArcView Spatial Analyst*© file (only if the reference file is created by *CrimeStat*).

Dual Kernel Density Estimate (DuelKernel)

The dual kernel density routine compares two different distributions involving the primary and secondary files. A 'first' file and 'second' file need to be defined. The comparison allows the ratio of the first file divided by the second file, the logarithm of the ratio of the first file divided by the second file, the difference between the first file and second file (i.e., first file – second file), or the sum of the first file and the second file.

File to be interpolated

Identify which file is to be the 'first file' (primary or secondary) and which is to be the 'second file (primary or secondary). The default is Primary for the first file and Secondary for the second file.

Method of interpolation

There are five types of kernel distributions that can be used to estimate the density points. Four of the five overlay a circle around each grid cell and assign weights to the points within the grid cell. The five types vary in the weights they assign to nearby points:

Kernel that assigns weights for entire study area

1. The **normal** kernel overlays a normal distribution over each point, which then extends over the entire study area defined by the reference file. This is the default kernel function. The distribution extends in all directions and is limited only by the study area.

Kernels that assign weights within a specific circle

2. The **uniform** kernel weights all points within the circle equally.

3. The **quartic** kernel overlays an inverted bell-shape surface that extends only for a limited distance from each point; the weights for points within the circle decline with distance, but gradually.

4. The **triangulated** (or conical) kernel overlays a cone over each grid cell; the weights for points within the circle decrease consistently with distance.

5. Finally, the **negative exponential** (or peaked) kernel overlays a sharply-decreasing function over the grid cell; the weights for points within the circle

decrease very rapidly with distance. The five methods produce similar results although the normal is generally smoother for any given bandwidth.

Choice of bandwidth

The kernels are applied to a limited search distance, called 'bandwidth'. For the normal kernel, bandwidth is the standard deviation of the normal distribution. For the uniform, quartic, triangular and negative exponential kernels, bandwidth is the radius of a circle for the search area. For all types, larger bandwidth will produce smoother density estimates and both adaptive and fixed bandwidth intervals can be selected.

Adaptive bandwidth

An adaptive bandwidth distance is identified by the minimum number of other points found within a circle drawn around a single point. A circle is placed around each point, in turn, and the radius is increased until the minimum sample size is reached. Thus, each point has a different bandwidth interval. This is the default bandwidth setting. The user can modify the minimum sample size. The default is 100 points.

Fixed bandwidth

A fixed bandwidth distance is a fixed interval for each point. The user must define the interval and the distance units by which it is calculated (miles, nautical miles, feet, kilometers, meters). The default is one mile.

Variable bandwidth

A variable bandwidth allows separate fixed intervals for both the first and second files. For each, the user must define the interval and the distance units by which it is calculated (miles, nautical miles, feet, kilometers, meters). The default is one mile for both the first and second files.

Output units

Specify the density units as points per square mile, per square nautical miles, per square feet, per square kilometers, or per square meters. The default is points per square mile.

Use intensity variable

For the first and second files separately, check the appropriate box if an intensity variable is interpolated.

Use weighting variable

For the first and second files separately, check the appropriate box if a weighting variable is used in the interpolation.

Calculate densities or probabilities

The density estimate for each cell can be calculated in one of six ways:

1. **Ratio of densities**. This is the ratio of the density for the first file divided by the density of the second file

2. **Log ratio of densities**. This is the natural logarithm of the ratio of the density for the first file divided by the density of the second file.

3. **Absolute difference in densities**. This is the difference between the absolute density of the first file and the absolute density of the second file. It is the *net* difference. The densities of each file are scaled so that the sum of the grid cells equals the sample size.

4. **Relative difference in densities**. This is the difference between the relative density of the first file and the relative density of the second file. It is the *relative* difference. The cell densities of each file are divided by the grid cell area to produce a measure of relative density in the specified output units (e.g., points per square mile). The relative density of the second file is then subtracted from the relative density of the first file.

5. **Absolute sum of densities**. This is the sum of the absolute density of the first file and the absolute density of the second file. The densities of each file are scaled so that the sum of the grid cells equals the sample size.

6. **Relative sum of densities**. This is the sum of the relative density of the first file and the relative density of the second file. It is the *relative* sumThe cell densities of each file are divided by the grid cell area to produce a measure of relative density in the specified output units (e.g., points per square mile). The relative density of the second file is then added to the relative density of the first file.

Select whether the ratio of densities, the log ratio of densities, the absolute difference in densities, the relative difference in densities, the absolute sum of densities, or the relative sum of densities are to be output for each cell. The default is the ratio of densities.

Output

The results can be output as a *Surfer for Windows* file (for both an external or created reference file) or as an *ArcView* 'shp', *MapInfo* 'mif', *Atlas*GIS* 'bna', or *ArcView Spatial Analyst* only if the reference file is created by *CrimeStat*).

Space-Time Analysis

The space-time analysis tab allows the analysis of the interaction between space and time. There are three routines. First, there is the Knox index that shows the simple binomial relationship between events occurring in space and in time. Second, there is the Mantel index that shows the correlation between closeness in space and closeness in time. Third, there is a spatial-temporal moving average that calculates a mean center for a temporal span. Fourth, there is a Correlated Walk Analysis that diagnoses the spatial and temporal sequencing of incidents committed by a serial offender.

For each of these routines, times **must** be defined by an integer or real variable, and **not** by a formatted date. For example, 3 days, 2.1 weeks, 4.3 months, or the number of days from January 1, 1900 (e.g., 37174) are all eligible time values. 'November 1, 2001', '07/30/01' or '19th October, 2001' are not eligible values. Convert all formatted dates into a real number. Time units must be consistent across all observations (i.e., all values are hours or days or weeks or months or years, but not two or more these units). If these conditions are violated, *CrimeStat* will calculate results, but they won't be correct.

Knox Index

The Knox index is an index showing the relationship between 'closeness in time' and 'closeness in distance'. Pairs of events are compared in distance and in time and are represented as a 2 x 2 table. If there is a relationship, it would normally be positive, that is events that are close together in space (i.e., in distance) are also occurring in a short time span. There are three methods for defining closeness in time or in distance:

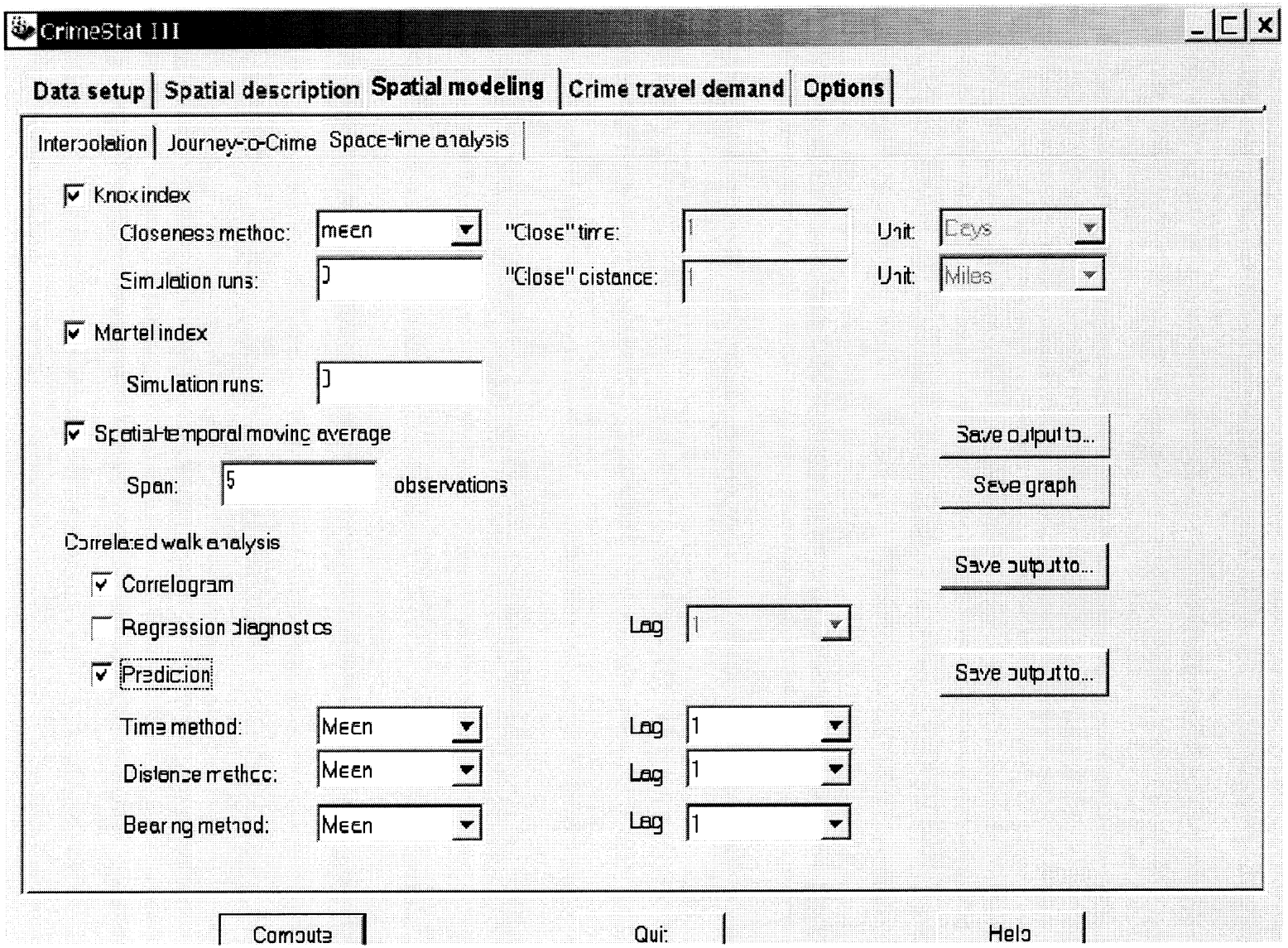
1. **Mean**. That is, events that are closer together than the mean time interval or are closer together than the mean distance are defined as 'Close' whereas events that are father together than mean time interval or are farther together than the mean distance are defined as 'Not close'.

2. **Median**. That is, events that are closer together than the median time interval or are closer together than the median distance are defined as 'Close' whereas events that are father together than median time interval or are farther together than the median distance are defined as 'Not close'.

3. **User defined**. The user can specify any value for distinguishing 'Close' and 'Not close' for either time or distance.

Figure 2.11:

Space-Time Analysis Screen



The output includes a 2 x 2 table of the distribution of pairs categorized as 'Close' or 'Not close' in time and in distance. Since pairs of events are being compared, there are N*(N-1)/2 pairs in a data set where N is the number of events. The output also includes a table of the expected of the distribution of pairs on the assumption that events in time are space are independent of each other. The output includes a Chi-square statistic. Since the observations are not independent, the usual p-values associated with Chi-square tests do not apply.

Knox simulation runs

A Monte Carlo simulation can be run to estimate approximate Type I error probability levels for the Knox Index. The user specifies the number of simulation runs. Data are randomly assigned and the chi-square value for the Knox index is calculated for each run. The random output is sorted and percentiles are calculated. Twelve percentiles are identified for these statistics:

1. The minimum for the spatially random simulations
2. The maximum for the spatially random simulations
3. The 0.5th percentile for the spatially random simulations
4. The 1st percentile for the spatially random simulations
5. The 2.5th percentile for the spatially random simulations
6. The 5th percentile for the spatially random simulations
7. The 10th percentile for the spatially random simulations
8. The 90th percentile for the spatially random simulations
9. The 95th percentile for the spatially random simulations
10. The 97.5th percentile for the spatially random simulations
11. The 99th percentile for the spatially random simulations
12. The 99.5th percentile for the spatially random simulations

These can allow eight a one-tail or two-tail significance test. For example, for a 5% one-tail test, use the 95th percentile whereas for a 5% two-tail test, use the 2.5th and 97.5th percentiles. The simulated data that is used can be viewed by checking the 'Dump simulation data' box on the Options tab.

Mantel Index

The Mantel index is the correlation between closeness in time and closeness in distance across pairs. Each pair of events is compared for the time interval and the distance between them. If there is a positive relationship between closeness in time and closeness in space (distance), then there should be a sizeable positive correlation between the two measures. Note, that since pairs of events are being compared, there are N*(N-1)/2 pairs in the data set where N is the number of events.

Mantel simulation runs

A Monte Carlo simulation can be run to estimate the approximate confidence intervals around the Mantel correlation. The user specifies the number of simulation runs and the Mantel index is calculated for randomly assigned data. The random output is sorted and percentiles are calculated. Twelve percentiles are identified for these statistics:

1. The minimum for the spatially random simulations
2. The maximum for the spatially random simulations
3. The 0.5th percentile for the spatially random simulations
4. The 1st percentile for the spatially random simulations
5. The 2.5th percentile for the spatially random simulations
6. The 5th percentile for the spatially random simulations
7. The 10th percentile for the spatially random simulations
8. The 90th percentile for the spatially random simulations
9. The 95th percentile for the spatially random simulations
10. The 97.5th percentile for the spatially random simulations
11. The 99th percentile for the spatially random simulations
12. The 99.5th percentile for the spatially random simulations

These can allow eight a one-tail or two-tail significance test. For example, for a 5% one-tail test, use the 95th percentile whereas for a 5% two-tail test, use the 2.5th and 97.5th percentiles. The simulated data that is used can be viewed by checking the 'Dump simulation data' box on the Options tab.

Spatial-Temporal Moving Average

This routine calculates the mean center as it changes over the sequence of the events. The routine sorts the incidents in the order in which they occur. The user defines a span of sequential incidents; the default is five observations. The routine places a window covering the span over the incidents and calculates the mean center (the mean X coordinate and the mean Y coordinate). It then moves the window one observation. Approximations are made at the beginning and end observations for the sequence. The result is a set of mean centers ordered from the first through last observations. This statistic is useful for identifying whether the central location for a set of incidents (perhaps committed by a serial offender) has moved over time.

There are four outputs for this routine:

1. The sample size
2. The number of observations making up the span
3. The span number
4. The X and Y coordinates for each span window.

The tabular results are output as a dBase 'dbf', Microsoft Access 'mdb', Ascii 'dat' or ODBC-compliant file. A line showing the sequential output is output as an ArcView '.shp',

MapInfo '.mif' or Atlas*GIS '.bna' file.

Correlated Walk Analysis

Correlated Walk Analysis (CWA) analyzes the sequential movements of a serial offender and makes predictions about the time and location of the next event. Sequential movements are analyzed in terms of three parameters: ***time difference*** between events (e.g., the number of days between two consecutive events), ***distance between events*** – the distance between two consecutive events, and ***bearing (direction) between events*** – the angular direction between two consecutive events in degrees (from 0 to 360).

There are three CWA routines for analyzing sequential events:

1. Correlogram
2. Regression diagnostics
3. Prediction

CWA - Correlogram

The correlogram presents the lagged correlations between events for time difference, distance, and bearing (direction). The lags are the sequential comparisons. A lag of 0 is the sequence compared with itself; by definition, the correlation is 1.0. A lag of 1 is the sequence compared with the previous sequence. A lag of 2 is the sequence compared with two previous sequences. A lag of 3 is the sequence compared with three previous sequences, and so forth. In total, comparisons are made up to seven previous sequences (a lag of 7).

Typically, for time difference, distance and location separately, the lag with the highest correlation is the strongest. However, with each consecutive lag, the sample size decreases by one and a high correlation associated with a high lag comparison can be unreliable if the sample size is small. Consequently, the *adjusted correlogram* discounts the correlations by the number of lags.

CWA- Regression diagnostics

The regression diagnostics presents the regression statistics for different lag models. The lag must be specified; the default is a lag of 1 (the sequential events compared with the previous events). Three regression models are run for time difference, direction, and bearing. The output includes statistics for:

1. The sample size
2. The distance and time units
3. The lag of the model (from 1 to 7)
4. The multiple R (correlation) between the lags
5. The squared multiple R (i.e., R-squared)
6. The standard error of estimate for the regression

7. The coefficient, standard error, t-value, and probability value (two-tail) for the constant.
8. The coefficient, standard error, t-value, and probability value (two-tail) for the coefficient.
9. The analysis of variance for the regression model, including the sum-of-squares and the mean-square error for the regression model and the residual (error), the F-test of the regression mean-square error divided by the residual mean-square error, and the probability level for the F-test.

In general, the model with the lowest standard error of estimate (and, consequently, highest multiple R) is best. However, with a small sample size, the model can be unreliable. Further, with each consecutive lag, the sample size decreases by one and a high multiple R associated with a high lag comparison can be unreliable if the sample size is small.

CWA- Prediction

The prediction routine allows the prediction of a next event, in time, distance, and direction. For each parameter – time difference, distance, and bearing, there are three models that can be used:

1. The mean difference (i.e., mean time difference, mean distance, mean bearing)
2. The median difference (i.e., median time difference, median distance, median bearing)
3. The regression model (i.e., the estimated regression coefficient and intercept)

For each of these, a different lag comparison can be used, from 1 to 7. The lag defines the sequence from which the prediction is made. Thus, for a lag of 1, the interval from the next-to-last to the last event is used as a reference (i.e., between events N-1 and N); for a lag of 2, the interval from the third-to-last to the next-to-last event is used as a reference (i.e.,between events N-2 and N-1); and so forth. The particular model selected is then added to the reference sequence. Note: if the regression model is used, the lag for distance and bearing must be the same.

Example 1: with a lag of 1 and the use of the mean difference, the mean time difference is added to the time of the last event, the mean distance is added to the location of the last event, and the mean bearing is added to the location of the last event.

Example 2: with a lag of 2 and the use of the regression model, the predicted time difference is added to the time of the next-to-last event; the predicted distance is added to the location of the next-to-last event and the prediction bearing is added to the location of the last event.

Example 3: with a lag of 1 for time and the use of the regression model, a lag of 2 for distance and the use of the mean distance, and a lag of 3 for bearing and the use of the median bearing, the predicted time difference is added to the last event, the mean distance

is added to the location of the next-to-last event, and the median bearing is added to the location of the third-from-last event.

The output includes:

1. The method used for time, distance, and bearing
2. The lag used for time, distance, and bearing
3. The predicted time difference
4. The predicted distance
5. The predicted bearing
6. The final predicted time
7. The X-coordinate of the final predicted location
8. The Y-coordinate of the final predicted location

Journey to Crime Analysis

The journey to crime (Jtc) routine estimates the likelihood that a serial offender lives at any location within the study area. Both a primary file and a reference file are required. The locations of the serial crimes are defined in the primary file while all locations within the study area are identified in the reference file. The Jtc routine can use two different travel distance functions: 1) An already-calibrated distance function; and 2) A mathematical formula. Either direct or indirect (Manhattan) distances can be used though the default is direct (see Measurement parameters).

Calibrate Journey to Crime Function

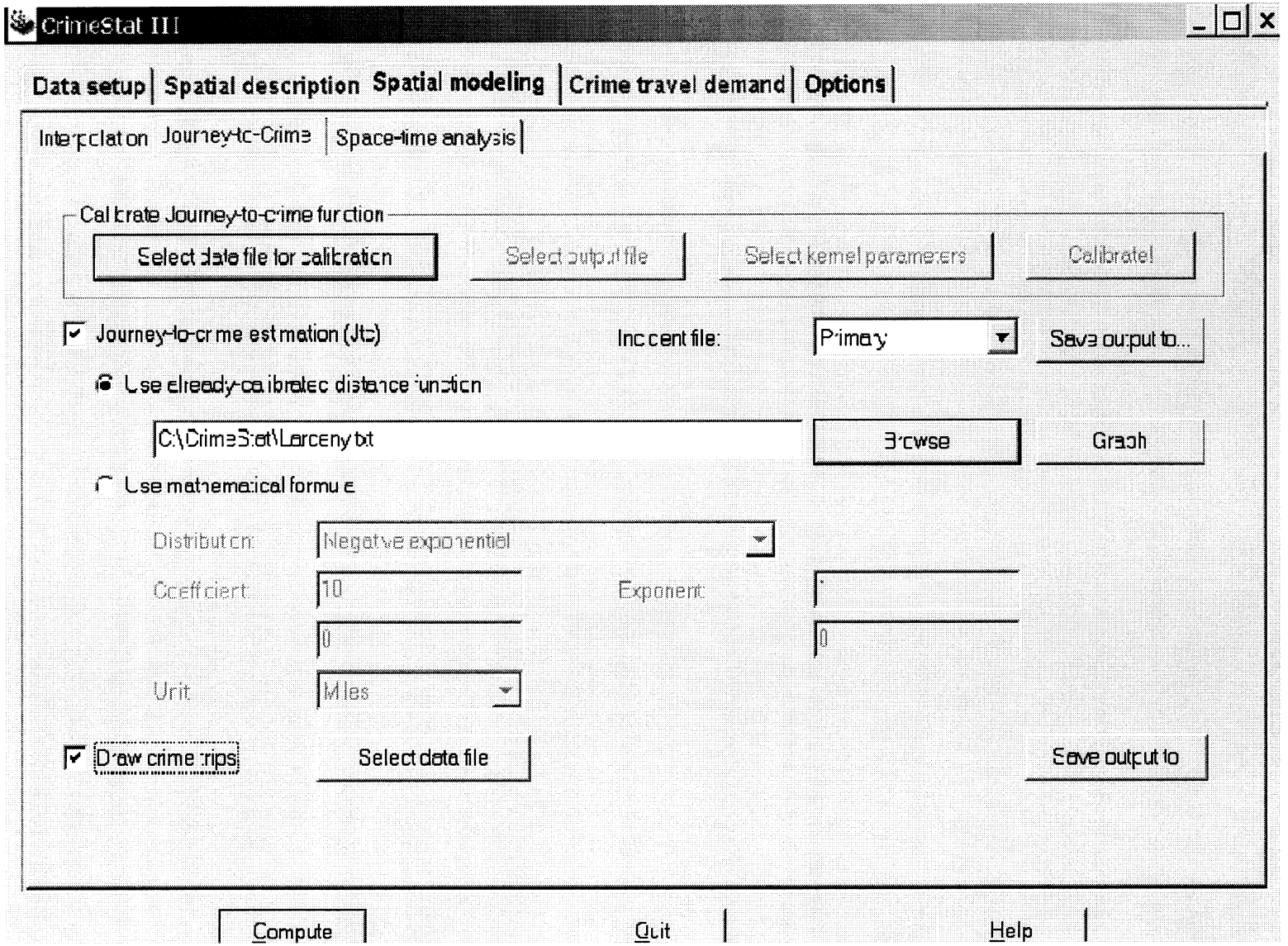
This routine calibrates a journey to crime distance function for use in the journey to crime estimation routine. A file is input which has a set of incidents (records) which includes both the X and Y coordinates for the location of the offender's residence (origin) and the X and Y coordinates for the location of the incident that the offender committed (destination). The routine estimates a travel distance function using a one-dimensional kernel density method. For each record, the distance between the origin location and the destination location is calculated and is represented on a distance scale. The maximum distance is calculated and divided into a number of intervals; the default is 100 equal sized intervals, but the user can modify this. For each distance (point) calculated, a one-dimensional kernel is overlaid. For each distance interval, the values of all kernels are summed to produce a smooth function of journey to crime distance. The results are saved to a file that can be used in the journey to crime estimation routine.

Select data file for calibration

Select the file that has the X and Y coordinates for the origin and destination locations. *CrimeStat* can read ASCII, dbase 'dbf', ArcView 'shp', MapInfo 'dat' files, and files that follow the ODBC standard interface. Select the tab and indicate the type of file to be selected. Use the browse button to search for the file. If the file type is ASCII, select the type of data separator (comma, semicolon, space, tab) and the number of columns.

Figure 2.12:

Journey-to-Crime Screen



is added to the location of the next-to-last event, and the median bearing is added to the location of the third-from-last event.

The output includes:

1. The method used for time, distance, and bearing
2. The lag used for time, distance, and bearing
3. The predicted time difference
4. The predicted distance
5. The predicted bearing
6. The final predicted time
7. The X-coordinate of the final predicted location
8. The Y-coordinate of the final predicted location

Journey to Crime Analysis

The journey to crime (Jtc) routine estimates the likelihood that a serial offender lives at any location within the study area. Both a primary file and a reference file are required. The locations of the serial crimes are defined in the primary file while all locations within the study area are identified in the reference file. The Jtc routine can use two different travel distance functions: 1) An already-calibrated distance function; and 2) A mathematical formula. Either direct or indirect (Manhattan) distances can be used though the default is direct (see Measurement parameters).

Calibrate Journey to Crime Function

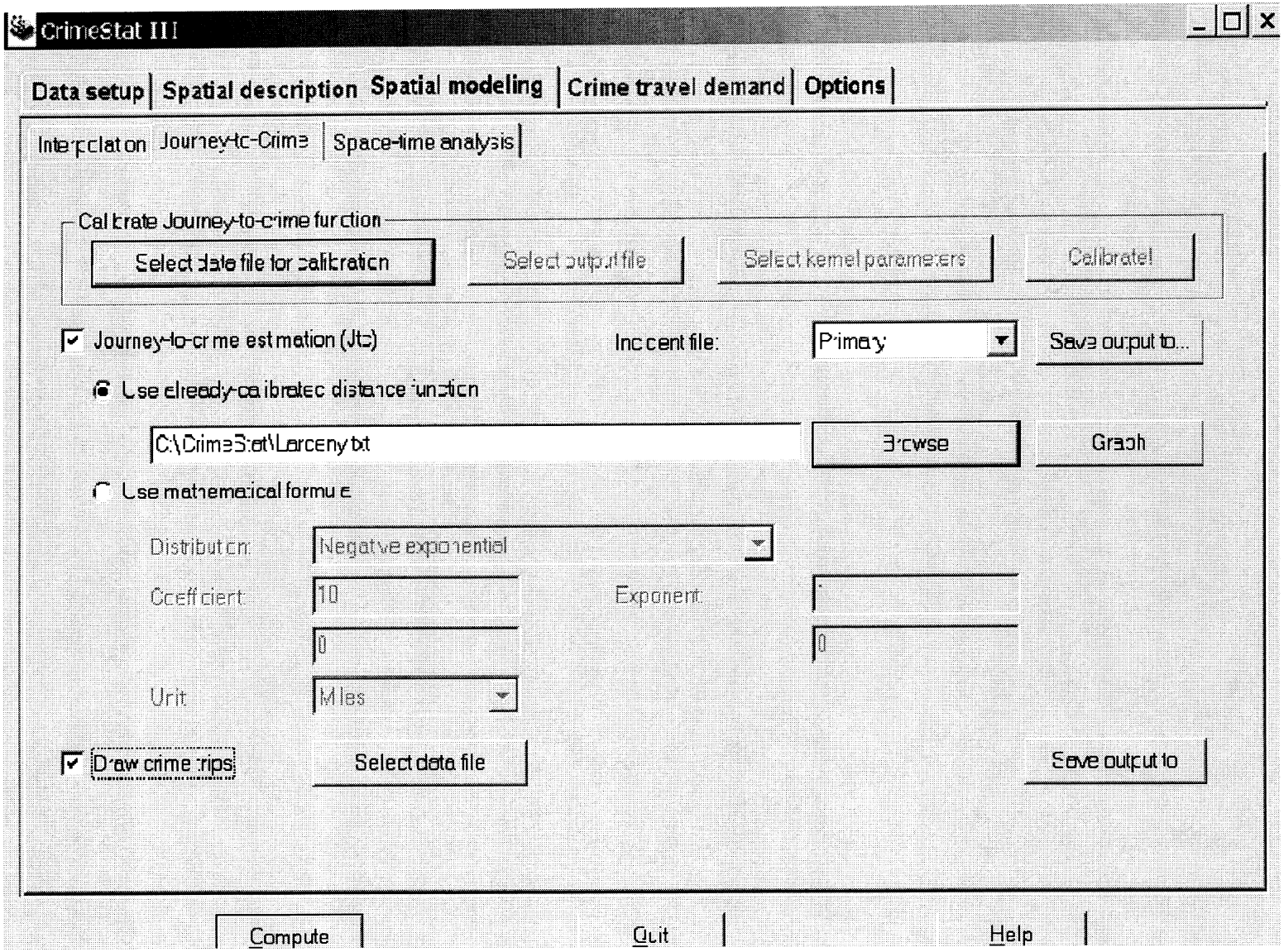
This routine calibrates a journey to crime distance function for use in the journey to crime estimation routine. A file is input which has a set of incidents (records) which includes both the X and Y coordinates for the location of the offender's residence (origin) and the X and Y coordinates for the location of the incident that the offender committed (destination). The routine estimates a travel distance function using a one-dimensional kernel density method. For each record, the distance between the origin location and the destination location is calculated and is represented on a distance scale. The maximum distance is calculated and divided into a number of intervals; the default is 100 equal sized intervals, but the user can modify this. For each distance (point) calculated, a one-dimensional kernel is overlaid. For each distance interval, the values of all kernels are summed to produce a smooth function of journey to crime distance. The results are saved to a file that can be used in the journey to crime estimation routine.

Select data file for calibration

Select the file that has the X and Y coordinates for the origin and destination locations. *CrimeStat* can read ASCII, dbase 'dbf', ArcView 'shp', MapInfo 'dat' files, and files that follow the ODBC standard interface. Select the tab and indicate the type of file to be selected. Use the browse button to search for the file. If the file type is ASCII, select the type of data separator (comma, semicolon, space, tab) and the number of columns.

Figure 2.12:

Journey-to-Crime Screen



Variables

Define the file which contains the X and Y coordinates for both the origin (residence) and destination (crime) locations.

Column

Select the variables for the X and Y coordinates respectively for *both* the origin and destination locations (e.g., Lon, Lat, HomeX, HomeY, IncidentX, IncidentY). Both locations must be defined for the routine to work.

Type of coordinate system and data units

Select the type of coordinate system. If the coordinates are in longitudes and latitudes, then a spherical system is used and data units will automatically be decimal degrees. If the coordinate system is projected (e.g., State Plane, Universal Transverse Mercator – UTM), then data units could be either in feet (e.g., State Plane) or meters (e.g., UTM). Directional coordinates are not allowed for this routine.

Kernel parameters

There are five parameters that must be defined.

Method of interpolation

There are fives types of kernel distributions that can be used to estimate point density. Four of the five distributions overlay a circle around each grid cell and assign weights to the points within the grid cell. The five types vary in the weights they assign to nearby points:

Kernel that assigns weights for entire study area

1. The **normal** kernel overlays a normal distribution over each point, which then extends over the entire study area defined by the reference file. This is the default kernel function. The distribution extends in all directions and is limited only by the study area.

Kernels that assign weights within a specific circle

2. The **uniform** kernel weights all points within the circle equally.

3. The **quartic** kernel overlays an inverted bell-shape surface that extends only for a limited distance from each point; the weights for points within the circle decline with distance, but gradually.

4. The **triangulated** (or conical) kernel overlays a cone over each grid cell; the weights for points within the circle decrease consistently with distance.

5. Finally, the **negative exponential** (or peaked) kernel overlays a sharply-decreasing function over the grid cell; the weights for points within the circle decrease very rapidly with distance. The five methods produce similar results although the normal is generally smoother for any given bandwidth.

Choice of bandwidth

The kernels are applied to a limited search distance, called 'bandwidth'. For the normal kernel, bandwidth is the standard deviation of the normal distribution. For the uniform, quartic, triangular and negative exponential kernels, bandwidth is the radius of a circle for the search area. For all types, larger bandwidth will produce smoother density estimates and both adaptive and fixed bandwidth intervals can be selected.

Fixed bandwidth

A fixed bandwidth distance is a fixed interval for each point. The user must define the interval, the interval size, and the distance units by which it is calculated (miles, nautical miles, feet, kilometers, meters). The default bandwidth setting is fixed with intervals of 0.25 miles each. The interval size can be changed.

Adaptive bandwidth

An adaptive bandwidth distance is identified by the minimum number of other points found within a symmetrical band drawn around a single point. A symmetrical band is placed over each distance point, in turn, and the width is increased until the minimum sample size is reached. Thus, each point has a different bandwidth size. The user can modify the minimum sample size. The default for the adaptive bandwidth is 100 points.

Specify interpolation bins

The interpolation bins are defined in one of two ways:

1. By the number of bins. The maximum distance calculated is divided by the number of bins specified. This is the default with 100 bins. The user can change the number of bins.

2. By the distance between bins. The user can specify a bin width in miles, nautical miles, feet, kilometers, and meters.

Output units

Specify the density units as points per mile, nautical mile, foot, kilometer, or meter. The default is points per mile.

Calculate densities or probabilities

The density estimate for each cell can be calculated in one of three ways:

1. **Absolute densities**. This is the number of points per grid cell and is scaled so that the sum of all grid cells equals the sample size. This is the default.

2. **Relative densities**. For each grid cell, this is the absolute density divided by the grid cell area and is expressed in the output units (e.g., points per square mile).
3. **Probabilities**. This is the proportion of all incidents that occur in the grid cell. The sum of all grid cells equals a probability of 1.

Select whether absolute densities, relative densities, or probabilities are to be output for each cell. The default is absolute densities.

Save calibration distance file

The output *must* be saved to a file. CrimeStat can save the calibration output to either a dbase 'dbf' or ASCII text 'txt' file.

Calibrate!

Click on 'Calibrate!' to run the routine. The output is saved to the specified file upon clicking on 'Close'.

Graphing the travel demand function

Click on 'View graph' to see the journey to crime travel demand function (journey to crime likelihood by distance). The screen view can be printed by clicking on 'Print'. For a better quality graph, however, the output should be imported into a graphics package.

Journey to Crime Estimation (Jtc)

The journey to crime (Jtc) routine estimates the likelihood that a serial offender lives at any location within the study area. Both a primary file and a reference file are required. The locations of the serial crimes are defined in the primary file while all locations within the study area are identified in the reference file. The Jtc routine can use two different travel distance functions: 1) An already-calibrated function; and 2) A mathematical formula.

Use an already-calibrated distance function

If a travel distance function has already been calibrated (see 'Calibrate journey to crime function'), the file can be directly input into the Jtc routine.

Input

The user selects the name of the already-calibrated travel distance function. *CrimeStat* reads dbase 'dbf', ASCII text 'txt', and ASCII data 'dat' files.

Output

The Jtc routine calculates a relative likelihood estimate for each cell of the reference file. Higher values indicate higher relative likelihoods. The results can be output as a *Surfer for Windows* file (for both an external or created reference file) or as an *ArcView* 'shp', *MapInfo* 'mif', *Atlas*GIS* 'bna', or *ArcView Spatial Analyst* 'asc', or ASCII grid 'grd' file (only if the reference file is created by *CrimeStat*). The output file is saved as Jtc<root name> with the root name being provided by the user.

Use a mathematical formula

A mathematical formula can be used instead of a calibrated distance function. To do this, it is necessary to specify the type of distribution. There are five mathematical models that can be selected:

1. Negative exponential
2. Normal distribution
3. Lognormal distribution
4. Linear distribution
5. Truncated negative exponential

For each mathematical model, two or three different parameters must be defined:

1. Negative exponential - coefficient and exponent;
2. Normal distribution - mean distance, standard deviation and coefficient;
3. Lognormal distribution - mean distance, standard deviation and coefficient;
4. Linear distribution - intercept and slope; and
5. Truncated negative exponential - peak distance, peak likelihood, intercept, and exponent.

Output

The Jtc routine calculates a relative likelihood estimate for each cell of the reference file. Higher values indicate higher relative likelihoods. The results can be output as a *Surfer for Windows* file (for both an external or created reference file) or as an *ArcView* 'shp', *MapInfo* 'mif', *Atlas*GIS* 'bna', or *ArcView Spatial Analyst* 'asc', or ASCII grid 'grd' file (only if the reference file is created by *CrimeStat*). The output file is saved as Jtc<root name> with the root name being provided by the user.

Draw Crime Trips

This routine is a utility for both the Journey-to-Crime routine and the Trip Distribution routine (in the Crime Travel Demand module). If given a file with origins and destinations, the routine will draw a line between the origin and destination for each record. It is useful for examining the actual trip links made by an offender.

Select data file

Select the file that has the X and Y coordinates for the origin and destination locations. CrimeStat can read ASCII, dbase '.dbf', ArcView '.shp' and MapInfo 'dat' files. Select the tab and specify the type of file to be selected. Use the browse button to search for the file. If the file type is ASCII, select the type of data separator (comma, semicolon, space, tab) and the number of columns.

Variables

Define the file which contains the X and Y coordinates for both the origin (residence) and destination (crime) locations

Columns

Select the variables for the X and Y coordinates respectively for both the origin and destination locations (e.g., Lon, Lat, HomeX, HomeY, IncidentX, IncidentY.) Both locations must be defined for the routine to work.

Type of coordinate system and data units

Select the type of coordinate system. If the coordinates are in longitudes and latitudes, then a spherical system is being used and data units will automatically be decimal degrees. If the coordinate system is projected (e.g., State Plane, Universal Transverse Mercator - UTM), then data units could be either in feet (e.g., State Plane) or meters (e.g., UTM.) Directional coordinates are not allowed for this routine.

Save output to

The graphical results can be output as lines in ArcView '.shp', MapInfo '.mif' or Atlas*GIS '.bna' format.

IV. Crime Travel Demand

The crime travel demand module is a sequential model of crime travel by zone over a metropolitan area. Crime incidents are allocated to zones, both by the location where the crime occurred (destinations) and the location where the offender started (origins). A crime trip is defined as a crime event that originates at one location and ends at another location; the two locations can be the same. For each zone, the number of crimes originating in the

zone and the number of crimes ending (occurring) in that zone are enumerated. Thus, the model is for count (or volumes), not rates. Other zonal data must be obtained to be used as predictor variables of the origin and destination counts.

The model is made up four sequential steps, each of which can involve smaller steps:

1. **Trip generation** - separate models are developed for predicting the number of crimes originating or ending in each zone. There are, therefore, two models. One is a model of the predicted number of crime trips that originate in each zone while the other is a model of the predicted number of crime trips that end in each zone. The number of origin zones can be greater than the number of destination zones.

2. **Trip distribution** - A model is developed for the number of crimes originating in each zone that go to each destination zone. The result is a prediction of the number of crimes originating in each zone that end in each zone (trip links).

3. **Mode split** - A model is developed that splits the number of predicted trips from each origin zone to each destination zone by travel mode (e.g., walking, bicycle, driving, bus, train). Thus, each zone-to-zone trip link is separated into different travel modes.

4. **Network assignment** - A model is developed for the route taken for each crime trip link (whether for all modes or by separate modes). Thus, the shortest path through a network is determined. Different travel modes will have different routes since bus and train, in particular, must use a separate network.

Crime Travel Demand Data Preparation

In order to run the crime travel demand module, particular data must be obtained and prepared. These involve:

1. A zonal framework that will be used for the modeling. In general, it is best to select the smallest zone size for which data can be obtained (e.g., block groups, census tracts, traffic analysis zones). However, it is often difficult to obtain data for the smallest units (e.g., blocks, grid cells). The larger the zone size, the more there will be intra-zonal trips and the greater the error in the model. Thus, the user must balance the need for small zones with the availability of data. Since crimes can occur outside a study area, the number of origin zones can be (and probably should be) greater than the number of destination zones. However, each destination zone should be included within the origin zone collection. Typically, there will be separate data sets for the origin zones and for the destination zones.

2. Data on crime origins and crime destinations are obtained (usually from arrest records) and are allocated to zones. The incidents are then summed by zone to produce a count. The "Assign primary points to secondary points" routine (under Distance analysis) can be used for this purpose. Thus, each origin zone has a count of the number of crimes originating in that zone and each destination zone has separate counts of the number of crimes originating in that zone and the number of crimes occurring (ending) in that zone. Crimes can be sub-divided into types (e.g., robbery, burglary, vehicle theft).

3. Additional data for the zones are obtained. These would include population (or households), sub-populations (e.g., age groups, race/ethnic groups), income levels, poverty levels, employment (retail and non-retail), land use, particular types of land use (e.g., drug locations, markets, parking lots), policing variables (e.g., personnel deployment, beat frequency), intervention variables (e.g., drug treatment centers), and other variables. It's important that all variables included must cover all zones for either the origin data set or the destination data set. For example, if poverty is used a variable in the origin model, then all origin zones must have an enumeration of poverty. Similarly, if retail employment is used as a variable in the destination model, then all destination zones must have an enumeration of retail employment.

4. Data on dummy variables and special generators are also obtained. Dummy variables would be a proxy for a condition that does or does not exist. Zones that have the condition are assigned a '1' whereas zones that do not have the condition are assigned a '0'. For example, if a freeway cross a zone, then a freeway dummy variable would assign '1' to that zone (and all others that the freeway crossed) whereas all other zones received a '0' for this variable. A special generator is a land use that attracts trips (e.g., a stadium, a railroad station). All zones that have the special generator are assigned a value whereas all other zones receive a '0'; the value can either be a dummy variable (i.e., a '1') or the actual count if that can be obtained (e.g., the number of patrons at a football stadium event).

Trip Generation

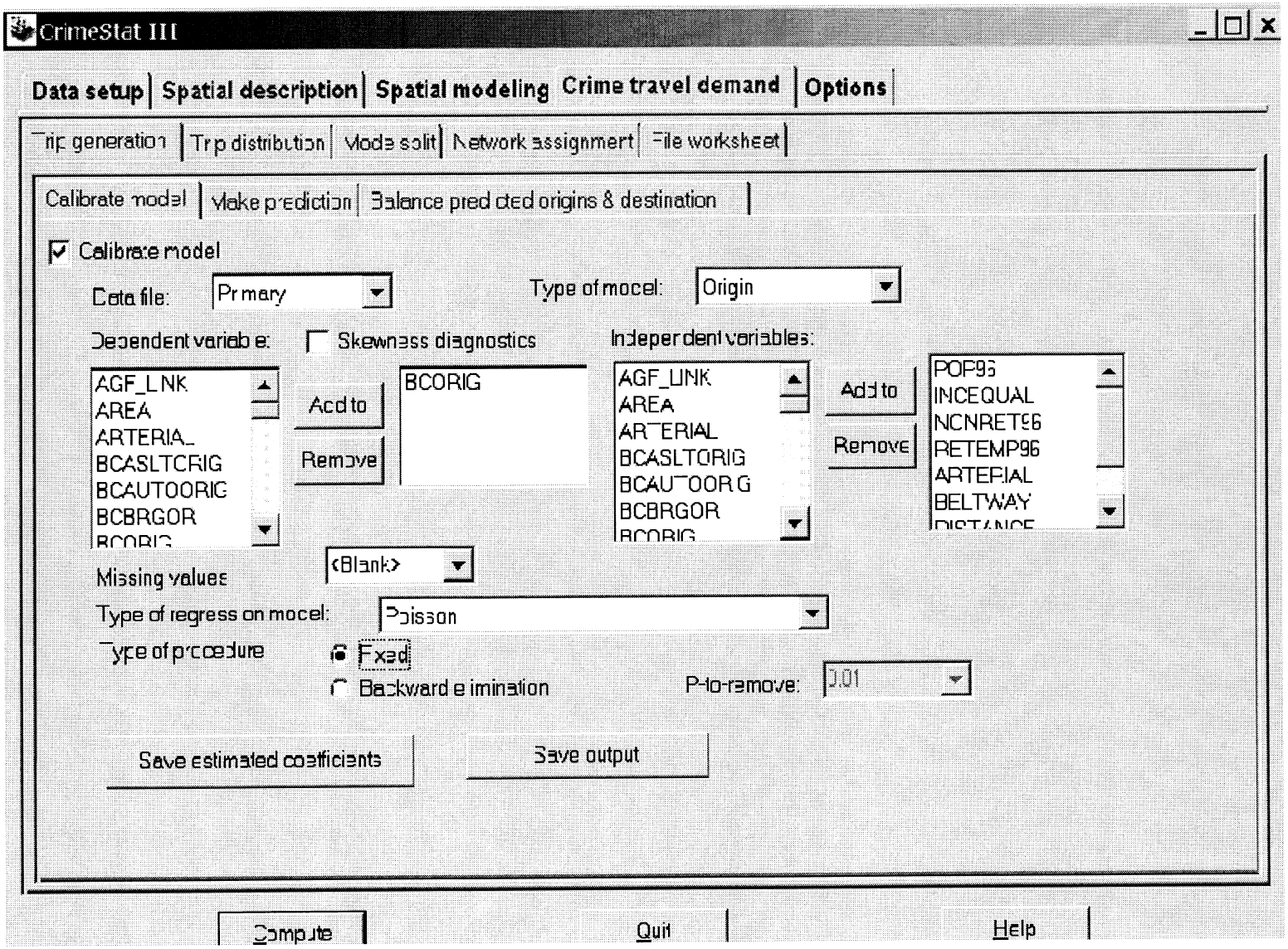
Trip generation involves the development of separate models for predicting the number of crimes originating in each zone and the number of crimes occurring (ending) in each zone. There are three steps to the trip generation:

1. **Calibrate model** - a step that calibrates the model against known data using regression techniques. The result is a prediction of the number of trips either originating in a zone (the origin model) or the number of trips ending in a zone (the destination model).

2. **Make prediction** - a step that applies the calibrated model to a data set and also allows the addition of trips from outside the study area (external trips).

Figure 2.13:

Trip Generation Screen



3. **Balance predicted origins & destinations** - a step that ensures that the number of predicted origins equals the number of predicted destinations. Since a trip involves an origin and a destination, it is essential that the number of origins equal the number of destinations.

Calibrate Trip Generation Model

This step involves calibrating a regression model against the zonal data. Two separate models are developed, one for trip origins and one for trip destinations. The dependent variable is the number of crimes originating in a zone (for the trip origin model) or the number of crimes ending in a zone (for the trip destination model). The independent variables are zonal variables that may predict the number of origins or destinations.

Data file

The data file is input as either the primary or secondary file. Specify whether the data file is the primary or secondary file.

Type of model

Specify whether the model is for origins or destinations. This will be printed out on the output header.

Dependent variable

Select the dependent variable from the list of variables. There can be only one dependent variable per model.

Skewness diagnostics

If checked, the routine will test for the skewness of the dependent variable. The output includes:

1. The "g" statistic
2. The standard error of the "g" statistic
3. The Z value for the "g" statistic
4. The probability level of a Type I error for the "g" statistic
5. The ratio of the simple variance to the simple mean

Error messages indicate whether there is probable skewness in the dependent variable. If there is skewness, use a Poisson regression model.

Independent variables

Select independent variables from the list of variables in the data file. Up to 15 variables can be selected.

Missing values

Specify any missing value codes for the variables. Blank records will automatically be considered as missing. If any of the selected dependent or independent variables have missing values, those records will be excluded from the analysis.

Type of regression model

Specify the type of regression model to be used. The default is a Poisson regression with over-dispersion correction. Other alternatives are a Poisson regression and an Ordinary Least Squares regression.

Type of regression procedure

Specify whether a fixed model (all selected independent variables are used in the regression) or a backward elimination stepwise model is used. The default is a fixed model. If a backward elimination stepwise model is selected, choose the P-to-remove value (default is .01). The backward elimination starts with all selected variables in the model (the fixed procedure). However, it proceeds to drop variables that fail the P-to-remove test, one at a time. Any variable that has a significance level in excess of the P-to-remove value is dropped from the equation.

Save estimated coefficients (parameters)

The estimated coefficients of the final model can be saved as a 'dbf' file. Specify e a file name. This would be useful in order to repeat the regression while adding in external trips to the predicted origins (see Make trip generation prediction below) or to apply the coefficients to another dataset (e.g., future values of the independent variable).

Save output

The output is saved as a 'dbf' file under a different file name. The output includes all the variables in the input data set plus two new ones: 1) the predicted values of the dependent variable for each observation (with the name PREDICTED); and 2) the residual error values, representing the difference between the actual /observed values for each observation and the predicted values (with the name RESIDUAL).

Poisson output

The output of the Poisson regression routines include 13 fields for the entire model:

1. The dependent variable
2. The type of model
3. The sample size (N)
4. The degrees of freedom (N - dependent variables - 1)

5. The type of regression model (Poisson, Poisson with over-dispersion correction)
6. The log-likelihood value
7. The Likelihood Ratio
8. The probability value of the Likelihood Ratio
9. The Akaike Information Criterion (AIC)
10. The Schwartz Criterion (SC)
11. The Dispersion Multiplier
12. The approximate R-square value
13. The deviance R-square value

and 5 fields for each estimated coefficient:

14. The estimated coefficient
15. The standard error of the coefficient
16. The pseudo-tolerance value of the coefficient (see below)
17. The Z-value of the coefficient
18. The p-value of the coefficient.

OLS Output

The output of the Ordinary Least Square (OLS) routine includes 9 fields for the entire model:

1. The dependent variable
2. The type of model
3. The sample size (N)
4. The degrees of freedom (N - dependent variables - 1)
5. The type of regression model (Norma/Ordinary Least Squares)
6. Squared multiple R
7. Adjusted squared multiple R
8. F test of the model
9. p-value of the model

and 5 fields for each estimated coefficient:

10. The estimated coefficient
11. The standard error of the coefficient
12. The tolerance value of the coefficient (see below)
13. The t-value of the coefficient
14. The p-value of the coefficient.

Multicolinearity among the independent variables

A major consideration in any model is that the independent variables are statistically independent. Non-independence is called multicolinearity. Non-independence

means that there is overlap in prediction among two or more independent variables. This can lead to uncertainty in interpreting coefficients as well as an unstable model that may not hold in the future. Generally, it is a good idea to reduce multicolinearity as much as possible. A tolerance test is given for each coefficient. This is defined as 1 - the R-square of the independent variable predicted by the remaining independent variables in the equation using an Ordinary Least Squares model. It is an indicator of how much the other independent variables in the equation account for the variance of any particular independent variable. Since the method uses the Ordinary Least Squares methods, it is an approximate (pseudo) test for the Poisson regression routines. A message is displayed that indicates probable or possible multicolinearity. A good idea is to drop one of the multicolinear independent variables and re-run the model. However, each of the coefficients should be inspected carefully before accepting a final model.

Graph of residual errors

While the output page is open, clicking on the graph button will display a graph of the residual errors (on the Y axis) against the predicted values (on the X axis).

Make Trip Generation Prediction

This routine applies an already-calibrated regression model to a data set. This would be useful for several reasons: 1) if external trips are to be added to the model (which is normally preferred); 2) if the model is applied to another data set; and 3) if variations on the coefficients are being tested with the same data set. The model will need to be calibrated first (see Calibrate trip generation model) and the coefficients saved as a parameters file. The coefficient parameter file is then re-loaded and applied to the data.

Data file

The data file is input as either the primary or secondary file. Specify whether the data file is the primary or secondary file.

Type of model

Specify whether the model is for origins or destinations. This will be printed out on the output header.

Trip generation parameters (coefficients) file

This is the saved coefficient parameter file. It is an Ascii file and can be edited if alternative coefficients were being tested (be careful about editing this without making a backup). Load the file by clicking on the Browse button and finding the file. Once loaded, the variable names of the saved coefficients are displayed in the "Matching parameters" box.

Independent variables

Select independent variables from the list of variables in the data file. Up to 15 variables can be selected.

Matching parameters

The selected independent variables need to be matched to the saved variables in the trip generation parameters file in the same order. Add the appropriate variables one by one in the order in which they are listed in the matching parameters box. It is essential that the order by the same otherwise the coefficients will be applied to the wrong variables.

> **Hint**: With your cursor placed in the list of independent variables, typing the first letter of matching variable name will take you to the first variable that starts with that letter. Repeating the letter will move down the list to the second, third, and so forth until the desired variable is reached.

Missing values

Specify any missing value codes for the variables. Blank records will automatically be considered as missing. If any of the selected dependent or independent variables have missing values, those records will be excluded from the analysis.

Add external trips

External trips are trips that start outside the modeled study area. Because they are crimes that originate outside the study area, they were not included in the zones used for the origin model. Therefore, they have to be independently estimated and added to the origin zone total to make the number of origins equal to the number of destinations. Click on the "Add external trips" button to enable this feature.

Number of external trips

Add the number of external trips to the box. This number will be added as an extra origin zone (the External zone).

Origin ID

Specify the origin ID variable in the data file. The external trips will be added as an extra origin zone, called the "External" zone. Note: all destination ID's should be in the origin zone file and must have the same names. This is necessary for subsequent modeling stages.

Type of regression model

Specify the type of regression model to be used. The default is a Poisson regression and the other alternative is a Normally-distributed/Ordinary Least Squares regression.

Save predicted values

The output is saved as a 'dbf' file under a different file name. The output includes all the variables in the input data set plus the predicted values of the dependent variable for each observation (with the name PREDICTED. In addition, if external trips were added, then there is a new record with the name EXTERNAL listed in the Origin ID column. This record lists the added trips in the PREDICTED column and zeros (0) for all other numeric fields.

Output

The tabular output includes summary information about file and lists the predicted values for each input zone.

Balance Predicted Origins & Destinations

Since, by definition, a 'trip' has an origin and a destination, the number of predicted origins must equal the number of predicted destinations. Because of slight differences in the data sets of the origin model and the destination model, it is possible that the total number of predicted origins (including any external trips - see Make trip generation prediction) may not equal the total number of predicted destinations. This step, therefore, is essential guarantee that this condition will be true. The routine adjusts either the number of predicted origins or the number of predicted destinations so that the condition holds. The trip distribution routines will not work unless the number of predicted origins equals the number of predicted destinations (within a very small rounding-off error).

Predicted origin file

Specify the name of the predicted origin file by clicking on the Browse button and locating the file.

Origin variable

Specify the name of the variable for the predicted origins (e.g., PREDICTED).

Predicted destination file

Specify the name of the predicted destination file by clicking on the Browse button and locating the file.

Destination variable

Specify the name of the variable for the predicted origins (e.g., PREDICTED).

Balancing method

Specify whether origins or destinations are to be held constant. The default is 'Hold destinations constant'.

Save predicted origin/destination file

The output is saved as a 'dbf' file under a different file name. The output includes all the variables in the input data set plus the adjusted values of the predicted values of the dependent variable for each observation. If destinations are held constant, the adjusted variable name for the predicted trips is ADJORIGIN. If origins are held constant, the adjusted variable name for the predicted trips is ADJDEST.

Output

The tabular output includes file summary information plus information about the number of origins and destinations before and after balancing. In addition, the predicted values of the dependent variable are displayed.

Trip Distribution

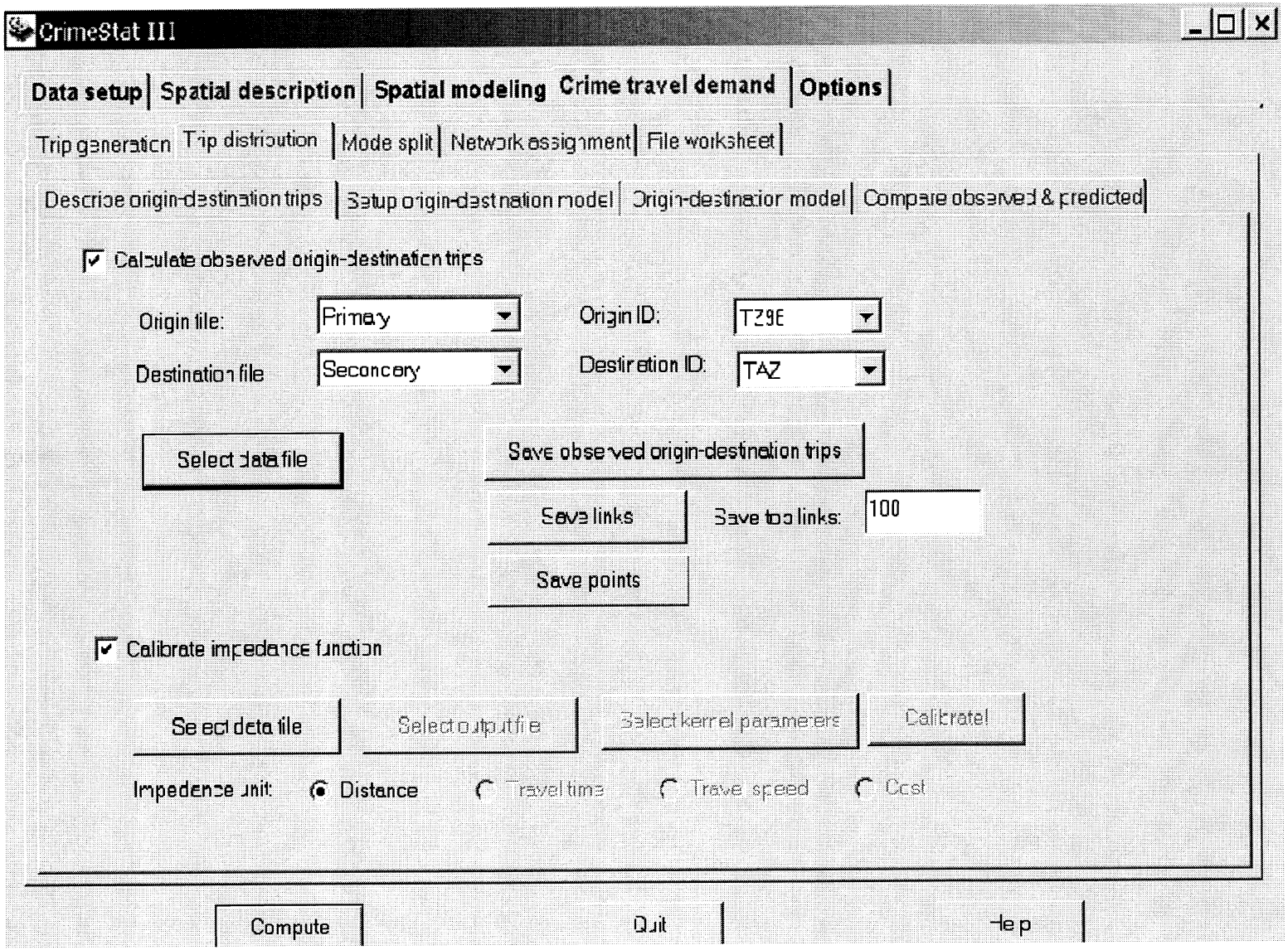
Trip distribution involves the estimation of the number of trips that travel from each origin zone (including the 'external' zone) to each destination zone. The estimation is based on a gravity-type model. The determining variables are the number of predicted origins, the number of predicted destinations, the impedance (or cost) of travel between the origin zone, coefficients for the origins and destinations, and exponents of the origins and destinations. The user inputs the number of predicted origins and predicted destinations and specifies an impedance model (which can be mathematical or calibrated from an existing data set). In addition, the user specifies exponents for the origin and destination values. The model iteratively estimates the coefficients. In addition, the routine can calculate the actual (observed) trip distribution with an existing data set that lists individual origin and destination locations. Finally, a comparison can be made between the observed distribution and that predicted by the model.

Describe Origin-Destination Trips

An empirical description of the actual trip distribution matrix can be made if there is a data set that includes individual origin and destination locations. The user defines the origin location and the destination location for each record and a set of zones from which to compare the individual origins and destinations. The routine matches up each origin

Figure 2.14:

Trip Distribution Screen



location with the nearest zone, each destination location with the nearest zone, and calculates the number of trips from each origin zone to each destination zone. This is an observed distribution of trips by zone.

Calculate observed origin-destination trips

Check if an empirical origin-destination trip distribution is to be calculated.

Origin file

The origin file is a list of origin zones with a single point representing the zone (e.g., the centroid). There can be more origin zones than destination zones, but all destination zones must be included among the origin zone list. The origin file must be input as either the primary or secondary file. Specify whether the data file is the primary or secondary file.

Origin ID

Specify the origin ID variable in the data file (e.g., CensusTract, Block, TAZ). Note: all destination ID's should be in the origin zone file and must have the same names.

Destination file

The destination file is a list of destination zones with a single point representing the zone (e.g., the centroid). It must be input as either the primary or secondary file. Specify whether the data file is the primary or secondary file.

Destination ID

Specify the destination ID variable in the data file (e.g., CensusTract, Block, TAZ). Note: the ID's used for the destination file zones must be the same as in the origin file.

Select data file

The data set must have individual origin and destination locations. Each record must have the X/Y coordinates of an origin location and the X/Y coordinates of a destination location. For example, an arrest file might list individual incidents with each incident having a crime location (the destination) and a residence or arrest location (the origin). Select the file that has the X and Y coordinates for the origin and destination locations. CrimeStat can read ASCII, dbase '.dbf', ArcView '.shp' and MapInfo 'dat' files. Select the tab and specify the type of file to be selected. Use the browse button to search for the file. If the file type is ASCII, select the type of data separator (comma, semicolon, space, tab) and the number of columns.

Variables

Define the file which contains the X and Y coordinates for both the origin (residence) and destination (crime) locations.

Columns

Select the variables for the X and Y coordinates respectively for both the origin and destination locations (e.g., Lon, Lat, HomeX, HomeY, IncidentX, IncidentY.) Both locations must be defined for the routine to work.

Missing values

Identify whether there are any missing values for these four fields (X and Y coordinates for both origin and destination locations). By default, CrimeStat will ignore records with blank values in any of the eligible fields or records with non-numeric values (e.g., alphanumeric characters, , *). Blanks will always be excluded unless the user selects <none>. There are 8 possible options:

1. **<blank>** fields are automatically excluded. This is the default
2. **<none>** indicates that no records will be excluded. If there is a blank field, CrimeStat will treat it as a 0
3. **0** is excluded
4. **-1** is excluded
5. **0** and **-1** indicates that both 0 and -1 will be excluded
6. **0**, **-1** and **9999** indicates that all three values (0, -1, 9999) will be excluded
7. **Any** other numerical value can be treated as a missing value by typing it (e.g., 99)
8. **Multiple** numerical values can be treated as missing values by typing them, separating each by commas (e.g., 0, -1, 99, 9999, -99)

Type of coordinate system and data units

The coordinate system and data units are listed for information. If the coordinates are in longitudes and latitudes, then a spherical system is being used and data units will automatically be decimal degrees. If the coordinate system is projected (e.g., State Plane, Universal Transverse Mercator - UTM), then data units could be either in feet (e.g., State Plane) or meters (e.g., UTM.)

Table output

The entire origin-destination matrix is output as a table to the screen including summary file information and:

1. The origin zone (ORIGIN)
2. The destination zone (DEST)

3. The number of observed trips (FREQ)

Save observed origin-destination trips

If specified, the full origin-destination output is saved as a 'dbf' file named by the user.

File output

The file output includes:

1. The origin zone (ORIGIN)
2. The destination zone (DEST)
3. The X coordinate for the origin zone (ORIGINX)
4. The Y coordinate for the origin zone (ORIGINY)
5. The X coordinate for the destination zone (DESTX)
6. The Y coordinate for the destination zone (DESTY)
7. The number of trips (FREQ)

Note: each record is a unique origin-destination combination and there are M x N records where M is the number of origin zones (including the external zone) and N is the number of destination zones.

Save links

The top observed origin-destination trip links can be saved as separate line objects for use in a GIS. Specify the output file format (ArcView '.shp', MapInfo '.mif' or Atlas*GIS '.bna') and the file name.

Save top links

Because the output file is very large (number of origin zones x number of destination zones), the user can select a sub-set of zone combinations with the most observed trips. Indicating the top K links will narrow the number down to the most important ones. The default is the top 100 origin-destination combinations. Each output object is a line from the origin zone to the destination zone with an ODT prefix. The prefix is placed before the output file name. The line graphical output for each object includes:

1. An ID number from 1 to K, where K is the number of links output (ID)
2. The feature prefix (ODT)
3. The origin zone (ORIGIN)
4. The destination zone (DEST)
5. The X coordinate for the origin zone (ORIGINX)
6. The Y coordinate for the origin zone (ORIGINY)
7. The X coordinate for the destination zone (DESTX)
8. The Y coordinate for the destination zone (DESTY)

9. The number of observed trips for that combination (FREQ)
10. The distance between the origin zone and the destination zone.

Save points

Intra-zonal trips (trips in which the origin and destination are the same zone) can be output as separate point objects as an ArcView '.shp', MapInfo '.mif' or Atlas*GIS '.bna' file. Again, the top K points are output (default=100). Each output object is a point representing an intra-zonal trip with an ODTPOINTS prefix. The prefix is placed before the output file name. The point graphical output for each object includes:

1. An ID number from 1 to K, where K is the number of links output (ID)
2. The feature prefix (POINTSODT)
3. The origin zone (ORIGIN)
4. The destination zone (DEST)
5. The X coordinate for the origin zone (ORIGINX)
6. The Y coordinate for the origin zone (ORIGINY)
7. The X coordinate for the destination zone (DESTX)
8. The Y coordinate for the destination zone (DESTY)
9. The number of observed trips for that combination (FREQ)

Calibrate Impedance Function

This function allows the calibration of an approximate travel impedance function based on actual trip distributions. It is used to describe the travel distance of an actual sample (the calibration sample). A file is input which has a set of incidents (records) that includes both the X and Y coordinates for the location of the offender's residence (origin) and the X and Y coordinates for the location of the incident that the offender committed (destination.) The routine estimates a travel distance function using a one-dimensional kernel density method. For each record, the distance between the origin location and the destination location is calculated and is represented on a distance scale. The maximum distance is calculated and divided into a number of intervals; the default is 100 equal sized intervals, but the user can modify this. For each distance (point) calculated, a one-dimensional kernel is overlaid. For each distance interval, the values of all kernels are summed to produce a smooth function of travel impedance. The results are saved to a file that can be used origin-destination model. Note, however, that this is an empirical distribution and represents the combination of origins, destinations, and costs. It is not necessarily a good description of the impedance (cost) function by itself. Many of the mathematical functions produce a better fit than the empirical impedance function.

Select data file for calibration

Select the file that has the X and Y coordinates for the origin and destination locations. CrimeStat can read ASCII, dbase '.dbf', ArcView '.shp' and MapInfo 'dat' files. Select the tab and select the type of file to be selected. Use the browse button to search for

the file. If the file type is ASCII, select the type of data separator (comma, semicolon, space, tab) and the number of columns.

Variables

Define the file which contains the X and Y coordinates for both the origin (residence) and destination (crime) locations

Columns

Select the variables for the X and Y coordinates respectively for both the origin and destination locations (e.g., Lon, Lat, HomeX, HomeY, IncidentX, IncidentY.) Both locations must be defined for the routine to work.

Missing values

Identify whether there are any missing values for these four fields (X and Y coordinates for both origin and destination locations). By default, CrimeStat will ignore records with blank values in any of the eligible fields or records with non-numeric values (e.g., alphanumeric characters, , *). Blanks will always be excluded unless the user selects <none>. There are 8 possible options:

1. **<blank>** fields are automatically excluded. This is the default
2. **<none>** indicates that no records will be excluded. If there is a blank field, CrimeStat will treat it as a 0
3. **0** is excluded
4. **-1** is excluded
5. **0** and **-1** indicates that both 0 and -1 will be excluded
6. **0**, **-1** and **9999** indicates that all three values (0, -1, 9999) will be excluded
7. **Any** other numerical value can be treated as a missing value by typing it (e.g., 99)
8. **Multiple** numerical values can be treated as missing values by typing them, separating each by commas (e.g., 0, -1, 99, 9999, -99)

Type of coordinate system and data units

Select the type of coordinate system. If the coordinates are in longitudes and latitudes, then a spherical system is being used and data units will automatically be decimal degrees. If the coordinate system is projected (e.g., State Plane, Universal Transverse Mercator - UTM), then data units could be either in feet (e.g., State Plane) or meters (e.g., UTM.) Directional coordinates are not allowed for this routine.

Select kernel parameters

There are five parameters that must be defined.

Method of interpolation

There are fives types of kernel distributions that can be used to estimate point density:

1. The normal kernel overlays a three-dimensional normal distribution over each point that then extends over the area defined by the reference file. This is the default kernel function.

2. The uniform kernel overlays a uniform function (disk) over each point that only extends for a limited distance.

3. The quartic kernel overlays a quartic function (inverse sphere) over each point that only extends for a limited distance.

4. The triangular kernel overlays a three-dimensional triangle (cone) over each point that only extends for a limited distance.

5. The negative exponential kernel overlays a three dimensional negative exponential function ('salt shaker') over each point that only extends for a limited distance

The methods produce similar results though the normal is generally smoother for any given bandwidth.

Choice of bandwidth

The kernels are applied to a limited search distance, called 'bandwidth'. For the normal kernel, bandwidth is the standard deviation of the normal distribution. For the uniform, quartic, triangular and negative exponential kernels, bandwidth is the radius of a circle defined by the surface. For all types, larger bandwidth will produce smoother density estimates and both adaptive and fixed bandwidth intervals can be selected.

Fixed bandwidth

A fixed bandwidth distance is a fixed interval for each point. The user must define the interval, the interval size, and the distance units by which it is calculated (miles, nautical miles, feet, kilometers, meters.) The default bandwidth setting is fixed with intervals of 0.25 miles each. The interval size can be changed.

Adaptive bandwidth

An adaptive bandwidth distance is identified by the minimum number of other points found within a symmetrical band drawn around a single point. A symmetrical band is placed over each distance point, in turn, and the width is increased until the minimum

sample size is reached. Thus, each point has a different bandwidth size. The user can modify the minimum sample size. The default for the adaptive bandwidth is 100 points.

Specify interpolation bins

The interpolation bins are defined in one of two ways:

1. By the number of bins. The maximum distance calculated is divided by the number of specified bins.This is the default with 100 bins. The user can change the number of bins.

2. By the distance between bins. The user can specify a bin width in miles, nautical miles, feet, kilometers, and meters.

Output (areal) units

Specify the areal density units as points per mile, nautical mile, foot, kilometer, or meter. The default is points per mile.

Calculate densities or probabilities

The density estimate for each cell can be calculated in one of three ways:

1. **Absolute densities**. This is the number of points per grid cell and is scaled so that the sum of all grid cells equals the sample size.

2. **Relative densities**. For each grid cell, this is the absolute density divided by the grid cell area and is expressed in the areal output units (e.g., points per square mile)

3. **Probabilities**. This is the proportion of all incidents that occur in the grid cell. The sum of all grid cells equals a probability of 1. Unlike the Jtc calibration routine, this is the default. In most cases, a user would want a proportional (probability) distribution as the relative differences in impedance for different costs are what is of interest.

Select whether absolute densities, relative densities, or probabilities are to be output for each cell. The default is probabilities.

Select output file

The output must be saved to a file. CrimeStat can save the calibration output to either a dbase 'dbf' or ASCII text 'txt' file.

Calibrate!

Click on 'Calibrate!' to run the routine. The output is saved to the specified file upon clicking on 'Close'.

Graphing the travel impedance function

Click on 'View graph' to see the travel impedance function. The screen view can be printed by clicking on 'Print'. For a better quality graph, however, the output should be imported into a graphics package.

Setup Origin-Destination Model

The page is for the setup of the origin-destination model. All the relevant files, models and exponents are input on the page.

Predicted origin file

The predicted origin file is a file that lists the origin zones with a single point representing the zone (e.g., the centroid) and also includes the predicted number of crimes by origin zone. The file must be input as either the primary or secondary file. Specify whether the data file is the primary or secondary file.

Origin variable

Specify the name of the variable for the predicted origins (e.g., PREDICTED, ADJ ORIGINS).

Origin ID

Specify the origin ID variable in the data file (e.g., CensusTract, Block, TAZ). Note: all destination ID's should be in the origin zone file and must have the same names.

Predicted destination file

The predicted destination file is a list of destination zones with a single point representing the zone (e.g., the centroid) and also includes the predicted number of crimes by destination zone. It must be input as either the primary or secondary file. Specify whether the data file is the primary or secondary file.

Destination variable

Specify the name of the variable for the predicted destination (e.g., PREDICTED, ADJ DEST).

Destination ID

Specify the destination ID variable in the data file (e.g., CensusTract, Block, TAZ). Note: the ID's used for the destination file zones must be the same as in the origin file.

Exponents

The exponents are power terms for the predicted origins and destinations. They indicate the relative strength of those variables. For example, compared to an exponent of 1.0 (the default), an exponent greater than 1.0 will strengthen that variable (origins or destinations) while an exponent less than 1.0 will weaken that variable. They can be considered 'fine tuning' adjustments.

Origins

Specify the exponent for the predicted origins. The default is 1.0.

Destinations

Specify the exponent for the predicted origins. The default is 1.0.

Impedance function

The trip distribution routine can use two different travel distance functions: 1) An already-calibrated distance function; and 2) A mathematical formula. The default is a mathematical formula.

Use an already-calibrated distance function

If a travel distance function has already been calibrated (see 'Calibrate impedance function' under trip distribution), the file can be directly input into the routine. The user selects the name of the already-calibrated travel distance function. CrimeStat reads dbase 'dbf', ASCII text 'txt', and ASCII data 'dat' files.

Use a mathematical formula

A mathematical formula can be used instead of a calibrated distance function. To do this, it is necessary to specify the type of distribution. There are five mathematical models that can be selected:

1. Negative exponential
2. Normal
3. Lognormal
4. Linear
5. Truncated negative exponential

The lognormal is the default. For each mathematical model, two or three different parameters must be defined:

1. For the negative exponential, the coefficient and exponent

2. For the normal distribution, the mean distance, standard deviation and coefficient

3. For lognormal distribution, the mean distance, standard deviation and coefficient

4. For the linear distribution, an intercept and slope

5. For the truncated negative exponential, a peak distance, peak likelihood, intercept, and exponent.

Measurement unit

The routine can calculate impedance in four ways, by:

1. Distance (miles, nautical miles, feet, kilometers, and meters)
2. Travel time (minutes, hours)
3. Speed (miles per hour, kilometers per hour)
4. General travel costs (unspecified units).

These must be setup under 'Network distance' on the Measurement Parameters page. Specify the appropriate units. In the Network Parameters dialogue, specify the measurement units. The default is distance in miles.

Assumed impedance for external zones

For trips originating outside the study area (external trips), specify the amount and the units that will be assumed for these trips. The default is 25 miles.

Assumed impedance for intra-zonal trips

For trips originating and ending in the same zone (intra-zonal trips), specify the amount and the units that will be assumed for these trips. The default is 0.25 miles.

Model constraints

In calibrating a model, the routine must constrain either the origins or the destinations (single constraint) or constrain both the origins and the destinations (double constraint). In the latter case, it is an iterative solution. The default is to constrain destinations as it is assumed that the destinations totals (the number of crimes occurring in

each zone) are probably more correct than the number of crimes originating in each zone. . Specify the type of constraint for the model.

Constrain origins

If constrain origins is selected, the total number of trips from each origin zone will be held constant.

Constrain destinations

If constrain destinations is selected, the total number of trips from each destination zone will be held constant.

Constrain both origins and destinations

If constrain both origins and destinations is selected, the routine iteratively works out a balance between the number of origins and the number of destinations.

Origin-Destination Model

The trip distribution (origin-destination) model is implemented in two steps. First, the coefficients are calculated according to the exponents and impedance functions specified on the setup page. Second, the coefficients and exponents are applied to the predicted origins and destinations resulting in a predicted trip distribution. Because these two steps are iterative, they cannot be run simultaneously.

Calibrate Origin-Destination Model

Check the 'Calibrate origin-destination model' box to run the calibration model.

Save modeled coefficients (parameters)

The modeled coefficients are saved as a 'dbf' file. Specify a file name.

Apply Predicted Origin-Destination Model

Check the 'Apply predicted origin-destination model' box to run the trip distribution prediction.

Modeled coefficients file

Load the modeled coefficients file saved in the 'Calibrate origin-destination model' stage.

Assumed coordinates for external zone

In order to model trips from the 'external' zone (trips from outside the study area), specify coordinates for this zone. These coordinates will be used in drawing lines from the predicted origins to the predicted destinations. There are four choices:

1. Mean center (the mean X and mean Y of all origin file points are taken). This is the default.

2. Lower-left corner (the minimum X and minimum Y values of all origin file points are taken).

3. Upper-right corner (the maximum X and maximum Y values of all origin file points are taken).

4. Use coordinates (user-defined coordinates). Indicate the X and Y coordinates that are to be used.

Table output

The table output includes summary file information and (with default names):

1. The origin zone (ORIGIN)
2. The destination zone (DEST)
3. The number of predicted trips (PREDTRIPS)

Save predicted origin-destination trips

Define the output file. The output is saved as a 'dbf' file specified by the user.

File output

The file output includes (with default names):

1. The origin zone (ORIGIN)
2. The destination zone (DEST)
3. The X coordinate for the origin zone (ORIGINX)
4. The Y coordinate for the origin zone (ORIGINY)
5. The X coordinate for the destination zone (DESTX)
6. The Y coordinate for the destination zone (DESTY)
7. The number of predicted trips (PREDTRIPS)

Note: each record is a unique origin-destination combination and there are M x N records where M is the number of origin zones (including the external zone) and N is the number of destination zones.

Save links

The top predicted origin-destination trip links can be saved as separate line objects for use in a GIS. Specify the output file format (ArcView '.shp', MapInfo '.mif' or Atlas*GIS '.bna') and the file name.

Save top links

Because the output file is very large (number of origin zones x number of destination zones), the user can select a sub-set of zone combinations with the most predicted trips. Indicating the top K links will narrow the number down to the most important ones. The default is the top 100 origin-destination combinations. Each output object is a line from the origin zone to the destination zone with an ODT prefix. The prefix is placed before the output file name. The graphical output includes (with default names):

1. An ID number from 1 to K, where K is the number of links output (ID)
2. The feature prefix (ODT)
3. The origin zone (ORIGIN)
4. The destination zone (DEST)
5. The X coordinate for the origin zone (ORIGINX)
6. The Y coordinate for the origin zone (ORIGINY)
7. The X coordinate for the destination zone (DESTX)
8. The Y coordinate for the destination zone (DESTY)
9. The number of predicted trips for that combination (PREDTRIPS)
10. The distance between the origin zone and the destination zone.

Save points

Intra-zonal trips (trips in which the origin and destination are the same zone) can be output as separate point objects as an ArcView '.shp', MapInfo '.mif' or Atlas*GIS '.bna' file. Again, the top K points are output (default=100). Each output object is a point representing an intra-zonal trip with an ODTPOINTS prefix. The prefix is placed before the output file name. The graphical output for each includes (with default names):

1. An ID number from 1 to K, where K is the number of links output (ID)
2. The feature prefix (POINTSODT)
3. The origin zone (ORIGIN)
4. The destination zone (DEST)
5. The X coordinate for the origin zone (ORIGINX)
6. The Y coordinate for the origin zone (ORIGINY)
7. The X coordinate for the destination zone (DESTX)
8. The Y coordinate for the destination zone (DESTY)
9. The number of predicted trips for that combination (PREDTRIPS)

Compare Observed & Predicted Origin-Destination Trip Lengths

The predicted trip distribution model can be compared with the observed (actual) trip distribution. Since there are many cells for this comparison (M origins x N destinations), a comparison is usually conducted for the trip length distributions. Each origin-destination link (whether the observed distribution or that predicted by the model) is converted into a trip length. The maximum distance between an origin and a destination is then divided into K bins (intervals), where K can be defined by the user; the default is 25. The two distributions are compared with two statistics: 1) the coincidence ratio (essentially a positive correlation index that varies between 0 and 1 with 0 representing little coincidence and 1 representing perfect coincidence) and 2) the Komolgorov-Smirnov two-sample test (a test of the difference between the cumulative proportions of the observed and predicted distributions). There is also a graph that compares the two distributions.

Observed trip file

Select the observed trip distribution file by clicking on the Browse button and finding the file.

Observed number of origin-destination trips

Specify the variable for the observed number of trips. The default name is FREQ.

Orig_ID

Specify the ID name for the origin zone. The default name is ORIGIN. Note: the origin ID's should be the same as in the predicted file in order to compare the top links.

Orig_X

Specify the name for the X coordinate of the origin zone. The default name is ORIGINX.

Orig_Y

Specify the name for the Y coordinate of the origin zone. The default name is ORIGINY.

Dest_ID

Specify the ID name for the destination zone. The default name is DEST. Note: the destination ID's should be the same as in the predicted file in order to compare the top links.

Dest_X

Specify the name for the X coordinate of the destination zone. The default name is DESTX.

Dest_Y

Specify the name for the Y coordinate of the destination zone. The default name is DESTY.

Predicted trip file

Select the predicted trip distribution file by clicking on the Browse button and finding the file.

Predicted number of origin-destination trips

Specify the variable for the predicted number of trips. The default name is PREDTRIPS

Orig_ID

Specify the ID name for the origin zone. The default name is ORIGIN. Note: the origin ID's should be the same as in the observed file in order to compare the top links.

Orig_X

Specify the name for the X coordinate of the origin zone. The default name is ORIGINX.

Orig_Y

Specify the name for the Y coordinate of the origin zone. The default name is ORIGINY.

Dest_ID

Specify the ID name for the destination zone. The default name is DEST. Note: the destination ID's should be the same as in the observed file in order to compare the top links.

Dest_X

Specify the name for the X coordinate of the destination zone. The default name is DESTX.

Dest_Y

Specify the name for the Y coordinate of the destination zone. The default name is DESTY.

Select bins

Specify how the bins (intervals) will be defined. There are two choices. One is to select a fixed number of bins. The other is to select a constant interval.

Fixed number

This sets a fixed number of bins. An interval is defined by the maximum distance between zone divided by the number of bins. The default number of bins is 25. Specify the number of bins.

Constant interval

This defines an interval of a specific size. If selected, the units must also be chosen. The default is 0.25 miles. Other distance units are nautical miles, feet, kilometers, and meters. Specify the interval size.

Save comparison

The output is saved as a 'dbf' file specified by the user.

Table output

The table output includes summary information and:

1. The number of trips in the observed origin-destination file
2. The number of trips in the predicted origin-destination file
3. The number of intra-zonal trips in the observed origin-destination file
4. The number of intra-zonal trips in the predicted origin-destination file
5. The number of inter-zonal trips in the observed origin-destination file
6. The number of inter-zonal trips in the predicted origin-destination file
7. The average observed trip length
8. The average predicted trip length
9. The median observed trip length
10. The median predicted trip length
11. The Coincidence Ratio (an indicator of congruence varying from 0 to 1)
12. The D value for the Komolgorov-Smirnov two-sample test
13. The critical D value for the Komolgorov-Smirnov two-sample test
14. The p-value associated with the D value of Komolgorov-Smirnov two-sample test relative to the critical D value.

and for each bin:

15. The bin number
16. The bin distance
17. The observed proportion
18. The predicted proportion

File output

The saved file includes (with default names):

1. The bin number (BIN)
2. The bin distance (BINDIST)
3. The observed proportion (OBSERVPROP)
4. The predicted proportion (PREDPROP)

Graph of observed and predicted trip lengths

While the output page is open, clicking on the graph button will display a graph of the observed and predicted trip length proportions on the Y-axis by the trip length distance on the X-axis.

Compare Top Links

As an alternative to a comparison of trip lengths for the observed and predicted distributions, the top links can be compared with a pseudo-Chi square test. Since the top links have the most trips, the Chi square distribution can be used for comparison. However, because the rest of the distribution is not being used, significance tests are invalid.

The statistic compares the number of trips for the top links in the observed distribution with the number of trips for the same links in the predicted model. The routine calulates a Chi square value.

The statistic is useful for comparing different models. The lower the Chi square value, the better the fit between the predicted model and the observed for the top links. The aim is to find the model that gives the lowest possible Chi square value.

Note: in order to use this routine, the origin and destination ID's must be the same for both the observed and predicted trip files.

Click the box and specify the number of links to be compared. The default value is 100. The output includes:

1. The number of links that are compared

and for each trip pair in order of the number of trips:

2. The zone ID of the origin zone (FromZone)
3. The zone ID of the destination zone (ToZone)
4. The observed (actual) number of trips
5. The predicted number of trips.

At the bottom of the page is a Chi-square test of the difference between the observed and predicted number of trips for the top links. Since not all trips have been included in this distribution, no significance test is conducted. The aim should be to find the model with the lowest Chi-square value.

Optimizing the Fit Between the Observed and Predicted Links

Ideally, the best model would fulfill three comparison tests. First, the number of intra-zonal tests (and, by implication, the number of inter-zonal trips) in the predicted trip distribution would be identical to the number of intra-zonal trips in the observed distribution. Second, the overall model would have a high coincidence ratio and a non-significant Komolgorov-Smirnov test for the trip length comparison. Third, the Chi square value for the top links would be the lowest possible. In practice, an optimal model may have to balance these three criteria, producing a good match in the number of intra-zonal trips, a reasonably low Chi square value for the top links, and a reasonably high coincidence ratio for the trip length comparison. There may not be a single, optimal model.

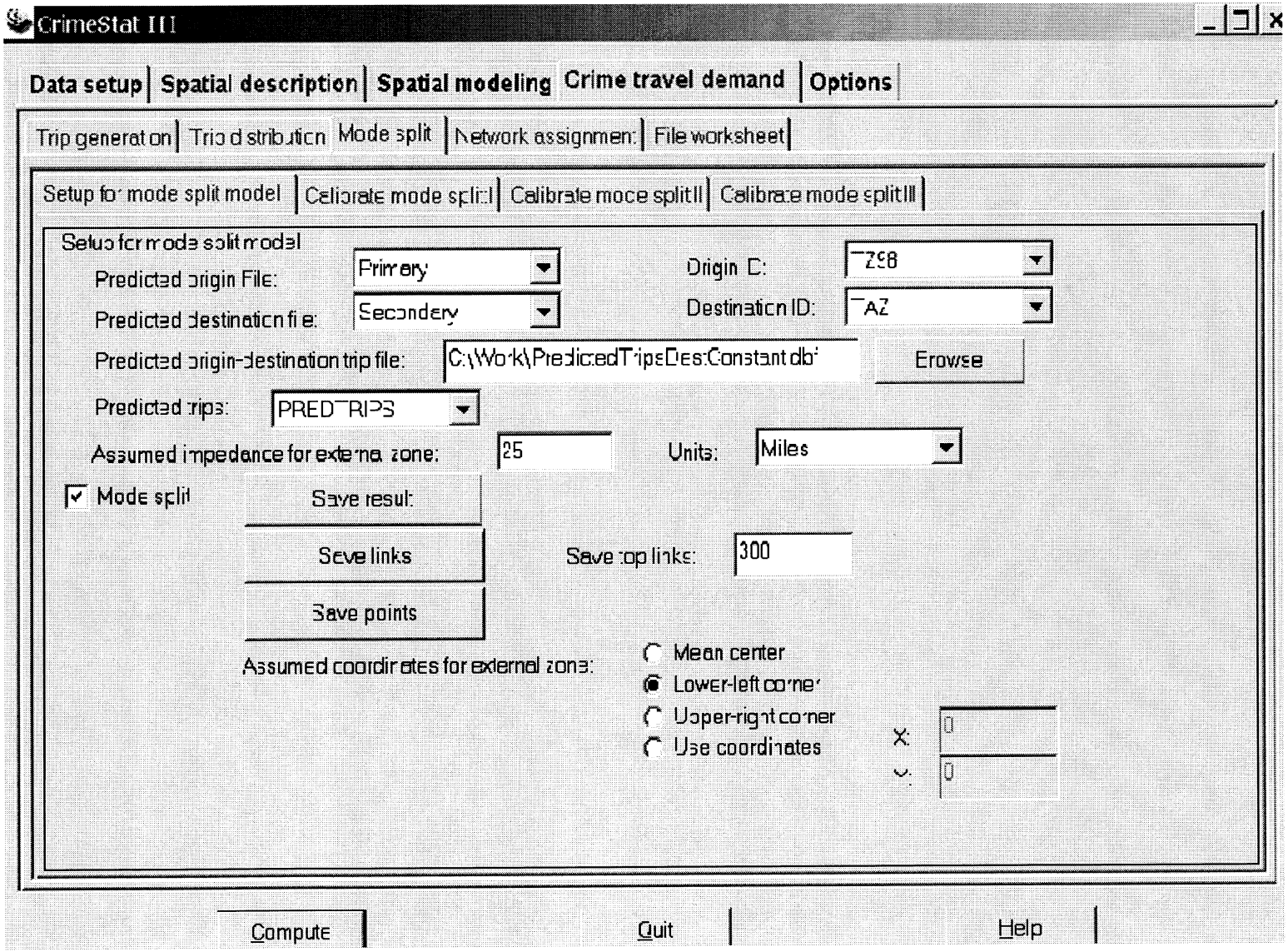
Mode Split

Mode split involves separating the predicted trips by link (i.e., the trips from any one origin zone, A, to any one destination zone, B) into distinct travel modes (e.g., walk, bicycle, drive, bus, train). The basis of the separation is an aggregate relative impedance function. This is, essentially, the 'cost' of traveling by any one mode relative to all modes, whether cost is defined in terms of distance, travel time, or generalized costs. The model can be determined by either an empirically-derived impedance function or a mathematical function. The empirically-derived impedance function would come from a calibration data set whereas the mathematical function is selected on the basis of either previous experience or other studies. The separate impedance functions can be constrained to a network in order to prevent trips from being allocated that are nearly impossible (e.g., train trips where there are no train lines and bus trips where there are no bus routes).

The steps of the routine are as follows. First, the user inputs a file of predicted trips (i.e., the number of predicted trips from every origin zone to every destination zone). Second, the user defines which travel modes are to be modeled. Up to five separate modes are allowed. Third, the user sets up an impedance model for each travel mode. Any of the impedance models can be constrained to a particular network (e.g., bus mode constrained to

Figure 2.15:

Mode Split Screen



a bus network; train mode constrained to a train network). This would normally be desired even for modes where travel in any direction is possible (e.g., walk, bicycle, drive modes). Fourth, and finally, after all impedance models have been defined, the routine is run and splits the predicted trips into the defined modes on the basis of the relative impedance of each mode to all impedances.

Setup for Mode Split Model

This page defines the predicted trip file and the output file. It also allows a definition of where external trips are assumed to come from.

Predicted origin file

The predicted origin file is a file that lists the origin zones with a single point representing the zone (e.g., the centroid) and also includes the predicted number of crimes by origin zone. The file must be input as either the primary or secondary file. Specify whether the data file is the primary or secondary file.

Origin variable

Specify the name of the variable for the predicted origins (e.g., PREDICTED, ADJORIGINS).

Origin ID

Specify the origin ID variable in the data file (e.g., CensusTract, Block, TAZ). Note: all destination ID's should be in the origin zone file and must have the same names.

Predicted destination file

The predicted destination file is a list of destination zones with a single point representing the zone (e.g., the centroid) and also includes the predicted number of crimes by destination zone. It must be input as either the primary or secondary file. Specify whether the data file is the primary or secondary file.

Destination variable

Specify the name of the variable for the predicted destination (e.g., PREDICTED, ADJDEST).

Destination ID

Specify the destination ID variable in the data file (e.g., CensusTract, Block, TAZ). Note: the ID's used for the destination file zones must be the same as in the origin file.

Predicted origin-destination trip file

The predicted origin-destination trip file is a file that lists the predicted number of trips from every origin zone to every destination zone. On the mode split setup page, select the predicted trip file (i.e., the predicted origin-destination trip file by clicking on the 'Browse' button.

Predicted trips

Specify the variable for the predicted number of trips. The default name is PREDTRIPS

Assumed impedance for external zone

In order to model trips from the 'external zone' (trips from outside the study area), specify an impedance to be assumed. The default is 25 miles.

Assumed coordinates for external zone

In order to model trips from the 'external' zone (trips from outside the study area), specify coordinates for this zone. These coordinates will be used in drawing lines from the predicted origins to the predicted destinations. There are four choices:

1. Mean center (the mean X and mean Y of all origin file points are taken). This is the default.

2. Lower-left corner (the minimum X and minimum Y values of all origin file points are taken).

3. Upper-right corner (the maximum X and maximum Y values of all origin file points are taken).

4. Use coordinates (user-defined coordinates). Indicate the X and Y coordinates that are to be used.

Run Mode Split

Check the "Mode split" box to enable the routine. It will run when the "Compute" button is clicked.

Mode split output

There are three types of output for the mode split routine. First, the zone-to-zone trip file for each mode separately can be output as a dbf file. Second, the most frequent inter-zonal (i.e., trips between different zones) trips for each mode separately can be output

as polylines. Third, the most frequent intra-zonal (i.e., trips within the same zone) trips for each mode separately can be output as points.

Save result

Define the output file. The output will be saved as a 'dbf' file specified by the user. For each mode, the prefix 'TMode' will be prefaced before the file. For example, if the name provided by the user is "robberies.dbf" and if there are three travel modes modeled, then there will be three output files (TMode1robberies.dbf; TMode2robberies.dbf; TMode3robberies.dbf).

File output

The file output includes:

1. The origin zone (ORIGIN)
2. The destination zone (DEST)
3. The X coordinate for the origin zone (ORIGINX)
4. The Y coordinate for the origin zone (ORIGINY)
5. The X coordinate for the destination zone (DESTX)
6. The Y coordinate for the destination zone (DESTY)
7. The number of predicted trips (PREDTRIPS)

Note: each record is a unique origin-destination combination and there are M x N records where M is the number of origin zones (including the external zone) and N is the number of destination zones.

Save links

The top predicted origin-destination trip links can be saved as separate line objects for use in a GIS. Specify the output file format (ArcView '.shp', MapInfo '.mif' or Atlas*GIS '.bna') and the file name. For each mode, the prefix 'TripMode' will be prefaced before the file. For example, if the name provided by the user is "robberies" and if there are three travel modes modeled, then there will be three graphical output files (TripMode1robberies.shp/mif/bna; TripMode2robberies.shp/mif/bna; TripMode3robberies.shp/mif/bna).

Save top links

Because the output file is very large (number of origin zones x number of destination zones), the user can select a sub-set of zone combinations with the most predicted trips. Indicating the top K links will narrow the number down to the most important ones. The default is the top 100 origin-destination combinations. Each output object is a line from the origin zone to the destination zone with a TripMode prefix where " is the mode number. The prefix is placed before the output file name. The graphical output includes:

1. An ID number from 1 to K, where K is the number of links output (ID)
2. The feature prefix (ODT)
3. The origin zone (ORIGIN)
4. The destination zone (DEST)
5. The X coordinate for the origin zone (ORIGINX)
6. The Y coordinate for the origin zone (ORIGINY)
7. The X coordinate for the destination zone (DESTX)
8. The Y coordinate for the destination zone (DESTY)
9. The number of predicted trips for that combination (PREDTRIPS)
10. The distance between the origin zone and the destination zone.

Save points

Intra-zonal trips (trips in which the origin and destination are the same zone) can be output as separate point objects as an ArcView '.shp', MapInfo '.mif' or Atlas*GIS '.bna' file. Again, the top K points are output (default=100). Each output object is a point representing an intra-zonal trip with a TripModePoints prefix where " is the mode number. The prefix is placed before the output file name. For example, if the name provided by the user is "robberies" and if there are three travel modes modeled, then there will be three graphical output files (TripModePoints1robberies.shp/mif/bna; TripModePoints2robberies.shp/mif/bna; TripModePoints3robberies.shp/mif/bna).

The graphical output for each includes:

1. An ID number from 1 to K, where K is the number of links output (ID)
2. The feature prefix (POINTSODT)
3. The origin zone (ORIGIN)
4. The destination zone (DEST)
5. The X coordinate for the origin zone (ORIGINX)
6. The Y coordinate for the origin zone (ORIGINY)
7. The X coordinate for the destination zone (DESTX)
8. The Y coordinate for the destination zone (DESTY)
9. The number of predicted trips for that combination (PREDTRIPS)

Calibrate Mode Split: I-III

For each mode (up to five), the impedance parameters have to be set. There are three pages for this. "Calibrate mode split: I" covers modes 1 and 2. "Calibrate mode split: II' covers modes 3 and 4. "Calibrate mode split: III" covers mode 5. For each mode, the user should indicate whether the mode is to be used, the name to be used for the mode, whether a default impedance will be calculated directly or if it should be constrained to a network, and the specific impedance model used. If any mode is not used, then it will not be part of the calculations. Use only those modes that are relevant, but, also, be sure not to leave out any important ones.

The following instructions apply to **each** of the five modes.

Mode #

Check the box if the mode is to be used.

Label

Put in a label for the mode. Default names are provided (walk, bicycle, drive, bus, train), but the user is not required to use those.

Impedance constraint

The impedance will be calculated either directly or is constrained to a network. The default impedance is defined with the type of distance measurement specified on the Measurement Parameters page (under Data setup). On the other hand, if the impedance is to be constrained to a network, then the network has to be defined.

Default

The default impedance is that specified on the Measurement parameters page. If direct distance is the default distance (on the measurement parameters page), then all impedances are calculated as a direct distance. If indirect distance is the default, then all impedances are calculated as indirect (Manhattan) distance. If network distance is the default, then all impedances are calculated using the specified network and its parameters; travel impedance will automatically be constrained to the network under this condition.

Constrain to network

An impedance calculation should be constrained to a network where there are limited choices. For example, a bus trip requires a bus route; if a particular zone is not near an existing bus route, then a direct distance calculation will be misleading since it will probably underestimate true distance. Similarly, for a train trip, there needs to be an existing train route. Even for walking, bicycling and driving trips, an existing network might produce a more realistic travel impedance than simply assuming a direct travel path. If the impedance calculation is to be constrained to a network, then the network must be defined.

Check the 'Constrain to network' box and click on the 'Parameters' button. The network file can be either a shape line or polyline file (the default) or another file, either dBase IV 'dbf', Microsoft Access 'mdb', Ascii 'dat', or an ODBC-compliant file. If the file is a shape file, the routine will know the locations of the nodes. All the user needs to do is identify a weighting variable, if used.

For a dBase IV or other file, the X and Y coordinate variables of the end nodes must be defined. These are called the "From" node and the "End" node, though there is no

particular order. An optional weight variable is allowed for both a shape or dbf file. The routine identifies nodes and segments and finds the shortest path. By default, the shortest path is in terms of distance though each segment can be weighted by travel time, travel speed, or generalized cost; in the latter case, the units are minutes, hours, or unspecified cost units.

Finally, the number of graph segments to be calculated is defined as the network limit. The default is 50,000 segments. Be sure that this number is greater than the number of segments in your network. Note: using network distance for distance calculations can be a very slow process (i.e., taking up to several days for calculating an entire matrix).

Minimum absolute impedance

If the mode is constrained to a network, an additional constraint is needed to ensure realistic allocations of trips. This is the minimum absolute impedance between zones. The default is 2 miles. For any zone pair (an origin zone and a destination zone) that is closer together (in distance, time interval, or cost) than the minimum specified, no trips will be allocated to that mode. This constraint is to prevent unrealistic trips being assigned to intra-zonal trips or trips between nearby zones. CrimeStat uses three impedances for a constrained network: 1) the impedance from the origin zone to the nearest node on the network (e.g., nearest rail station); b) the impedance along the network to the node nearest to the destination; and c:) the impedance from that node to the destination zone. Since most impedance functions for a mode constrained to a network will have the highest likelihood some distance from the origin, it's possible that the mode would be assigned to, essentially, very short trips (e.g., the distance from an origin zone to a rail network and then back again might be modeled as a high likelihood of a train trip even though such a trip is very unlikely).

For each mode that is constrained to a network, specify the minimum absolute impedance. The units will be the same as that specified by the measurement units. The default is 2 miles. If the units are distance, then trips will only be allocated to those zone pairs that are equal to or greater in distance than the minimum specified. If the units are travel time or speed, then trips will only be allocated to those zone pairs that are farther apart than the distance that would be traveled in that time at 30 miles per hour. If the units are cost, then the routine calculates the average cost per mile along the network and only allocates trips to those zone pairs that are farther apart than the distance that would be traveled at that average cost.

Impedance function

The model split routine can use two different travel distance functions: 1) An already-calibrated distance function; and 2) A mathematical formula. The default is a mathematical formula.

Use an already-calibrated distance function

If a travel distance function for the specific mode has already been calibrated (see 'Calibrate impedance function' under trip distribution), the file can be directly input into the routine. That routine can be used to calibrate a function if there are data on origins and destinations for individual travel modes.

Browse

The user selects the name of the already-calibrated travel distance function. CrimeStat reads dbase 'dbf', ASCII text 'txt', and ASCII data 'dat' files.

Use a mathematical formula

A mathematical formula can be used instead of a calibrated distance function. To do this, it is necessary to specify the type of distribution. There are five mathematical models that can be selected:

1. Negative exponential - the default
2. Normal distribution
3. Lognormal distribution
4. Linear distribution
5. Truncated negative exponential

For each mathematical model, two or three different parameters must be defined:

1. For the negative exponential, the coefficient and exponent. This is the default and default values are provided.

2. For the normal distribution, the mean distance, standard deviation and coefficient,

3. For lognormal distribution, the mean distance, standard deviation and coefficient,

4. For the linear distribution, an intercept and slope; and

5. For the truncated negative exponential, a peak distance, peak likelihood, intercept, and exponent.

Measurement unit

The routine can calculate impedance in four ways, by:

1. Distance (miles, nautical miles, feet, kilometers, and meters)
2. Travel time (minutes, hours)

3. Speed (miles per hour, kilometers per hour)
4. General travel costs (unspecified units).

Specify the appropriate units. The default is distance in miles.

Network Assignment

Network assignment involves assigning predicted trips (either all trips or by separate travel modes) to a particular route on a network. That is, for every origin-destination trip link, a particular route is found along a network (roadway, transit). The routine does this using a shortest path algorithm. The user must provide the network with its parameters. The routine allows the definition of one-way streets in order to produce a more realistic representation. In the current version, the assignment routine works on one predicted trip file at a time.

Predicted Origin-Destination File

The predicted origin-destination trip file is a file that lists the predicted number of trips from every origin zone to every destination zone. Select the predicted trip file (i.e., the predicted origin-destination trip file) by clicking on the 'Browse' button.

Origin ID

Specify the origin zone ID variable in the data file. The default name is ORIGIN.

Origin_X

Specify the name of the variable for the X coordinate of the origin zone. The default name is ORIGINX.

Origin_Y

Specify the name of the variable for the Y coordinate of the origin zone. The default name is ORIGINY.

Destination ID

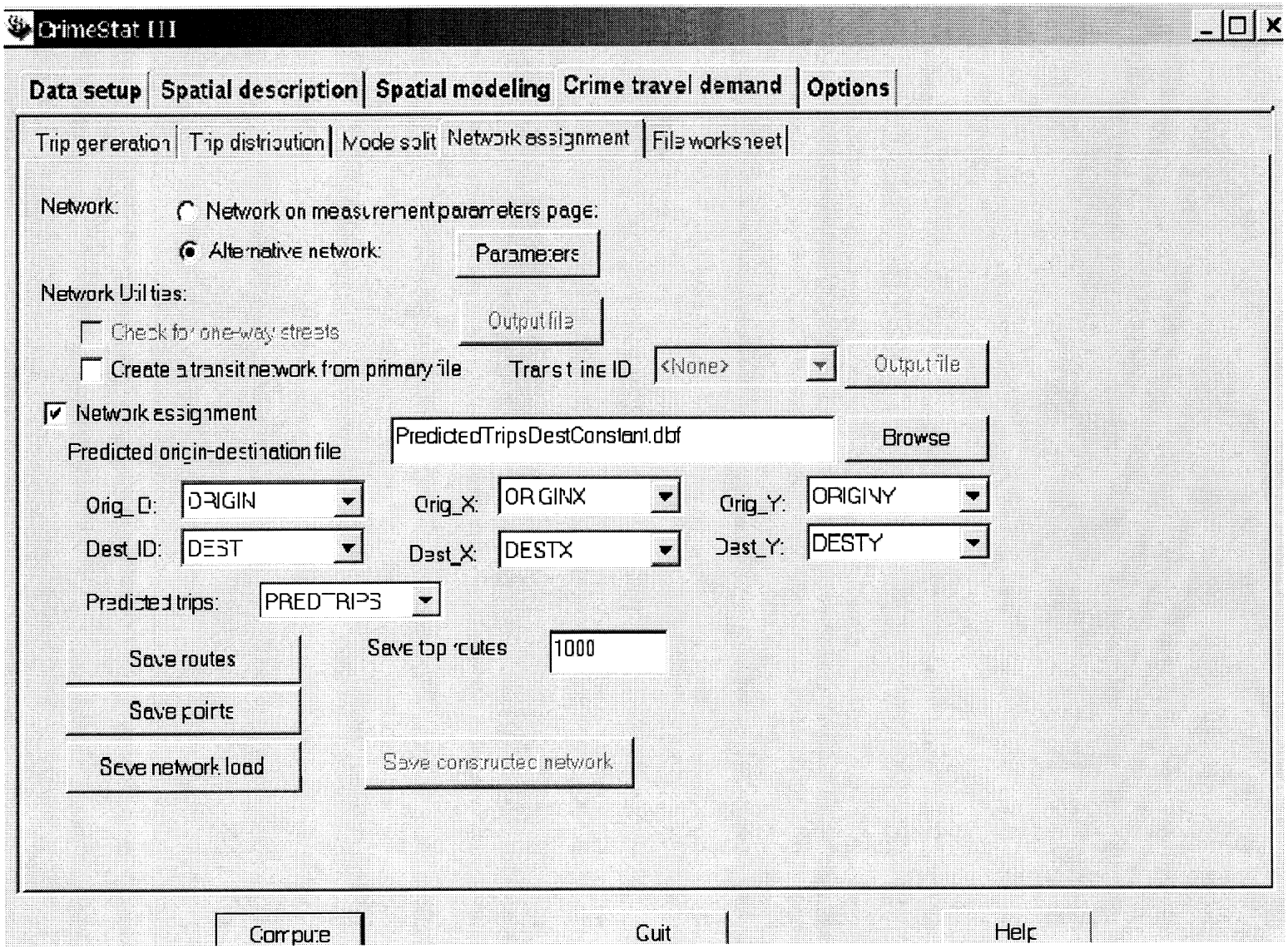
Specify the destination zone ID variable in the data file. The default name is DEST.

Destination_X

Specify the name of the variable for the X coordinate of the destination zone. The default name is DESTX.

Figure 2.16:

Network Assignment Screen



Destination_Y

Specify the name of the variable for the Y coordinate of the destination zone. The default name is DESTY.

Predicted trips

Specify the variable for the predicted number of trips. The default name is PREDTRIPS

Network Used

The network assignment routine requires a network from which the shortest path from every origin zone to every destination zone can be computed. To run this routine, check the 'Network assignment' box at the top of the page.

The user must specify the network that is to be used. There are two choices. First, if a network was defined on the Measurement parameters page (Data setup), that network can be used to calculate the shortest path. Second, whether a network has been defined on the Measurement parameters page or not, an alternative network can be selected.

Network on measurement parameters page

Check the 'Network on Measurement parameters page' box to use that network. All the parameters will have been defined for that setup (see Measurement parameters page).

Alternative network

If an alternative network is to be used, it must be defined. Check the 'Alternative network' box and click on the 'Parameters' button.

> Note: if a network is also used on the Measurement Parameters page, then it must be defined there as well. CrimeStat will check whether that file exists; if it does not, the routine will stop and an error message will be issued. Therefore, if an alternative network is used, the user should probably change the distance measurement on the Measurement Parameters page to direct or indirect distance.

Type of network

Network files can bi-directional (e.g., a TIGER file) or single directional (e.g., a transportation modeling file). In a bi-directional file, travel can be in either direction. In a single directional file, travel is only in one direction. Specify the type of network to be used.

Network input file

The network file can either be a shape file (line, polyline, or polylineZ file) or another file, either dBase IV 'dbf', Microsoft Access 'mdb', Ascii 'dat', or an ODBC-compliant file. The default is a shape file. If the file is a shape file, the routine will know the locations of the nodes. For a dBase IV or other file, the X and Y coordinate variables of the end nodes must be defined. These are called the "From" node and the "End" node. An optional weight variable is allowed for all types of file0073. The routine identifies nodes and segments and finds the shortest path. If there are one-way streets in a bi-directional file, the flag fields for the "From" and "To" nodes should be defined.

Network weight field

Normally, each segment in the network is not weighted. In this case, the routine calculates the shortest distance between two points using the distance of each segment. However, each segment can be weighted by travel time, speed or travel costs. If travel time is used for weighting the segment, the routine calculates the shortest time for any route between two points. If speed is used for weighting the segment, the routine converts this into travel time by dividing the distance by the speed. Finally, if travel cost is used for weighting the segment, the routine calculates the route with the smallest total travel cost. Specify the weighting field to be used and be sure to indicate the measurement units (distance, speed, travel time, or travel cost) at the bottom of the page. If there is no weighting field assigned, then the routine will calculate using distance.

From one-way flag and To one-way flag

One-way segments can be identified in a bi-directional file by a 'flag' field (it is not necessary in a single directional file). The 'flag' is a field for the end nodes of the segment with values of '0' and '1'. A '0' indicates that travel can pass through that node in either direction whereas a '1' indicates that travel can only pass from the other node of the same segment (i.e., travel cannot occur from another segment that is connected to the node). The default assumption is for travel to be allowed through each node (i.e., there is a '0' assumed for each node). For each one-way street, specify the flags for each end node. A '0' allows travel from any connecting segments whereas a '1' only allows travel from the other node of the same segment. Flag fields that are blank are assumed to allow travel to pass in either direction.

FromNode ID and ToNode ID

If the network is single directional, there are individual segments for each direction. Typically, two-way streets have two segments, one for each direction. On the other hand, one-way streets have only one segment. The FromNode ID and the ToNode ID identify from which end of the segment travel should occur. If no FromNode ID and ToNode ID is defined, the routine will chose the first segment of a pair that it finds, whether travel is in the right or wrong direction. To identify correctly travel direction, define the FromNode and ToNode ID fields.

Type of coordinate system

The type of coordinate system for the network file is the same as for the primary file.

Measurement unit

By default, the shortest path is in terms of distance. However, each segment can be weighted by travel time, travel speed, or travel cost.

1. For travel time, the units are minutes, hours, or unspecified cost units.

2. For speed, the units are miles per hour and kilometers per hour. In the case of speed as a weighting variable, it is automatically converted into travel time by dividing the distance of the segment by the speed, keeping units constant.

3. For travel cost, the units are undefined and the routine identifies routes by those with the smallest total cost.

Network graph limit

Finally, the number of graph segments to be calculated is defined as the network limit. The default is 50,000 segments. Be sure that this number is slightly greater than the number of segments in your network. Note: using network distance for distance calculations can be a slow process, for example taking up to several hours for calculating an entire matrix. Use only if more precision is needed or for the network assignment routine in the crime travel demand module.

Network Utilities

There are two network utilities that can be used.

Check for one-way streets

First, there is a routine that will identify one-way streets if the network is single directional. In a single directional file, one-way streets do not have a reciprocal pair (i.e., a segment traveling in the opposite direction). This is indicated by a reciprocal pair of ID's for the "From" and "To" nodes. If checked, the routine identifies those segments that do not have reciprocal node ID's. The network is saved with a new field called "Oneway". One-way segments are assigned a value of '1' value and two-way segments are assigned a value of '0'. The output is saved as an ArcView '.shp', MapInfo '.mif' or Atlas*GIS '.bna' file.

Create a transit network from primary file

Second, there is a routine that will create a network from the primary file. This is useful for creating a transit network from a collection of bus stops (bus network) or rail stations (rail network). If checked, the routine will read the primary file and will draw lines

from one point to another in the order in which the points appear in the primary file. Note, it is essential to order the points in the same order in which the network should be drawn (otherwise, an illogical network will be obtained). It is easy to do this in a spreadsheet program.

Transit Line ID

The routine can handle multiple lines, for example different rail lines or bus routes (e.g., Line A, Line B, Route 1, Route 2). In the primary file, the points must be grouped by lines, however, and must be classified by an ID field. Within each group, the points must be arranged in order of occurrence; the routine will draw a lines from one point to another in that order. In the Transit Line ID field, indicate which variable is the classification variable.

The output is saved as an ArcView '.shp', MapInfo '.mif' or Atlas*GIS '.bna' file.

Network Output

There are three types of output for the network assignment routine. First, the most frequent inter-zonal (i.e., trips between different zones) routes can be output as polylines. Second, the most frequent intra-zonal (i.e., trips within the same zone) routines can be output as points. Third, the entire network can be output in terms of the total number of trips that occur on each segment (network load).

Save routes

The shortest routes can be saved as separate polyline objects for use in a GIS. Specify the output file format (ArcView '.shp', MapInfo '.mif' or Atlas*GIS '.bna') and the file name.

Save top routes

Because the output file is very large (number of origin zones x number of destination zones), the user can select a zone-to-zone route with the most predicted trips. The default is the top 100 origin-destination combinations. Each output object is a line from the origin zone to the destination zone with a Route prefix. The prefix is placed before the output file name. The graphical output includes:

1. An ID number from 1 to K, where K is the number of links output (ID)
2. The feature prefix (ROUTE)
3. The origin zone (ORIGIN)
4. The destination zone (DEST)
5. The X coordinate for the origin zone (ORIGINX)
6. The Y coordinate for the origin zone (ORIGINY)
7. The X coordinate for the destination zone (DESTX)
8. The Y coordinate for the destination zone (DESTY)

9. The number of trips on that particular route (FREQ)
10. The distance between the origin zone and the destination zone (DIST).

Save points

Intra-zonal trips (trips in which the origin and destination are the same zone) can be output as separate point objects as an ArcView '.shp', MapInfo '.mif' or Atlas*GIS '.bna' file. Again, the top K points are output (default=100). Each output object is a point representing an intra-zonal trip with a RoutePoints. The prefix is placed before the output file name.

The graphical output for each includes:

1. An ID number from 1 to K, where K is the number of links output (ID)
2. The feature prefix (ROUTEPoints)
3. The origin zone (ORIGIN)
4. The destination zone (DEST)
5. The X coordinate for the origin zone (ORIGINX)
6. The Y coordinate for the origin zone (ORIGINY)
7. The X coordinate for the destination zone (DESTX)
8. The Y coordinate for the destination zone (DESTY)
9. The number of trips on that particular route (FREQ)
10. The distance between the origin zone and the destination zone (DIST).

Save network load

It is also possible to save the total network load as an ArcView '.shp', MapInfo '.mif' or Atlas*GIS '.bna' file. This is the total number of trips on each segment of the network. The routine takes every origin zone to destination zone combination and sums the number of trips that occur on each segment of the network.

Click on the "Save output network" box and specify a file name for the output.

File Worksheet

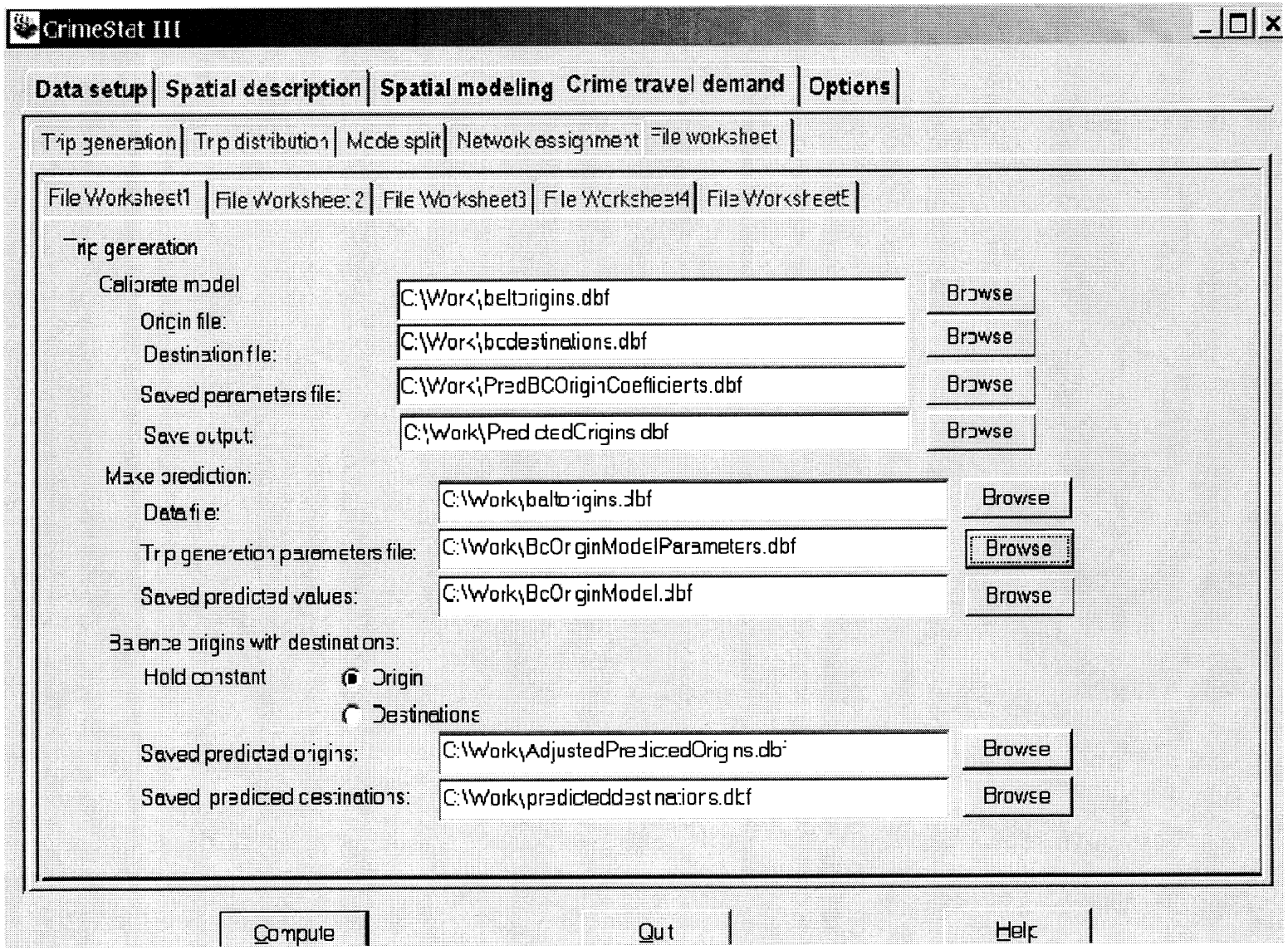
The file worksheet allows the saving of names for the files in the crime travel demand module. Because there are a large number of files used (many used in multiple routines), saving the names will make it easier to keep track of the files. The file worksheet is not required for use of the crime travel demand module. But it is recommended for remembering the files in a particular travel demand model. There are five worksheets for keeping track of the different routines.

File Worksheet 1

This worksheet keeps track of the files used in the trip generation step. These include:

Figure 2.17:

File Worksheet Screen



Trip generation

Calibrate model
Make prediction
Balance origins with destinations

File Worksheet 2

This worksheet keeps track of some used in the trip distribution step, in particular the observed trip distribution and trip distribution model setup. These include:

Trip distribution
Describe origin-destination trips
Setup origin-destination model

File Worksheet 3

This worksheet also keeps track files used in the trip distribution step, in particular the trip distribution model and the comparison between the observed and predicted trip length distributions. These include:

Origin-destination model
Compare observed and predicted origin-destination trip lengths

File Worksheet 4

This worksheet keeps track of the files used in the mode split step, including the mode split setup and modes 1-3. These include:

Mode split
Setup for mode split
Modes modeled
Modes 1-3

File Worksheet 5

This worksheet keeps track of the remaining files used in the mode split step (modes 4-5) as well as network assignment routine. These include:

Mode split (continued)
Modes modeled
Modes 4-5
Network assignment

V. Options

The options allow the saving of parameters, the changing of tab colors for the three sections and the outputing of simulated data for the Monte Carlo simulation routines.,

Saving Parameters

All the input parameters can be saved. In the options section, there is a 'Save parameters' button. A parameters file must have a 'param' extension. A saved parameters file can be re-loaded with the 'Load parameters' button.

Colors

The colors for each of the three sections can be changed by selecting the appropriate tab and choosing a color from the color spectrum.

Dump Simulation Data

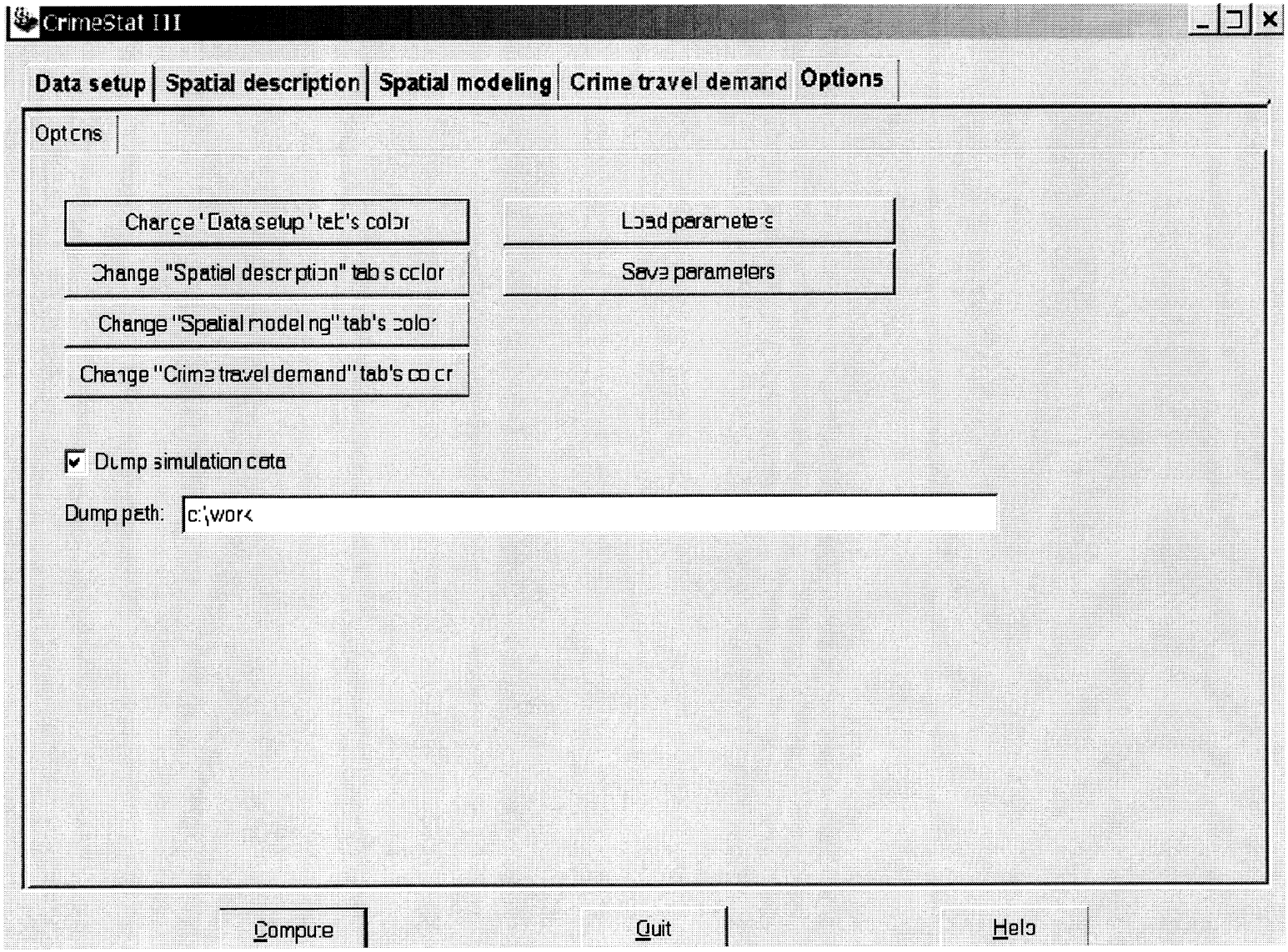
When running a Monte Carlo simulation with the Ripley's K, the Nearest Neighbor Hierarchical Clustering, the Risk-adjusted Nearest Neighbor Hierarchical Clustering, the STAC, the Mantel, or the Knox routines, the data can be output to dbf files. Each simulation run is output with the name Sim_data<*I*>.dbf where <*I*> is the run number (e.g., Sim_data4.dbf).

VI. Dynamic Data Exchange (DDE) Support

CrimeStat supports Dynamic Data Exchange (DDE). See Appendix A in the documentation or the online help screens for more information.

Figure 2.18:

Options Page Screen



Chapter 3
Entering Data into *CrimeStat*

The graphic user interface of *CrimeStat* is a tabbed form (figure 3.1). There are five groups of functions: Data setup, Spatial description, Spatial modeling, Crime Travel Demand, and Options. Each group, in turn is made up of several sets of routines:

Data Setup

Primary file	Data file of incident/point locations (Required)
Secondary file	Secondary data file of incident/point locations
Reference file	File for referencing interpolations
Measurement Parameters	Areal and linear characteristics of study area

Spatial Description

Spatial Distribution	Basic characteristics of the incident distribution
Distance Analysis I	Characteristics of the distances between points
Distance Analysis II	Matrix distances
'Hot Spot' Analysis I	Tools for identifying 'Hot Spots'
'Hot Spot' Analysis II	More tools for identifying 'Hot Spots'

Spatial Modeling

Interpolation	Three-dimensional density analysis
Journey-to-crime Analysis	Analyzing the travel behavior of serial offenders
Space-time Analysis	The interaction between space and time

Crime Travel Demand

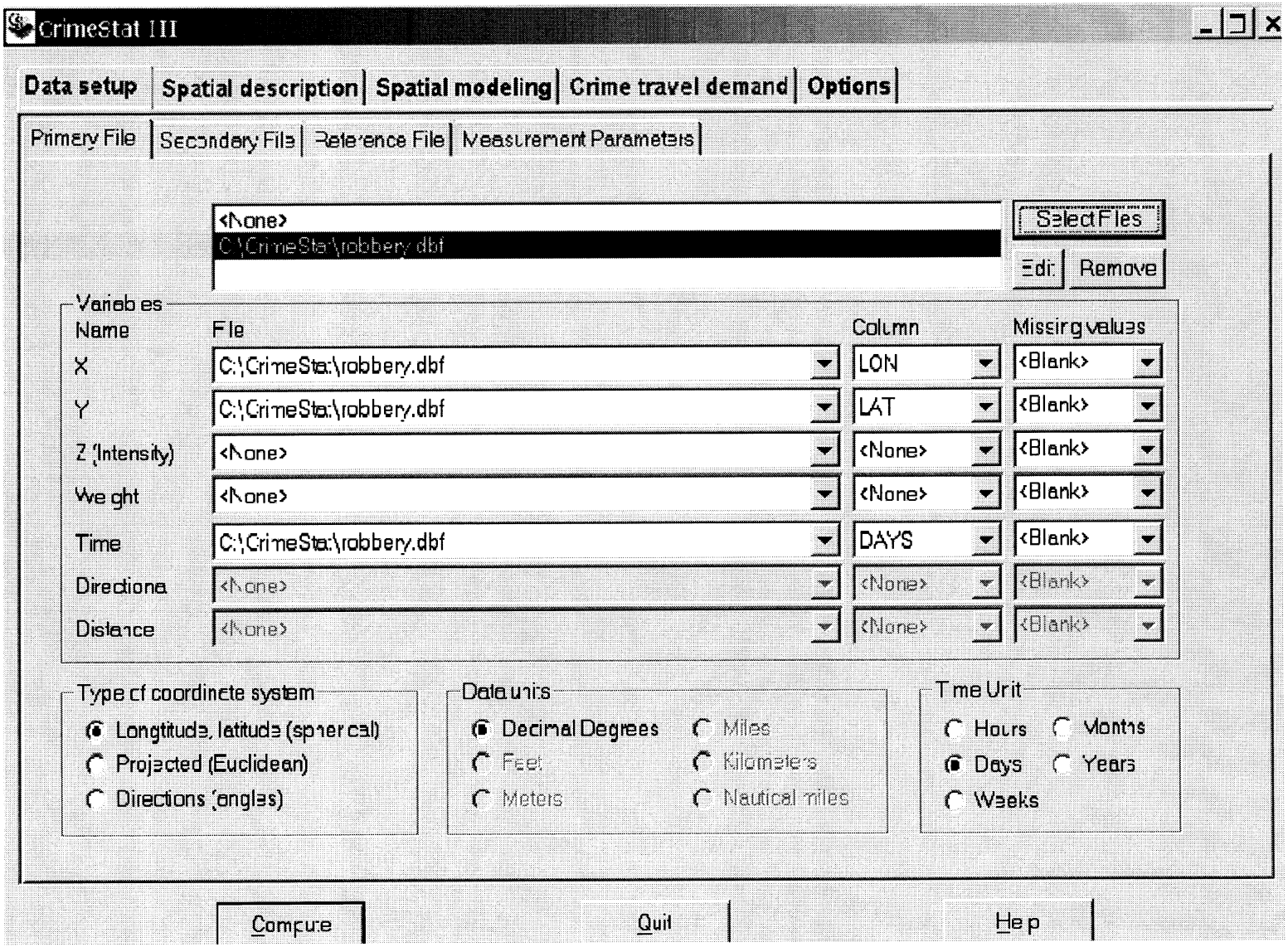
Trip generation	Models of crime origins and crime destinations
Trip distribution	Model of trips between origins and destinations
Mode split	Model of travel mode used for trips
Network assignment	Model of route taken for trips
File worksheet	Worksheet of file names

Options

Save parameters	Save the data setup parameters
Load parameters	Load already-saved parameters file
Colors	Change the color of tabs
Simulation	Output simulation data

This section discusses the Data Setup tabs.

Figure 3.1: *CrimeStat* User Interface



Required Data

CrimeStat can input data in several formats - ASCII, *dbase III/ IV* 'dbf', *ArcView* 'shp', *MapInfo* 'dat', and files that support the ODBC standard, such as *Excel®*, *Lotus 1-2-3®*, *Microsoft Access®*, and *Paradox®*. It is essential that the files have X and Y coordinates as part of their structure. The program assumes that the assigned X and Y coordinates are correct. It reads a file - ASCII, 'dbf' or 'shp' and takes the given X and Y coordinates.

If you read an *ArcView* shape file, the incident's X and Y coordinates are automatically added as the first fields in the primary file by *CrimeStat*. If you use any other type of file you must add X and Y coordinates to the file. To automate this in

ArcView, add the Avenue extension *Coordinate Utility V1.0* (available in Arc Scripts) to your extension list. To do this in *MapInfo* add the KGM utility *Table Geography* as a tool. Both work great. It is a good idea to add the X and Y coordinates to any file. They are useful for analysis in other programs and allow for easy reconstruction of the file if the geo-coding is lost.

Coordinates

CrimeStat analyzes point data, defined geographically by X and Y coordinates. These X/Y coordinates represent a single location where either an incident occurred (e.g., a burglary) or where a building or other object can be represented as a single point. A point will have X and Y coordinates in a spherical or Cartesian system. In a spherical coordinate system, each point can be defined by longitude (for X) and latitude (for Y). In a projected coordinate system, such as State Plane or UTM, each X and Y is defined by feet or meters from an arbitrary reference origin. *CrimeStat* can handle both spherical and projected points. For some uses, coordinates can be polar, that is defined as angles from an arbitrary reference vector, usually direct north.[1] One of the routines in the program calculates the angular mean and variance of a collection of angles.

Point data can be obtained from a number of sources. The most frequent would be the various incident data bases stored by a police department, which could include calls for service, crime reports, or closed cases. Other sources of incident data can include secondary data from other agencies (e.g., hospital records, emergency medical service records, locations of businesses) or even sampled data (Levine and Wachs, 1986a; 1986b). There are also point data from broadcast sources, such as radios, televisions, or microwaves.

To read projected coordinates into *CrimeStat*, the user doesn't need to define the particular projection (other than to indicate that the coordinates are projected). *ArcView* will output the objects in the projected units so that they can be read directly into that program or into *ArcGIS*. However, to output calculated objects to *MapInfo* requires the definition of the specific projection used.[2] See chapter 4 for the first examples of outputing objects.

Intensities and weights

For some uses, points can have *intensity* values or *weights*. These are optional inputs in *CrimeStat*. An *intensity* is a value assigned to a point location aside from the X/Y coordinates. It is another variable, typically denoted as a Z-value. For example, if the point location is the location of a police station, then the intensity could be the number of calls for service over a month at that station. Or, to use census geography, if the point is the centroid of a census tract, then the intensity could be the population of that census tract. In other words, an intensity is a variable assigned to a particular location.

Some of the routines in *CrimeStat* require an intensity value (e.g., the spatial autocorrelation indices) and others can utilize a point location with an intensity value assigned (e.g., kernel density interpolation). If no intensity value is assigned, the routines which require it cannot be run while the routines which can utilize it will assume that the intensity is 1 (i.e., that all points have equal intensity).

A *weight* occurs when different point locations are to receive differential statistical treatment. For example, if a police department has designated different areas for service, for example 'urban' and 'rural', a value can be assigned for each of these areas (e.g., '1' for urban and '2' for rural). Most of the routines in *CrimeStat* will use the weights in the calculations. Weights would be useful if different zones are to be evaluated on the basis of another variable. For example, suppose a police department has divided its service area into urban and rural. In the rural part, there are twice as many patrol officers assigned per capita than in the urban areas; the higher population densities in the urban areas are assumed to compensate for the longer travel distances in the rural areas. Let's assume that all crimes occurring in the rural areas receive a weight of 2 while those in the urban area receive a weight of 1. The police department then wants to estimate the density of household burglaries relative to the population using the duel kernel density function (see Chapter 7). But, to reflect the differential assignment of police officers, the analysts use the service area as a weight. The result would be a per capita estimate of burglary density (i.e., burglaries per person), but weighted by the service area. It would provide an estimate of burglary risk adjusted for differential service in rural and urban areas. In most cases, there will no weights, in which case, all points are assumed to have an equal weight of '1'.

It is possible to have both intensities and weights, although this would be rare. For example, if the X and Y coordinates are the centroids of census tracts, a third variable - the total population of each census tract could be an intensity. There could also be an weighting based on service area. In calculating the Moran's I spatial autocorrelation index, the total population is used as an intensity while the service area is used as a weight. In this case, *CrimeStat* calculates a weighted Moran's I spatial autocorrelation.

But the use of both an intensity *and* a weight would be less common. For most of the statistics, a variable could be used as *either* a weight or an intensity, and the results will be the same. However, be careful in assigning the same variable as both an intensity and a weight. In such instances, cases may end up being weighted twice, which will produce distorted results.[3]

Time Measures

CrimeStat now includes routines for analyzing spatial characteristics in relation to time. Many serial crime incidents occur in a short period of time. For example, a group of car thieves may steal cars from a neighborhood over a very short period of time, for example a few days. Thus, there is often an interaction between a concentrated spatial pattern of events occurring in a short time period. Because of this, police departments routine collect information on the time of the event, the day and time.

There are three routines which analyze spatial concentration in relation to time: the Knox index, the Mantel index, and a correlated walk model. But for using any of these routines, the user has to define time in a consistent manner. Both the primary and secondary files can allow a time variable. However, these have to be defined in a *consistent* manner for all records in a file. There are five time periods that are allowed:

Hour
Day (default)
Week
Month
Year

The default is 'day'. That is, the program will assume that any time variable is in days, either an arbitrary number of days (e.g., days from January 1st) or the number of days from January 1, 1900, which is the default time reference for most computer systems. If the time unit is not in days, the user needs to indicate the appropriate unit.

Missing Value Codes

Unfortunately, data is frequently messy. In most police departments, the crime incident data base is being continually updated, daily and, perhaps, hourly. At any one time, many of the records will not have been geocoded or will have been incompletely geocoded.

Blank records

CrimeStat allows the inclusion of codes for missing values, that is values of eligible fields that are not complete or are not correct. These codes are applied to the fields defined on the primary or secondary data sets (X, Y, weight, intensity). Automatically, *CrimeStat* will exclude records with blank fields or with fields having any non-numeric value (e.g., alphanumeric characters, #, *) for the eligible fields. The statistics will be calculated only on those records which have eligible numerical values. Fields for other variables in the data base that are not defined in the primary and secondary data sets will be ignored.

Other missing value codes

In addition to blank and non-numeric values, *CrimeStat* can exclude any other value that has been used for a missing values code (e.g., 0, -1, 99). That is, if the program encounters a field with a missing value code, it will exclude that record from the calculations. Next to the X, Y, weight and intensity fields on both the primary and secondary files is a missing values code box. The default has been set to blank. That is, if *CrimeStat* finds no information in a field, it will ignore that record. However, there are eight options that can be selected:

1. ***<blank>*** fields are automatically excluded. This is the default;
2. ***<none>*** indicates that no records will be excluded. If there is a blank field, *CrimeStat* will treat it as a 0;
3. ***0*** is excluded;
4. ***-1*** is excluded;
5. ***0 and -1*** indicates that both 0 and -1 will be excluded;
6. ***0, -1 and 9999*** indicates that all three values (0, -1, 9999) will be excluded;
7. ***Any*** other numerical value can be treated as a missing value by typing it (e.g., 99); and
8. ***Multiple*** numerical values can be treated as missing values by typing them, separating each by commas (e.g., 0, -1, 99, 9999, -99).

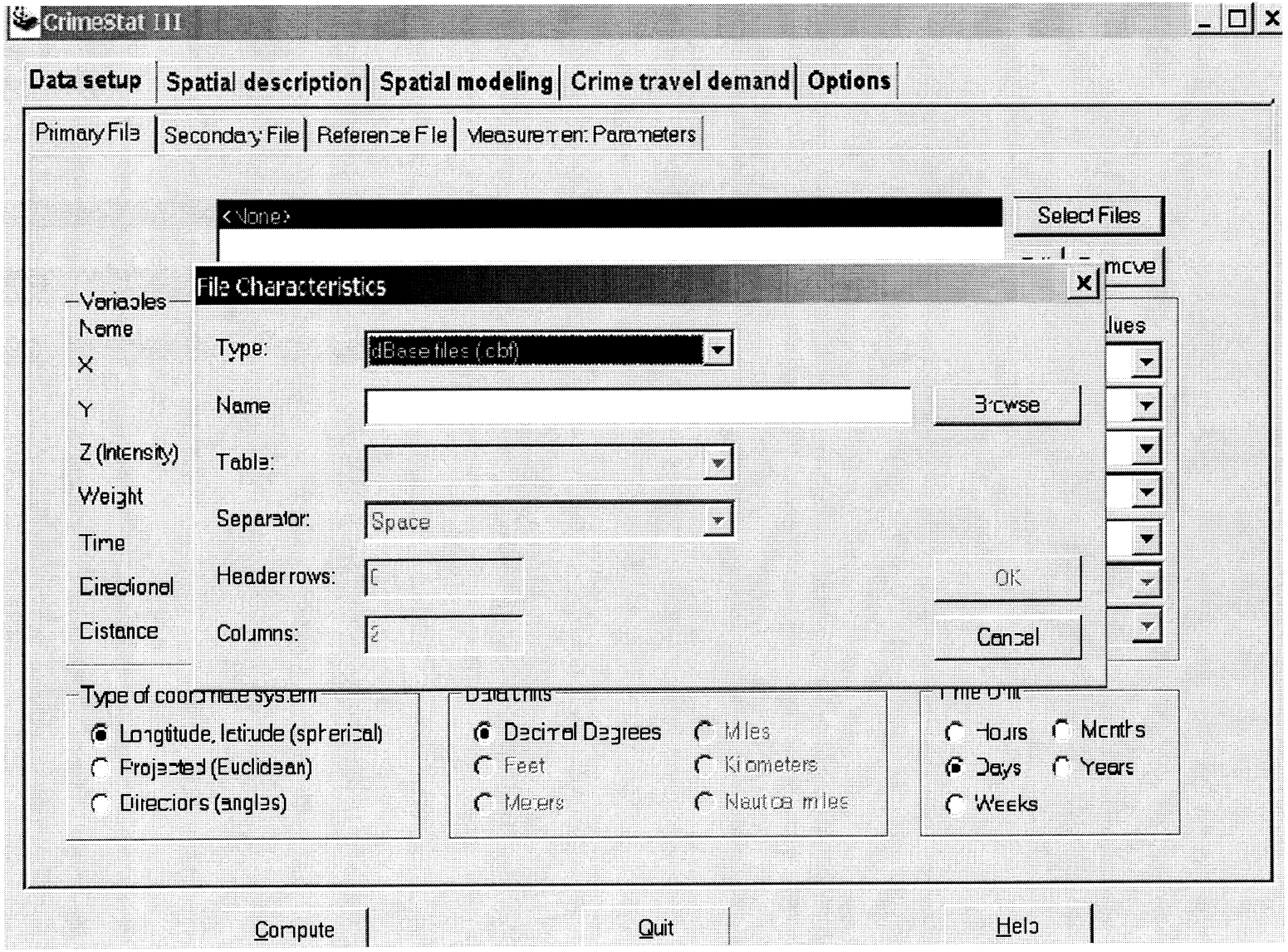
It is important for users to understand their data sets prior to using *CrimeStat*. If the data are 'clean', that is all X/Y fields are populated with correct values as are all weight/intensity fields (if used), then the program will have no problems running routines. On the other hand, in large administrative data bases, such as in most police departments, there will be many records that are incomplete or have missing values codes (e.g., 0). Unless *CrimeStat* is told what are the missing value codes, with the exception of blank or non-numeric values, it will include them in the calculations. For example, some data base programs put a 0 for an X or Y field which has not been geocoded. *CrimeStat* doesn't know that the 0 is a missing value and will use it in calculations since 0 is a perfectly good number. It is important that users either clean their data thoroughly or define the missing value codes completely for the primary and secondary files.

Primary File

The *Primary File* is required and provides the coordinates of points of incidents. On the primary file tab, the user must first click on *Select Files*. A dialog box appears that allows the user to select which of six file formats applies to the primary file (Figure 3.2). For each of the file formats, the user must define two characteristics - the type of file (ASCII, '.dbf', 'dat', '.shp', 'mdb', or ODBC) and the name of the file. There is a browse window which allows the user to find the file.

In developing this program, we have targeted it towards users of *ArcView*, *MapInfo* and *Atlas*GIS*. These GIS programs either store their attribute data in *dBase III/IV* format in a file with a 'dbf' extension (e.g., precinct1.dbf) or can read and write directly 'dbf'

Figure 3.2: File Format Selection



Linking *CrimeStat III* to *MapInfo*

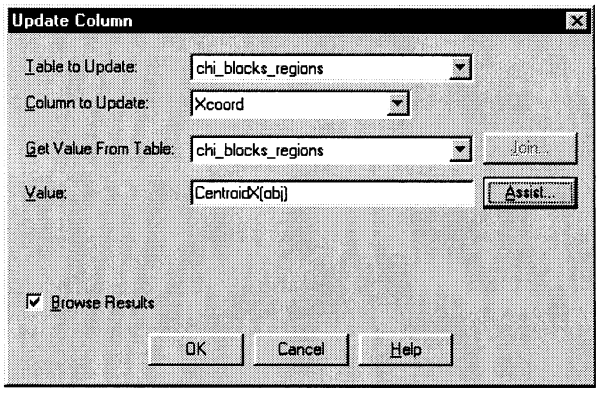
Richard Block
Professor of Sociology and Criminal Justice
Loyola University of Chicago

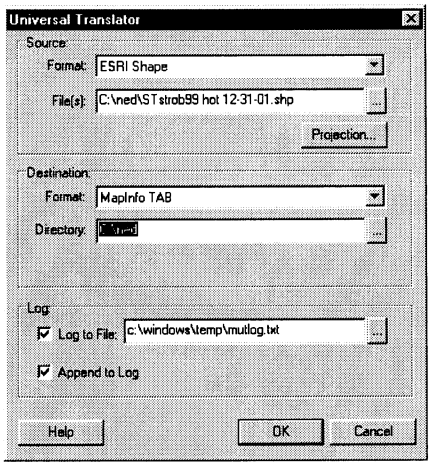
MapInfo point 'dat' files can be inputted to *CrimeStat* as primary or secondary files. However, x and y coordinates need to be added to the file. If the point data are in latitude/longitude, this is easily done with a free extension, *Table Geography*, available through the Directions Magazine website as part of the KGM utilities at: http://www.directionsmag.com/tools/Default.asp?a=file&ID=11 . Add this extension to your MapInfo toolbox. Click on the tool. You will first be asked for a table to add coordinates. The program automatically adds columns for longitude and latitude.

If you are using another projection, you will need to add and update columns to your file. To do this, add columns for x and y coordinates to your table (Table–>Maintenance–>Table Structure–>Add Field) in an appropriate numeric format for your projection. As shown in left figure, update these new columns with the coordinates (Table–>update column). Choose the data file and column that you want to update. Next, click assist and then functions. Choose *centroidx* to update the horizontal field and *centroidy* to update the vertical field. Within *CrimeStat*, identify the file type as *MapInfo* 'dat'.

For some *CrimeStat* require a reference file. These are identified by the lower-left and upper-right coordinates of a rectangle. To derive these coordinates, make the top map (cosmetic) layer editable. Draw a rectangle identifying the study area. Select the rectangle. Convert it to a region (objects-→ convert to region). Double click on the rectangle, and the appropriate coordinates and area of the rectangle will appear.

Several *CrimeStat* routines output geographic features that can be added as a layer in *MapInfo*. To output these graphics, first designate an output file. If you are working in longitude/latitude, choose a *MapInfo* 'mif 'file as output. In *MapInfo*, import the mif file (Table–>Import), and open the file as a layer in your map. For any other projection, output to an *ESRI* shape file and use the Universal Translator tool (right figure) to import your file (Tools---->Universal Translator). Choose *ESRI* shape and the file that you designated in CrimeStat. Next, choose the appropriate projection. Identify the destination format–chose *MapInfo tab* and, finally, identify the directory for storage of the file. The table can then be opened as a layer on your map. *CrimeStat* graphic output is brought into *MapInfo* as regions and has all the functionality of a regions layer. Figure 7.6 includes STAC and single kernel density output.





files. Many other GIS programs, however, also can read 'dbf' files. For *ArcView* and *MapInfo*, the X and Y coordinates which define crime incident points are not directly part of the 'dbf' file, but instead exist on the geographic file.

Input File Formats

ArcView

In *ArcView* the coordinates are stored on the 'shp' file, not the 'dbf' file. *CrimeStat* can read directly a 'shp' file so the 'dbf' file is not required to have the X and Y coordinates.

MapInfo

However, in *MapInfo*, the coordinates are stored in 'tab' files. To use *CrimeStat* with *MapInfo*, therefore, requires that the X and Y coordinates be assigned to two fields in the 'tab' file and then saved as a 'dbf' file. See the endnotes for directions on doing this.[4] Even in *ArcView*, some users may wish to export the points as a 'dbf' file because of other information that are on the records. The endnotes also list these directions.[5] *MapInfo* also uses a 'dat' format, which is similar to 'dbf'. This can be ready by *CrimeStat*.

Atlas*GIS

In *Atlas*GIS*, on the other hand, a point file is already a 'dbf' file and will have fields for the X and Y coordinates.

Microsoft Access

'Mdb' Files from *Microsoft Access® 97* (or earlier) can also be read by *CrimeStat*. The user will have to ensure that the file has an X and Y coordinate.

ODBC

Similarly, *CrimeStat* can read any file that uses Open Database Connectivity (ODBC). ODBC is a programming interface that enables programs to access data in database management systems that use Structured Query Language (SQL) as a data access standard, such as Excel®, Paradox®, Microsoft Access, Lotus 1-2-3®, and FoxPro®.

ASCII

For an ASCII file, however, three additional attributes must be defined. The first is the type of character that is used to separate the variables in the file. There are four possibilities:[6]

Space (one or more, the default)
Comma
Semicolon

Tab

The second characteristic is the number of rows which have labels on them (*Header Rows*). Some ASCII files will have rows which label the names of the variables. The user should indicate the number if this is the case otherwise *CrimeStat* will produce an error code. The default is 0, that is the program assumes that there are no headers unless instructed otherwise. To change this, the user should insert the cursor in the appropriate cell, backspace to erase the default number and type in the correct number.

The third characteristic of an ASCII file that must be defined is the number of variables (columns or fields) in the file. With spherical or projected coordinates, there will be at least two variables (the X and Y coordinate) and there may be more if other variables are included in the file. However, with directional coordinates (see below), there may be only one. *CrimeStat* assumes that the number of columns in the ASCII file is two unless instructed otherwise. Again, the user should insert the cursor in the appropriate cell, backspace to erase the default number and type in the correct number. After defining the file type and name, the user should click on *OK*.

Identifying Variables

After defining a file, either '.dbf', ASCII, 'dat', or '.shp', it is necessary to identify the variables. Two variables are required and two are optional. The required variables are the X and Y coordinates. The user should indicate the file name that contains the coordinates by clicking on the drop down menu and highlighting the correct name. After having identified which file contains the X and Y coordinates, it is necessary to identify the variable name. Click on the drop down menu under *Column* and highlight the name of the variable for the X and Y coordinates respectively.[7] Figure 3.3 shows a correct defining of file and variable names for the primary file.

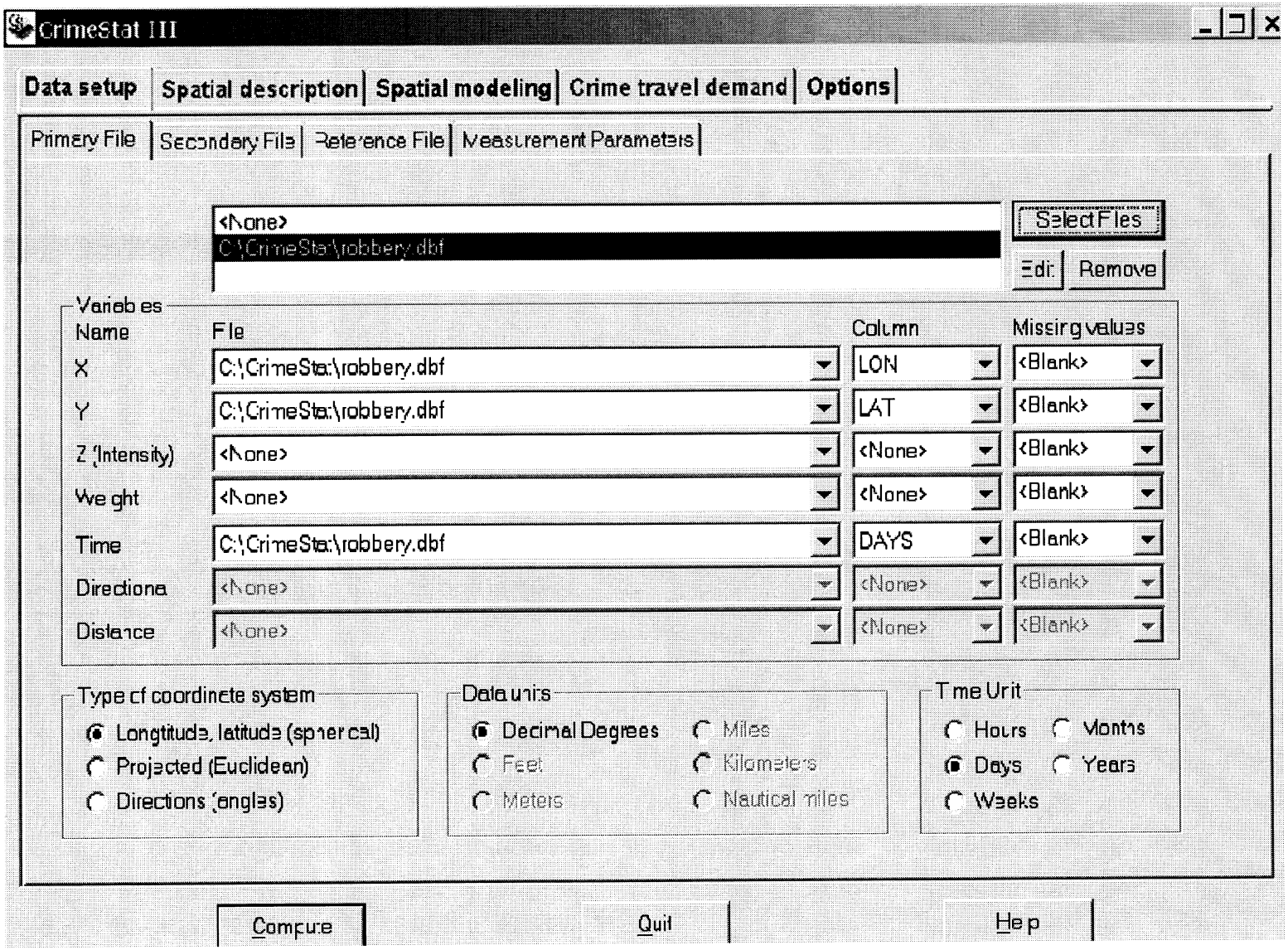
Multiple files can be entered on the primary file tab. However, only one can be utilized at a time. In theory, one can have separate files containing the X and Y coordinates, though in practice this will rarely occur.

Weight Variable

Sometimes, a point location is weighted. As mentioned above, weights are used when points represents areas and the areas are statistically treated differently. For most of the statistics, *CrimeStat* can weight the statistics during the calculation (e.g., the weighted mean center, the weighted nearest neighbor index).

By default, *CrimeStat* assigns a weight of 1 to each point. If the user does not define a weight variable, then the program assumes that each point has equal weight (i.e., 1). On the other hand, if there are weights, then the weight variable should be defined on the primary file screen and its name listed.

Figure 3.3: Primary File Definition



Intensity Variable

Similarly, a point location can have an intensity assigned to it. Most of the statistics in *CrimeStat* can use an intensity variable and some statistics require it (Moran's I, Geary's C and Local Moran). If no intensity is defined, *CrimeStat* will not calculate statistics requiring an intensity variable and, in statistics where an intensity is optional (e.g., interpolation), will assume a default intensity of 1. On the other hand, if there is an intensity variable, then this should be defined on the primary file screen and its variable name identified.

In general, be very careful about using *both* an intensity variable *and* a weighting variable. Use both only when there are separate weights and intensities. Most of the routines can use both intensities and weighting and may, consequently, double-weight cases. Figure 3.4 shows a primary file screen with an intensity variable defined.

Time Variable

Finally, a time variable can be defined for use in the special Space-time analysis tools under Spatial modeling. *CrimeStat* allows five different time references:

Hours
Days
Weeks
Months
Years

The default is 'days' but the user can choose one of the other four categories. However, the program assumes that all records are consistent defined. For example, all records must be in days or in hours. If some records are in days, for example, and other records are in hours, the program will not know that there is an inconsistency and will treat each of the records in the way they have been defined. It's important, therefore, that a user ensure that all records are consistent in the way that time is defined. Figure 3.5 illustrates the defining of a time variable on the primary file page.

Coordinate System

In addition to the primary file name and variable assignment, it is necessary to identify the type of coordinate system used and the units of measurement. *CrimeStat* recognizes three coordinate systems:

Spherical coordinates (longitude and latitude)

This is a universal coordinate system that measures location by angles from reference points on Earth.[8]

Figure 3.4: Primary File With Intensity Variable Defined

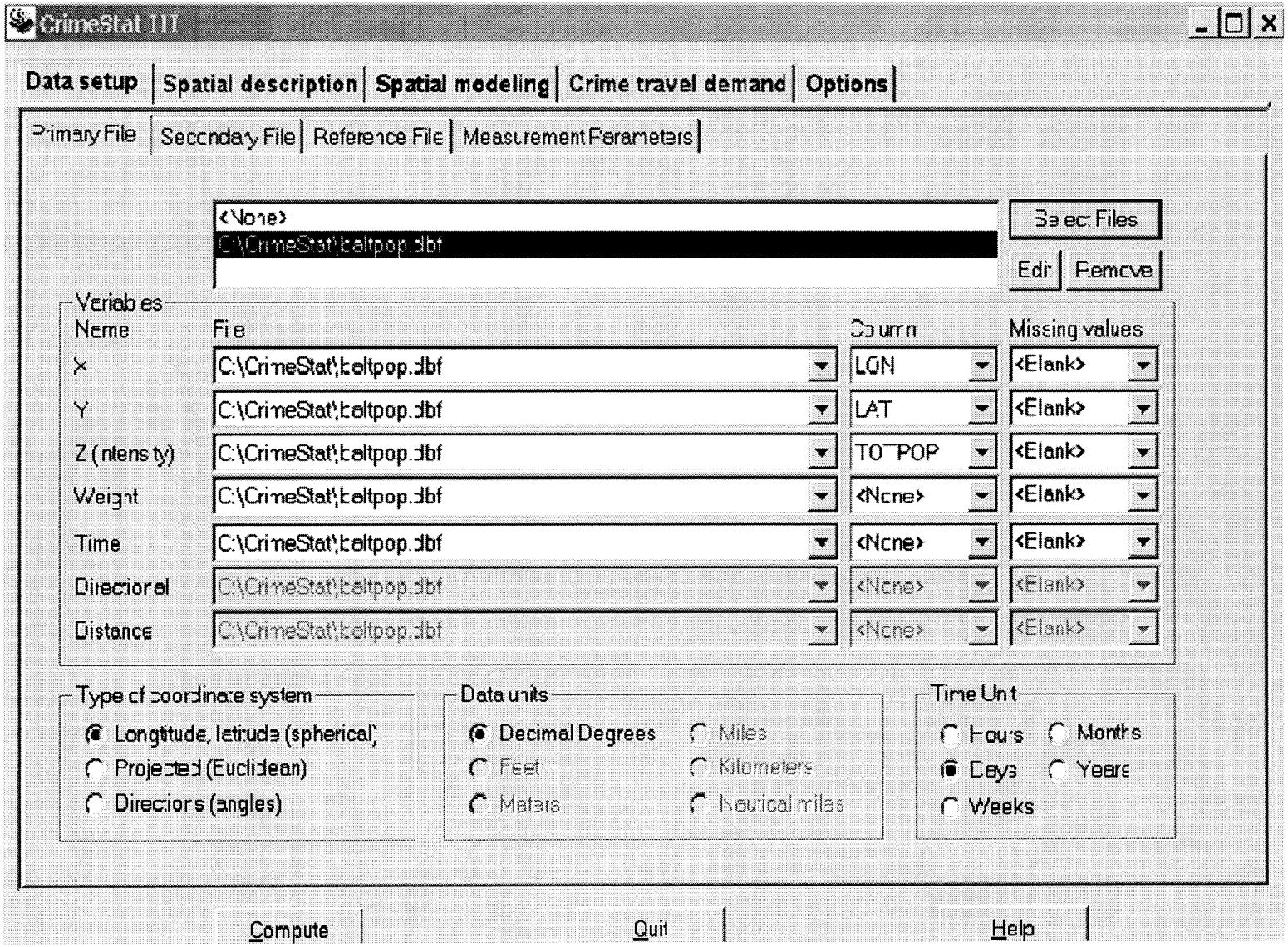
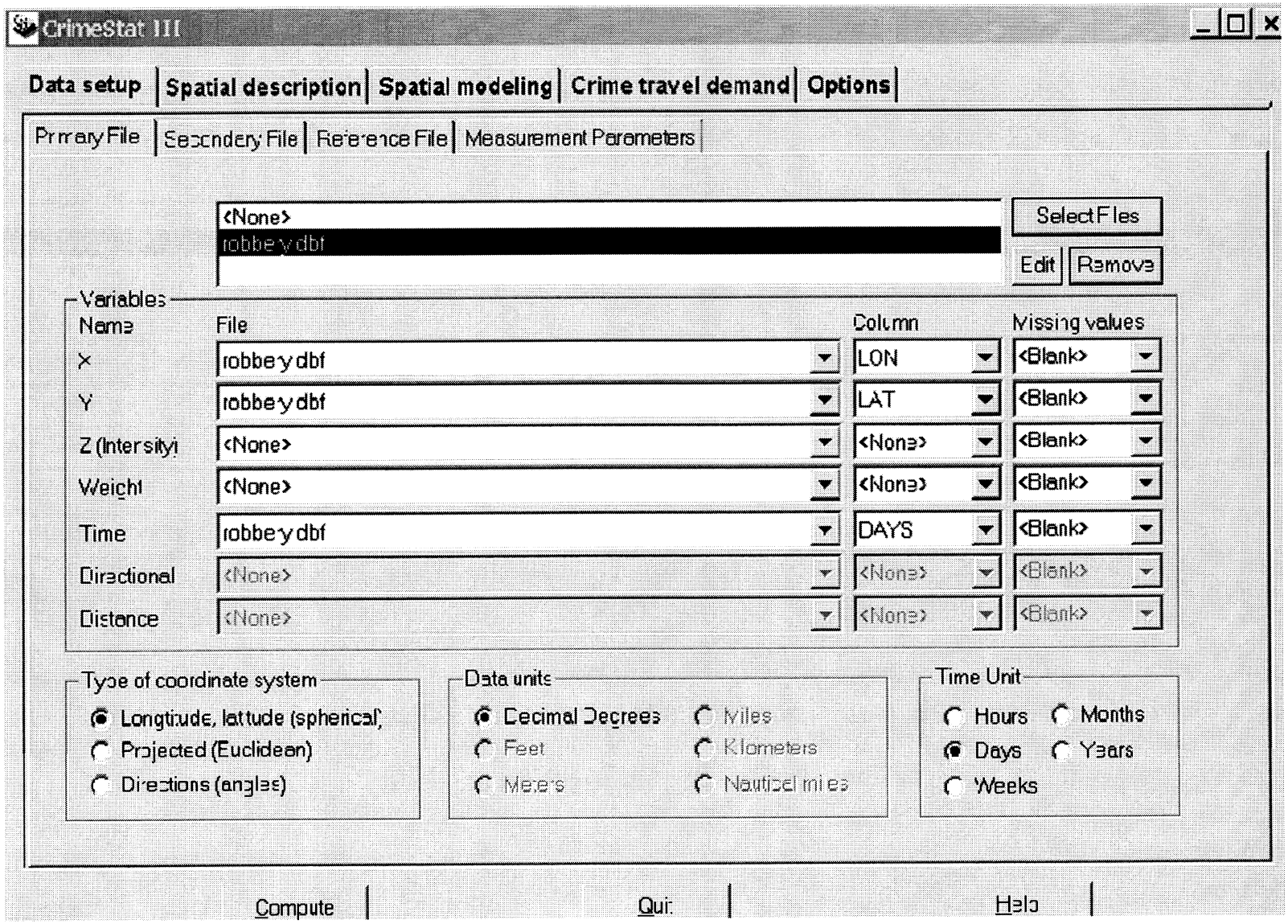


Figure 3.5: Time Variable Definition



Projected coordinates

Projected coordinates are arbitrary coordinates based on a particular projection of the earth to a flat plane. They have an arbitrary origin (the place where X=0 and Y=0) and are almost always defined in units of feet or meters.[9]

CrimeStat can work with either spherical or projected coordinates. On the primary file tab, the user indicates which coordinate system is being used. If the coordinate system is spherical, then units are automatically assumed to be latitude and longitude in decimal degrees. If the coordinate system is projected, then it is necessary to specify whether the measurement units are feet or meters.

Directional coordinates

For some uses, a polar coordinate system can be used. Point locations are defined by angles from an arbitrary reference line, usually true north and vary between 0^0 and 360^0 in a clockwise rotation. All locations are measured as an angular deviation from the reference point and with distance being measured from a central location. *CrimeStat* has the ability to read in angles for use in calculating the angular mean and variance. In addition, if directional coordinates are used, an optional distance variable for each measurement can be used.

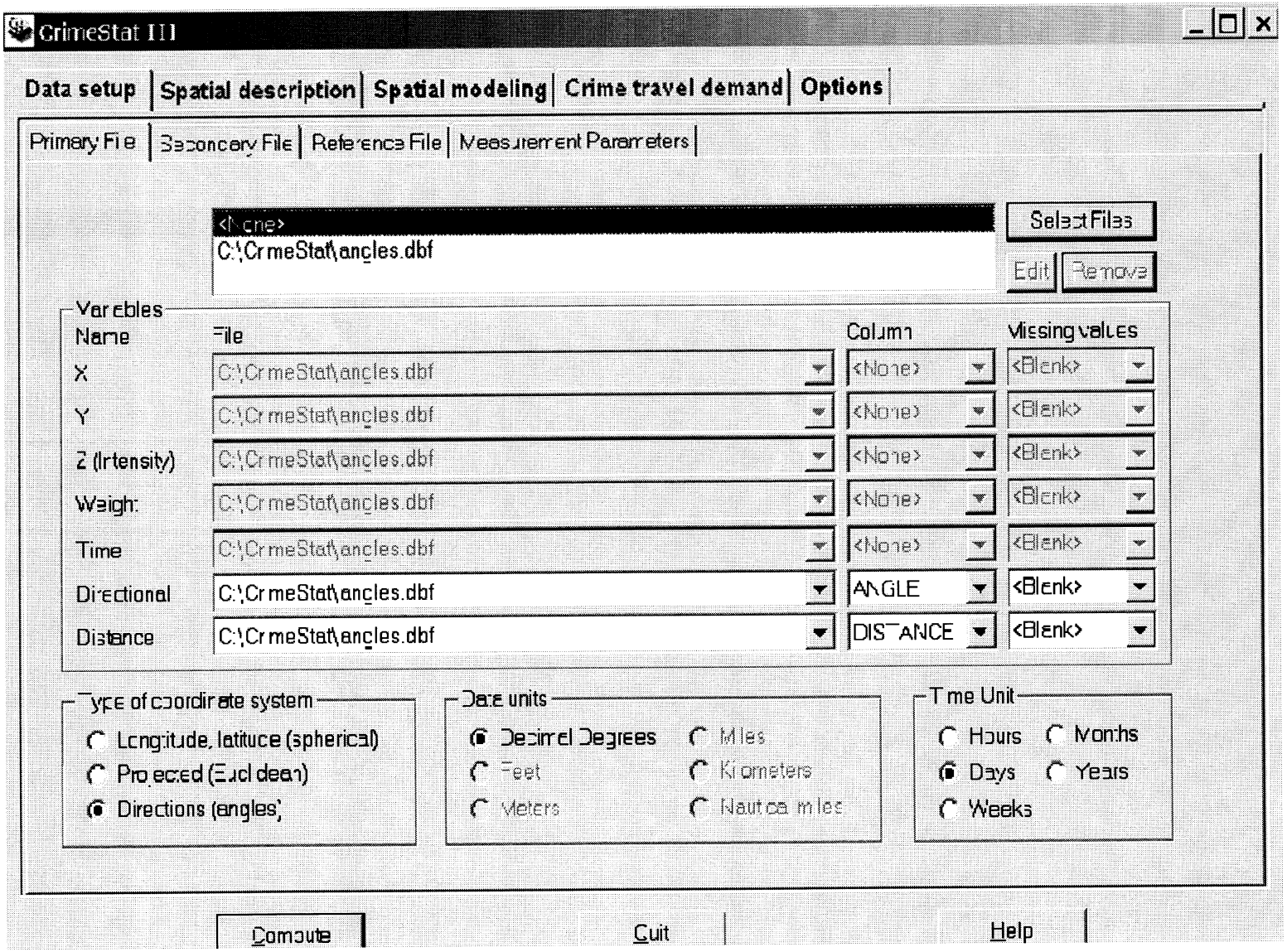
If the file contains directional coordinates (angles), define the file name and variable name (column) that contains the directional measurements. If used, define the file name and variable name (column) that contains the distance variable. Figure 3.6 shows the primary file definition using directions.

Secondary File

CrimeStat also allows for the inputting of a secondary file. For example, the primary file could be locations where motor vehicles were stolen while the secondary file could be the location where stolen vehicles were recovered. Alternatively, the primary file could be burglary locations while the secondary file could be police stations. *CrimeStat* can construct two different types of indices with a secondary file. First, it can calculate the distance from every primary file point to every secondary file point. For example, this might be useful in assessing where to place police cars in order to minimize travel distance in response to calls for service. Second, *CrimeStat* can utilize both primary and secondary files in estimating a three-dimensional density surface (see Chapter 7). For example, if the primary file are residential burglaries and the secondary file contains the centroids of census block groups with the population within each block group assigned as an intensity variable, then *CrimeStat* can estimate the density of burglaries relative to the density of population (i.e., burglary risk).

The secondary file can also be either a '.dbf', '.shp' or ASCII. As with a primary file, there must be an X and Y variable defined, but it must be in the same coordinate system and data units as the primary file. The secondary file can also have weights and intensities

Figure 3.6: File Definition With Angles (Directions)



assigned, but not a time variable.. Figure 3.7 shows the inputting of an ASCII file for the secondary data set while figure 3.8 shows a correct definition of the secondary file.

Reference File

Several of the routines in *CrimeStat* generalize the point data to all locations in the study area, in particular the one-variable and two-variable density interpolation routines (chapter 8), and the risk-adjusted nearest neighbor hierarchical clustering routine (chapter 6). The generalization uses a reference file placed over the study area. The STAC program also uses a reference file for searching (chapter 7). Typically, the reference file is a rectangular grid file (true grid), that is a rectangle with cells defined by columns and rows.; each grid cell is a rectangle and column-row combinations are used. It is possible to use a non-rectangular grid file under special circumstances (e.g., a grid with water, mountains or other jurisdictions removed), but a rectangular grid would be used in most cases. *CrimeStat* can create a grid file directly or can read in an external grid file. Figure 3.9 shows a grid placed over both the County of Baltimore and the City of Baltimore.

Creating a Reference Grid

CrimeStat can also create a true grid. There are two steps:

1. The user selects *Create Grid* from the Reference File tab and inputs the X and Y coordinates of the lower-left and upper-right coordinates of the grid. These coordinates must be the same as for the primary file.

Thus, if the primary file is using spherical (lat/lon) coordinates, then the grid file coordinates must also be lat/lon. Conversely, if the primary file coordinates are projected, then the grid file coordinates must also be projected, using the same measurement units (feet or meters). The lower-left and upper-right coordinates are those from a grid which covers the geographical area. A user should identify these with a GIS program or from a properly indexed map. In *MapInfo*, this is easily done by either drawing a rectangle around the study area and double clicking to get information about the area or by checking the cursor position. In *ArcView*, you can draw a shape file of the appropriate reference rectangle and then use the Coordinate Utility script to get the X and Y coordinates.

2. The user selects whether the grid is to be created by cell spacing or by the number of columns.

With *By cell spacing*, the size of the cell is defined by its horizontal width, in the same units as the measurement units of the primary file. This would be used to maintain a certain size of spacing for a cell. For example, if the coordinate system is spherical and the lower-left coordinates are -76.90 and 39.20 degrees and the upper-right coordinates are -76.32 and 39.73 degrees (a grid which overlaps Baltimore City and Baltimore County), then the horizontal distance - the difference in the two longitudes (0.58 degrees) must be divided into appropriate sized intervals. At this latitude, the difference in longitudes is 34.02 miles. If a user wanted cell spacing of 0.01 degrees, then this would be entered and

Figure 3.7: Ascii File Selection of Secondary File

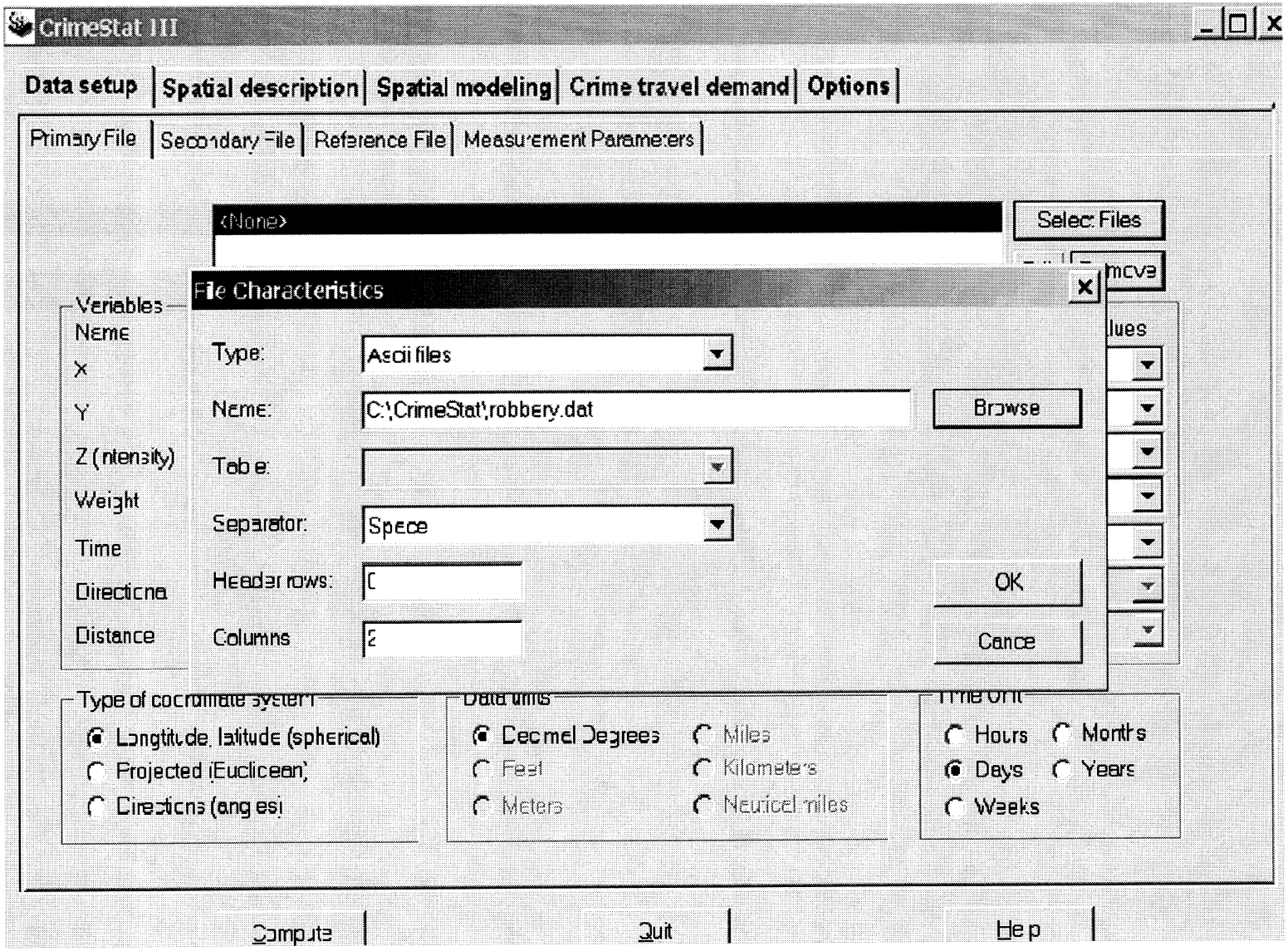


Figure 3.8: Secondary File Definition

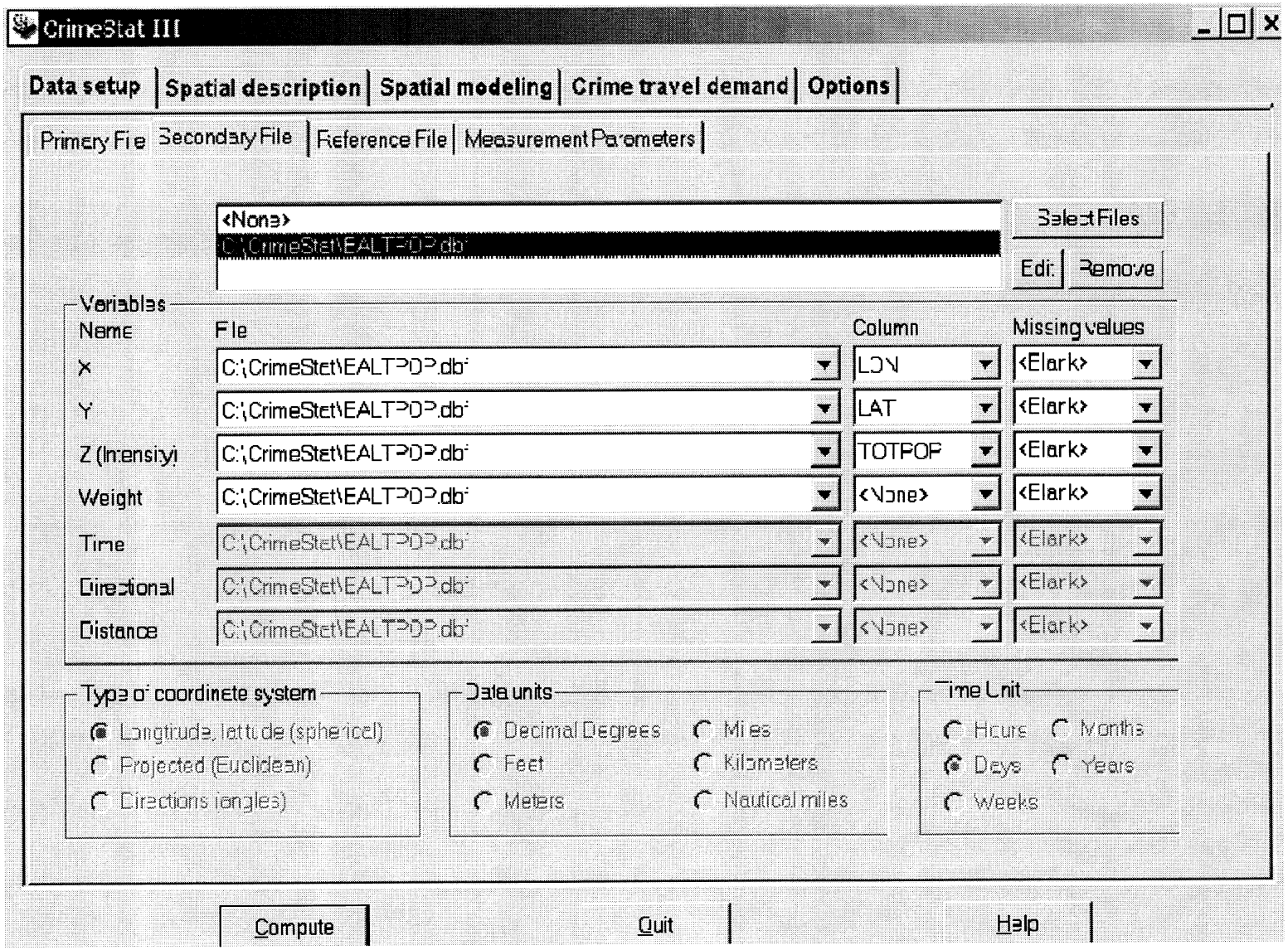
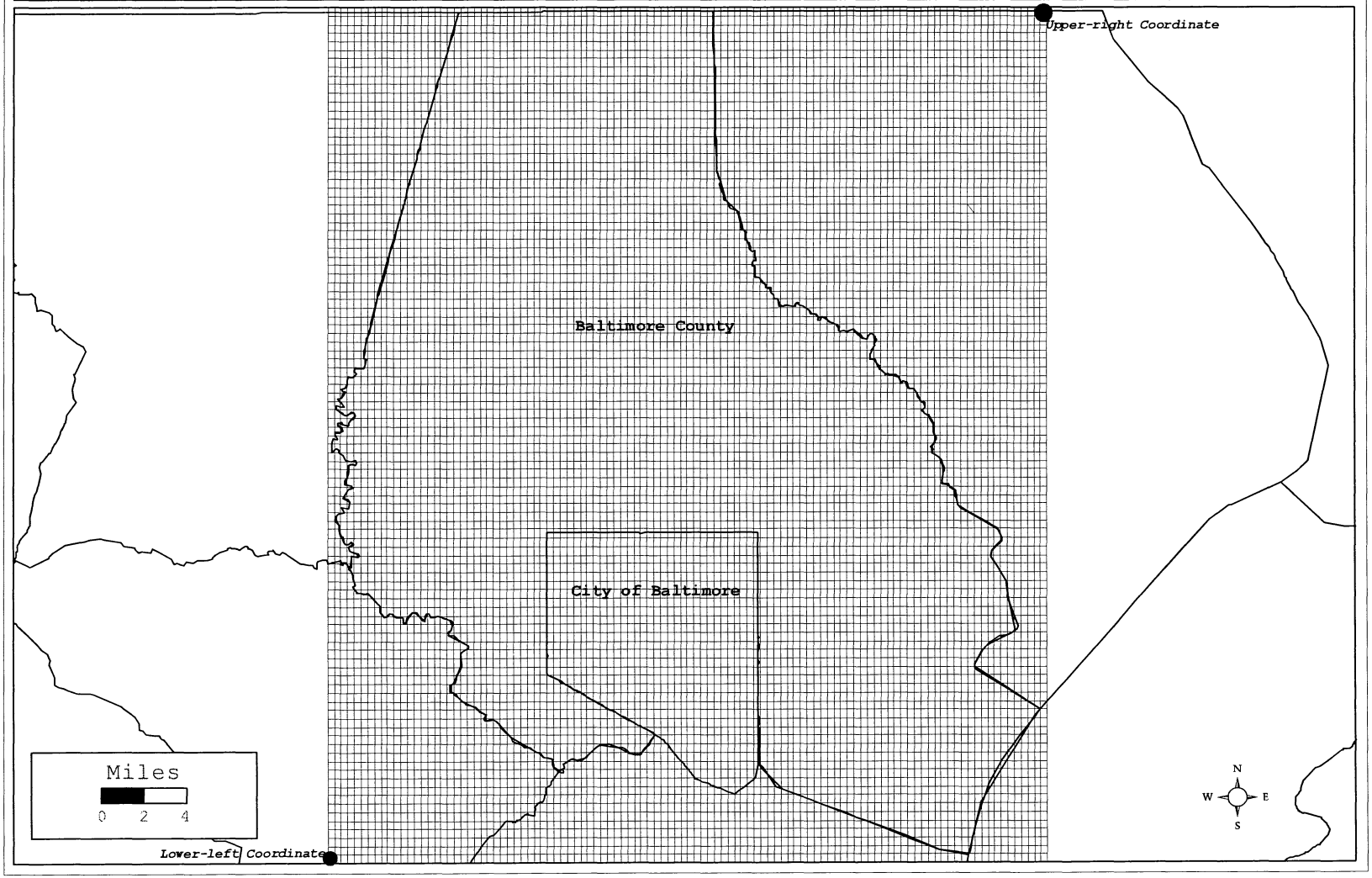


Figure 3.9: Grid Cell Structure for Baltimore Region

108 Width x 100 Height Grid Cells



CrimeStat would calculate 59 columns (cells) in the horizontal direction, one for each interval of 0.01 and one for the fractional remainder. If the coordinate system is projected, then similar calculations would be made using the projected units (feet or meters).

Probably an easier way to specify the grid is to indicate the number of columns. By checking *By number of columns*, the user defines the number of columns to be calculated. *CrimeStat* will automatically calculate the cell spacing needed and will calculate the required number of rows. For example, using the same coordinates as above, if a user wanted half mile squares for the cells, then they would need approximately 68 cells in the horizontal direction since 34.02 miles divided by 0.5 mile squares equals about 68 cells. Figure 3.10 shows a correctly defined reference file where *CrimeStat* creates the reference grid with the number of columns being defined; in the example, 100 columns are requested.

Saving a Reference File

The user can save the lower-left and upper-right coordinates of a defined reference grid and the number of columns. Type **S**ave <filename>. The coordinates and column sizes will be saved in the system registry. To load an already defined reference file, type **L**oad and then check the appropriate filename, followed by clicking on 'Load'.

In addition, the user can save the reference parameters to an external file. To do this, it has to be already saved in the system registry. Type **L**oad and then check the appropriate filename, followed by clicking on 'Save to File". Define the directory and file name and click 'Save'. The file will be saved with an 'ref' extension (e.g., BaltimoreCounty.ref).

Use an External Grid File

Many GIS programs can create uniform grids which cover a geographical area. As with the primary and secondary files, these need to be converted to either '.dbf', ASCII or '.shp' files. To use an existing grid file created in a GIS or another program, the user clicks on *From File* on the Reference File tab and selects the file.

There are three characteristics which should be identified for an existing grid file:

1. The name of the file. The user selects the file from a dialog box similar to the primary file.

2. If the existing reference file is a true grid, the *True Grid* box should be checked.

3. If it is a true grid, the number of columns should be entered. *CrimeStat* will automatically count the number of records in the file and place it in the *Cells* box. When the number of columns is entered, *CrimeStat* will automatically calculate the number of rows.

Figure 3.10: Create Reference Grid Setup

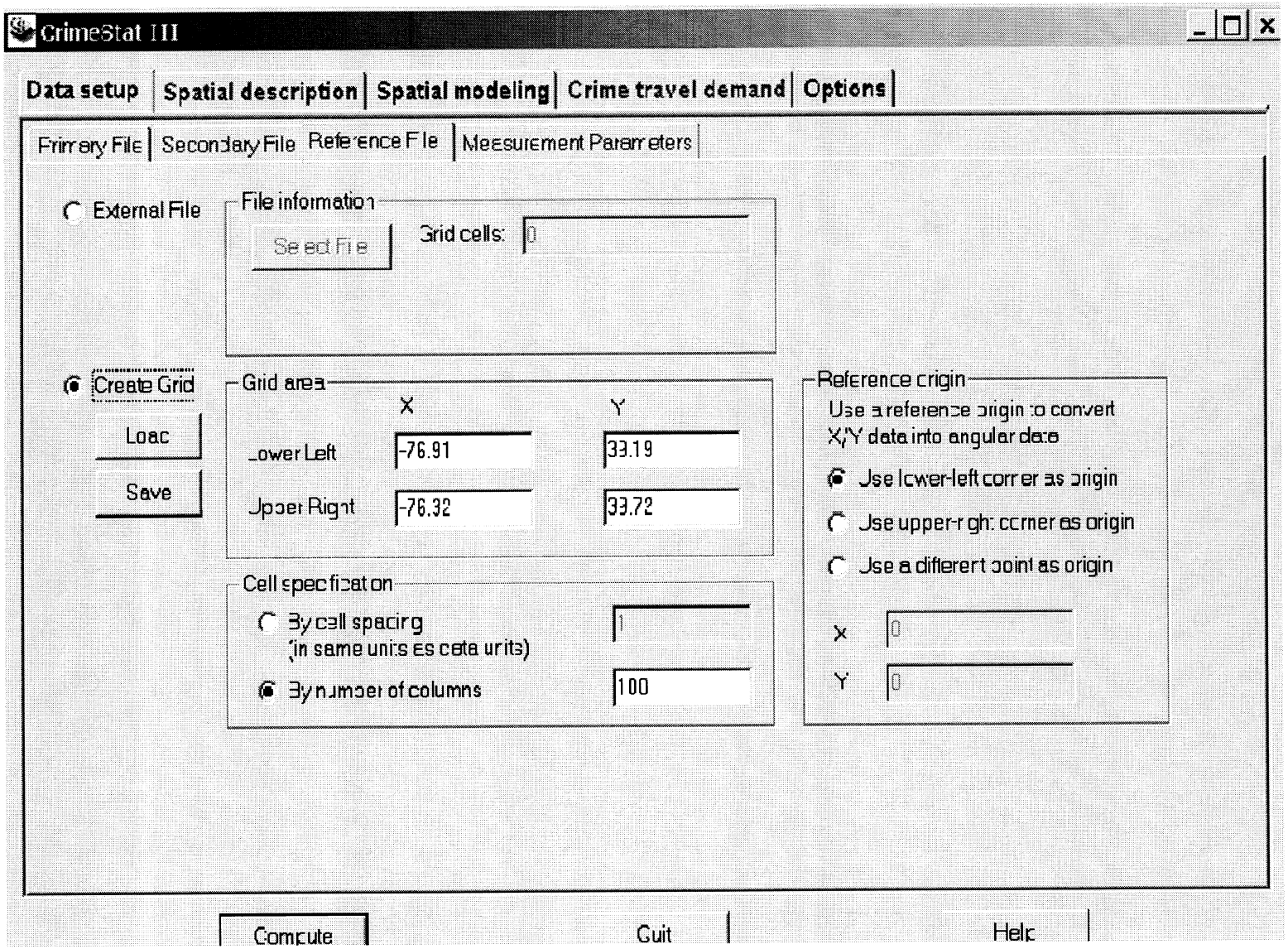


Figure 3.11 shows a correctly defined reference file using an existing grid file. One must be careful in using a file which is not a grid. *CrimeStat* can output the results of the interpolation routines in several GIS formats - *Surfer for Windows*, *ArcView Spatial Analyst*, *ArcView*, *MapInfo* and *Atlas*GIS*. Of these, only the output to *Surfer for Windows* will allow the reference to be a shape other than a true grid. For the interpolation outputs of *ArcView Spatial Analyst*, *ArcView*, *MapInfo* and *Atlas*GIS*, it is essential that the reference file be a true grid.

Use of Reference File

A reference grid can be very useful. First, a number of the routines use it for either interpolation (single and duel kernel routines; nearest neighbor hierarchical clustering routine) or keying a search radius (STAC). Second, a grid produced by *CrimeStat* can be used as a separate layer in a GIS program in order to reference other data that is displayed, aside from statistical calculations. Historically, many map uses are referenced to a grid in order to produce a systematic inventory (e.g., parcel maps; tax assessor maps; U.S. Geological Survey 7.5" 'quad' maps). In short, it is a routine with multiple purposes.

Measurement Parameters

The final properties that complete data definition are the measurement parameters. On the Measurement Parameters tab, the user defines the geographical area and the length of street network for the study area, and indicates whether direct or indirect distances are to be used. Figure 3.12 shows the measurement parameters tab page.

Area and Length of Street Network

In calculating distances between points for two of the statistics - the nearest neighbor index and the Ripley 'K' index, the area for which the points fall within needs to be defined (the study area). The user indicates the area of the geographical coverage and the measurement units that distances are calculated (feet, meters, miles, nautical miles, kilometers). Unlike the data units for the coordinate system, which must be consistent, CrimeStat can calculate distances in any of these units. In some cases, analysis will be conducted on a subset of the study area, rather than the entire area. For each analysis, the user should identify the area of the subset for which distance statistics are to be calculated.

In addition, the linear nearest neighbor statistic uses the total length of the street network as a baseline for comparison (see chapter 5). If this statistic is to be used, the total length of the street network should be defined. Most GIS programs can sum the total length of the street network. Again, if subsets of the study are used, the user should indicate the appropriate length of street network for the subset so that the comparison is appropriate.

Figure 3.11: Reference File Definition With An External File

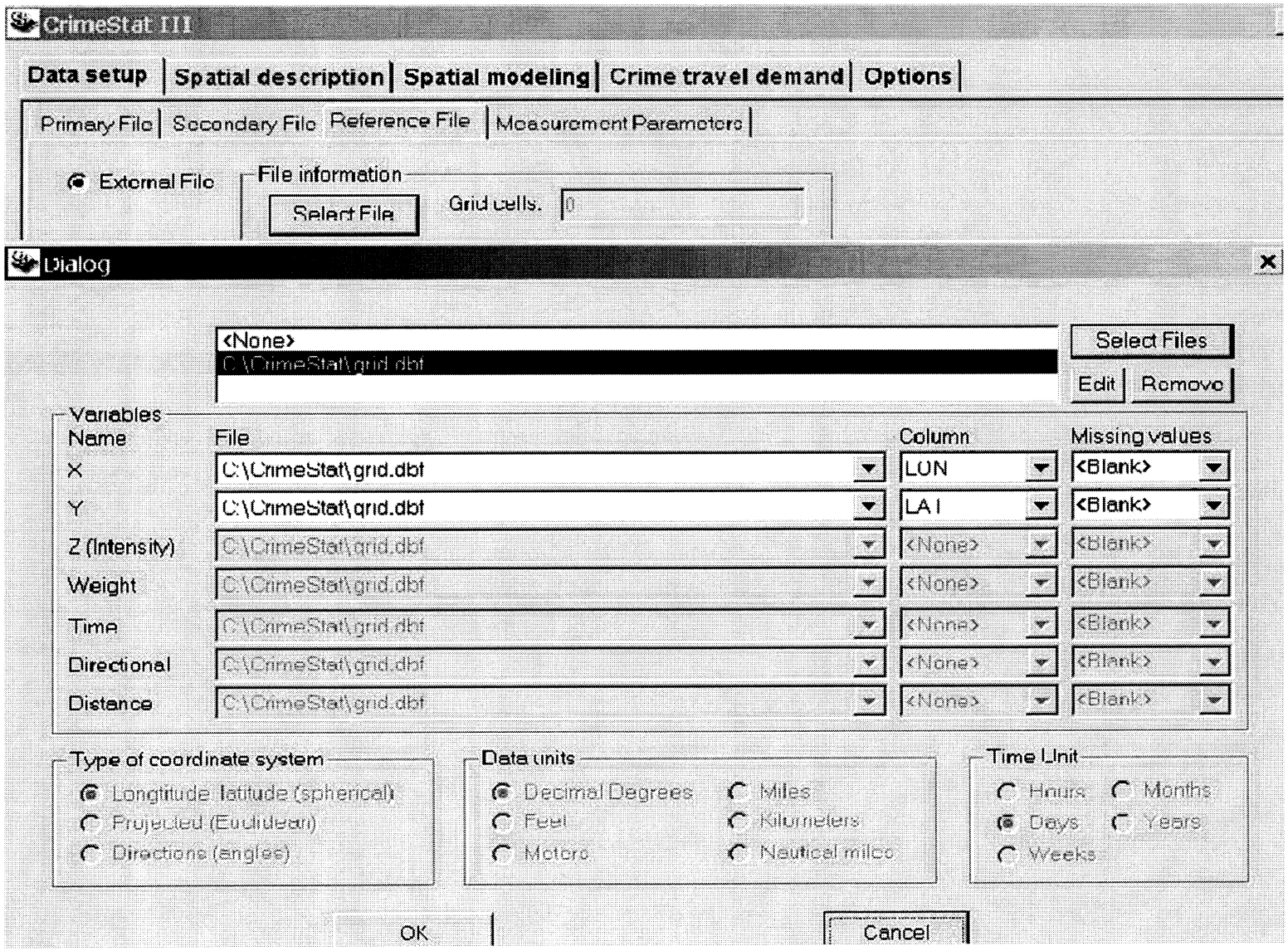
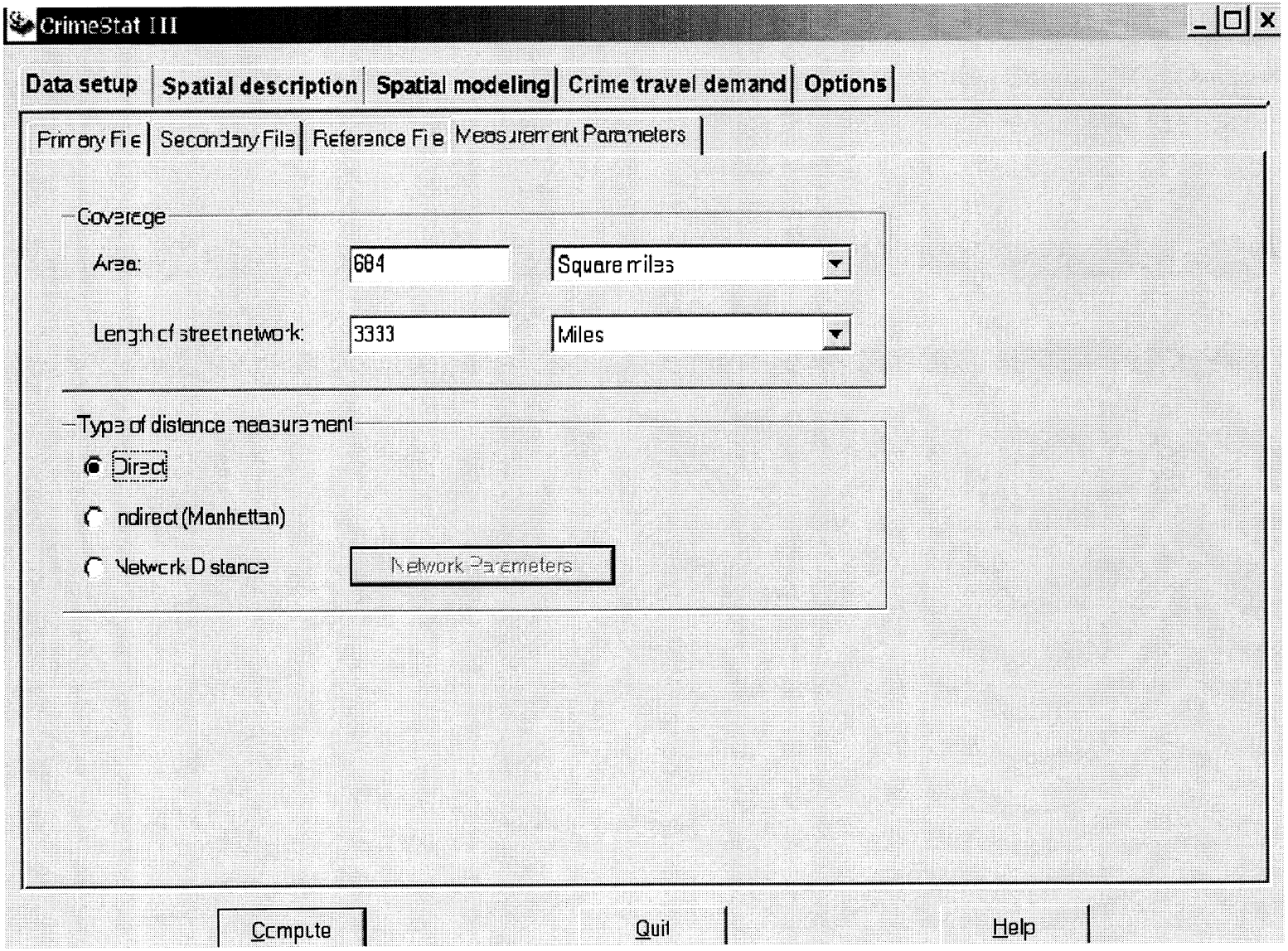


Figure 3.12: Measurement Parameters Page



Type of Distance Measurement

Direct distance

CrimeStat can calculate distance in three different ways: direct, indirect, and network distances. Direct distances are the shortest distance between two points. On a flat plane, that is with a projected coordinate system, the shortest distance between two points is a straight line. However, on a spherical coordinate system, the shortest distance between two points is a Great Circle line. Depending on the coordinate system, CrimeStat will calculate Great Circle distances using spherical geometry for spherical coordinates and Euclidean distances for projected coordinates. The drawings in figure 3.13 illustrate direct distances with a projected and spherical coordinate system. The shortest distance between point A and point B is either a straight line (projected) or a Great Circle (spherical). For details see McDonnell, 1979 (chapter 1) or Snyder, 1987 (pp. 29-33).

Indirect distance

Indirect distances are an approximation of travel on a rectangular road network. This is frequently called Manhattan distance, referring to the grid-like structure of Manhattan. Many cities, but certainly not all, lay out their streets in grids. The degree in which this is true varies. Older cities will not usually have grid structures whereas newer cities tend to use grid layouts more. Of course, no real city is a perfect grid, though some come close (e.g., Salt Lake City). Distances measured over a street network are always longer than a direct line or arc. In a perfect grid, travel can only occur in horizontal or vertical directions so that distances are the sum of the horizontal and vertical street lengths that have been traveled (i.e., one cannot cut diagonally across a block). Distances are measured as the sum of horizontal and vertical distances traveled between two points.

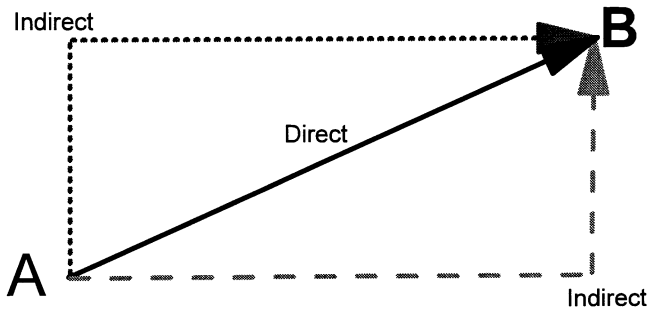
Indirect distance approximate actual travel pattern for a city where streets are arranged in grid pattern. This is why this type of distance is frequently called *Manhattan Distance.* In this case, indirect distances would be a more appropriate distance measurement than direct distances. Also, there is a linear nearest neighbor index which measures the distribution of point locations in relation to the street network rather than the geographical area and uses indirect distances. This will be discussed in Chapter 5. In this case, the use of indirect distances would be preferable than direct distances.[10]

Network distance

Network distances are travel on an actual network. The network can be a road network, a transit network, or a rail network. Travel is constrained to the network which usually will make it longer than direct distance measurement. However, the advantage is that travel is measured along the available routes, rather than as an abstract 'straight line' or even abstract 'grid'. Another advantage of network distance is that the network can be weighted by travel time, travel speed or travel cost. Thus, it's possible to measure approximate travel time or travel cost through the network, and not just distance. It is generally recognized that travel time is a more realistic dimension than distance since it

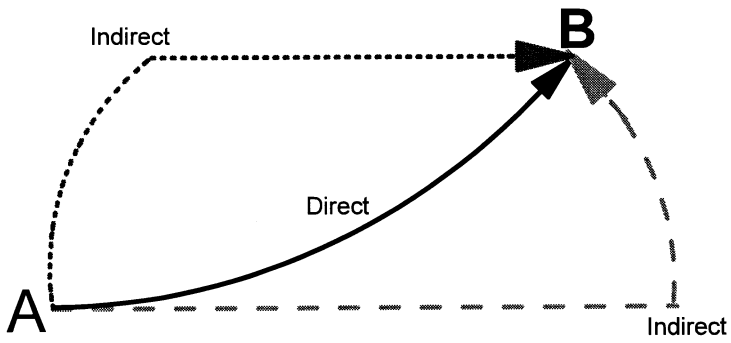
Figure 3.13: Direct and Indirect Distances

Two-dimensional Projected Geometry: Euclidean distance



A-B distance ('dotted route') =
A-B distance('dashed route')

Three-dimensional Spherical Geometry: Great Circle distance



A-B distance ('dotted route') <
A-B distance ('dashed route')

will vary by time of day. For example, it generally takes a lot longer to travel any distance in an urban area during the peak evening 'rush hours' (4-7 PM) than at, say, 3 AM in the morning. Distance is always invariant whereas travel time varies. An even more realistic dimension travel is cost. Trips over a metropolitan area are governed by a number of variables aside from travel time - vehicle operating costs, parking costs and, even, likely risk costs (e.g., likelihood of being caught). For an offender who is traveling, those other cost factors may be as important as the actual time it takes in determining whether to make a crime trip. In chapter 15, there is a discussion of travel costs in the context of travel decisions.

There are two major disadvantages in using network distance, however. First, there are errors in networks. For example, a network may not have incorporated all new roads or converted roads. Thus, the network algorithm will not choose a particular route when, in fact, it actually exists and people use it. It's critical that networks be updated to ensure accuracy. See chapter 12 for a discussion of network errors and the need to thoroughly clean them.

Second, it can take a long time to calculate distance along a network. The shortest path algorithm that is used must explore many alternative routines, a time consuming process. For simple statistics, this is not liable to be a problem. But, for some of the more complicated matrix operations (e.g., the distance from every point to every other point), calculation time increases exponentially with the number of cases. I've had runs that took five days on a fast computer. For any complex calculation, it becomes impractical to have to wait a long time just for a little extra precision. In short, it may not be worth the trouble. At some point in the near future, we will have 64 bit operating systems and super-fast computers. At that point, running all calculations on a network may be a much more practical proposition. For now, I highly recommend that network distance be used sparingly for calculations.

Distance Calculations

Distances in CrimeStat are calculated with the following formulas:

Direct, Projected Coordinate System

Distance is measured as the hypotenuse of a right triangle in Euclidean geometry.

$$d_{AB} = \sqrt{(X_A + X_B)^2 + (Y_A + Y_B)^2} \quad (3.1)$$

where d_{AB} is the distance between two points, A and B, X_A and X_B are the X-coordinates for points A and B in a projected coordinate system, Y_A and Y_B are the Y-coordinates for points A and B in a projected coordinate system.

Direct, Spherical Coordinate System

Distance is measured as the Great Circle distance between two points. All latitudes (ϕ) and longitudes (λ) are first converted into radians using:

$$\text{Radians } (\phi) = \frac{2\pi}{360} \tag{3.2}$$

$$\text{Radians } (\lambda) = \frac{2\pi}{360} \tag{3.3}$$

Then, the distance between the two points is determined from

$$d_{AB} = 2^* \text{ Arcsin } \{\text{Sin}^2[(\phi_B - \phi_A)/2] + \text{Cos } \phi_A * \text{Cos}\phi_B * \text{Sin}^2[(\lambda_B - \lambda_A)/2]^{1/2} \} \tag{3.4}$$

with all angles being defined in radians where d_{AB} is the distance between two points, A and B, ϕ_A and ϕ_B are the latitudes of points A and B, and λ_A and λ_B are the longitudes of points A and B (Snyder, 1987, p. 30, 5-3a).

Indirect, Projected Coordinate System

Distance is measured as the sides of a right triangle using Euclidean geometry.

$$d_{AB} = (X_A - X_B) + (Y_A - Y_B) \tag{3.5}$$

where d_{AB} is the distance between two points, A and B, X_A and X_B are the X-coordinates for points A and B in a projected coordinate system, Y_A and Y_B are the Y-coordinates for points A and B in a projected coordinate system.

Indirect, Spherical Coordinate System

Distance is measured by the average of summed Great Circle distances of two routes, one in the east-west direction followed by a north-south direction and the other in the north-south direction followed by an east-west direction.

$$d_{AB} = \frac{[d_{AB}(1) + d_{AB}(2)]}{2} \tag{3.6}$$

where d_{AB} is the distance between two points, A and B, $d_{AB}(1)$ is the sum of distances between points A and B by measuring the Great Circle distance of the east or west direction from a particular latitude first, and adding this to the Great Circle distance of the north or south direction from that same latitude, and $d_{AB}(2)$ is the sum of distances between points A and B by measuring the Great Circle distance of the north or south direction from a particular longitude first, and adding this to the Great Circle distance of the east or west direction from that same longitude.

Network Distance

Network distance is calculated with a shortest path algorithm. Chapters 12 and 16 provide more information on networks and how distance is calculated on them. A short summary will be given here. In general, distance is calculated by a shortest path algorithm. In a *shortest path* for a single trip (from a single origin to a single destination), the route with the lowest overall *impedance* is selected. Impedance can be defined in terms of distance, travel time, speed, or generalized cost.

There are a number of shortest path algorithms that have been developed (Sedgewick, 2002). They differ in terms of whether they are breadth-first (i.e., search all possibilities) or depth-first (i.e., go straight to the target) algorithms and whether they examine a one-to-many relationship (i.e., from a single origin node to many nodes) or a many-to-many relationship (All pairs; from each node to every other node).

The algorithm that is most commonly used for shortest path analysis of moderate-sized data sets (up to a million cases) is called *A**, which is pronounced "A-star" (Nilsson, 1980; Stout, 2000; Rabin 2000a, 2000b; Sedgewick, 2002). It is a one-to-many algorithm but is an improvement over another commonly-used algorithm called *Dijkstra* (Dijkstra, 1959). Therefore, I'll start first by describing the Dijkstra algorithm before explaining the A* algorithm.

Dijkstra algorithm

The Dijkstra algorithm is a one-to-many search strategy in which a shortest path from a single node to all other nodes is calculated. The routine is a breadth-first algorithm in that it searches all possible paths, but it builds the path one segment at a time. Starting from an origin location (node), it identifies the node that is nearest to it **and** which has not already been identified on the shortest path. After each node has been identified to be on the shortest path, it is removed from the search possibilities. The algorithm proceeds until the shortest path to all nodes has been determined.

The algorithm can also be structured to find the shortest path between a particular origin node and a particular destination node. In this case, it will quit once the destination node has been identified on the shortest path. The algorithm can also be structured to find the shortest path from each origin node to each destination node. It does this one path at a time (e.g., it finds the shortest path from node A to all other nodes; then it finds the shortest path from node B to all other nodes; and so forth).

A Algorithm*

The biggest problem with the Dijkstra algorithm is that it searches the path to every single node. If the purpose were to find the shortest path from a single node to all other nodes, then this would produce the best solution. However, with a matrix of distance from one set of points to another set of points (an origin-destination matrix), we really want to know the distance between a pair of nodes (one origin and one destination). Consequently, the Dikjstra algorithm is very, very slow compared to what we need. It would be a lot quicker if we could find the distance from each origin-destination pair one at a time, but quit the algorithm as soon as that distance has been determined.

This is where the A* algorithm comes in. A* was developed within the artificial intelligence research area as a means for developing a *heuristic* rule for solving a problem (Nilsson, 1980). In this case, the heuristic rule is the remaining distance from a solved node to the final destination. That is, at every step in the Dijkstra routine, an estimate is made of the remaining distance from each possible choice to the final destination. The node that is chosen for the shortest path is that which has the least total *combined* distance from the previously determined node to the final goal. Thus, for any step, if D_{i1} is the distance to a node, i, which has not already been put on the shortest path and D_{i2} is an estimate of the distance from that node to the final destination, the estimated total distance for that node is:

$$D_i = D_{i1} + D_{i2} \quad (3.7)$$

Of all the nodes that could be chosen, the node, i, which has the shortest total distance is selected next for the shortest path. There are two caveats to this statement. First, the node, i, cannot have already been selected for the shortest path; this is just re-stating the rules by which we search for nodes which have not yet been put on the shortest path list. Second, the estimate of the remaining distance to the final destination must be less than or equal to the actual distance to the final destination. In other words, the estimated distance, D_{i2}, cannot be an overestimate (Nilsson, 1980). However, the closer the estimated distance is to the real distance, the more efficient will be the search.

How then do we determine a reasonable estimate for D_{i2}? The answer is a straight line from the possible node to the final destination since the shortest distance between two points is a straight line (or, on a sphere, a Great Circle distance since the shortest distance between two points is an arc). If we simply calculate the straight-line from the node that we are exploring to the final node, then the heuristic will work. The effect of this simplifying heuristic is to cut down substantially on the number of nodes that have to be searched. As with the Dijkstra algorithm, A* can be applied to multiple origins. It does it one origin-destination combination at a time.

In general, if V is the number of nodes in the network, the Dijkstra algorithm requires V^2 searches whereas the A* algorithm requires only V searches (Sedgewick, 2002). As can be seen, this is much more efficient than having to search every single possible node, which is what Dijkstra requires.

As mentioned, chapters 12 and 16 discuss in more detail networks and how shortest path is calculated in them.

Saving Parameters

All data setup parameters can be saved. In the Options section, there is a 'Save parameters' button. The parameter file must be saved with a 'param' extension. To re-load a saved parameters file, use the 'Load parameters' button.

Automating Parameter Setup

CrimeStat has the ability to be automatically configured through Microsoft's Dynamic Data Exchange (DDE) code. DDE is an operating system language that allow one

application to call up another. The DDE code in CrimeStat allows the defining of the primary variable, the secondary variable, the reference file, and the measurement parameters. Appendix A gives the specific code instructions. Ron Wilson's example below illustrates how CrimeStat can be linked to another application.

Statistical Routines and Output

Statistical routines are selected from the two groupings of statistics - Spatial Description and Spatial Modeling. The user selects the routines and inputs any parameters, if required. Clicking on the Compute button all the routines that have been selected. Since CrimeStat is multi-threaded, different routines run in separate threads and may finish at different times. When a routine is finished, a Finished message will be displayed at the bottom of the screen.

Virtually all the routines output to either GIS packages or to standard 'dbf' files which can be read by spreadsheet, data base, and graphics programs. While each output table can be printed as an Ascii file to a printer, it is recommended that the user output the results in 'dbf' and read it into a program that has better output capabilities. For example, the nearest neighbor and Ripley's K routines output columns can be saved as standard 'dbf' files which can be read by spreadsheet programs, such as Excel or Lotus 1-2-3. The spreadsheet data, in turn, can be imported into most graphics programs, such as PowerPoint or Freelance, for creating better quality graphics. For 'cut-and-paste' operations, user can copy portions of the output tables and paste them into word processing programs. One should see CrimeStat as a collection of specialized statistical routines that can produce output for other programs, rather than as a full-blown package.

A Tutorial with the Sample Data Set

Let's run through the data setup and running of several routines with one of the sample data sets that were provided (SampleData.zip). Unzipping this file reveals two files called *Incident.dbf* and *BaltPop.dbf*. The incident file is a collection of incident locations that have been randomly simulated with the other file includes the 1990 population of census block groups in the Baltimore region.

1. Start the *CrimeStat* program by either double-clicking on the *CrimeStat* icon on the desktop (if installed) or else opening Windows Explorer and locating the directory where *CrimeStat* is stored and double-clicking on the file called *crimestat.exe*.

2. Once the program splash page closes, the user will be looking at the **Data Setup** page with the Primary File page open.

3. Click on 'Select Files' followed by 'Browse'. Locate the file called Incident.dbf and click on 'Open' followed by 'OK'.

4. The file name will now be listed for the X, Y, Z(intensity), Weight, and Time fields. This variable, however, only has three fields - ID, Lon, Lat, indicating an record number, the longitude and latitude of the incident location.

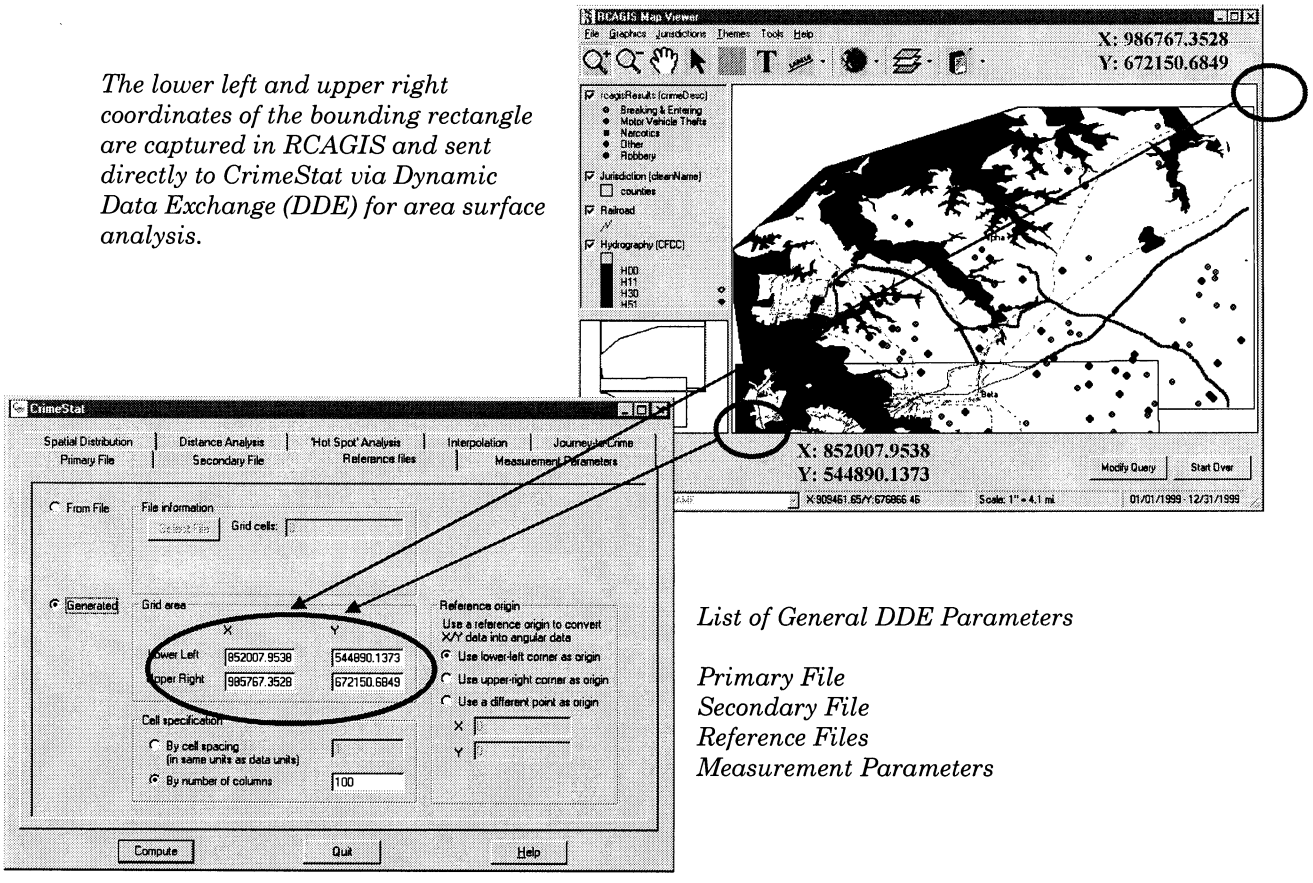
Using Dynamic Data Exchange (DDE) to Develop Software for Interfacing with *CrimeStat*

Ronald E. Wilson
Mapping and Analysis for Public Safety Program
National Institute of Justice
Washington, DC

CrimeStat has the capability to allow software developers to write programs that interface directly with it via Dynamic Data Exchange (DDE). The purpose is to allow for the population of *CrimeStat's* input parameters directly from a separate software application, such as a GIS. Parameters can be specified for automatic population such as the primary and secondary files along with key field variables or reference file coordinates for the area under which *CrimeStat's* algorithms will run an analysis. In addition, measurement parameters can be calculated to provide *CrimeStat* with coverage area or length of street network of an entire region or subset.

Coordinates are often difficult to work with, especially when trying to capture them for measurement or analysis. The Regional Crime Analysis GIS (RCAGIS) program, developed by the U.S. Department of Justice, was designed to interface with *CrimeStat* to provide the coordinates of the bounding rectangle of the area under analysis in order to populate the grid area input boxes of the Reference File with precise coordinates. Instead of writing them down by hand and typing them in manually, the interface between the two applications automates this process easily and more accurately.

The lower left and upper right coordinates of the bounding rectangle are captured in RCAGIS and sent directly to CrimeStat via Dynamic Data Exchange (DDE) for area surface analysis.



List of General DDE Parameters

Primary File
Secondary File
Reference Files
Measurement Parameters

5. Identify the appropriate fields under the Column heading by clicking on the cell and scrolling down to the appropriate name. For the X variable, the relevant name is Lon. For the Y variable, the relevant name is Lat (i.e., that's the names used in this file. However, the variables will not always be simply named). For this example, there are no intensity, weight or time variables.

6. Under Type of Coordinate System, be sure that 'Longitude/latitude (spherical)' is checked since this data set use spherical coordinates.

7. Because the coordinate system are spherical, the data units are automatically decimal degrees. If they were projected, one would have to choose the particular units - feet, meters, miles, kilometers, or nautical miles.

8. This finishes the setup for the primary file. Click on the Secondary File tab.

9. Again, click on select files, locate and open the BaltPop.dbf file.

10. Once loaded, this file has six variables: Blockgroup, lon, lat, area, and density.

11. Define the particular variables. For this file, the X variable is Lon and the Y variable is Lat. Also, define a Z (intensity) variable with Totpop. Note, that you could also assign this name to the Weight variable. Whether the population variable is assigned to the Intensity or Weight variable does not matter to the calculation. However, do <u>not</u> assign this name to <u>both</u> the intensity and the weight (i.e., only use one). This finishes the setup for the secondary variable.

12. Click on the Reference File tab. For these data, you will define a rectangle that covers the study area by identifying the X and Y coordinates for the lower-left corner of the rectangle and the upper-right corner of the rectangles. The following coordinates will work:

	X	Y
Lower-left corner	-76.91	39.19
Upper-right corner	-76.32	39.72

13. You will also need to tell the program how many columns you want it to calculate. The default value of 100 is fine. If you want it finer, type in a larger number. If you want it cruder, type in a smaller number. This finishes the Reference File setup.

14. Clock on the Measurement Parameters tab. There are three parameters that have to be defined.

A. For many routines, an area estimate is needed. For this sample set, 684 square miles works.

B. For the linear nearest neighbor statistic only, the program needs the total length of the street network. In this data, the total street length of the Tiger Files for Baltimore City and Baltimore county are 4868.9 miles.

C. Finally, the type of distance measurement has to be defined, direct or indirect. For this example, use direct measurement.

15. The data setup is now finished. If you want to re-use this data setup, click on the Options page and 'Save parameters'. Define a file name and be sure to give it a 'param' extension (e.g., SampleData.param). The next time you want to run this data set, all you'll need to do is click on the **Options** page, click on 'Load parameters', and click on the name of the parameters file that you saved.

16. You are now ready to run some statistics. For this example, we'll run only four statistics.

17. First, click on the **Spatial Description** page and then click on the Spatial Distribution tab.

    1. Check the Mean center and standard distance (Mcsd) box. Then, click on the 'Save result to' button and identify which GIS program you are writing to (ArcView/ArcGis 'shp'; Atlas*GIS 'BNA'; or MapInfo 'MIF') and give it a name (e.g., SampleData).

    2. Also, check the Standard deviational ellipse (Sde) box and, similarly, choose a file output with a name. You can use the same name (e.g., SampleData). *CrimeStat* will assign a unique prefix to each graphical object.

18. Second, click on the 'Hot Spot' Analysis I tab. Then, check the Nearest Neighbor Hierarchical Clustering (Nnh) box. For this example, keep the default search radius, minimum points per cluster, and number of standard deviations for the ellipses. Also, click on 'Save ellipses to', select a GIS file output, and give it a name. Again, you can use the same name as with the other statistics.

19. Third, click on the **Spatial Modeling** page and then the Interpolation tab. Check the duel kernel density interpolation box. This routine will interpolate the incident distribution (primary file) relative to the population distribution (secondary file). For this example, keep the default kernel parameters (these are explained in more detail in chapter 8). However, be sure to check the Use intensity variable box towards the bottom. This ensures that the duel kernel routine will use the population variable that you assigned when you set up the secondary file.

20. You are now ready to run the statistics. Click on the 'Compute' button. The routine will run until all four routines that you selected are finished; the time will depend on the speed of your computer.

21. Each of the outputs are displayed on a separate results tab. You can print any of these results by clicking on 'Save to text file' (one at a time).

22. You can also display the graphical objects created by the routine in your GIS. Click on 'Close' to close the results window. Then, bring up your GIS and find the objects created by this run. There will be a number of graphical objects associated with the mean center routine (having prefixes of Mc, Xyd, Sdd, Gm, and Hm; see chapter 4 for details). There will be two graphical objects associated with the nearest neighbor clustering routine (with prefixes of Nnh1 and Nnh2). Finally, there will be a grid object created by the duel kernel routine with a Dk prefix. You can load these objects in and display them along with the data file. For the duel kernel grid, you will need to graph the variable called "Z" to see the pattern.

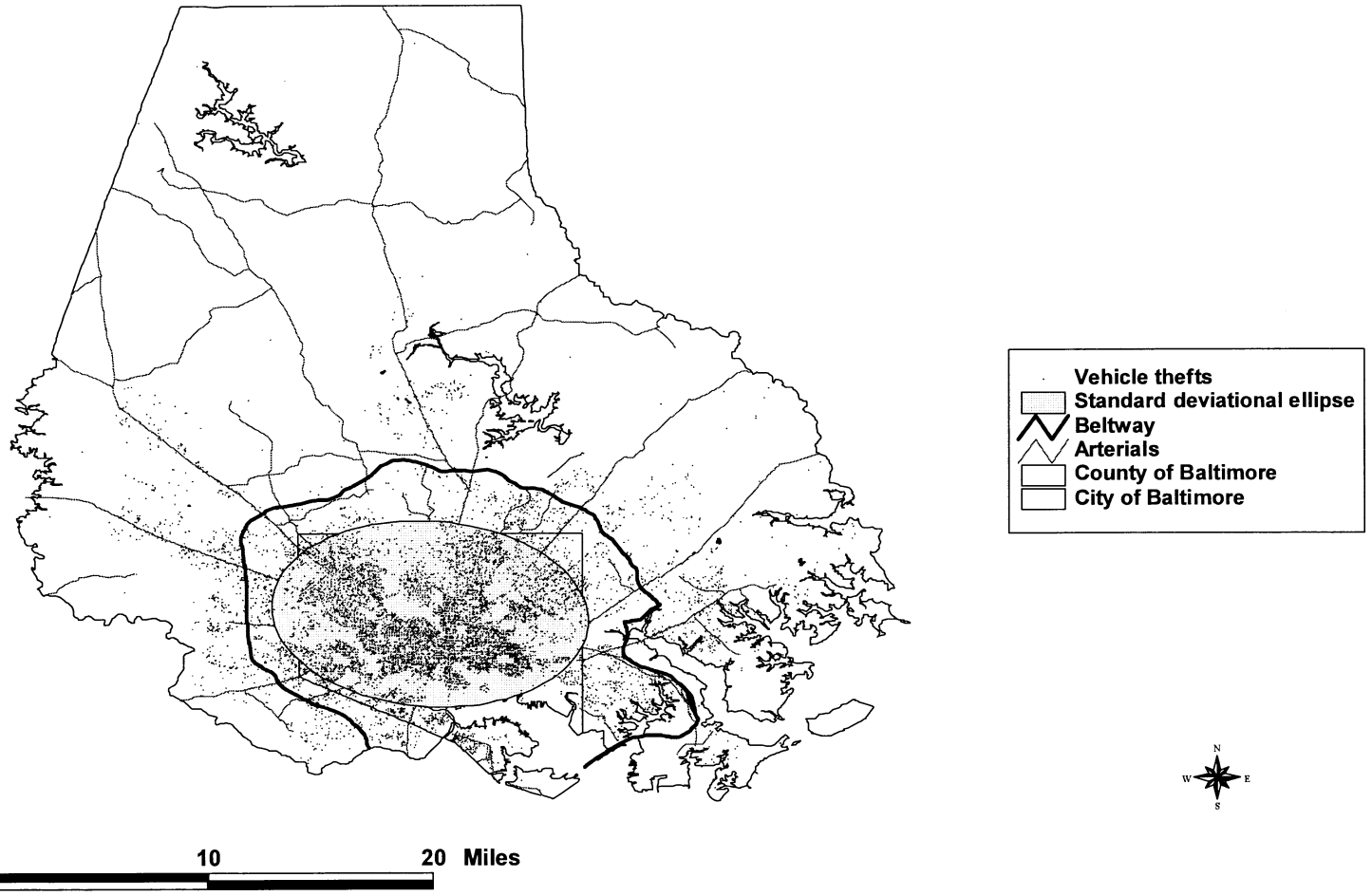
23. For example, figure 3.14 shows an *ArcView*® map of 1996 vehicle thefts in Baltimore City and Baltimore County along with the standard deviational ellipse of the vehicle thefts, calculated with *CrimeStat*. *CrimeStat* outputs the ellipse as a shape file, which is then brought directly into *ArcView*. A similar output could have been done for *MapInfo*®. Most of the statistics in *CrimeStat* have similar visual representations that can be displayed in a GIS program.

24. When you are finished with *CrimeStat*, click on 'Quit' to exit the program.

This finishes the quick tutorial. *CrimeStat* is very easy to set up and to run. In the next chapter, the focus will be on the statistics in the program, starting with the analysis of spatial distributions.

Figure 3.14:

Baltimore Vehicle Thefts: 1996
Location of Incidents and Standard Deviational Ellipse



Endnotes for Chapter 3

1. The spherical 'lat/lon' system is, of course, one type of polar coordinate system. But, it is a polar coordinate system with particular restrictions. Latitudes are angles up to 90^0, north or south of the Equator. Longitudes are angles from 0^0 to 180^0, east and west of the Greenwich Meridian. In the usual polar coordinate system, angles can vary from 0^0 to 360^0.

2. Some *MapInfo* users in Europe have found difficulty in directly reading MIF/MID files from *CrimeStat* and converting them to the particular national coordinate system (e.g., British National Grid, French National Geographic Institute). For example, in the United Kingdom, Pete Jones of the North Wales Police Department has developed a way around this problem. He writes

   "To save the result as a *MapInfo* (.mif) format the following is required:

   MIF Options
   Name of Projection: Earth Projection
   Projection Number: 8
   Datum Number 79

   Before importing the .mif table into *MapInfo* you need to edit it. Open the .mif file with a text editor. You know need to change the following line:

   CoordSys Earth Projection 8, 79

   Change it to

   CoordSys Earth Projection 8, 79, 7, -2, 49, 0.9996012717, 400000, -100000

   Now save the .mif file. You can now import the file into *MapInfo*."

   In France, J. Marc Zaninetti of the University of Orléans figured out how to import graphical objects into *MapInfo* using the French coordinate system. He writes

   "First convert with *MapInfo* your map to the international European Latitude/Longitude ED87 projection system.

   Second, produce the X and Y coordinates and export the data table in Dbase.

   Third, with *CrimeStat II*, modify the Save Output parameter in order to change the origin of the projection. By default, the MIF Options are the following:

   Name of projection: Earth projection
   Projection number: 1 (Latitude longitude)
   Datum number: 33 (international GRS80 origin 0°E, 0°N)

The European norm ED87 has the Datum number 108, so you have to change only this parameter. The new options are the following :

Name of projection: Earth projection
Projection number: 1 (Latitude longitude)
Datum number: 108 (European data ED87).

Finally, you can now import the MIF output tables directly into your *MapInfo* maps."

3. An alternative way to thinking about intensities and weights is to treat both as two different weights - weight #1 and weight #2. For example, weight #1 could be the population in a surrounding zone while weight #2 could be the employment in that same zone. Thus, incidents (e.g., burglaries) could be weighted both by the surrounding population and the surrounding employment. The analogy with double weights is not quite correct since several of the statistics (Moran's I, Geary's C and Local Moran) use only an intensity, but not a weight. The distinction between intensities and weights is historical, relating to the manner in which the statistics have been derived.

4. In *MapInfo*, point data are stored in a table. If the X and Y coordinates are not already part of the table, it will be necessary to add these fields.

   A. Click on *Table Maintenance TableStructure* <tablename>

   B. Click on *Add Field*

   C. Define the X field. If the coordinates are spherical, then an appropriate name might be Longitude or Lon. If the coordinates are projected, then X or XCoord might be appropriate names.

   D. Fill in the parameters of the new name.

      i. The type should be decimal.

      ii. The width should be sufficient to handle the longest string. With spherical coordinates, 12 would be sufficient.

      iii. Be sure to define an appropriate number of decimals places. With longitude, there should be at least 4 decimals places with 6 providing more accuracy. In a projected coordinate system, the number of decimal places would be usually 0 or 1.

   E. Click *OK* when finished.

F. If a Map Basic Window is not already open, click on *Options ShowMapBasicWindow*.

G. Make the Map Basic Window active by clicking on its top border.

H. Inside the window, type

```
update <tablename> set <Xvariablename> = centroidX(obj)
update <tablename> set <Yariablename> = centroidY(obj)
```

After each line, hit <Enter>. The appropriate names would be chosen. For example, if the point table was named robberies and the coordinates were spherical, then the statements would be

```
update robberies set lon=centroidX(obj)
 <Enter>
update robberies set lat=centroidY(obj)
<Enter>
```

I. The X and Y field names should be populated with the correct values for each point. To view the table, click on *Window NewBrowserWindow* <filename>.

J. Save the table as a 'dbf' with 'Save Copy As <name>'. Be sure to specify that the file is to be saved in 'dbf' format.

5. The following steps would be followed to add X and Y coordinates to a 'dbf' file of point locations in *ArcView*.

A. Make the point table active by clicking on it.

B. Open the theme table by clicking on the *Open Theme Table* button.

C. Click on *Table StartEditing*.

D. Click on *Edit AddField*.

E. In the Field Definition window, define a name for the X field (e.g., X, Longitude, Lon).

F. Define the parameters for the X field.

a. Make sure that the type is *Number*

b. Be sure that the width is large enough to handle the largest value. For spherical coordinates (i.e., longitude, latitude), 12 columns should be sufficient. For a projected coordinate system, the number of

columns should be two larger than the largest value.

c. Be sure that there are sufficient decimal places. With a spherical coordinate system, the minimum should be 4 decimal places with 6 being more accurate. With a projected coordinate system, 0 or 1 decimal places would be sufficient.

G. Click *OK* when finished.

H. Repeat steps E through G for the Y field.

I. For the X and Y variable in turn, click on the field name to highlight it.

J. Click on the *Calculate* button.

K. Double-click on the *[Shape]* field name.

L. In the dialog box, type *.GetX* for the X field and *.GetY* for the Y field after *[Shape]*, that is

[Shape].GetX
[Shape].GetY

M. Click *OK* when finished. The field will be populated with the X and Y values for the points in the same units as the data (e.g., lat/lon, feet or meters for UTM or State Plane Coordinates).

6. Note that in an ASCII file, a tab *looks like* it is separated by spaces. However, the underlying ASCII code is different and *CrimeStat* will treat these characteristics differently. That is, if the separator is a tab but the user indicates that it is a space, *CrimeStat* will not properly read the data.

7. Hint: If you type the first letter of the name (e.g., 'L' for longitude), then the program will find the first name that begins with that letter). Typing the letter again will find the second name, and so forth.

8. Since the world is approximately round, all lines are actually circles that eventually come back on to themselves. These are called *Great Circles* because they divide the Earth into two equal halves (Greenhood, 1964). On a sphere, such as the Earth, the shortest distance between any two points is a Great Circle. There are an infinite number of Great Circles, but coordinates are only referenced to two Great Circles. North-south lines are called *Meridians* (and are half Great Circles) and east-west lines are called *Parallels*. The basic reference parallel is the Equator, which is a Great Circle, and the two reference meridians are the Greenwich Meridian and the International Date Line (which is actually the same Great Circle on two sides of the earth).

There are two coordinates - *Longitude* and *Latitude*. For longitude, all east-west directions are defined as an angle from 0^0 to 180^0 with 0^0 being at the Greenwich Meridian and 180^0 being the International Date Line. All directions east of the Greenwich Meridian have a positive longitude whereas all directions west of this meridian have a negative longitude. For example, in the United States, Washington, DC, has a longitude of approximately -77.03 degrees because it is west of the Greenwich Meridian whereas New Delhi, India has a longitude of approximately +77.20 degrees because it is east of the Greenwich Meridian. These locations are approximate because cities cover areas and only a single point within the city has been classified (the center or *centroid* of the city).

For latitude, all north-south directions are defined in terms of an angle from the equator, which has a latitude of 0^0. The maximum is the North or South Poles which have latitudes of $+90^0$ and -90^0 respectively. Locations that are north of the Equator have a positive latitude while locations that are south have a negative latitude. Thus, in the United States, Los Angeles has a latitude of approximately +34.06 degrees whereas Buenos Aires in Argentina has an approximate latitude of -34.60 degrees.

To measure variations between degrees, subdivision of the angles are necessary. The traditional use of spherical coordinates divides angles into multiples of 60 and defines angles in relation to the reference Great Circles. Thus, each degree is subdivided into 60 minutes and each minute, in turn, can be divided into 60 seconds. For example, New York City has an approximate longitude of 73 degrees 58 minute 22 seconds West and an approximate latitude of 40 degrees 52 minutes 46 seconds North. However, with the advent of computers, most coordinates are now converted into decimal degrees. Thus, New York City has an approximate longitude of -77.973 degrees and an approximate latitude of +40.880 degrees. The conversion is simply

$$\text{Decimal degrees} = \text{Degrees} + \text{Minutes}/60 + \text{Seconds}/3600$$

9. Because the Earth is curved, any two dimensional representation produces distortion. The spherical latitude/longitude system (called 'lat/lon' for short) is a universal coordinate system. It is universal because it utilizes the spherical nature of the Earth and each location has a unique set of coordinates. Most other coordinate systems are projected because they are portrayed on a two-dimensional flat plane. Strictly speaking, spherical coordinates - longitudes and latitudes, are not X and Y coordinates since the world is round. However, by convention, they are often referred to as X and Y coordinates, particularly if a small section of the Earth is projected on a flat plane (a computer screen or a printed map).

Projections differ in how they 'flatten' or *project* a sphere onto a two dimensional plane. Typically, there are four properties of maps which cannot all be maintained in any two dimensional representation:

Shape - maintaining correct shape of a land body

Area - if the space represented on a map covers the same area throughout the map, it is called an equal-area map. The proportionality is maintained.

Distance - the distance between two points is in constant scale (i.e., the scale does not change)

Direction - the direction from a point towards another point is true.

Any projection creates one or more types of distortion and particular projections are chosen in order to have accuracy in one or two of these properties. Different projections portray different types of information. Most projections assume that the Earth is a sphere, a situation that is not completely true. The Earth's diameter at the equator is slightly greater than the distance between the poles (Snyder, 1987). The circumference of the Earth between the Poles is about 24,860 miles on a meridian; the circumference at the Equator is about 75 miles more.

There is an infinite number of projections. However, only a couple dozen have been used in practice (Greenhood, 1964; Snyder, 1987; Snyder and Voxland, 1989). They are based on projections of the sphere onto a cylinder, cone or flat plane. In the United States, several common coordinate systems are used. Theoretically, the projection and the coordinate system can be distinguished (i.e., a particular projection could use one of several coordinate systems, e.g. meters or feet). However, in practice, particular projections use common coordinates. Among the most common in use in the United States are:

A. Mercator - The *Mercator* is an early projection, and one of the most famous, which is used for world maps. The projection is done on a cylinder, which is vertically centered on a meridian, but touching a parallel. The globe is projected on the cylinder as if light is emanating from the center of the globe while the Earth turns. The meridians cut the equator at equal intervals. However, they maintain parallel lines, unlike the globe where they converge at the poles. The longitudes are stretched with increasing latitude (in both north and south directions) up until the 80^{th} parallel. The effect is that shape is approximately correct and direction is true. Distance, however, is distorted. For example, on a Mercator map, Greenland appears as big as the United States, which it is not. Distances can be measured in any units for a Mercator though usually they are measured in miles or kilometers.

B. Transverse Mercator - If the Mercator is rotated 90^0 so that the cylinder is centered on a parallel, rather than a meridian, it is called a *Transverse Mercator*. The cylinder is projected as being horizontal but is touching a meridian. The Transverse Mercator is divided into narrow north-south zones in order to reduce distortion. The meridian that the cylinder is touching is called the *Central Meridian* of the zone. Distances are accurate within a

limited distance from the central meridian. Thus, the boundaries of zones are selected in order to maintain reasonable distance accuracy. In the U.S., many states use the Transverse Mercator as the basis for their state plane coordinate system including Arizona, Hawaii, Illinois, and New York.

C. Universal Transverse Mercator (UTM) - In 1936, the International Union of Geodesy and Geophysics established a standard use of the Transverse Mercator, called the *Universal Transverse Mercator* (or UTM). In order to reduce distortion, the globe is divided into 60 zones, 6 degrees of longitude wide. For latitude, each zone is divided further into strips of 8 degrees latitude, from 84° N to 80° S. Within each band, there is a central meridian which, in theory, would be geodetically true. But, to reduce distortion across the area covered by each zone, scale along the central meridian is reduced to 0.9996. This produces two parallel lines of zero distortion approximately 180 km away from the central meridian. Scale at the boundary of the zone is approximately 1.0003 at U.S. latitudes. Coordinates are expressed in meters. By convention, the origin is the lower left corner of the zone. From the origin, *Eastings* are displacements eastward and from the origin, *Northings* are displacements northward. The central meridian is given an Easting of 500,000 meters. The Northing for the equator varies depends on the hemisphere. For the northern hemisphere, the equator has a Northing of 0 meters. For the southern hemisphere, the Equator has a Northing of 10,000,000 meters. The UTM system was adopted by the U.S. Army in 1947 and has been adopted by many national and international mapping agencies. Distances are always measured in meters in UTM.

D. Oblique Mercator - There are a number of cylindrical projections which are neither centered on a meridian (as in the Mercator) or on a parallel (as in the Transverse Mercator). These are called *Oblique Mercator* projections because the cylinder is centered on a line which is oblique to parallels or meridians. In the U.S., the *Hotine Oblique Mercator* is used for Alaska.

E. Lambert Conformal Conic - The *Lambert Conformal Conic* is a projection made on a cone, rather than a cylinder. Lambert's conformal projection centers the cone over a central location (usually the North Pole) and the cone 'cuts' through the globe at parallels chosen to be standards. Within those standards, shapes are true and meridians are straight. Outside those standards, parallels are spaced at increasing intervals the further north or south they go to reduce distance distortion. The projection is the basis of many state plane coordinate systems, including California, Connecticut, Maryland, Michigan, and Virginia.

F. Alber's Equal-Area - Another projection on a cone is the *Albers Equal-Area* except that parallels are spaced at decreasing intervals the further north or south they are placed from the standard parallels. The map is an equal-area projection and scale is true in the east-west direction.

G. State Plane Coordinates - Every state in the United States has an official coordinate system, called the *State Plane Coordinate System*. Each state is divided into one or more zones and a particular projection is used for each zone. With the exception of Alaska, which uses the Hotine Oblique Mercator for one of its eight zones, all state plane coordinate systems use either the Transverse Mercator or the Lambert Conformal Conic. Each state's shape determines which projection is chosen to represent that state. Typically, states extending in a north-south direction use Transverse Mercator projections while states extending in an east-west direction use Lambert Conformal Conic projections. But, there are exceptions, such as California which uses the Lambert. Projections are chosen to minimize distortion over the state. Several states use both projections (Florida, New York) and Alaska uses all three. Distances are measured in feet.

See Snyder (1987) and Snyder and Voxland (1989) for more details on these and other projections including the mathematical transformations used in the various projections. Other good references are Maling (1973), Robinson, Sale, Morrison and Muehrcke (1984), and the Committee on Map Projections (1986).

10. With a projected coordinate system, indirect distances can be measured by perpendicular horizontal or vertical lines on a flat plane because all direct paths between two points have equal distances. For example in figure 3.13, whether the distance is measured from point A north to the Y-coordinate of point B and then eastward until point B is reached or, alternatively, from point A eastward to the X-coordinate of point B, then northward until point B is reached, the distances will be the same. One of the advantages of a Manhattan geometry is that travel distances that are direct (i.e., that are pointed towards the final direction) are equal.

With a spherical coordinate system, however, Manhattan distances are not equal with different routes. Because the distance between two points at the same latitude decreases with increasing latitude (north or south) from the equator, the path between two points will differ on the route with Manhattan rules. In figure 3.13, for example, it is a longer distance to travel from point A eastward to the longitude of point B, before traveling north to point B than to travel northward from point A to the same latitude as point B before traveling eastward to point B. Consequently, *CrimeStat* modifies the Manhattan rules for a spherical coordinate system by calculating both routes between two points and averaging them. This is called a *Modified Spherical Manhattan Distance*.

CrimeStat III

Part II: Spatial Description

Chapter 4
Spatial Distribution

In this chapter, the spatial distribution of crime incidents will be discussed. The statistics that are used in describing the spatial distribution of crime incidents will be explained and will be illustrated with examples from *CrimeStat® III*. For the examples, crime incident data from Baltimore County and Baltimore City will be used. Figure 4.1 shows the user interface for the spatial distribution statistics in *CrimeStat*. For each of these, the statistics will first be presented followed by examples of their use in crime analysis.

Centrographic Statistics

The most basic type of descriptors for the spatial distribution of crime incidents are *centrographic statistics*. These are indices which estimate basic parameters about the distribution (Lefever, 1926; Furfey, 1927; Bachi, 1957; Neft, 1962, Hultquist, Brown and Holmes, 1971; Ebdon, 1988). They include:

1. Mean center
2. Median center
3. Center of minimum distance
4. Standard deviation of X and Y coordinates
5. Standard distance deviation
6. Standard deviational ellipse

They are called centrographic in that they are two dimensional correlates to the basic statistical moments of a single-variable distribution - mean, standard deviation, skewness, and kurtosis (see Bachi, 1957). They have been applied to crime analysis by Stephenson (1980) and, more recently, by Langworthy and Jefferis (1998).

Because two dimensions adds complexity not seen in one dimension, these statistical moments have been modified to be appropriate. Figure 4.2 shows how the centrographic statistics are selected in *CrimeStat*.

Mean Center

The simplest descriptor of a distribution is the *mean center*. This is merely the mean of the X and Y coordinates. It is sometimes called a *center of gravity* in that it represents the point in a distribution where all other points are balanced if they existed on a plane and the mean center was a fulcrum (Ebdon, 1988; Burt and Barber, 1996).

For a single variable, the mean is the point at which the sum of all differences between the mean and all other points is zero. Unfortunately, for two variables, such as the location of crime incidents, the mean center is not necessarily the point at which the sum of all distances to all other points is minimized. That property is attributed to the

Figure 4.1: Spatial Distribution Screen

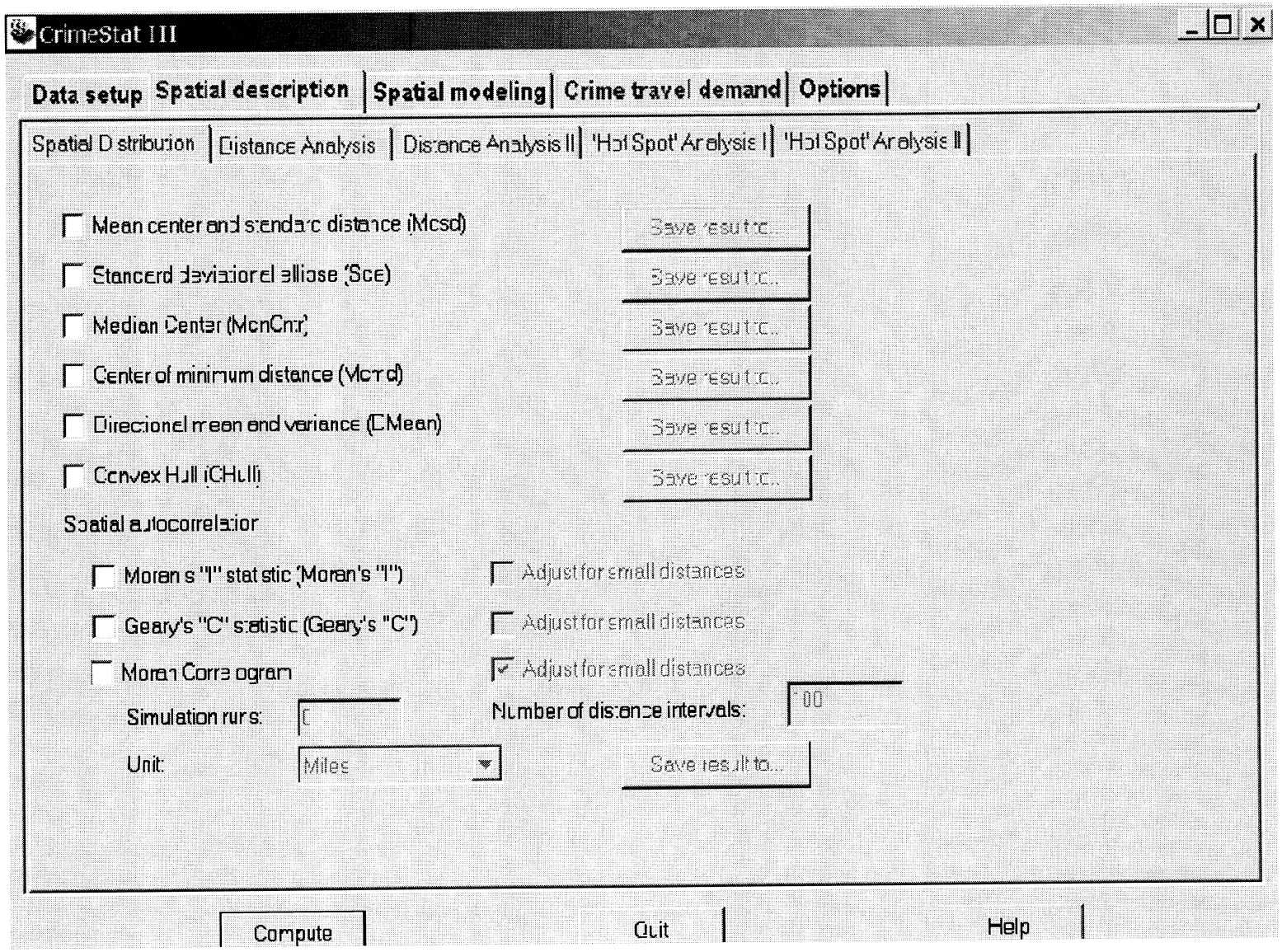
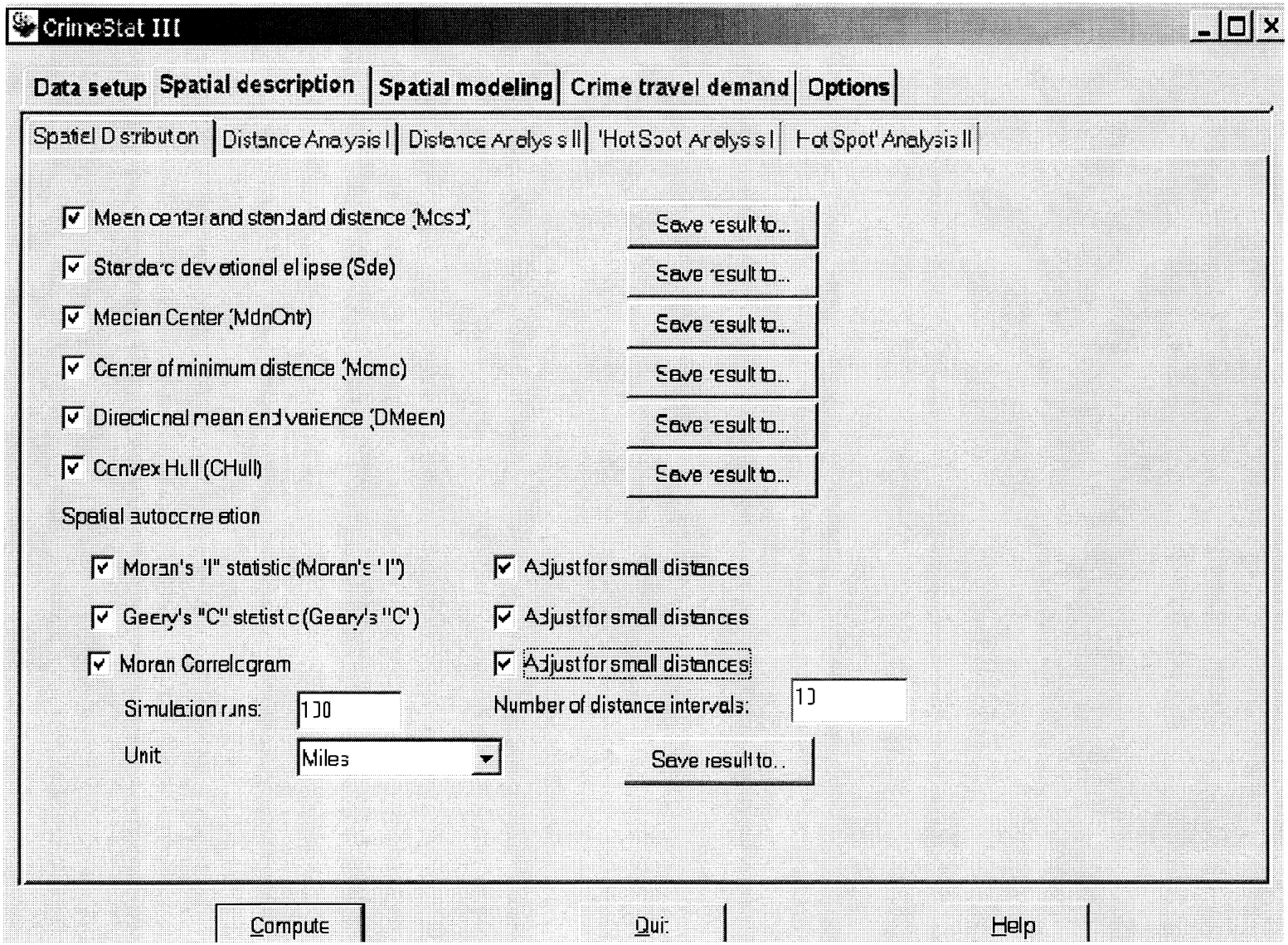


Figure 4.2: Selecting Centrographic Statistics



center of minimum distance (see below). However, the mean center can be thought of as a point where both the sum of all differences between the mean X coordinate and all other X coordinates is zero and the sum of all differences between the mean Y coordinate and all other Y coordinates is zero.

The formula for the mean center is:

$$\bar{X} = \sum_{i=1}^{N} \frac{X_i}{N} \qquad \bar{Y} = \sum_{i=1}^{N} \frac{Y_i}{N} \tag{4.1}$$

where X_i and Y_i are the coordinates of individual locations and N is the total number of points.

To take a simple example, the mean center for burglaries in Baltimore County has spherical coordinates of longitude -76.608482, latitude 39.348368 and for robberies longitude -76.620838, latitude 39.334816. Figure 4.3 illustrates these two mean centers.

Weighted Mean Center

A weighted mean center can be produced by weighting each coordinate by another variable, W_i. For example, if the coordinates are the centroids of census tracts, then the weight of each centroid could be the population within the census tract. Formula 4.1 is extended slightly to include a weight.

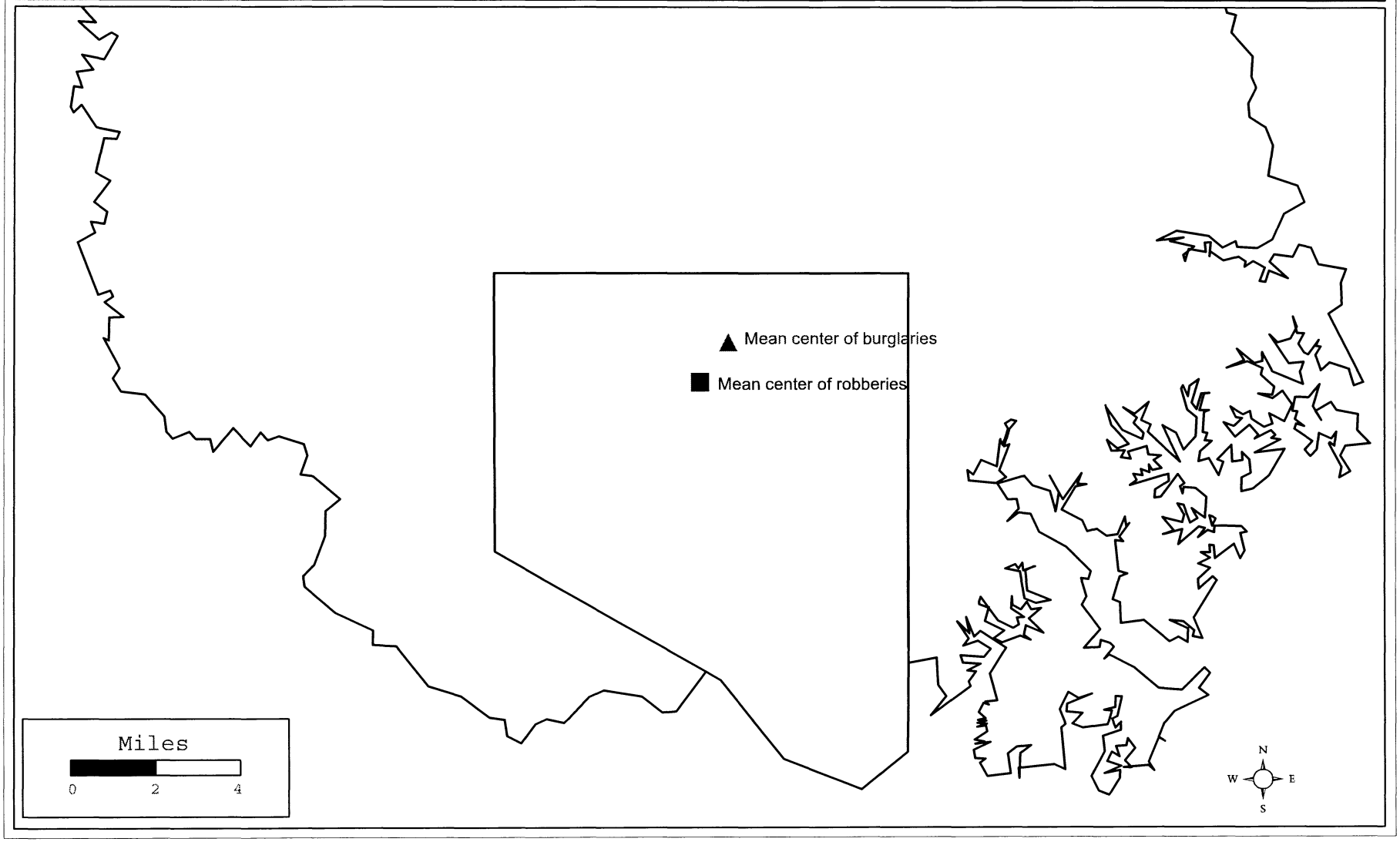
$$\bar{X} = \sum_{i=1}^{N} \frac{W_i X_i}{N} \qquad \bar{Y} = \sum_{i=1}^{N} \frac{W_i Y_i}{N} \tag{4.2}$$

The advantage of a weighted mean center is that points associated with areas can have the characteristics of the areas included. For example, if the coordinates are the centroids of census tracts, then the weight of each centroid could be the population within the census tract. This will produce a different center of gravity than, say, the unweighted center of all census tracts. *CrimeStat* allows the mean to be weighted by either the weighting variable or by the intensity variable. Users should be careful, however, not to weight the mean with both the weighting and intensity variable unless there is an explicit distinction being made between weights and intensities.

To take an example, in the six jurisdictions making up the metropolitan Baltimore area (Baltimore City, and Baltimore, Carroll, Harford, Howard and Anne Arundel counties), the mean center of all census block groups is longitude -76.619121, latitude 39.304344. This would be an *unweighted* mean center of the block groups. On the other hand, the mean center of the 1990 population for the Baltimore metropolitan area had coordinates of longitude -76.625186 and latitude 39.304186, a position slightly southwest of the unweighted mean center. Weighting the block groups by median household income

Figure 4.3: Burglary and Robbery in Baltimore County

Comparison of Mean Centers



produces a mean center which is still more southwest. Figure 4.4 illustrates these three mean centers.

Weighted mean centers can be useful because they describe spatial differentiation in the metropolitan area and factors that may correlate with crime distributions. Another example is the weighted mean centers of different ethnic groups in the Baltimore metropolitan area (figure 4.5). The mean center of the White population is almost identical to the unweighted mean center. On the other hand, the mean center of the African-American/Black population is southwest of this and the mean center of the Hispanic/Latino population is considerably south of that for the White population. In other words, different ethnic groups tend to live in different parts of the Baltimore metropolitan area. Whether this has any impact on crime distributions is an empirical question. As we will see, there is not a simple spatial correlation between these weighted mean centers and particular crime distributions.

When the *Mcsd* box is checked, *CrimeStat* will run the routine. *CrimeStat* has a status bar that indicates how much of the routine has been run (Figure 4.6).[1] The results of these statistics are shown in the *Mcsd* output table (figure 4.7).

Median Center

The median center is the intersection between the median of the X coordinate and the median of the Y coordinate. The concept is simple. However, it is not strictly a median. For a single variable, such as median household income, the median is that point at which 50% of the cases fall below and 50% fall above. On a two dimensional plane, however, there is not a single median because the location of a median is defined by the way that the axes are drawn. For example, in figure 4.8, there are eight incident points shown. Four lines have been drawn which divide these eight points into two groups of four each. However, the four lines do not identify an exact location for a median. Instead, there is an area of non-uniqueness in which any part of it could be considered the 'median center'. This violates one of the basic properties of a statistic is that it be a unique value.

Nevertheless, as long as the axes are not rotated, the median center can be a useful statistic. The *CrimeStat* routine outputs three statistics:

1. The sample size
2. The median of X
3. The median of Y

The tabular output can be printed and the median center can be output as a graphical object to ArcView 'shp', MapInfo 'mif' or Atlas*GIS 'bna' files. A root name should be provided. The median center is output as a point (MdnCntr<root name>).

Figure 4.4: Center of Baltimore Metropolitan Population

Mean Center of Block Groups Weighted By Selected Variables

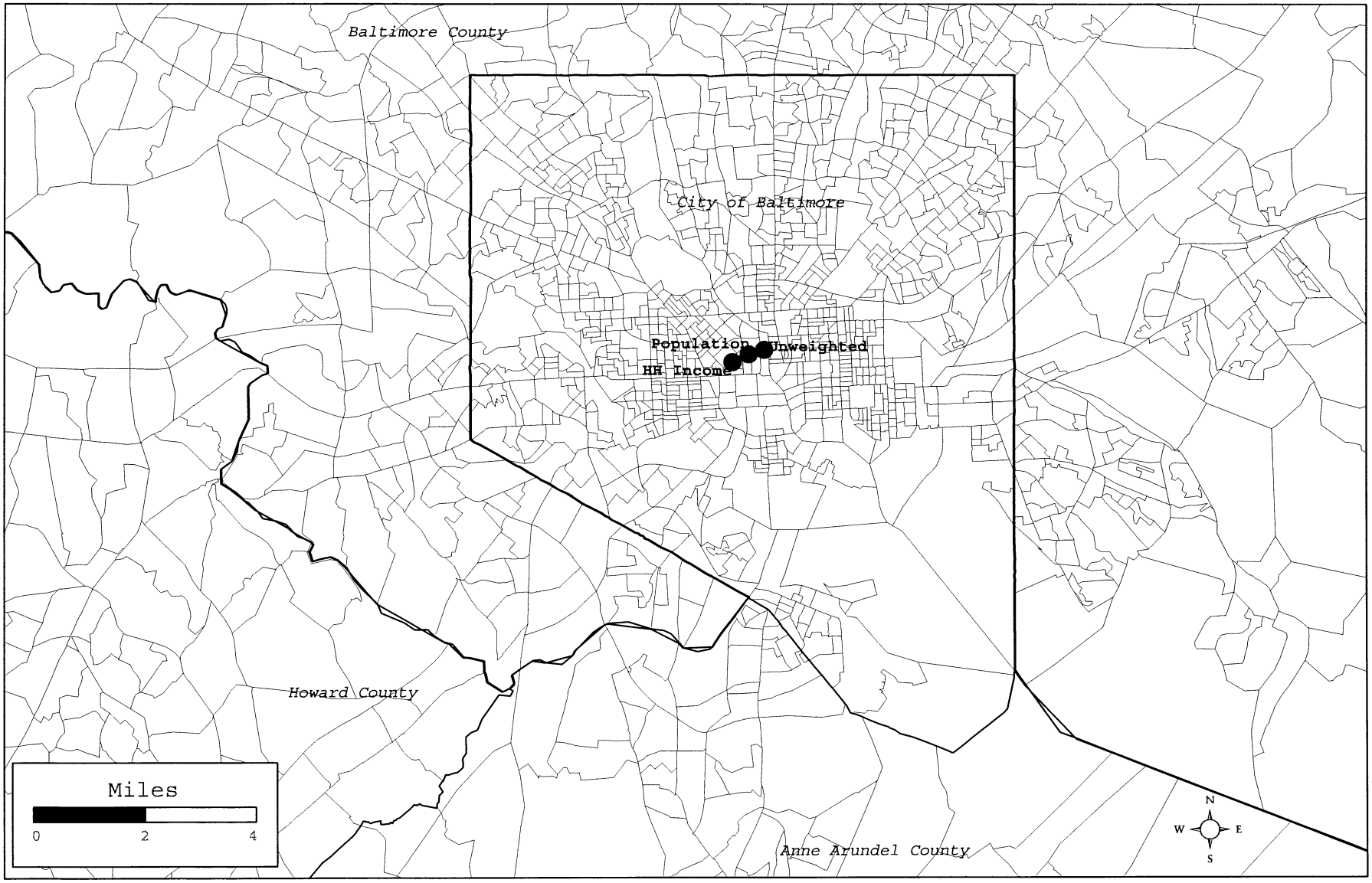


Figure 4.5: Center of Baltimore Metropolitan Population

Mean Center of Block Groups Weighted By Selected Variables

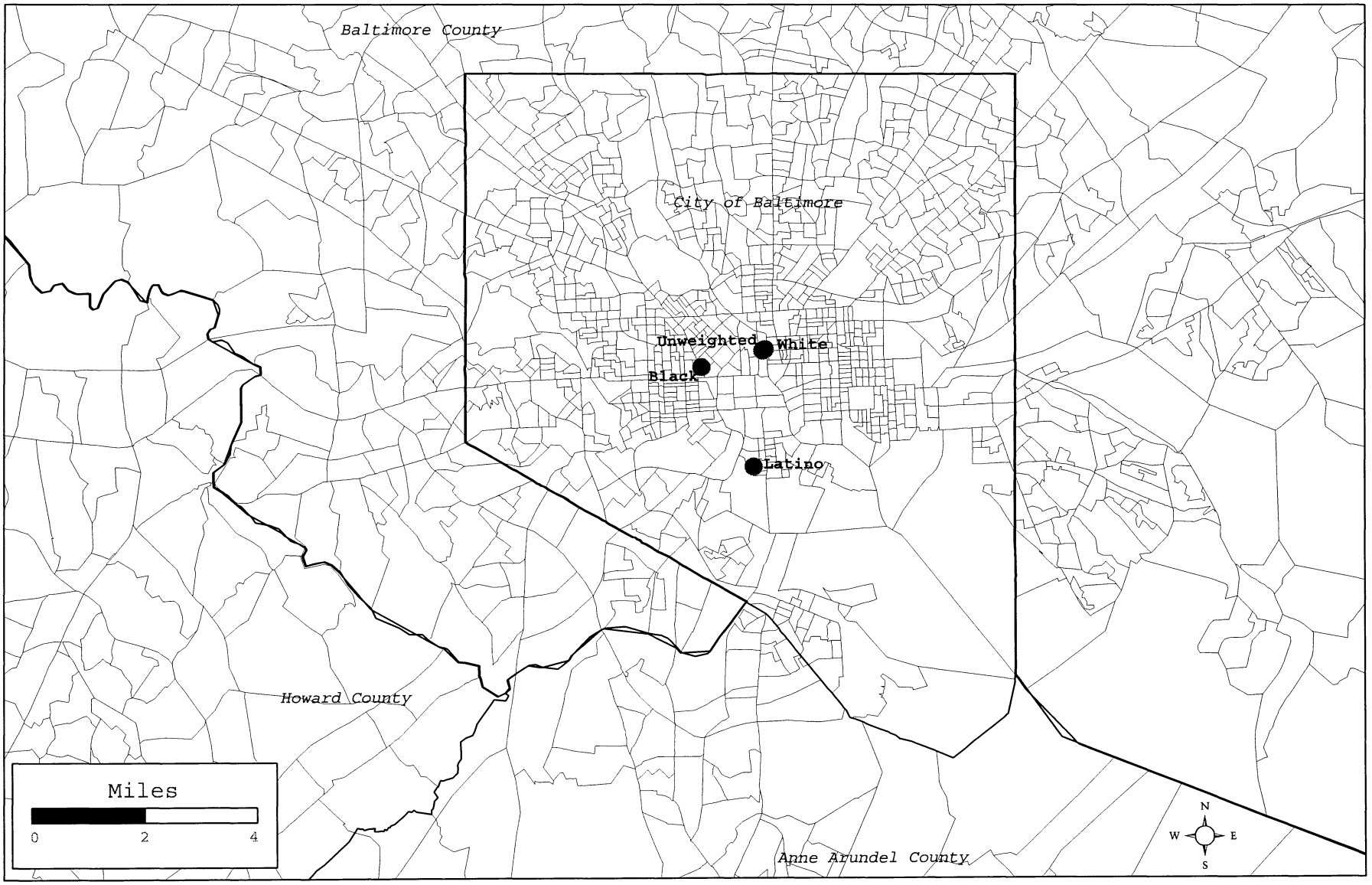


Figure 4.6: *CrimeStat* Calculating A Routine

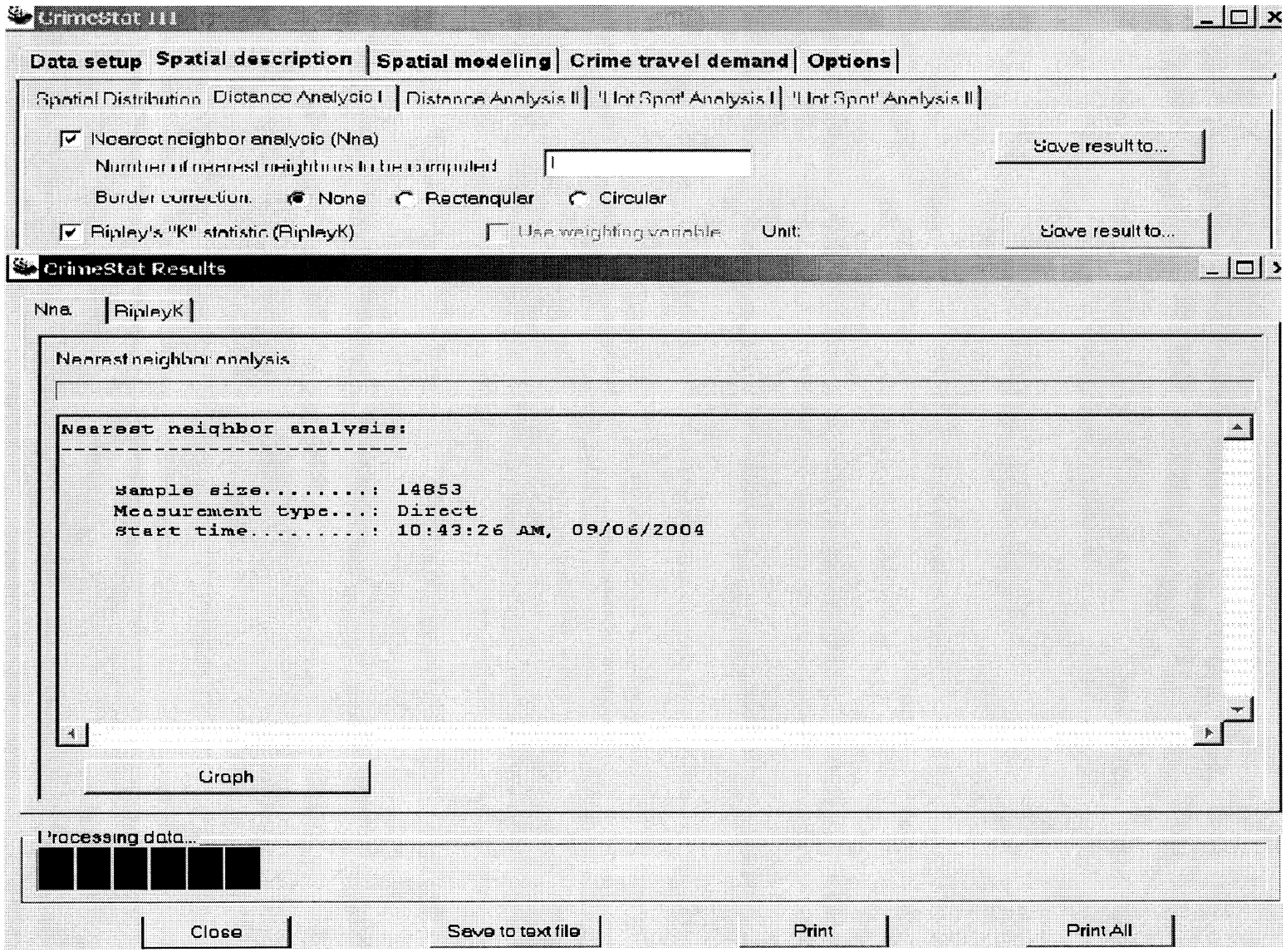


Figure 4.7: Mean Center and Standard Distance Deviation Output

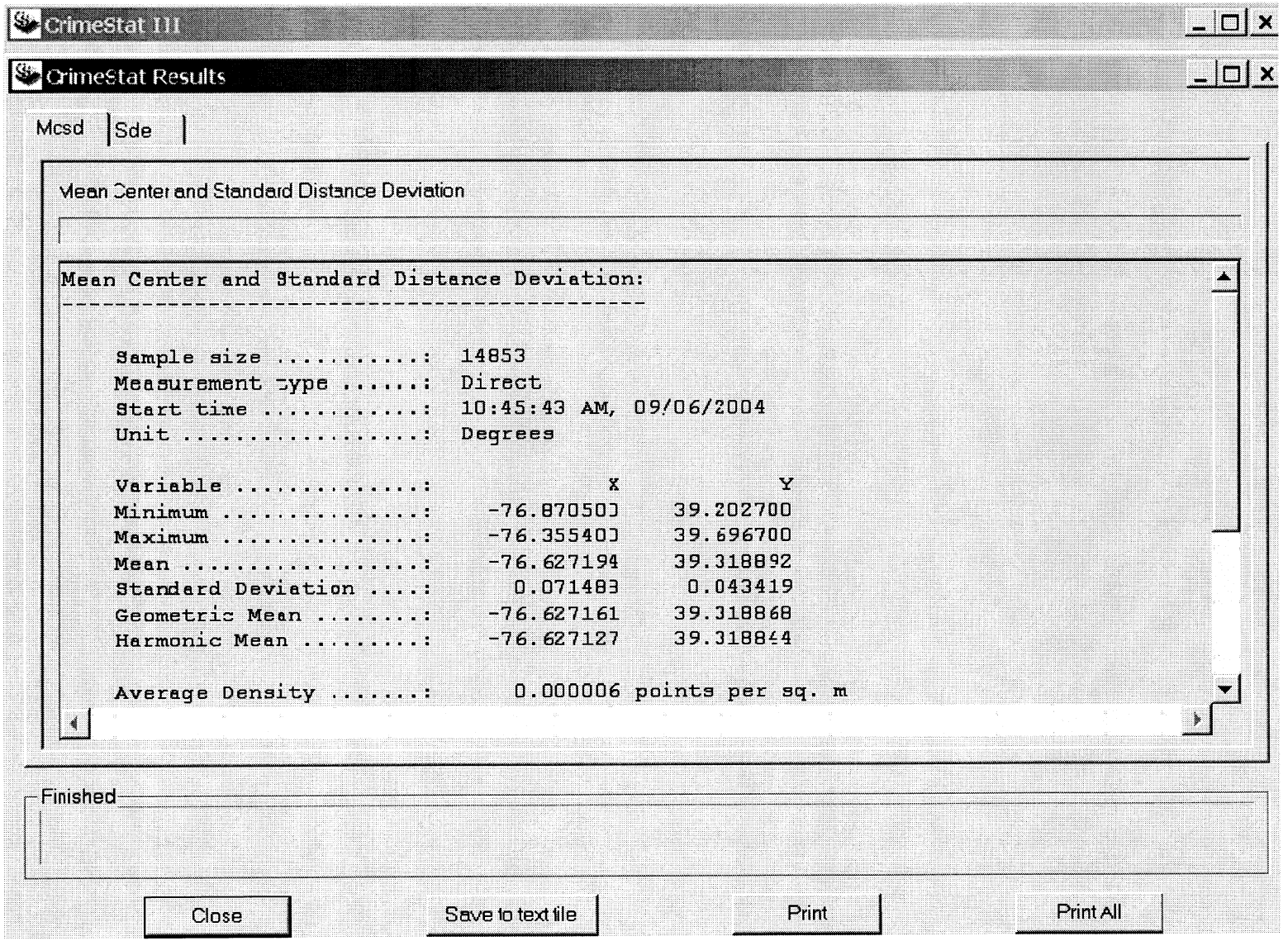
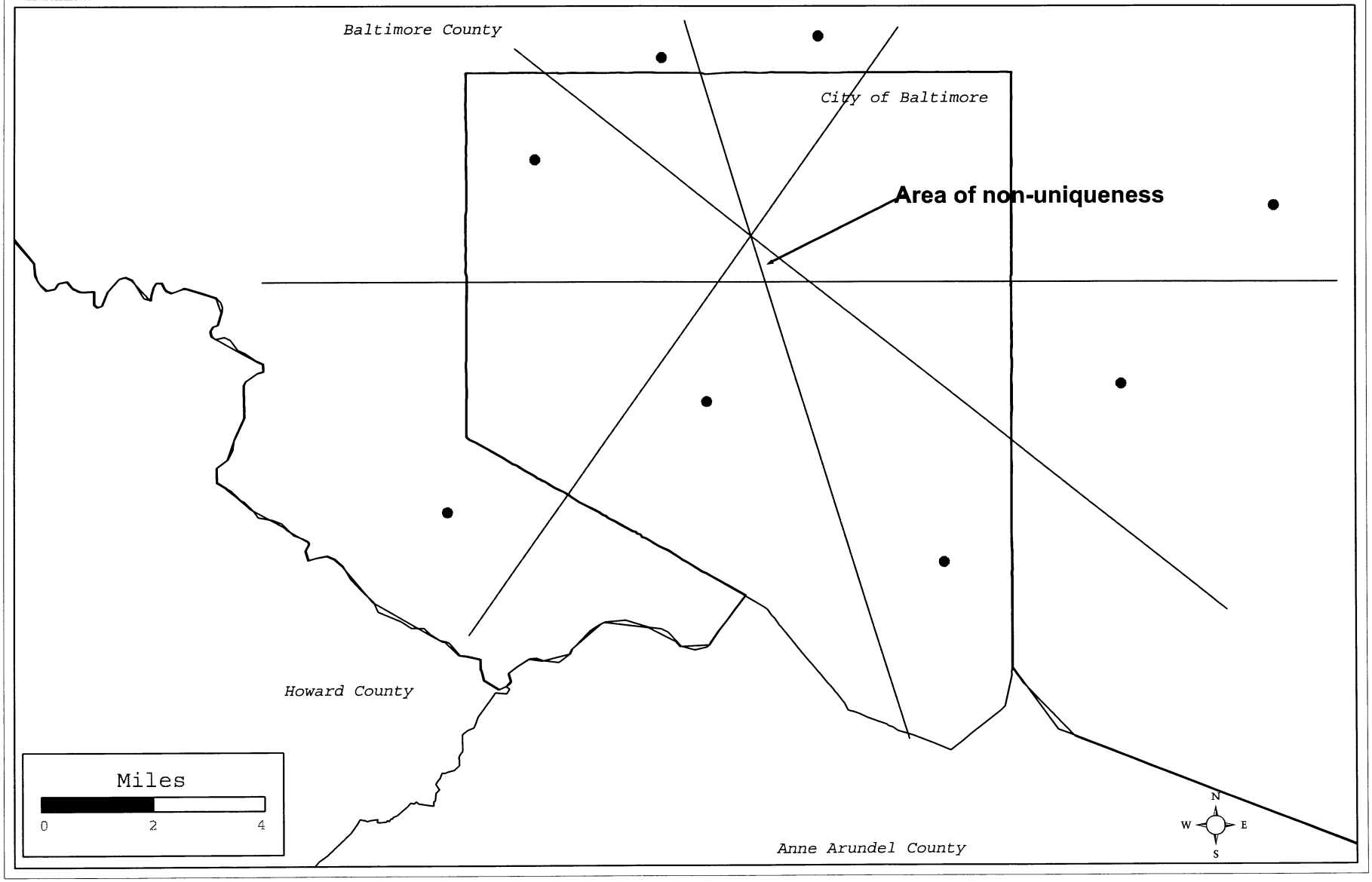


Figure 4.8: Non-Uniqueness of a Median Center

Lines Splitting Incident Locations Into Two Halves



Center of Minimum Distance

Another centrographic statistic is the *center of minimum distance*. Unfortunately, this statistic is sometimes also called the *median center*, which can make it confusing since the above statistic has the same name. Nevertheless, unlike the median center above, the center of minimum distance is a unique statistic in that it defines the point at which the sum of the distance to all other points is the smallest (Burt and Barber, 1996). It is defined as:

$$\text{Center of Minimum Distance} = C = \sum_{i=1}^{N} d_{ic} \text{ is a minimum} \qquad (4.3)$$

where d_{ic} is the distance between a single point, i, and C, the center of minimum distance (with an X and Y coordinate). Unfortunately, there is not a formula that can calculate this location.

Instead, an iterative algorithm is used that approximates this location (Kuhn and Kuenne, 1962; Burt and Barber, 1996). Depending on whether the coordinates are spherical or projected, *CrimeStat* will calculate distance as either Great Circle (spherical) or Euclidean (projected), as discussed in the previous chapter.[2] The results are shown in the *Mcmd* output table (figure 4.9).

The importance of the center of minimum distance is that it is a location where distance to all the defining incidents is the smallest. Since *CrimeStat* only measures distances as either direct or indirect, actual travel time is not being calculated. But in many jurisdictions, the minimum distance to all points is a good approximation to the point where travel distances are minimized. For example, in a police precinct, a patrol car could be stationed at the center of minimum distance to allow it to respond quickly to calls for service.

For example, figure 4.10 maps the center of minimum distance for 1996 auto thefts in both Baltimore City and Baltimore County and compares this to both the mean center and the median center statistic. As seen, both the center of minimum distance and the median center are south of the mean center, indicating that there are slightly more incidents in the southern part of the metropolitan area than in the northern part. However, the difference in these three statistics is very small, especially the median center and the center of minimum distance.

Standard Deviation of the X and Y Coordinates

In addition to the mean center and center of minimum distance, *CrimeStat* will calculate various measures of spatial distribution, which describe the dispersion, orientation, and shape of the distribution of a variable (Hammond and McCullogh 1978; Ebdon 1988). The simplest of these is the raw standard deviations of the X and Y

Figure 4.9: Center of Minimum Distance Output

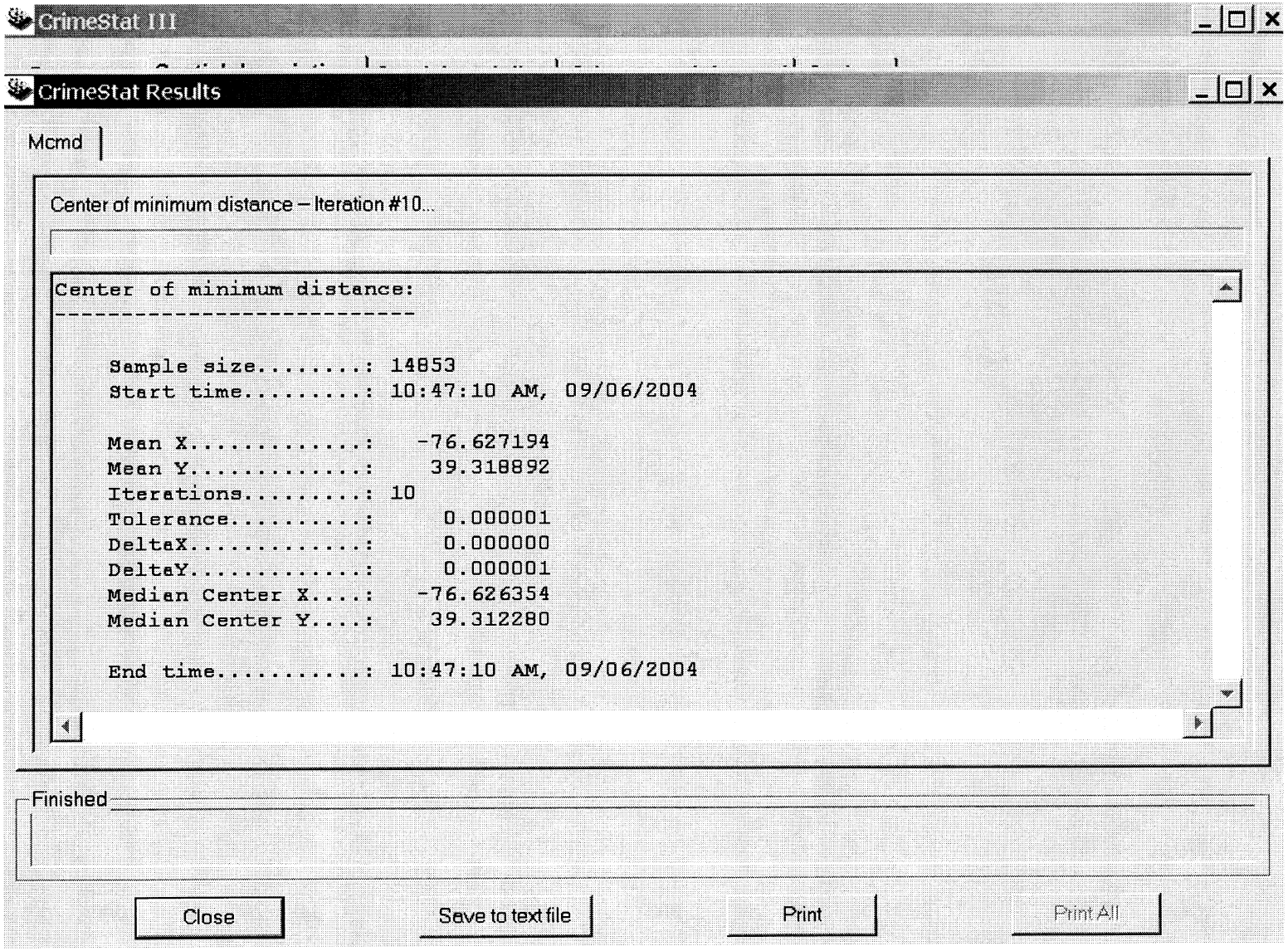
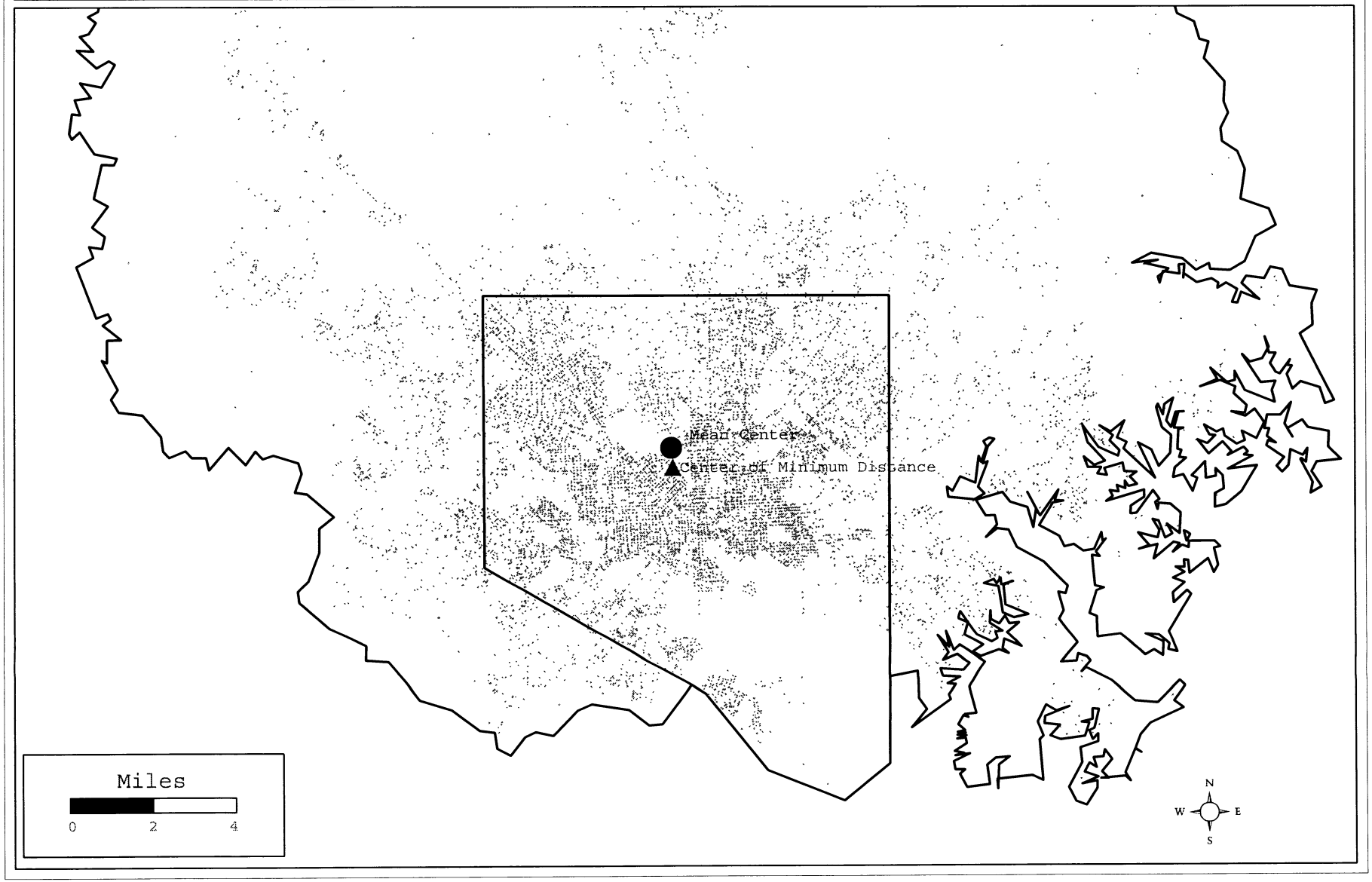


Figure 4.10: 1996 Metropolitan Baltimore Auto Thefts

Mean Center and Center of Minimum Distance for 1996 Auto Thefts



coordinates, respectively. The formulas used are the standard ones found in most elementary statistics books:

$$S_X = \text{SQRT}\left[\sum_{i=1}^{N} \frac{(X_i - \bar{X})^2}{N-1}\right] \quad (4.4)$$

$$S_Y = \text{SQRT}\left[\sum_{i=1}^{N} \frac{(Y_i - \bar{Y})^2}{N-1}\right] \quad (4.5)$$

where X_i and Y_i are the X and Y coordinates for individual points, $\bar{X}$ and $\bar{Y}$ are the mean X and mean Y, and N is the total number of points. Note that 1 is subtracted from the number of points to produce an unbiased estimate of the standard deviation.

The standard deviations of the X and Y coordinates indicate the degree of dispersion. Figure 4.11 shows the standard deviation of the coordinates for auto thefts and represents this as a rectangle. As seen, the distribution of auto thefts spreads more in an east-west direction than in a north-south direction.

Standard Distance Deviation

While the standard deviation of the X and Y coordinates provides some information about the dispersion of the incidents, there are two problems with it. First, it does not provide a single summary statistic of the dispersion in the incident locations and is actually two separate statistics (i.e., dispersion in X and dispersion in Y). Second, it provides measurements in the units of the coordinate system. Thus, if spherical coordinates are being used, then the units will be decimal degrees.

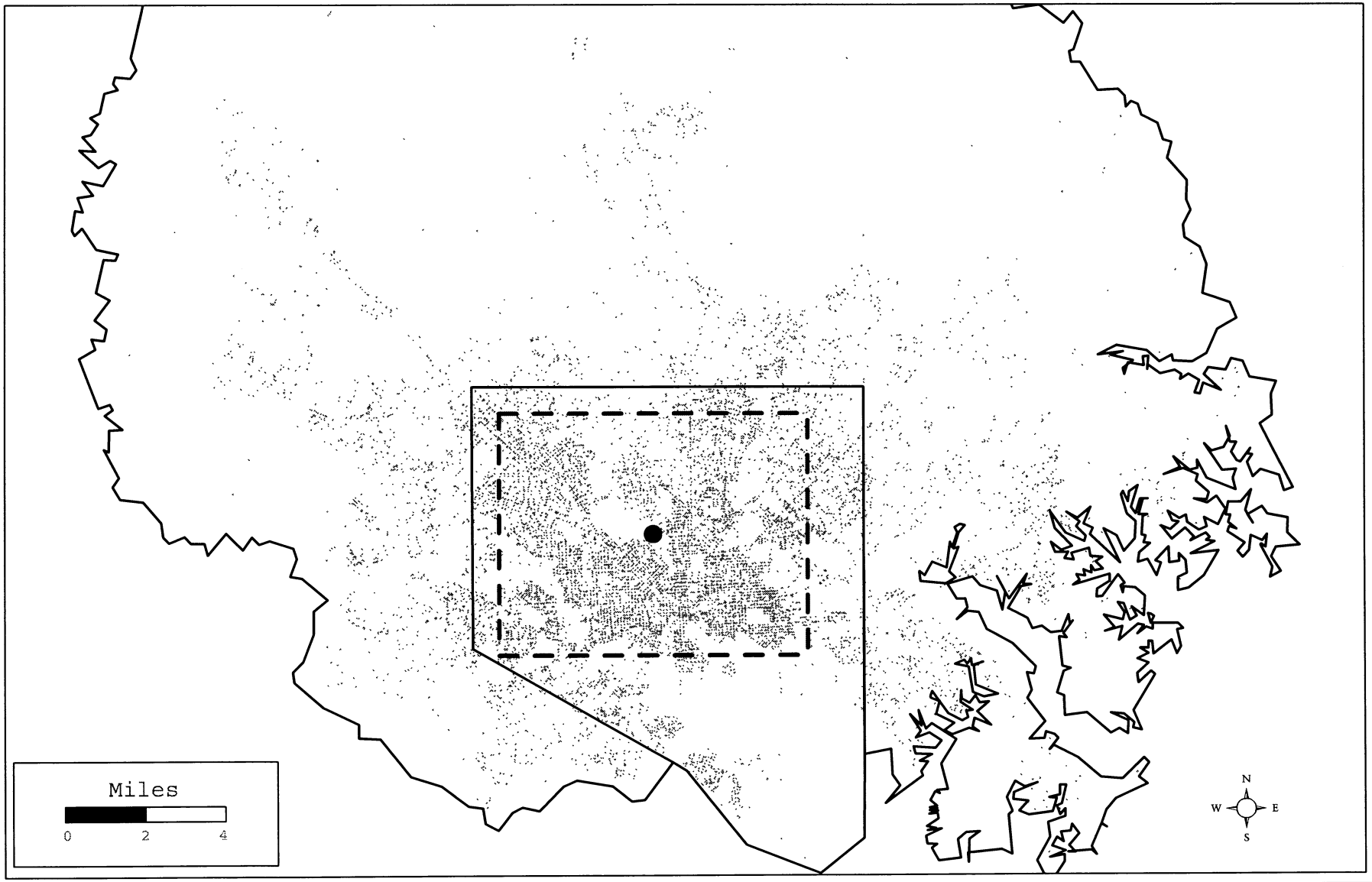
A measure which overcomes these problems is the *standard distance deviation* or *standard distance*, for short. This is the standard deviation of the *distance* of each point from the mean center and is expressed in measurement units (feet, meters, miles). It is the two-dimensional equivalent of a standard deviation.

The formula for it is

$$S_{XY} = \text{Sqrt}\left[\sum_{i=1}^{N} \left\{\frac{(d_{iMC})^2}{N-2}\right\}\right] \quad (4.6)$$

Figure 4.11: 1996 Metropolitan Baltimore Auto Thefts

Mean Center and Standard Deviations of X and Y Coordinates



where d_{iMC} is the distance between each point, i, and the mean center and N is the total number of points. Note that 2 is subtracted from the number of points to produce an unbiased estimate of standard distance since there are two constants from which this distance is measured (mean of X, mean of Y).[3]

The standard distance can be represented as a single vector rather than two vectors as with the standard deviation of the X and Y coordinates. Figure 4.12 shows the mean center and standard distance deviation of both robberies and burglaries for 1996 in Baltimore County represented as circles. It is clear that the spatial distributions of these two types of crime vary with robberies being slightly more concentrated.

Standard Deviational Ellipse

The standard distance deviation is a good single measure of the dispersion of the incidents around the mean center. However, with two dimensions, distributions are frequently skewed in one direction or another (a condition called *anisotropy*). Instead, there is another statistic which gives dispersion in two dimensions, the *standard deviational ellipse* or *ellipse*, for short (Ebdon, 1988; Cromley, 1992).

The standard deviational ellipse is derived from the bivariate distribution (Furfey, 1927; Neft, 1962; Bachhi, 1957) and is defined by

$$\text{Bivariate Distribution} = \text{SQRT}\ \frac{[\sigma^2_x + \sigma^2_y]}{2} \qquad (4.7)$$

The two standard deviations, in the X and Y directions, are orthogonal to each other and define an ellipse. Ebdon (1988) rotates the X and Y axis so that the sum of squares of distances between points and axes are minimized. By convention, it is shown as an ellipse.

Aside from the mean X and mean Y, the formulas for these statistics are as follows:

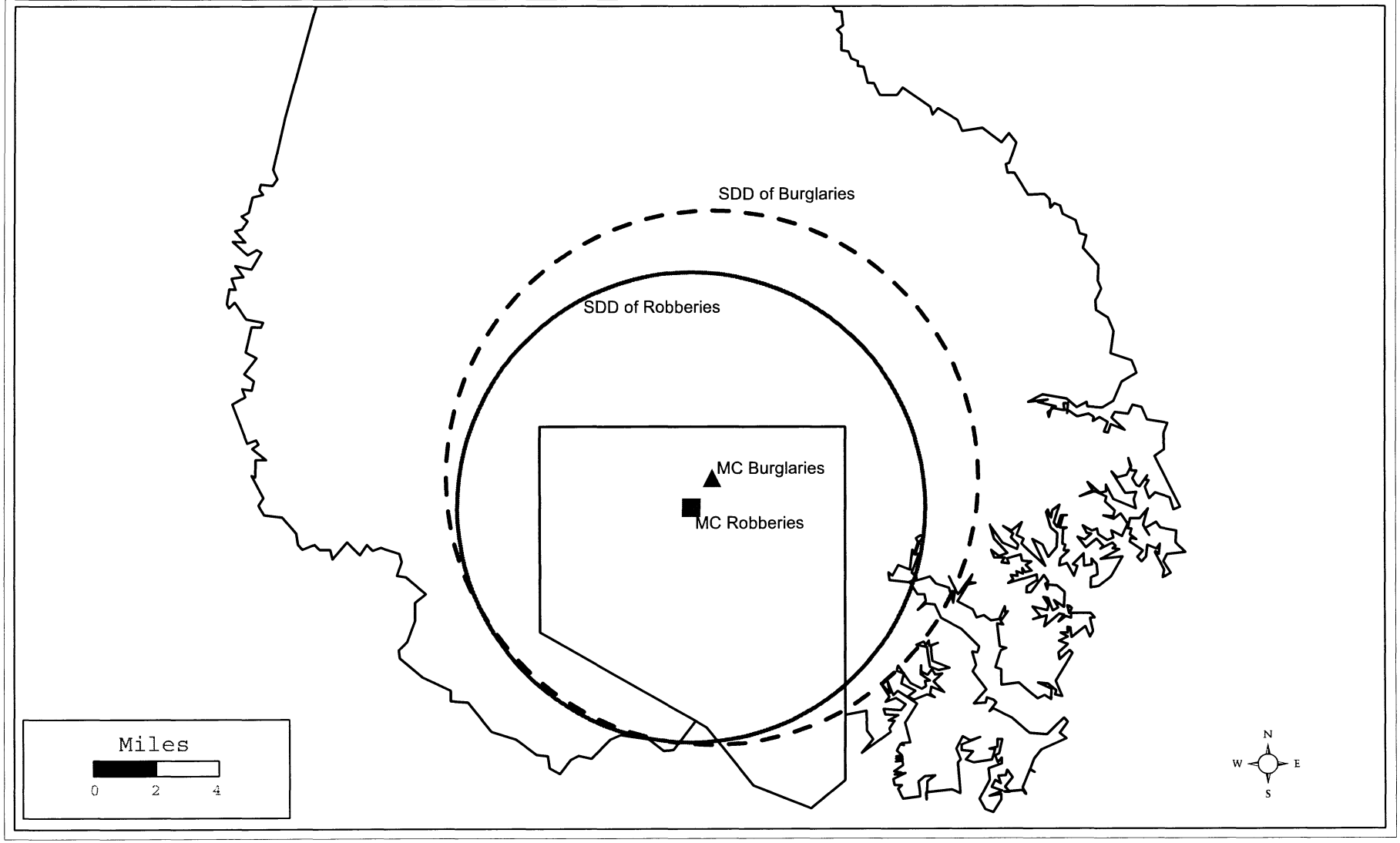
1. The Y-axis is rotated *clockwise* through an angle, θ, where

$$\theta = \text{ARCTAN}\ \{(\Sigma(X_i-\bar{X})^2 - \Sigma(Y_i-\bar{Y})^2) + [(\Sigma(X_i-\bar{X})^2 - \Sigma(Y_i-\bar{Y})^2)^2 + 4(\Sigma(X_i-\bar{X})(Y_i-\bar{Y}))^2]^{1/2}\}/(2\Sigma(X_i-\bar{X})(Y_i-\bar{Y})) \qquad (4.8)$$

where all summations are for i=1 to N (Ebdon, 1988).

Figure 4.12: 1996 Baltimore County Burglaries and Robberies

Comparison of Mean Centers and Standard Distance Deviations



2. Two standard deviations are calculated, one along the transposed X-axis and one along the transposed Y-axis.

$$S_x = SQRT(2)\{\sum_{i=1}^{N}[(X_i - \bar{X})Cos\theta - (Y_i - \bar{Y})Sin\theta]^2/(N-2)\}^{1/2} \quad (4.9)$$

$$S_y = SQRT(2)\{\sum_{i=1}^{N}[(X_i - \bar{X})Sin\theta - (Y_i - \bar{Y})Cos\theta]^2/(N-2)\}^{1/2} \quad (4.10)$$

where N is the number of points. Note, again, that 2 is subtracted from the number of points in both denominators to produce an unbiased estimate of the standard deviational ellipse since there are two constants from which the distance along each axis is measured (mean of X, mean of Y).[4]

3. The X-axis and Y-axis of the ellipse are defined by

$$Length_x = 2S_x \quad (4.11)$$

$$Length_y = 2S_y \quad (4.12)$$

4. The area of the ellipse is

$$A = \pi S_x S_y \quad (4.13)$$

Figure 4.13 shows the output of the ellipse routine and figure 4.14 maps the standard deviational ellipse of auto thefts in Baltimore City and Baltimore County for 1996.

Geometric Mean

The mean center routine (Mcsd) includes two additional means. First, there is the geometric mean, which is a mean associated with the mean of the logarithms. It is defined as:

$$\text{Geometric Mean of X} = GM(X) = \prod_{i=1}^{N} (X_i)^{1/N} \quad (4.14)$$

$$\text{Geometric Mean of Y} = GM(Y) = \prod_{i=1}^{N} (Y_i)^{1/N} \quad (4.15)$$

Figure 4.13: Standard Deviational Ellipse Output

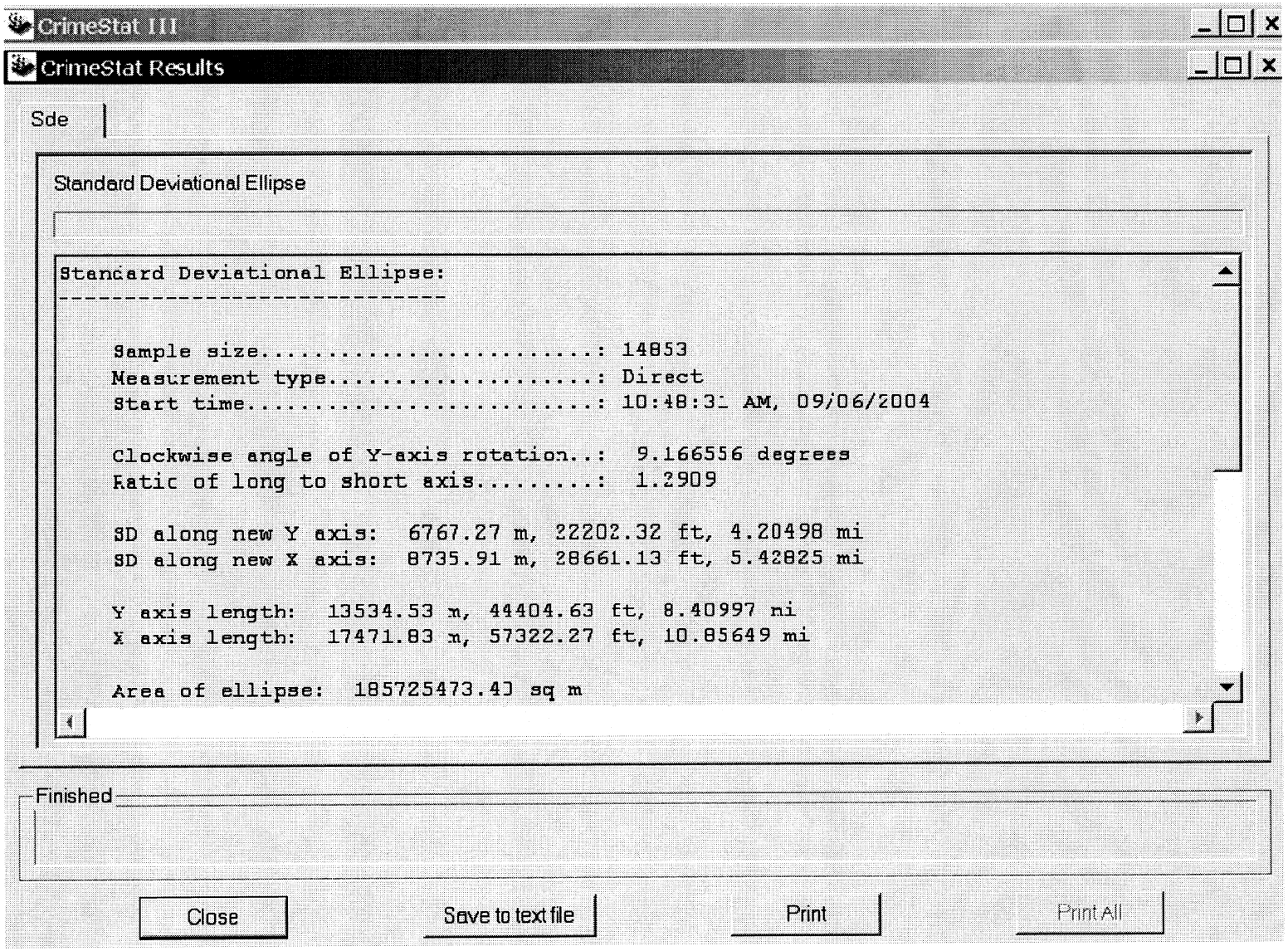
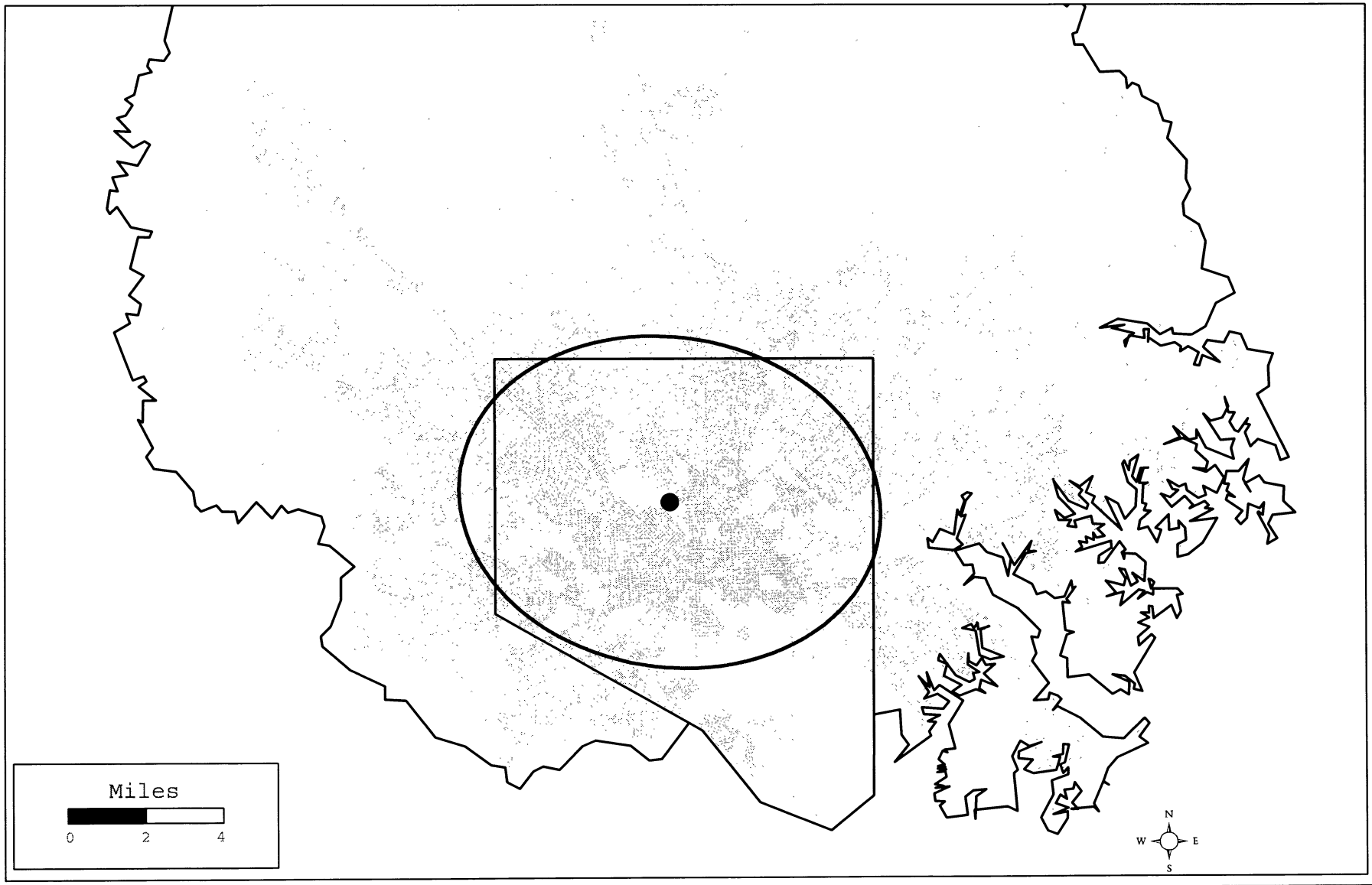


Figure 4.14: 1996 Metropolitan Baltimore Auto Thefts

Mean Center and Standard Deviational Ellipse



where Π is the product term of each point value, i (i.e., the values of X or Y are multiplied times each other), and N is the sample size (Everitt, 1995). The equation can be evaluated by logarithms.

$$\text{Ln}[GM(X)] = \frac{1}{N}[\text{Ln}(X_1) + \text{Ln}(X_2) + ... + \text{Ln}(X_N)] = \frac{1}{N}\Sigma \text{Ln}(X_i) \quad (4.16)$$

$$\text{Ln}[GM(Y)] = \frac{1}{N}[\text{Ln}(Y_1) + \text{Ln}(Y_2) + ... + \text{Ln}(Y_N)] = \frac{1}{N}\Sigma \text{Ln}(Y_i) \quad (4.17)$$

$$GM(X) = e^{\text{Ln}(GM(X)} \quad (4.18)$$

$$GM(Y) = e^{\text{Ln}(GM(Y)} \quad (4.19)$$

The geometric mean is the anti-log of the mean of the logarithms. Because it first converts all X and Y coordinates into logarithms, it has the effect of discounting extreme values. The geometric mean is output as part of the Mcsd routine and has a 'Gm' prefix before the user defined name.

Harmonic Mean

The harmonic mean is also a mean which discounts extreme values, but is calculated differently. It is defined as

$$\text{Harmonic mean of X} = HM(X) = \frac{N}{\Sigma (1/X_i)} \quad (4.20)$$

$$\text{Harmonic mean of Y} = HM(Y) \quad \frac{N}{\Sigma (1/Y_i)} \quad (4.21)$$

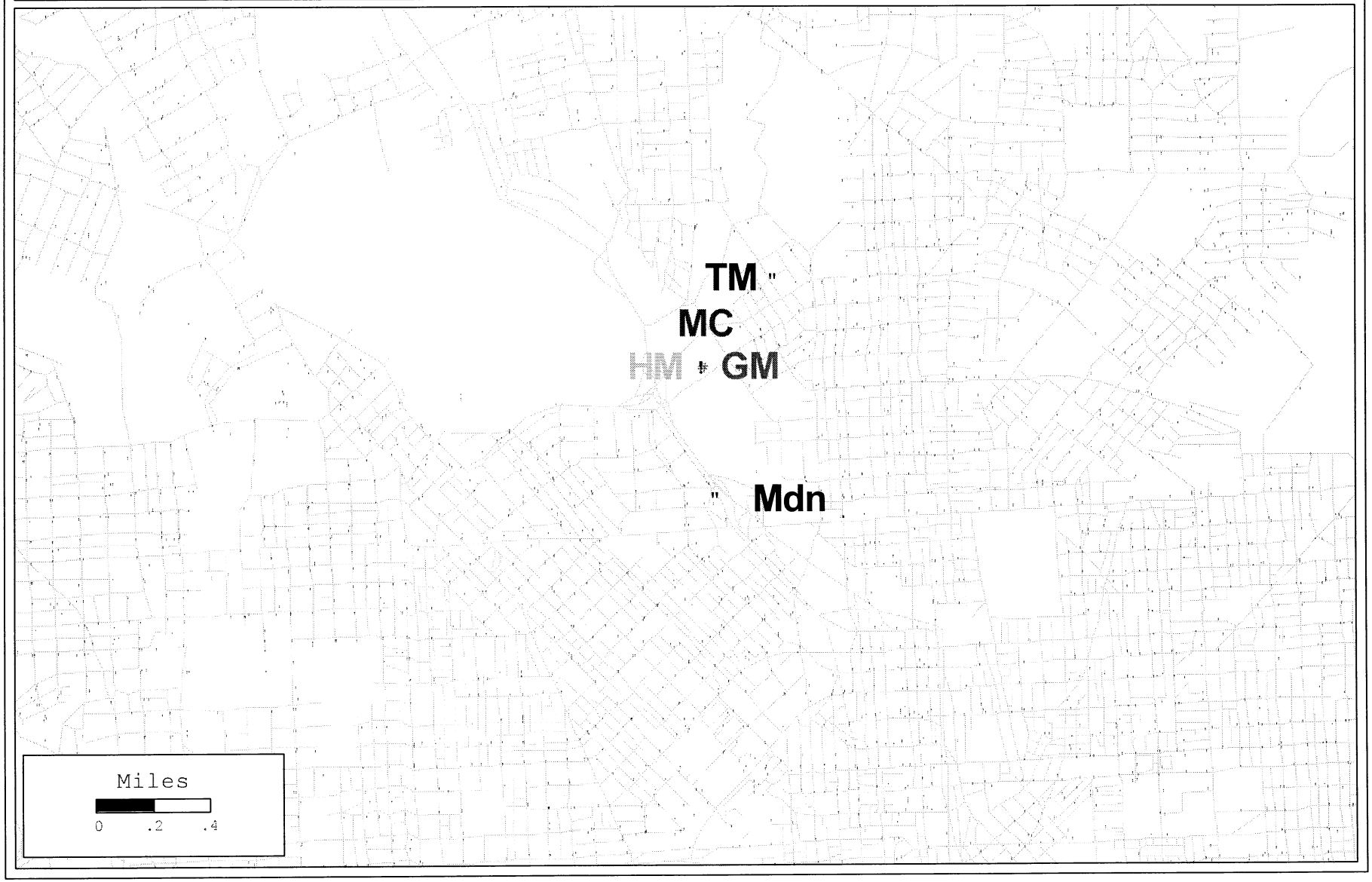
In other words, the harmonic mean of X and Y respectively is the inverse of the mean of the inverse of X and Y respectively (i.e., take the inverse; take the mean of the inverse; and invert the mean of the inverse). The harmonic mean is output as part of the Mcsd routine and has a 'Hm' prefix before the user defined name.

The geometric and harmonic means are discounted means that 'hug' the center of the distribution. They differ from the mean center when there is a very skewed distribution. To contrast the different means, figure 4.15 below shows five different means for Baltimore County motor vehicle thefts:

Figure 4.15:

Five Mean Centers for 1996 Baltimore Vehicle Thefts

Five Different Means Compared



1. Mean center;
2. Center of minimum distance;
3. Geometric mean;
4. Harmonic mean; and
5. Triangulated mean (discussed below)

In the example, the mean center, geometric mean, and harmonic mean fall almost on top of each other; however, they will not always be so. The center of minimum distance approximates the geographical center of the distribution. The triangulated mean is defined by the angularity and distance from the lower-left and upper-right corners of the data set (see below).

Centrographic descriptors can be very powerful tools for examining spatial patterns. They are a first step in any spatial analysis, but an important one. The above example illustrates how they can be a basis for decision-making, even with small samples. A couple of other examples can be illustrated.

Average Density

The average density is the number of incidents divided by the area. It is a measure of the average number of events per unit of area; it is sometimes called the *intensity*. If the area is defined on the measurement parameters page, the routine uses that value; otherwise, it takes the rectangular area defined by the minimum and maximum X and Y values (the bounding rectangle).

Output Files

Calculating the Statistics

Once the statistics have been selected, the user clicks on *Compute* to run the routine. The results are shown in a results table.

Tabular Output

For each of these statistics, *CrimeStat* produces tabular output. In *CrimeStat*, all tables are labeled by symbols, for example Mcsd for the mean center and standard distance deviation or Mcmd for the center of minimum distance. All tables present the sample size.

Graphical Objects

The six centrographic statistics can be output as graphical objects. The mean center and center of minimum distance are output as single points. The standard deviation of the X and Y coordinates is output as a rectangle. The standard distance deviation is output as a circle and the standard deviational ellipse is output as an ellipse.

CrimeStat currently supports graphical outputs to *ArcView* '.shp' files, to *MapInfo* '.mif' and to *Atlas*GIS* '.bna' files. Before running the calculation, the user should select the desired output files and specify a root name (e.g., Precinct1Burglaries). Figure 4.16 shows a dialog box for selecting for the GIS program output. For *MapInfo* output only, the user has to also indicate the name of the projection, the projection number and the datum number. These can be found in the *MapInfo* users guide. By default, *CrimeStat* will use the standard parameters for a spherical coordinate system (Earth projection, projection number 1, and datum number 33). If a user requires a different coordinate system, the appropriate values should be typed into the space. Figure 4.17 shows the selection of the *MapInfo* coordinate parameters.

If requested, the output files are saved in the specified directory under the specified (root) name. For each statistic, *CrimeStat* will add prefix letters to the root name.

MC<*root*> for the mean center
MdnCntr<root> for the median center
Mcmd<*root*> for center of minimum distance
XYD<*root*> for the standard deviation of the X and Y coordinates
SDD<*root*> for the standard distance deviation
SDE<*root*> for the standard deviational ellipse.

The '.shp' files can be read directly into *ArcView* as themes. The '.mif' and '.bna' files have to be imported into *MapInfo* and *Atlas*GIS,* respectively.[5]

Statistical Testing

While the current version of *CrimeStat* does not conduct statistical tests that compare two distributions, it is possible to conduct such tests. Appendix B presents a discussion of the statistical tests that can be used. Instead, the discussion here will focus on using the outputs of the routines without formal testing.

Decision-making Without Formal Tests

Formal significance testing has the advantage of providing a consistent inference about whether the difference in two distributions is likely or unlikely to be due to chance. Almost all formal tests compare the distribution of a statistic with that of a random distribution. However, police departments frequently have to make decisions based on small samples, in which case the formal tests are less useful than they would with larger samples. Still, the centrographic statistics calculated in *CrimeStat* can be useful and can help a police department make decision even in the absence of formal tests.

Example 1: June and July Auto Thefts in Precinct 11

We want to illustrate the use of these statistics to make decisions with two examples. The first is a comparison of crimes in small geographical areas. In most metropolitan areas, most analysts will concentrate on particular sub-areas of the

Figure 4.16: Outputting Objects to A GIS Program

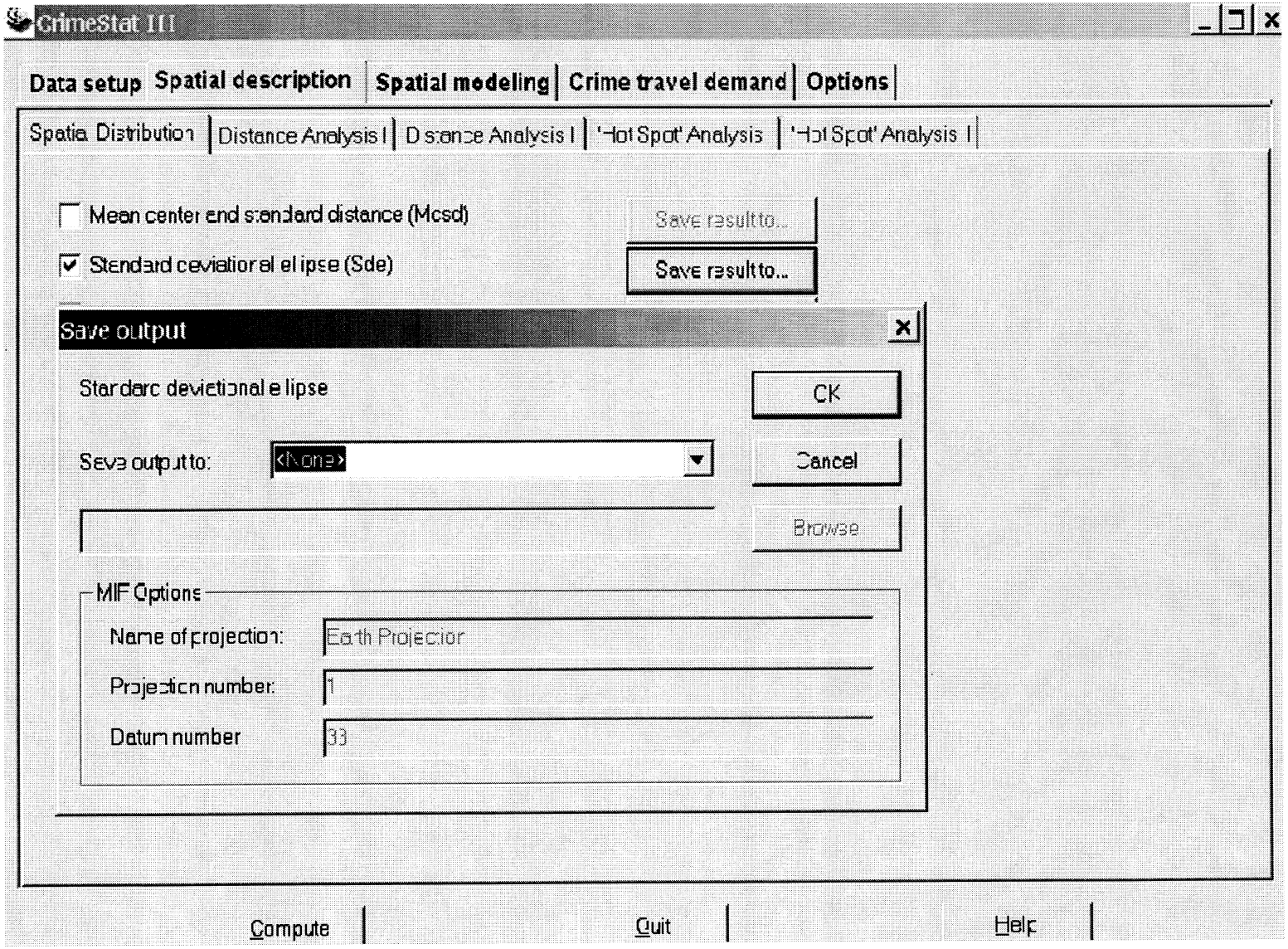
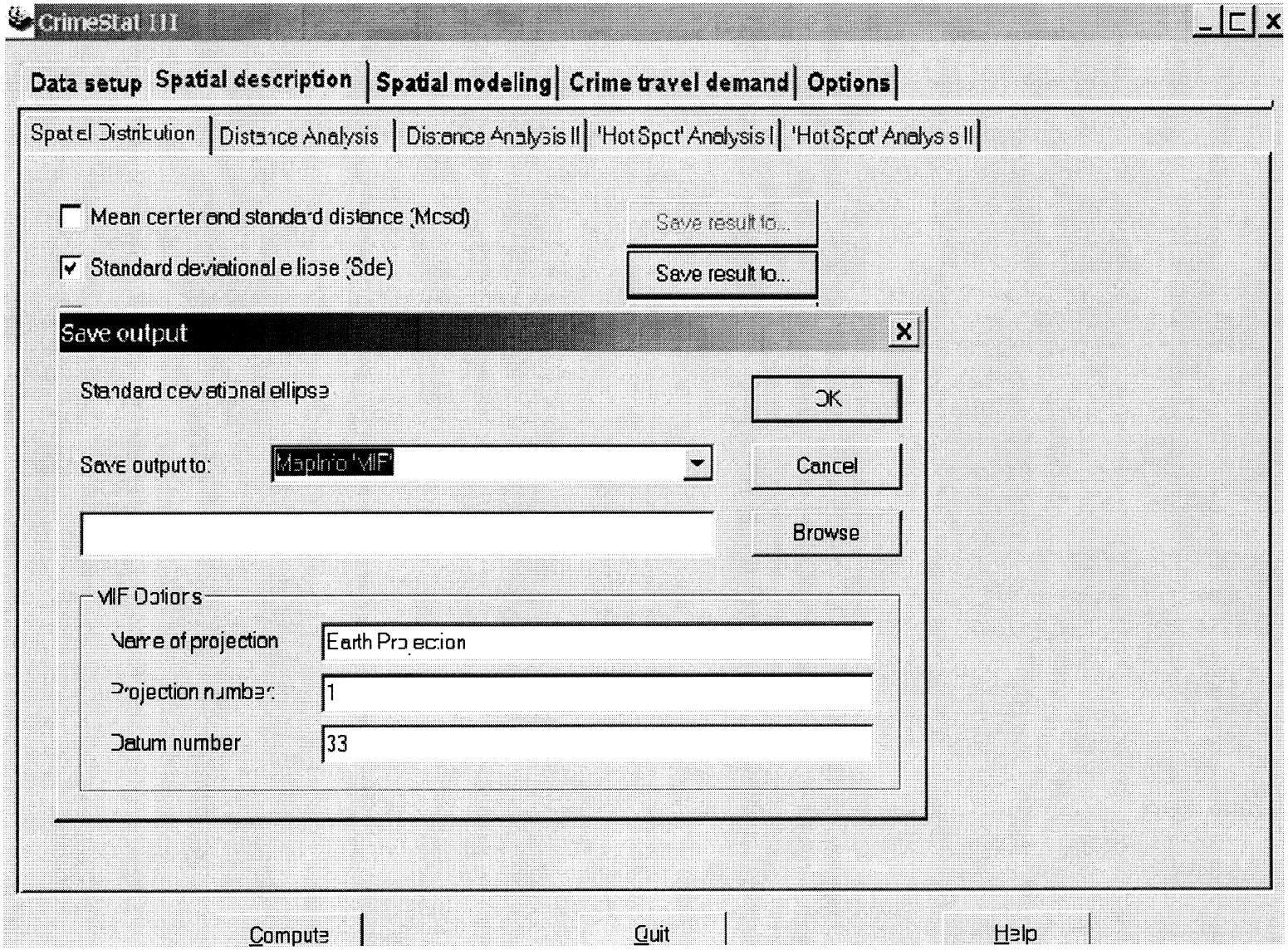


Figure 4.17: *MapInfo* Output Options



jurisdiction, rather than on the jurisdiction itself. In Baltimore County, for instance, analysis is done both for the jurisdiction as a whole as well as by individual precincts. Below in Figure 4.18 are the standard deviational ellipses for 1996 auto thefts for June and July in Precinct 11 of Baltimore County. As can be seen, there was a spatial shift that occurred between June and July of that year, the result most probably of increased vacation travel to the Chesapeake Bay. While the comparison is very simple, involving looking at the graphical object created by *CrimeStat*, such a month to month comparison can be useful for police departments because it points to a shift in incident patterns, allowing the police department to reorient their patrol units.

Example 2: Serial Bburglaries in Baltimore City and Baltimore County

The second example illustrates a rash of burglaries that occurred on both sides of the border of Baltimore City and Baltimore County. On one hand there were ten residential burglaries that occurred on the western edge of the City/County border within a short time period of each other and, on the other hand, there were 13 commercial burglaries that occurred in the central part of the metropolitan areas. Both police departments suspected that these two sets were the work of a serial burglar (or group of burglars). What they were not sure about was whether the two sets of burglaries were done by the same individuals or by different individuals.

The number of incidents involved are too small for significance testing; only one of the parameters tested was significant and that could easily be due to chance. However, the police do have to make a guess about the possible perpetrator even with limited information. Let's use *CrimeStat* to try and make a decision about the distributions.

Figure 4.19 illustrates these distributions. The thirteen commercial burglaries are shown as squares while the ten residential burglaries are shown as triangles. Figure 4.20 plots the mean centers of the two distributions. They are close to each other, but not identical. An initial hunch would suggest that the robberies are committed by two perpetrators (or groups of perpetrators), but the mean centers are not different enough to truly confirm this expectation. Similarly, figure 4.21 plots the center of minimum distance. Again, there is a difference in the distribution, but it is not great enough to truly rule out the single perpetrator theory.

Figure 4.22 plots the raw standard deviations, expressed as a rectangle by *CrimeStat*. The dispersion of incidents overlaps to a sizeable extent and the area defined by the rectangle is approximately the same. In other words, the search area of the perpetrator or perpetrators is approximately the same. This might argue for a single perpetrator, rather than two. Figure 4.23 shows the standard distance deviation of the two sets of incidents. Again, there is sizeable overlap and the search radiuses are approximately the same.

Only with the standard deviational ellipse, however, is there a fundamental difference between the two distributions (figure 4.24). The pattern of commercial robberies is falling along a northeast-southwest orientation while that for residential robberies along

Figure 4.18:

Vehicle Theft Change in Precinct 11

Standard Deviational Ellipses for June and July 1996

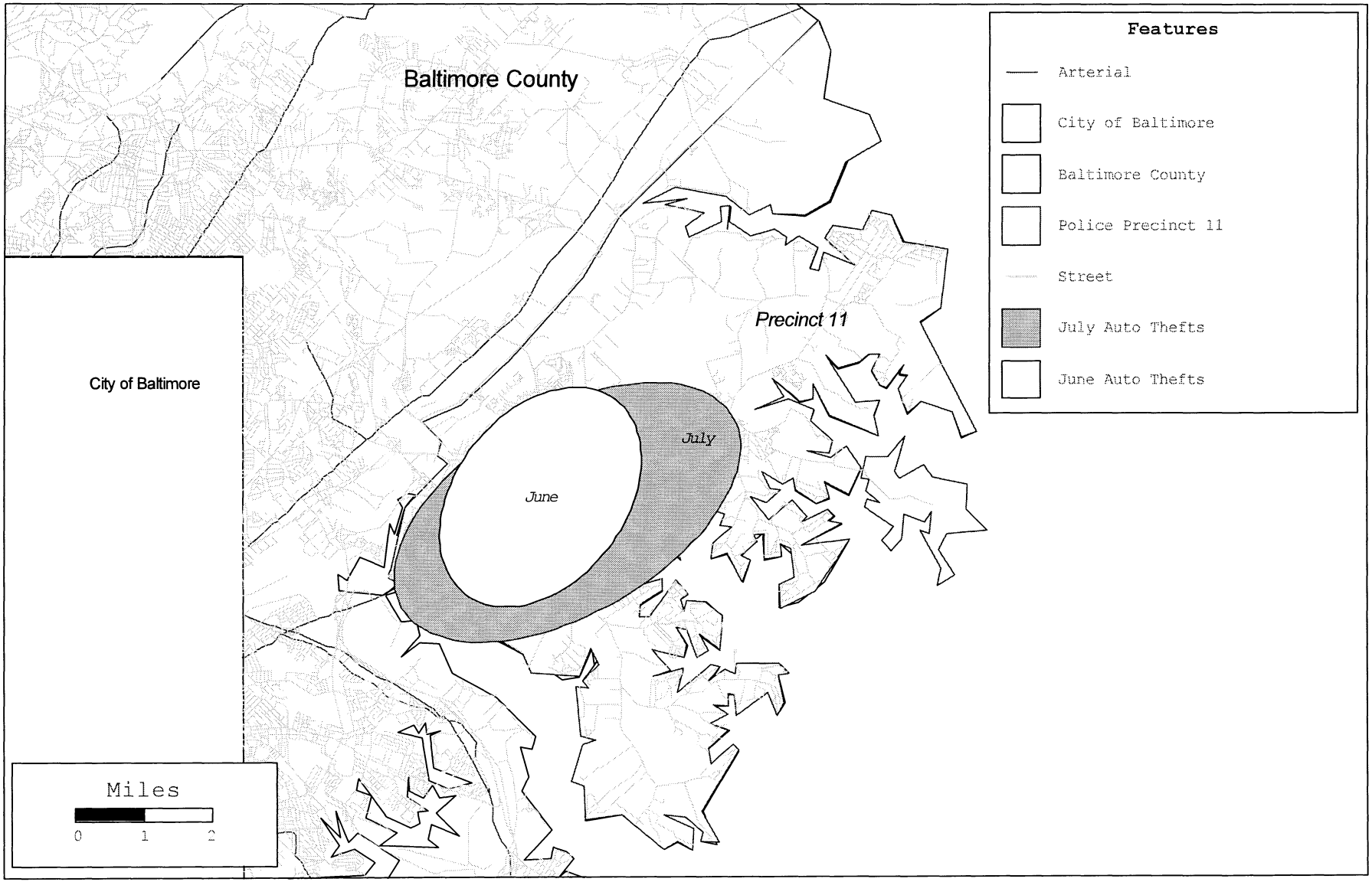


Figure 4.19:

Identifying Serial Burglars

Incident Distribution of Two Serial Offenders

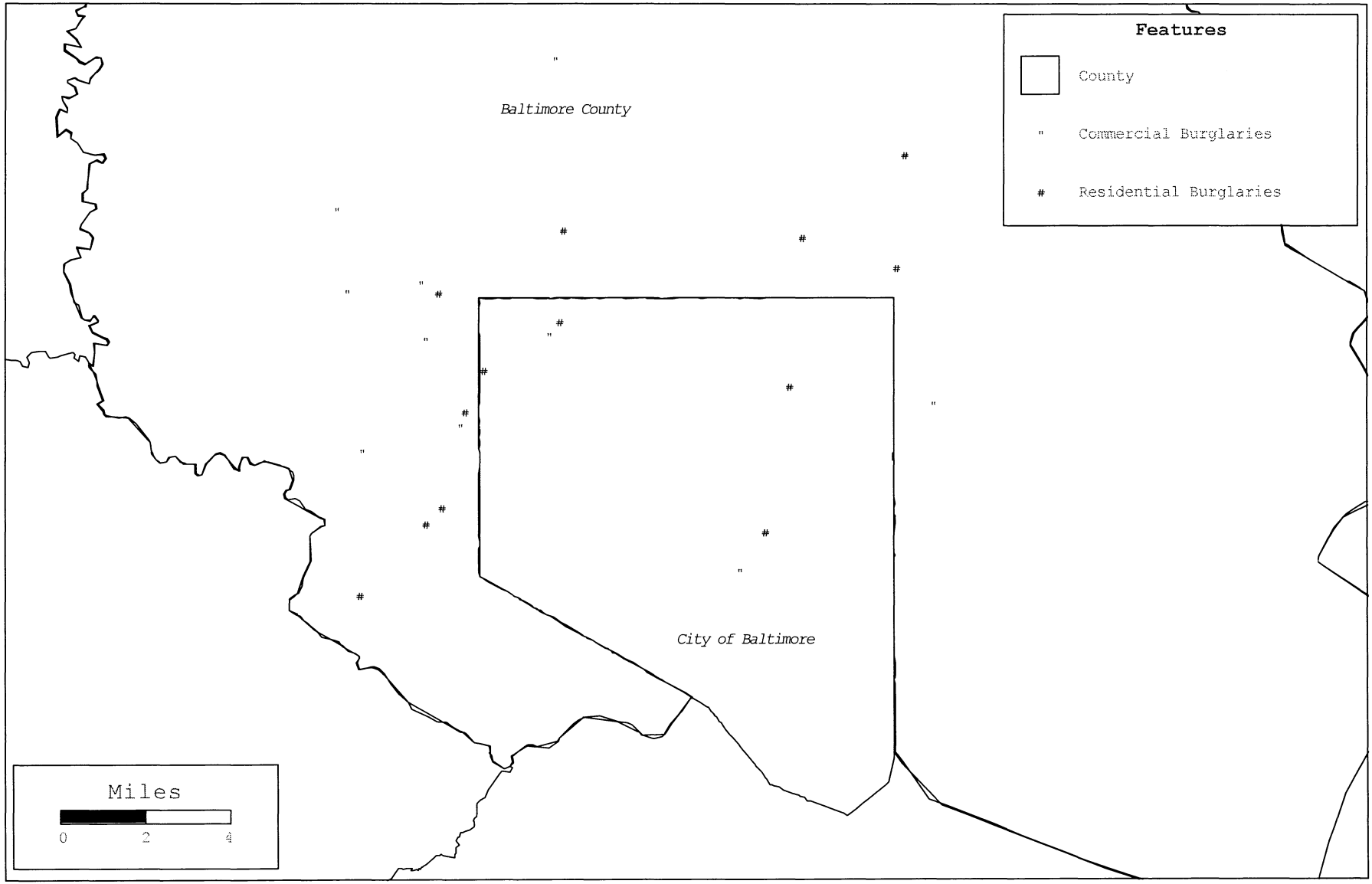


Figure 4.20:

Identifying Serial Burglars

Mean Centers of Incidents for Two Serial Offenders

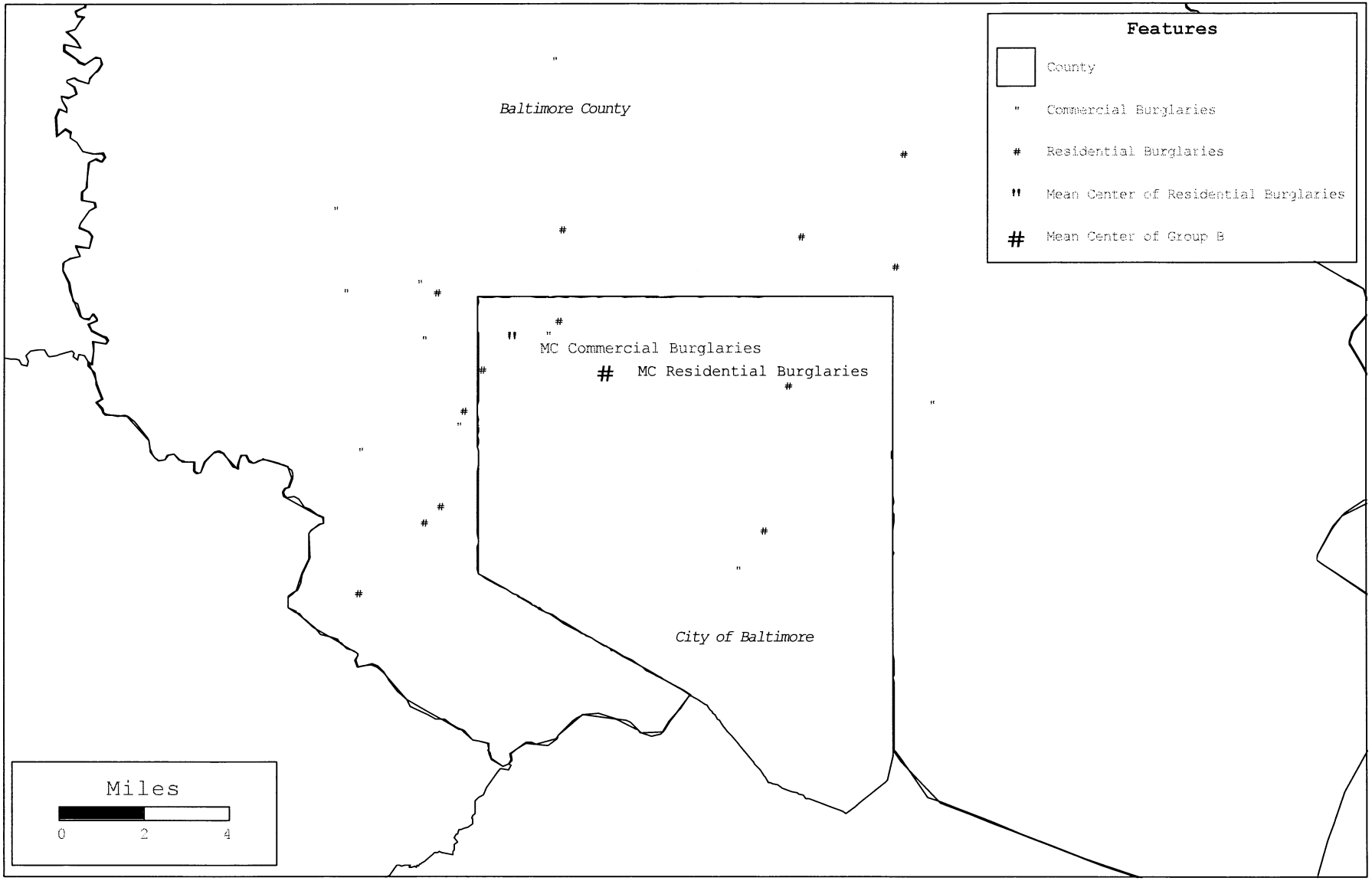


Figure 4.21:

Identifying Serial Burglars

Center of Minimum Distances for Incidents for Two Serial Offenders

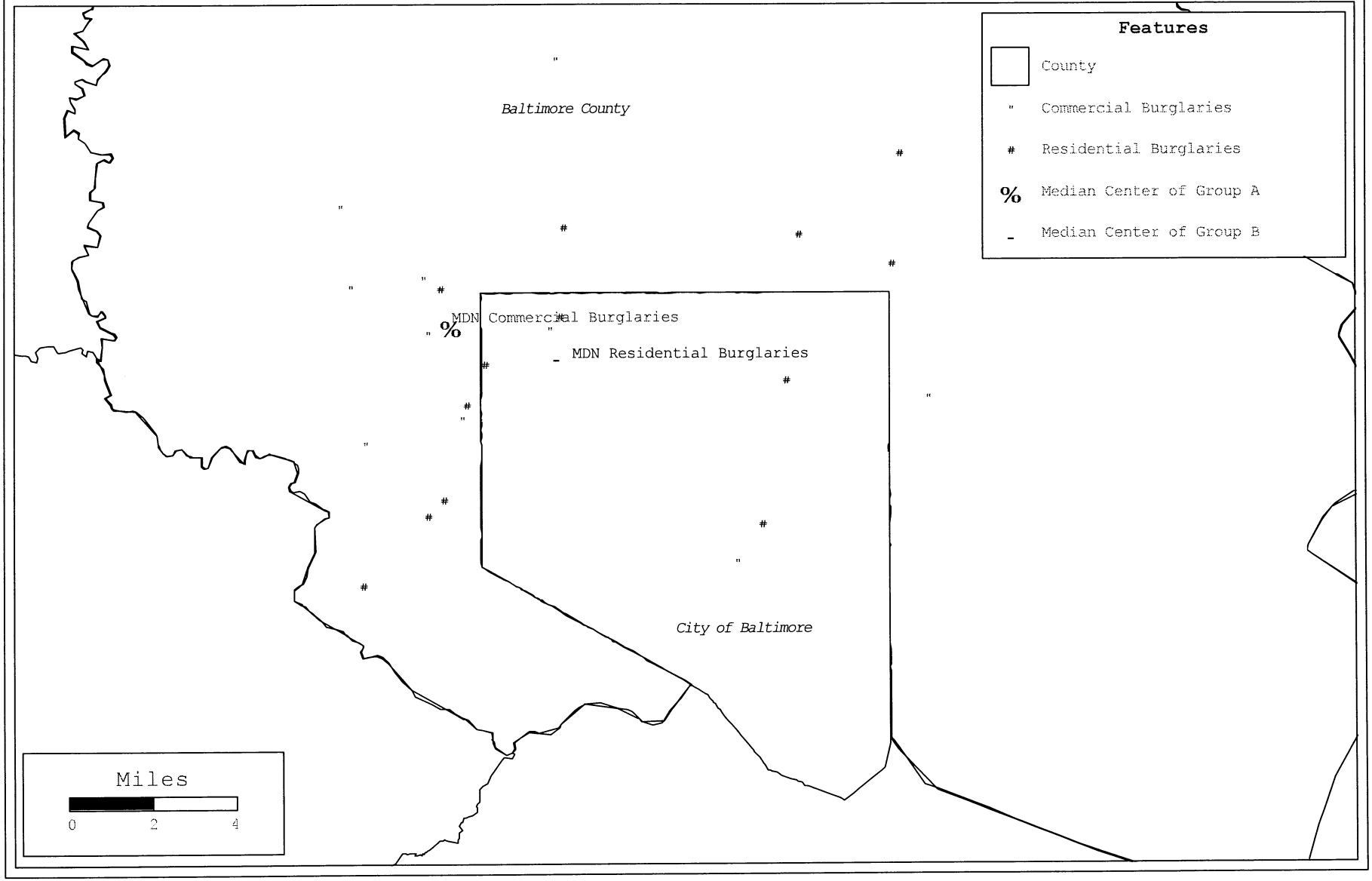


Figure 4.22:

Identifying Serial Burglars

Standard Deviations of Incidents for Two Serial Offenders

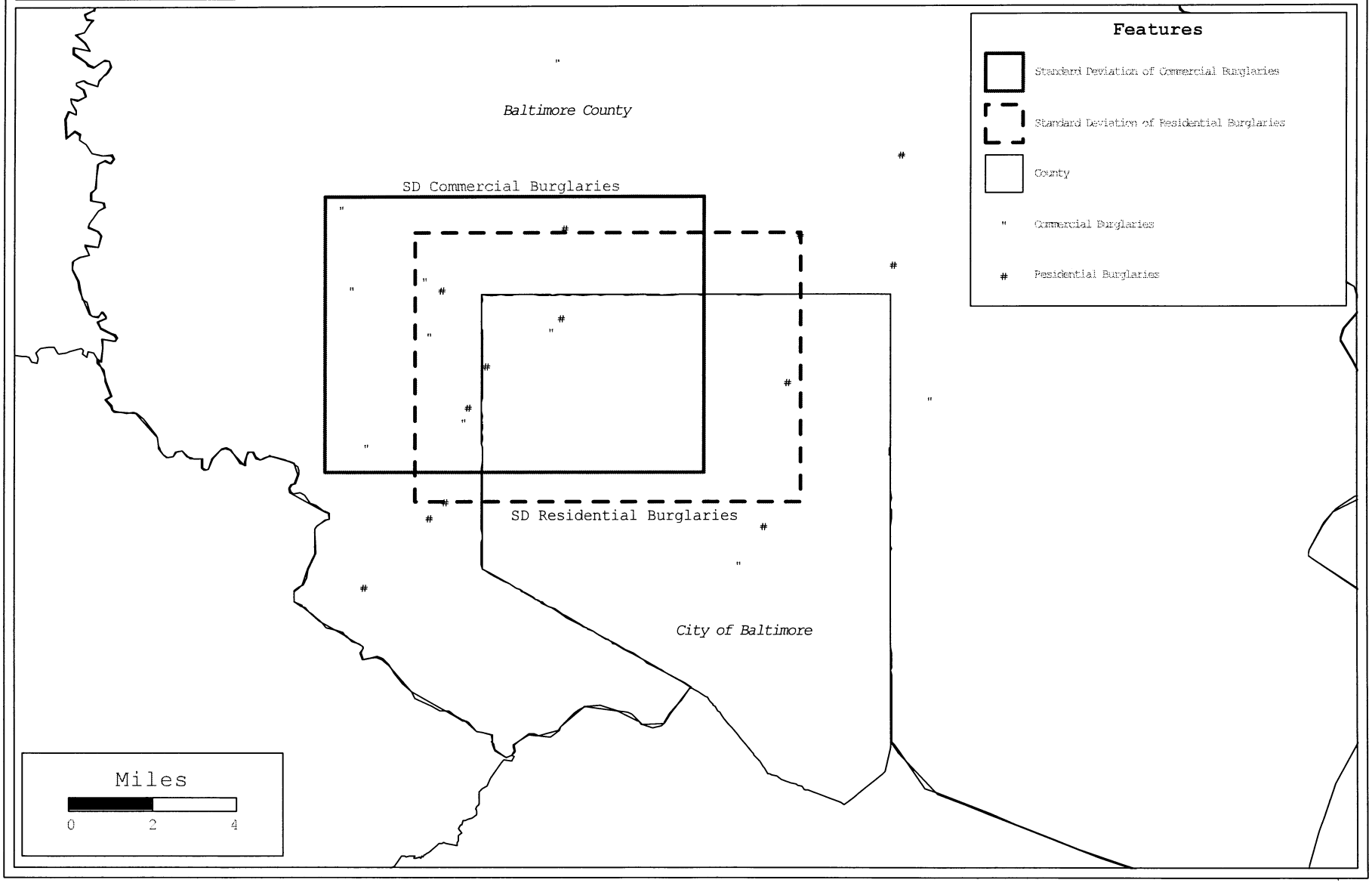


Figure 4.23:

Identifying Serial Burglars

Standard Distance Deviation of Incidents for Two Serial Offenders

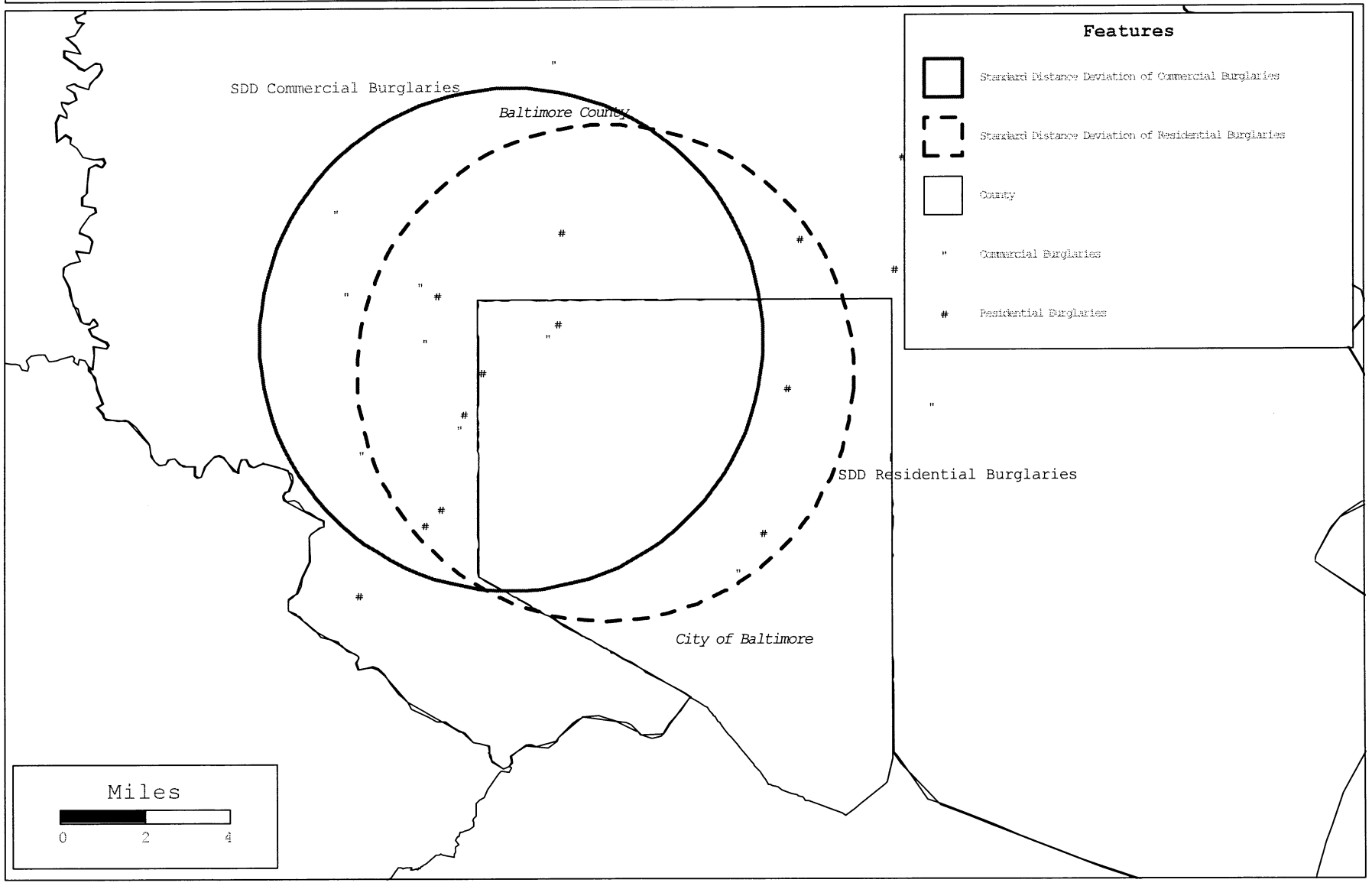
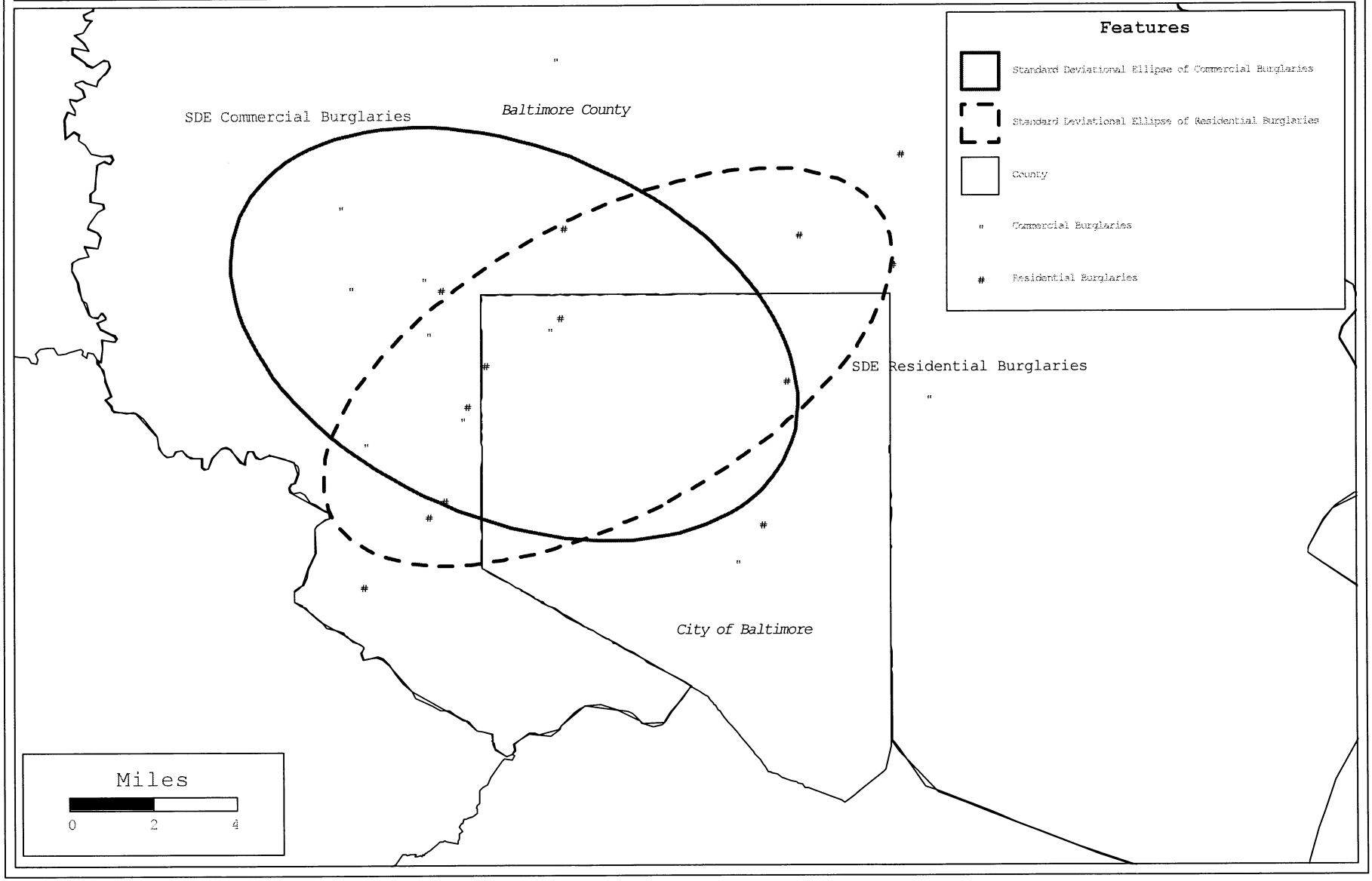


Figure 4.24:

Identifying Serial Burglars

Standard Deviational Ellipse of Incidents for Two Serial Offenders



a northwest-southeast axis. In other words, when the orientation of the incidents is examined, as defined by the standard deviational ellipse, there are two completely opposite patterns. Unless this difference can be explained by an obvious factor (e.g., the distribution of commercial establishments), it is probable that the two sets of robberies were committed by two different perpetrators (or groups of perpetrators).

Directional Mean and Variance

Centrographic statistics utilize the coordinates of a point, defined as an X and Y value on either a spherical or projected/Cartesian coordinate system. There is another type of metric that can be used for identifying incident locations, namely a *polar coordinate* system. A *vector* is a line with direction and length. In this system, there is a reference vector (usually 0^0 due North) and all locations are defined by angular deviations from this reference vector. By convention, angles are defined as deviations from 0^0, clockwise through 360^0. Note the measurement scale is a circle which returns back on itself (i.e. 0^0 is also 360^0). Point locations can be represented as vectors on a polar coordinate system.

With such a system, ordinary statistics cannot be used. For example, if there are five points which on the northern side of the polar coordinate system and are defined by their angular deviations as 0^0, 10^0, 15^0, 345^0, and 350^0 from the reference vector (moving clockwise from due North), the statistical mean will produce an erroneous estimate of 144^0. This vector would be southeast and will lie in an opposite direction from the distribution of points.

Instead, statistics have to be calculated by trigonometric functions. The input for such a system is a set of vectors, defined as angular deviations from the reference vector and a distance vector. Both the angle and the distance vector are defined with respect to an origin. The routine can calculate angles directly or can convert all X and Y coordinates into angles with a bearing from an origin. For reading angles directly, the input is a set of vectors, defined as angular deviations from the reference vector. *CrimeStat* calculates the mean direction and the circular variance of a series of points defined by their angles. On the primary file screen, the user must select Direction (angles) as the coordinate system.

If the angles are to be calculated from X/Y coordinates, the user must define an origin location. On the reference file page, the user can select among three origin points:

1. The lower-left corner of the data set (the minimum X and Y values). This is the default setting.

2. The upper-right corner of the data set (the maximum X and Y values); and

3. A user-defined point.

Users should be careful about choosing a particular location for an origin, either lower-left, upper-right or user-defined. If there is a point at that origin, *CrimeStat* will drop that case since any calculations for a point with zero distance are indeterminate.

Users should check that there is no point at the desired origin. If there is, then the origin should be adjusted slightly so that no point falls at that location (e.g., taking slightly smaller X and Y values for the lower-left corner or slightly larger X and Y values for the upper right corner).

The routine converts all X and Y points into an angular deviation from true North relative to the specified origin and a distance from the origin. The bearing is calculated with different formulae depending on the quadrant that the point falls within.

First Quadrant

With the lower-left corner as the origin, all angles are in the first quadrant. The clockwise angle, θ_i is calculated by

$$\theta_i = \text{Arctan}\left[\frac{\text{Abs}(X_i - X_o)}{\text{Abs}(Y_i - Y_o)}\right] \qquad (4.22)$$

where X_i is the X-value of the point, Y_i is the Y-value of the point, X_o is the X-value of the origin, and Y_o is the Y-value of the origin.

The angle, θ_i, is in radians and can be converted to polar coordinate degrees using:

$$\theta_i \text{ (degrees)} = \theta_i \text{ (radians)} * 180/\pi \qquad (4.23)$$

Third Quadrant

With the upper-right corner as the origin, all angles are in the third quadrant. The clockwise angle, θ_i, is calculated by

$$\theta_i = \pi + \text{Arctan}\left[\frac{\text{Abs}(X_i - X_o)}{\text{Abs}(Y_i - Y_o)}\right] \qquad (4.24)$$

where the angle, θ_i, is again in radians. Since there are 2π radians in a circle, π radians is 180^0. Again, the angle in radians can be converted into degrees with formula 4.23 above.

Second and Fourth Quadrants

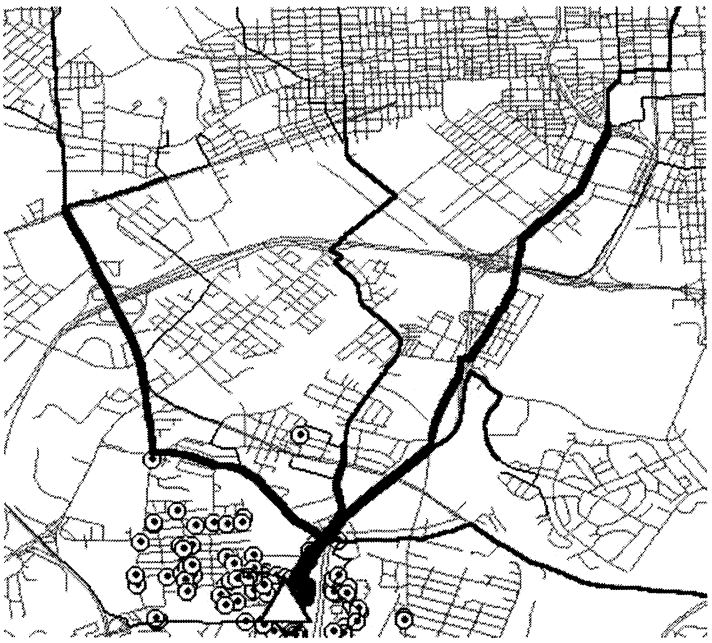
When the origin is user-defined, each point must be evaluated as to which quadrant it is in. The second and fourth quadrants define the clockwise angle, θ_i, differently

Using Spatial Measures of Central Tendency with Network Analyst to Identify Routes Used by Motor Vehicle Thieves

Philip R. Canter
Baltimore County Police Department
Towson, Maryland

Motor vehicle thefts have been steadily declining countywide over the last 5 years, but one police precinct in southwest Baltimore County was experiencing significant increases over several months. Cases were concentrated in several communities, but directed deployment and saturated patrols had minimal impact. In addition to increasing patrols in target communities, the precinct commander was interested in deploying police on roads possibly used by motor vehicle thieves. Police analysts had addresses for theft and recovery locations; it was a matter of using the existing highway network to connect the two locations.

To avoid analyzing dozens of paired locations, analysts decided to set up a database using one location representing the origin of motor vehicle thefts for a particular community. The origin was computed using *CrimeStat*'s median center for motor vehicle theft locations reported for a particular community. The median center is the position of minimum average travel and is less affected by extreme locations compared to the arithmetic mean center. The database consisted of the median center paired with a recovery location. Using Network Analyst, a least-effort route was computed for cases reported by community. A count was assigned to each link along a roadway identified by Network Analyst. Analysts used the count to thematically weight links in ArcView. The precinct commander deployed resources along these routes with orders to stop suspicious vehicles. This operation resulted in 27 arrests, and a reduction in motor vehicle thefts.

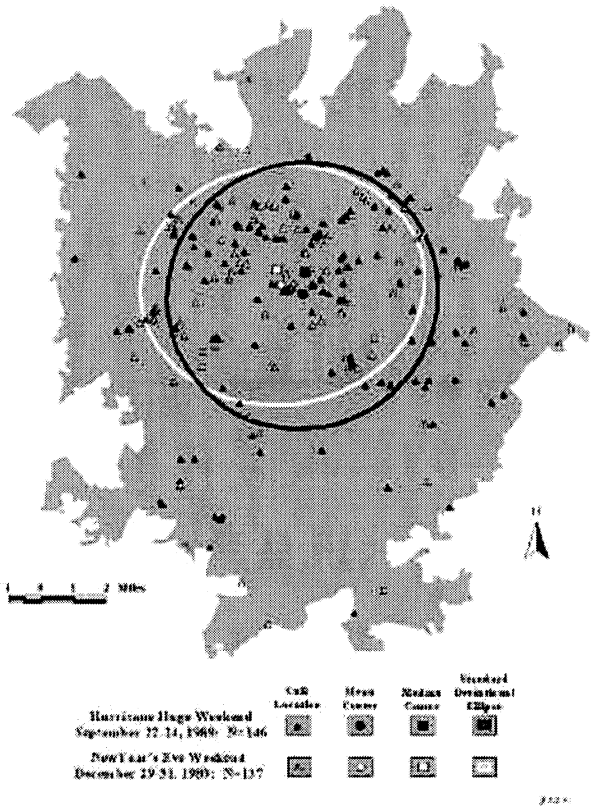


Distance Analysis
Man With A Gun Calls For Service
Charlotte, N.C., 1989

James L. LeBeau
Administration of Justice
Southern Illinois University – Carbondale

Hurricane Hugo arrived on Friday, September 22, 1989 in Charlotte, North Carolina. That weekend experienced the highest counts of *Man With A Gun* calls for service for the year. The locations of the calls during the Hugo Weekend are compared with the following New Year's Eve weekend.

CrimeStat was used to compare the two weekends. Compared to the New Year's Eve weekend: 1) Hugo's mean and median centers are more easterly; 2) Hugo's ellipse is larger and more circular; and 3) Hugo's ellipse shifts more to the east and southeast. The abrupt spatial change of *Man With A Gun* calls during a natural disaster might indicate more instances of defensive gun use for protection of property.



$$\theta_i = 0.5\pi + \text{Arctan}\left[\frac{\text{Abs}(Y_i - Y_o)}{\text{Abs}(X_i - X_o)}\right] \quad (4.25)$$

$$\theta_i = 1.5\pi + \text{Arctan}\left[\frac{\text{Abs}(Y_i - Y_o)}{\text{Abs}(X_i - X_o)}\right] \quad (4.26)$$

Once all X/Y coordinates are converted into angles, the mean angle is calculated.

Mean Angle

With either angular input or conversion from X/Y coordinates, the *Mean Angle* is the resultant of all individual vectors (i.e., points defined by their angles from the reference vector). It is an angle that summarizes the mean direction. Graphically, a *resultant* is the sum of all vectors and can be shown by laying each vector end to end. Statistically, it is defined as

$$\text{Mean angle} = \bar{\theta} = \text{Abs}\left\{\text{Arctan}\left[\frac{\Sigma\, d_i \sin\theta_i}{\Sigma\, d_i \cos\theta_i}\right]\right\} \quad (4.27)$$

where the summation of sines and cosines is over the total number of points, i, defined by their angles, θ_i. Each angle, θ_i, can be weighted by the length of the vector, d_i. In an unweighted angle, d_i is assumed to be of equal length, 1. The absolute value of the ratio of the sum of the weighted sines to the sum of the weighted cosines is taken. All angles are in radians. In determining the mean angle, the quadrant of the resultant must be identified:

1. If $\Sigma \sin\theta_i > 0$ and $\Sigma \cos\theta_i > 0$, then $\bar{\theta}$ can be used directly as the mean angle
2. If $\Sigma \sin\theta_i > 0$ and $\Sigma \cos\theta_i < 0$, then the mean angle is $\pi/2 + \bar{\theta}$.
3. If $\Sigma \sin\theta_i < 0$ and $\Sigma \cos\theta_i < 0$, then the mean angle is $\pi + \bar{\theta}$.
4. If $\Sigma \sin\theta_i < 0$ and $\Sigma \cos\theta_i > 0$, then the mean angle is $1.5\pi + \bar{\theta}$.

Formulas 4.22, 4.24, 4.25 and 4.26 above are then used to convert the directional mean back to an X/Y coordinate, depending on which coordinate it falls within.

Circular Variance

The dispersion (or variance) of the angles are also defined by trigonometric functions. The unstandardized variance, R, is sometimes called the *sample resultant length* since it is the resultant of all vectors (angles).

$$R = \text{SQRT}\left[\left(\Sigma\, d_i \sin \theta_i\right)^2 + \left(\Sigma\, d_i \cos \theta_i\right)^2 \right] \tag{4.28}$$

where d_i is the length of vector, i, with an angle (bearing) for the vector of θ_i. For the unweighted sample resultant, d_i is 1.

Because R increases with sample size, it is standardized by dividing by N to produce a *mean resultant length*.

$$\bar{R} = \frac{R}{N} \tag{4.29}$$

where N is the number points (sample size).

Finally, the average distance from the origin, D, is calculated and the *circular variance* is calculated by

$$\text{Circular variance} = \frac{1}{D}\left\{D - \frac{R}{N}\right\} = (D - \bar{R})/D = 1 - \frac{\bar{R}}{D} \tag{4.30}$$

This is the standardized variance which varies from 0 (no variability) to 1 (maximum variability). The details of the derivations can be found in Burt and Barber (1996) and Gaile and Barber (1980).

Mean Distance

The mean distance, $\bar{d}$, is calculated directly from the X and Y coordinates. It is identified in relation to the defined origin.

Directional Mean

The directional mean is calculated as the intersection of the mean angle and the mean distance. It is not a unique position since distance and angularity are independent dimensions. Thus, the directional mean calculated using the minimum X and minimum Y location as the reference origin (the 'lower left corner') will yield a different location from the directional mean calculated using the maximum X and maximum Y location as the origin (the 'upper right corner'). There is a weighted and unweighted directional mean.

Though *CrimeStat* calculates the location, users should be aware of the non-uniqueness of the location. The unweighted directional mean can be output with a 'Dm' prefix. The weighted directional mean is not output.

Triangulated Mean

The triangulated mean is defined as the intersection of the two vectors, one from the lower-left corner of the study area (the minimum X and Y values) and the other from the upper-right corner of the study area (the maximum X and Y values). It is calculated by estimating mean angles from each origin (lower left and upper right corners), translating these into equations, and finding the point at which these equations intersect (by setting the two functions equal to each other).

Directional Mean Output

The directional mean routine outputs nine statistics:

1. The sample size;
2. The unweighted mean angle;
3. The weighted mean angle;
4. The unweighted circular variance;
5. The weighted circular variance;
6. The mean distance;
7. The intersection of the mean angle and the mean distance;
8. The X and Y coordinates for the triangulated mean; and
9. The X and Y coordinates for the weighted triangulated mean.

The directional mean and triangulated mean can be saved as an *ArcView* 'shp', *MapInfo* 'mif', or *Atlas*GIS* 'bna' file. The unweighted directional mean - the intersection of the mean angle and the mean distance is output with the prefix 'Dm' while the unweighted triangulated mean location is output with a 'Tm' prefix. The weighted triangulated mean is output with a 'TmWt' prefix. The directional mean can be saved as an *ArcView* 'shp', *MapInfo* 'mif', or *Atlas*GIS* 'bna' file. The letters 'Dm' are prefixed to the user defined file name. See the example below.

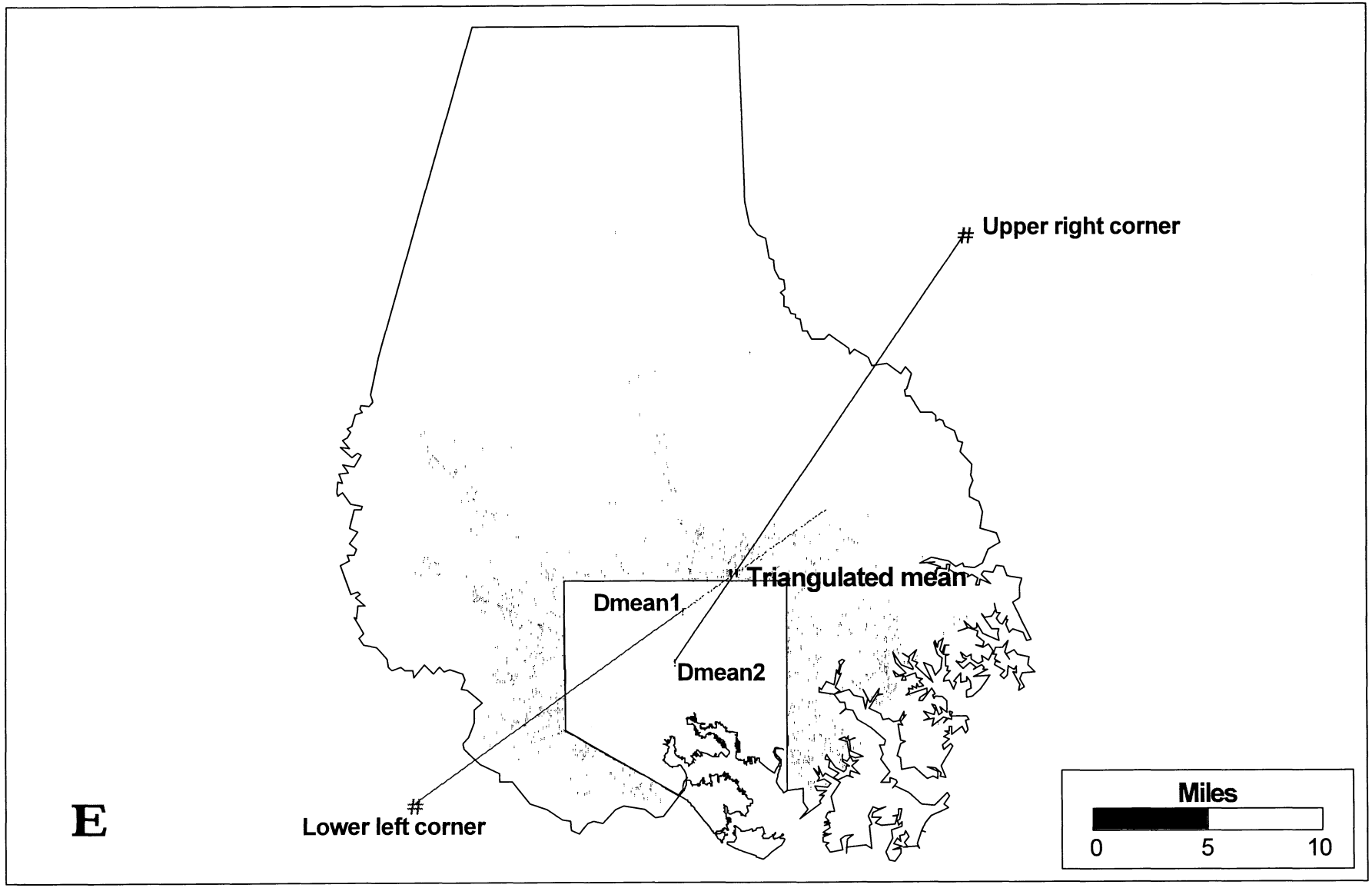
Figure 4.25 shows the unweighted triangular mean for 1996 Baltimore County robberies and compares it to the two directional means calculated using the lower-left corner (Dmean1) and the upper-right corner (Dmean2) respectively as origins. As can be seen, the two directional means fall at different locations. Lines have been drawn from each origin point to their respective directional means and are extended until they intersect. As seen, the triangulated mean falls at the location where the two vectors (i.e., mean angles) intersect.

Because the triangulated mean is calculated with vector geometry, it will not necessarily capture the central tendency of a distribution. Asymmetrical distributions can cause it to be placed in peripheral locations. On the other hand, if the distribution is

Figure 4.25:

Triangulated Mean for Baltimore County Robberies

Defined by the Intersection of Two Mean Angles



relatively balanced in each direction, it can capture the center of orientation perhaps better than other means, as figure 4.25 shows.

Appendix B includes a discussion of how to formally tests the mean direction between two different distributions.

Convex Hull

The convex hull is a boundary drawn around the distribution of points. It is a relatively simple concept, at least on the surface. Intuitively, it represents a polygon that circumscribes all the points in the distribution such that no point lies outside of the polygon.

The complexity comes because there are different ways to define a convex hull. The most basic algorithm is the *Graham scan* (Graham, 1972). Starting with one point known to be on the convex hull, typically the point with the lowest X coordinate, the algorithm sorts the remaining points in angular order around this in a counterclockwise manner. If the angle formed by the next point and the last edge is less than 180 degrees, then that point is added to the hull. If the angle is greater than 180 degrees, then the chain of nodes starting from the last edge must be deleted. The routine proceeds until the hull closes back on itself (de Berg, van Kreveld, Overmans, and Schwarzkopf, 2000).

Many alternative algorithms have been proposed. Among these are the 'gift wrap' (Chand and Kapur, 1970; Skiena, 1997), the Quick Hull, the "Divide and conquer" (Preparata and Hong, 1977), and the incremental (Kallay, 1984) algorithms. Even more complexity has been introduced by the mathematics of fractals where an almost infinite number of borders could be defined (Lam and De Cola, 1993). In most implementations, though, a simplified algorithm is used to produce the convex hull.

CrimeStat implements a 'gift wrap' algorithm. Starting with the point with the lowest Y coordinate, A, it searches for another point, B, such that all other points lie to the left of the line AB. It then finds another point, C, such that all remaining points lie to the left of the line BC. It continues in this way until it reaches the original point A again. It is like 'wrapping a gift' around the outside of the points.

The routine outputs three statistics:

1. The sample size;
2. The number of points in the convex hull
3. The X and Y coordinates for each of the points in the convex hull

The convex hull can be saved as an ArcView 'shp', MapInfo 'mif', or Atlas*GIS 'bna' file with a 'Chull' prefix.

Figure 4.26 shows the convex hull of Baltimore County robberies for 1996. As seen, the hull occupies a relatively smaller part of Baltimore County. Figure 4.27, on the other

Figure 4.26:
Convex Hull of Baltimore County Robberies: 1996

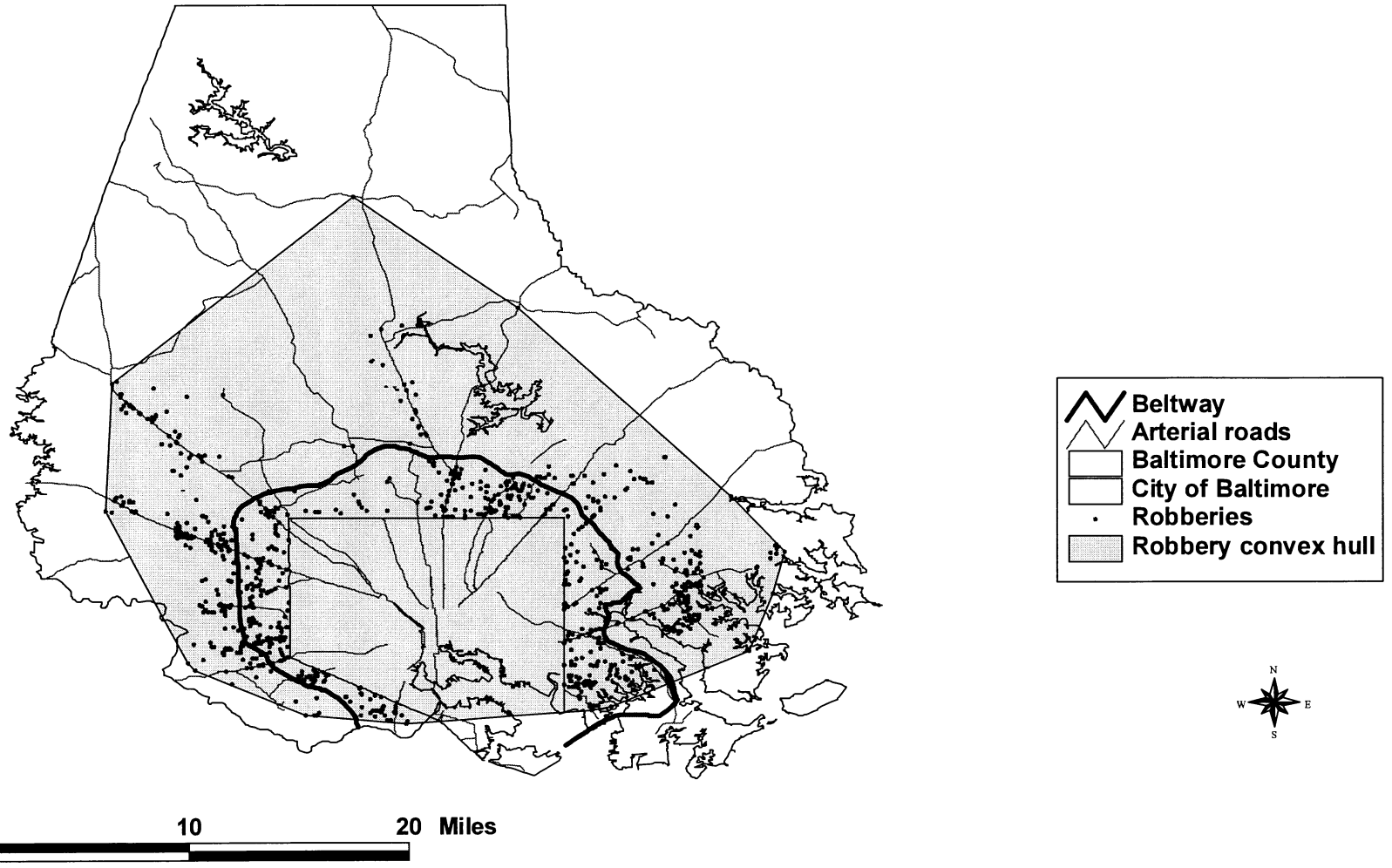


Figure 4.27:
Convex Hull of Baltimore County Burglaries: 1996

hand, shows the convex hull of 1996 Baltimore County burglaries. As seen, the convex hull of the burglaries cover a much larger area than for the robberies.

Uses and Limitations of a Convex Hull

A convex hull can be useful for displaying the geographical extent of a distribution. Simple comparisons, such as in figures 4.26 and 4.27, can show whether one distribution has a greater extent than another. Further, as we shall see, a convex hull can be useful for describing the geographical spread of a crime hot spot, essentially indicating where the crimes are distributed.

On the other hand, a convex hull is vulnerable to extreme values. If one incident is isolated, the hull will of necessity be large. The mean center, too, is influenced by extreme values but not to the same extent since it averages the location of all points. The convex hull, on the other hand, is defined by the most extreme points. A comparison of different crime types or the same crime type for different years using the convex hull may only show the variability of the extreme values, rather than any central property of the distribution. Therefore, caution must be used in interpreting the meaning of a hull.

Spatial Autocorrelation

The concept of *spatial autocorrelation* is one of the most important in spatial statistics. Spatial *independence* is an arrangement of incident locations such that there are no spatial relationships between any of the incidents. The intuitive concept is that the location of an incident (e.g., a street robbery, a burglary) is unrelated to the location of any other incident. The opposite condition - spatial autocorrelation, is an arrangement of incident locations where the location of points are related to each other, that is they are not statistically independent of one another. In other words, spatial autocorrelation is a spatial arrangement where spatial independence has been violated.

When events or people or facilities are clustered together, we refer to this arrangement as *positive* spatial autocorrelation. Conversely, an arrangement where people, events or facilities are dispersed is referred to as *negative* spatial autocorrelation; it is a rarer arrangement, but does exist (Levine, 1999).

Many, if not most, social phenomena are spatially autocorrelated. In any large metropolitan area, most social characteristics and indicators, such as the number of persons, income levels, ethnicity, education, employment, and the location of facilities are not spatially independent, but tend to be concentrated.

There are practical consequences. Police and crime analysts know from experience that incidents frequently cluster together in what are called 'hot spots'. This non-random arrangement allows police to target certain areas or zones where there are high concentrations as well as prioritize areas by the intensity of incidents. Many of the incidents are committed by the same individuals. For example, if a particular neighborhood had a concentration of street robberies over a time period (e.g., a year), many

of these robberies will have been committed by the same perpetrators. Statistical dependence between events often has common causes.

Statistically, however, non-spatial independence suggests that many statistical tools and inferences are inappropriate. For example, the use of correlation coefficients or Ordinary Least Squares regression (OLS) to predict a consequence (e.g., the correlates or predictors of burglaries) assumes that the observations have been selected randomly. If the observations, however, are spatially clustered in some way, the estimates obtained from the correlation coefficient or OLS estimator will be biased and overly precise. They will be biased because the areas with higher concentration of events will have a greater impact on the model estimate and they will overestimate precision because, since events tend to be concentrated, there are actually fewer number of independent observations than are being assumed. This concept of spatial autocorrelation underlies almost all the spatial statistics tools that are included in *CrimeStat*.

Indices of Spatial Autocorrelation

There are a number of formal statistics which attempt to measure spatial autocorrelation. This include simple indices, such as the Moran's I" or Geary's C statistic; derivatives indices, such as Ripley's K statistic (Ripley, 1976) or the application of Moran's I to individual zones (Anselin, 1995); and multivariate indices, such as the use of a spatial autocorrelation parameter in a bivariate regression model (Cliff and Ord, 1973; Griffith, 1987) or the use of a spatially-lagged dependent variable in a multiple variable regression model (Anselin, 1992). The simple indices attempt to identify whether spatial autocorrelation exists for a single variable, while the more complicated indices attempt to estimate the effect of spatial autocorrelation on other variables.

CrimeStat includes two global indices - Moran's I statistic and Geary's C statistic, and an application of Moran's I to different distance intervals. Moran and Geary are *global* in that they represent a summary value for all the data points. They are also very similar indices and are often used in conjunction. The Moran statistic is slightly more robust than the Geary, but the Geary is often used as well.

Moran's I Statistic

Moran's I statistic (Moran, 1950) is one of the oldest indicators of spatial autocorrelation. It is applied to zones or points which have continuous variables associated with them (intensities). For any continuous variable, X_i, a mean can be calculated and the deviation of any one observation from that mean can also be calculated. The statistic then compares the value of the variable at any one location with the value at all other locations (Ebdon, 1985; Griffith, 1987; Anselin, 1992). Formally, it is defined as

$$I = \frac{N \sum_i \sum_j W_{ij} (X_i - \bar{X})(X_j - \bar{X})}{(\sum_i \sum_j W_{ij}) \sum_i (X_i - \bar{X})^2} \quad (4.31)$$

where N is the number of cases, X_i is the variable value at a particular location, i, X_j is the variable value at another location (where $i \neq j$), $\bar{X}$ is the mean of the variable and W_{ij} is a weight applied to the comparison between location i and location j.

In Moran's initial formulation, the weight variable, W_{ij}, was a contiguity matrix. If zone j is adjacent to zone i, the interaction receives a weight of 1. Otherwise, the interaction receives a weight of 0. Cliff and Ord (1973) generalized these definitions to include any type of weight. In more current use, W_{ij} is a distance-based weight which is the inverse distance between locations i and j ($1/d_{ij}$). *CrimeStat* uses this interpretation. Essentially, it is a *weighted* Moran's I where the weight is an inverse distance.

The weighted Moran's I is similar to a correlation coefficient in that it compares the sum of the cross-products of values at different locations, two at a time weighted by the inverse of the distance between the locations, with the variance of the variable. Like the correlation coefficient, it typically varies between -1.0 and + 1.0. However, this is not absolute as an example later in the chapter will show. When nearby points have similar values, the cross-product is high. Conversely, when nearby points have dissimilar values, the cross-product is low. Consequently, an "I" value that is high indicates more spatial autocorrelation than an "I" that is low.

However, unlike the correlation coefficient, the theoretical value of the index does not equal 0 for lack of spatial dependence, but instead a number which is negative but very close to 0.

$$E(I) = -\frac{1}{N-1} \quad (4.32)$$

Values of "I" above the theoretical mean, E(I), indicate positive spatial autocorrelation while values of "I" below the theoretical mean indicate negative spatial autocorrelation.

Adjustment for Small Distances

CrimeStat calculates the weighted Moran's I formula using equation 4.31. However, there is one problem with this formula that can lead to unreliable results. The distance weights between two locations, W_{ij}, is defined as the reciprocal of the distance between the two points:

$$W_{ij} = \frac{1}{d_{ij}} \quad (4.33)$$

Unfortunately, as d_{ij} becomes small, then W_{ij} becomes very large, approaching infinity as the distance between the points approaches 0. If the two zones were next to each other, which would be true for two adjacent blocks for example, then the pair of observations would have a very high weight, sufficient to distort the "I" value for the entire sample. Further, there is a scale problem that alters the value of the weight. If the zones are police precincts, for example, then the minimum distance between precincts will be a lot larger than the minimum distance between a smaller type of geographical unit, such as blocks. We need to take into account these different scales.

CrimeStat includes an adjustment for small distances so that the maximum weight can never be greater than 1.0. The adjustment scales distances to one mile, which is a typical distance unit in the measurement of crime incidents. When the small distance adjustment is turned on, the minimal distance is automatically scaled to be one mile. The formula used is

$$W_{ij} = \frac{\text{one mile}}{\text{one mile} + d_{ij}} \quad (4.34)$$

in the units are specified. For example, if the distance units, d_{ij}, are calculated as feet, then

$$W_{ij} = \frac{5{,}280}{5{,}280 + d_{ij}}$$

where 5,280 is the number of feet in a mile. This has the effect of insuring that the weight of a particular pair of point locations will not have an undue influence on the overall statistic. The traditional measure of "I" is the default condition in *CrimeStat* (figure 4.28), but the user can turn on the small distance adjustment.

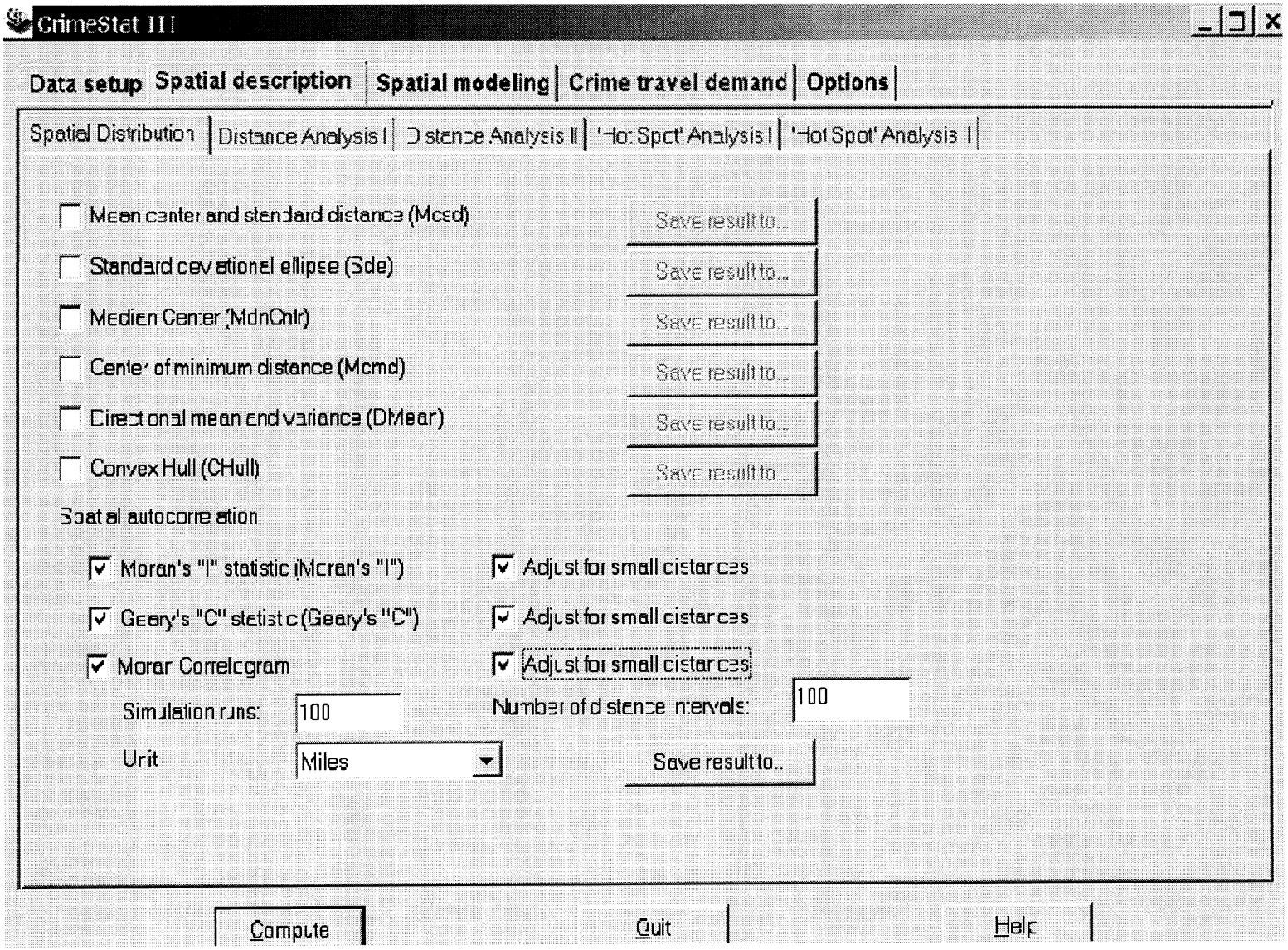
Testing the Significance of the Weighted Moran's I

The empirical distribution can be compared with the theoretical distribution by dividing by an estimate of the theoretical standard deviation

$$Z(I) = \frac{I - E(I)}{S_{E(I)}} \quad (4.35)$$

where "I" is the empirical value calculated from a sample, E(I) is the theoretical mean of a random distribution and $S_{E(I)}$ is the theoretical standard deviation of E(I).

Figure 4.28: Selecting Spatial Autocorrelation Statistics



There are several interpretations of the theoretical standard deviation which affect the particular statistic used for the denominator as well as the interpretation of the significance of the statistic (Anselin, 1992). The most common assumption is to assume that the standardized variable, Z(I), has a sampling distribution which follows a standard normal distribution, that is with a mean of 0 and a variance of 1. This is called the *normality* assumption.[6] A second interpretation assumes that each observed value could have occurred at any location, that is the location of the values and their spatial arrangement is assumed to be unrelated. This is called the *randomization* assumption and has a slightly different formula for the theoretical standard deviation of I.[7] *CrimeStat* outputs the Z-values and p-values for both the normality and randomization assumptions (figure 4.29).

Example 3: Testing Auto Thefts with the Weighted Moran's I

To illustrate the use of Moran's I with point locations requires data to have intensity values associated with each point. Since most crime incidents are represented as a single point, they do not naturally have associated intensities. It is necessary, therefore, to adapt crime data to fit the form required by Moran's I. One way to do this is assign crime incidents to geographical zones and count the number of incidents per zone.

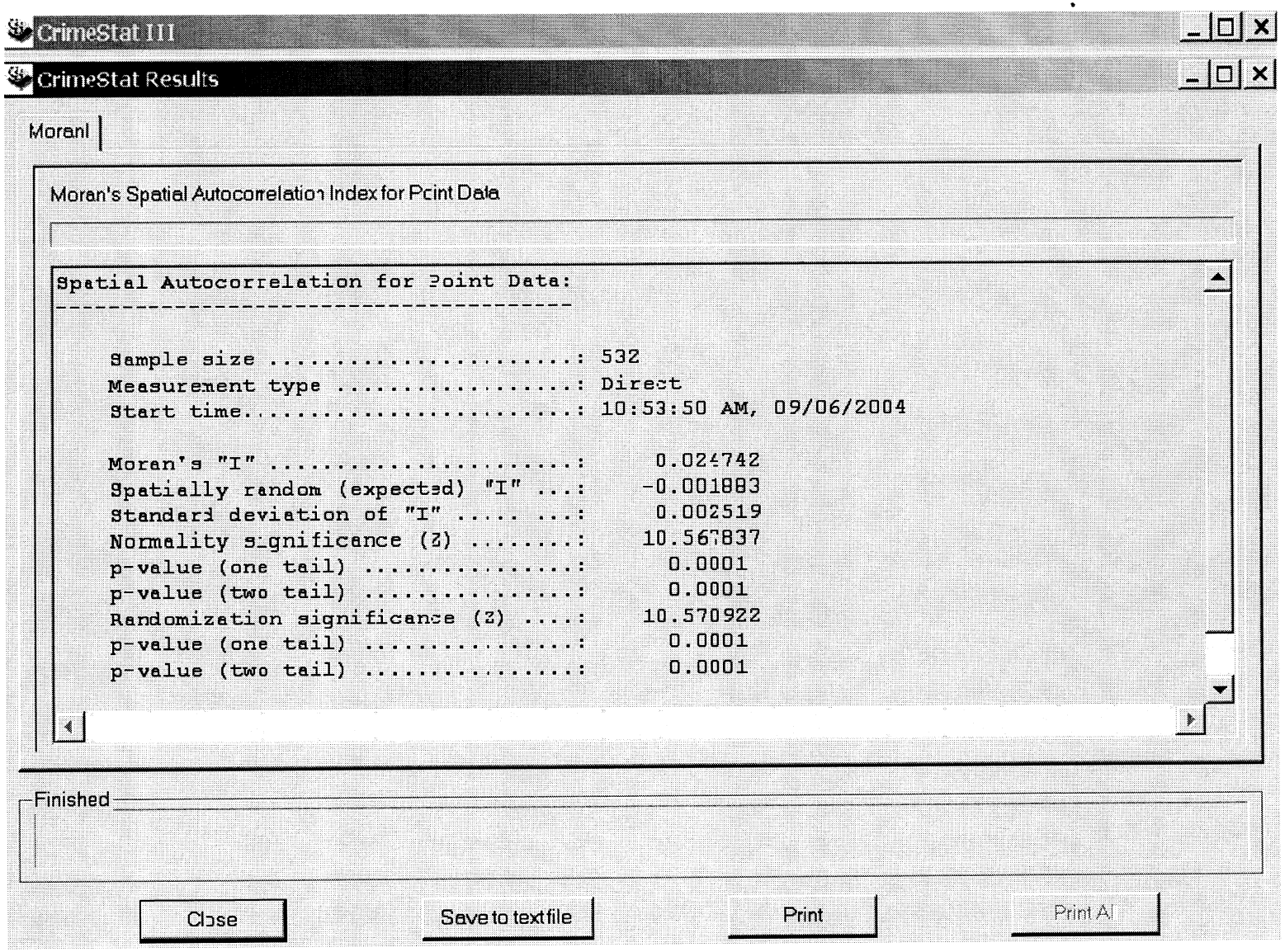
Figure 4.30 shows 1996 motor vehicle thefts in both Baltimore County and Baltimore City by individual blocks. With a GIS program, 14,853 vehicle theft locations were overlaid on top of a map of 13,101 census blocks and the number of motor vehicle thefts within each block were counted and then assigned to the block as a variable (see the 'Assign primary points to secondary points' routine in chapter 5). The numbers varied from 0 incidents (for 7,675 blocks) up to 46 incidents (for 1 block). The map shows the plot of the number of auto thefts per block.

Clearly, aggregating incident locations to zones, such as blocks, eliminates some information since all incidents within a block are assigned to a single location (the centroid of the block). The use of Moran's I, however, requires the data to be in this format. Using data in this form, Moran's I was calculated using the small distance adjustment because many blocks are very close together. *CrimeStat* calculated "I" as 0.012464 and the theoretical value of "I" as -0.000076. The test of significance using the normality assumption gave a Z-value of 125.13, a highly significant value. Below are the calculations.

$$Z(I) = \frac{I - E(I)}{S_{E(I)}} = \frac{0.012464 - (-0.00076)}{0.000100} = 125.13 \ (p \leq .001)$$

In other words, motor thefts are highly and positively spatially autocorrelated. Blocks with many incidents tend to be located close to blocks which also have many incidents and, conversely, blocks with few or no incidents tend to be located close to blocks which also have few or no incidents.

Figure 4.29: Moran's I Statistic Output



How does this compare with other distributions? Finding positive spatial autocorrelation for auto thefts is not surprising given that there is such a high concentration of population (and, hence, motor vehicles) towards the metropolitan center. For comparison, we ran Moran's I for the population of the blocks (Figure 4.31).[8] With these data, Moran's I for population was 0.001659 with a Z-value of 17.32; the theoretical "I" is the same since the same number of blocks is being used for the statistic (n=13,101).

Comparing the "I" value for motor vehicle thefts (0.012464) with that of population (0.00166) suggests that motor vehicle thefts are slightly more concentrated than would be expected on the basis of the population distribution. We can set up an approximate test of this hypothesis. The joint sampling distribution for two variables, such as motor vehicle thefts and population, is not known. However, if we assume that the standard error of the distribution follows a spatially random distribution under the assumption of normality, then equation 4.35 can be applied:

$$Z(I) = \frac{I_{MV} - I_P}{S_{E(I)}} = \frac{0.012464 - 0.001659}{0.000100} = 108.05 \ (p \leq .001)$$

where I_{MV} is the "I" value for motor vehicle thefts, I_P is the "I" value for population, and $S_{E(I)}$ is the standard deviation of "I" under the assumption of normality. The high Z-value suggests that motor vehicle thefts are much more clustered than the clustering of population. To put it another way, they are more clustered than would be expected from the population distribution. As mentioned, this is an approximate test since the joint distribution of "I" for two empirical distributions of "I" is not known.

Geary's C Statistic

Geary's C statistic is similar to Moran's I (Geary, 1954). In this case, however, the interaction is not the cross-product of the deviations from the mean, but the deviations in intensities of each observation location with one another. It is defined as

$$C = \frac{(N-1)\left[\sum_i \sum_j W_{ij} (X_i - X_j)^2\right]}{2\left(\sum_i \sum_j W_{ij}\right) \sum_i (X_i - \bar{X})^2} \qquad (4.36)$$

The values of C typically vary between 0 and 2, although 2 is not a strict upper limit (Griffith, 1987). The theoretical value of C is 1; that is, if values of any one zone are spatially unrelated to any other zone, then the expected value of C would be 1. Values less than 1 (i.e., between 0 and 1) typically indicate positive spatial autocorrelation while values greater than 1 indicate negative spatial autocorrelation. Thus, this index is inversely related to Moran's I. It will not provide identical inference because it emphasizes the differences in values between pairs of observations comparisons rather than the covariation between the pairs (i.e., product of the deviations from the mean). The

Figure 4.30:

1996 Baltimore Region Motor Vehicle Thefts

Number of Vehicle Thefts Per Block Group

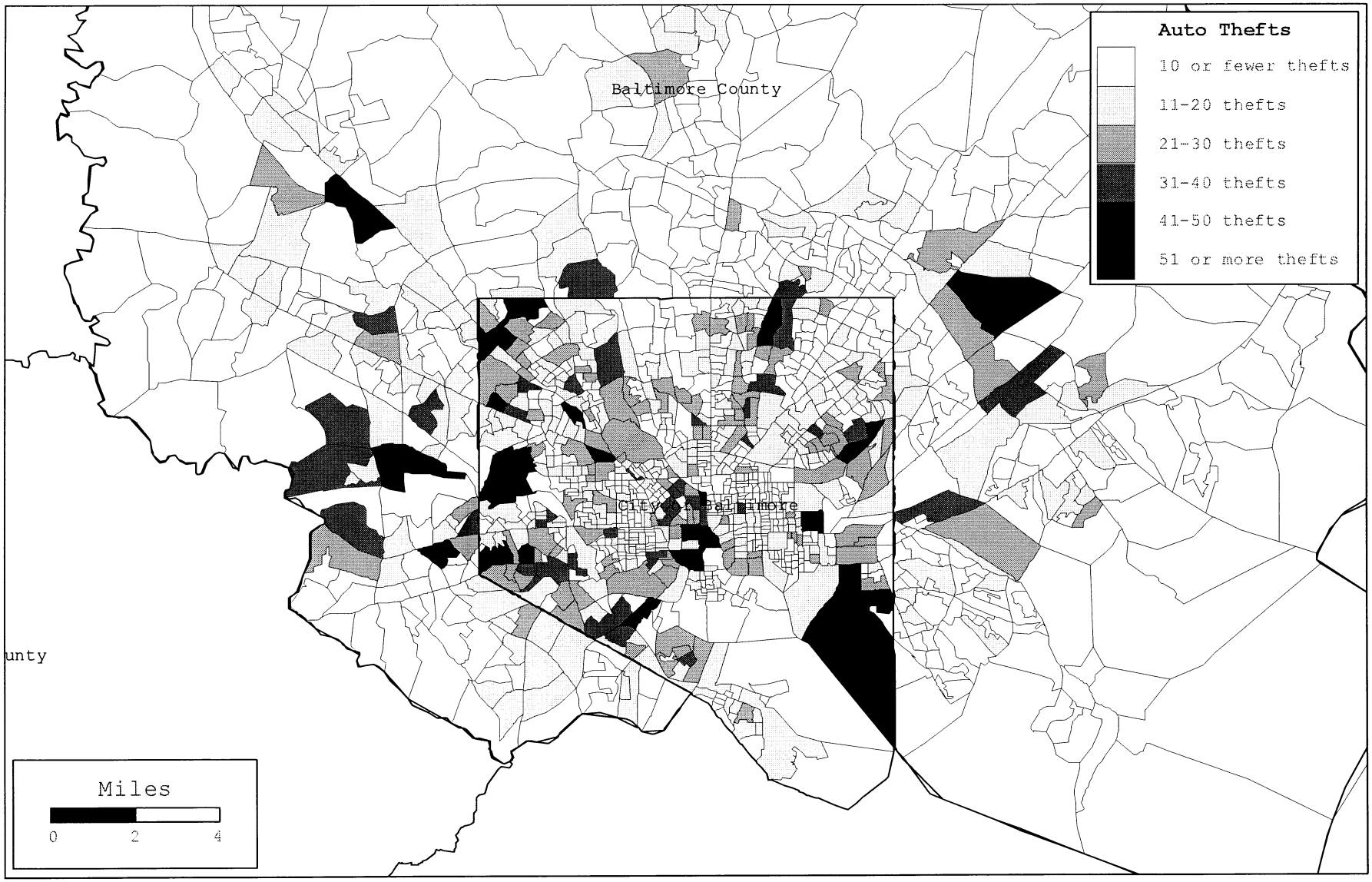
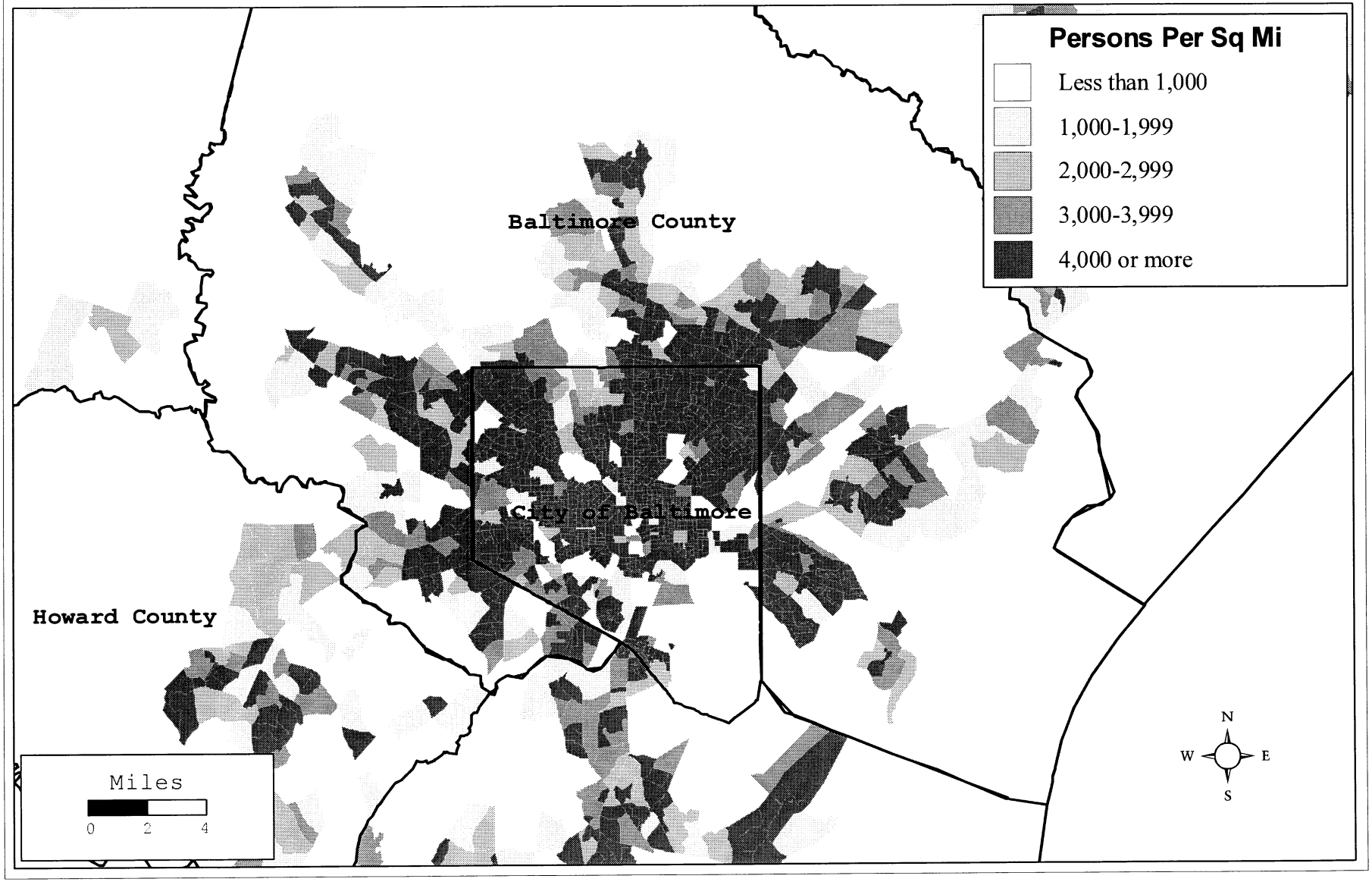


Figure 4.31:

1990 Baltimore Population Density

Number of Persons Per Square Mile by Block



Moran coefficient gives a more global indicator whereas the Geary coefficient is more sensitive to differences in small neighborhoods.

Adjustment for Small Distances

Like Moran's I, the weights are defined as the inverse of the distance between the paired points:

$$W_{ij} = \frac{1}{d_{ij}} \quad \text{repeat} \tag{4.33}$$

However, the weights will tend to increase substantially as the distance between points decreases. Consequently, a small distance adjustment is allowed which ensures that no weight is greater than 1.0. The adjustment scales the distances to one mile

$$W_{ij} = \frac{\text{one mile}}{\text{one mile} + d_{ij}} \quad \text{repeat} \tag{4.34}$$

in the units are specified. This is the default condition although the user can calculate all weights as the reciprocal distance by turning off the small distance adjustment.

Testing the Significance of Geary's C

The empirical C distribution can be compared with the theoretical distribution by dividing by an estimate of the theoretical standard deviation

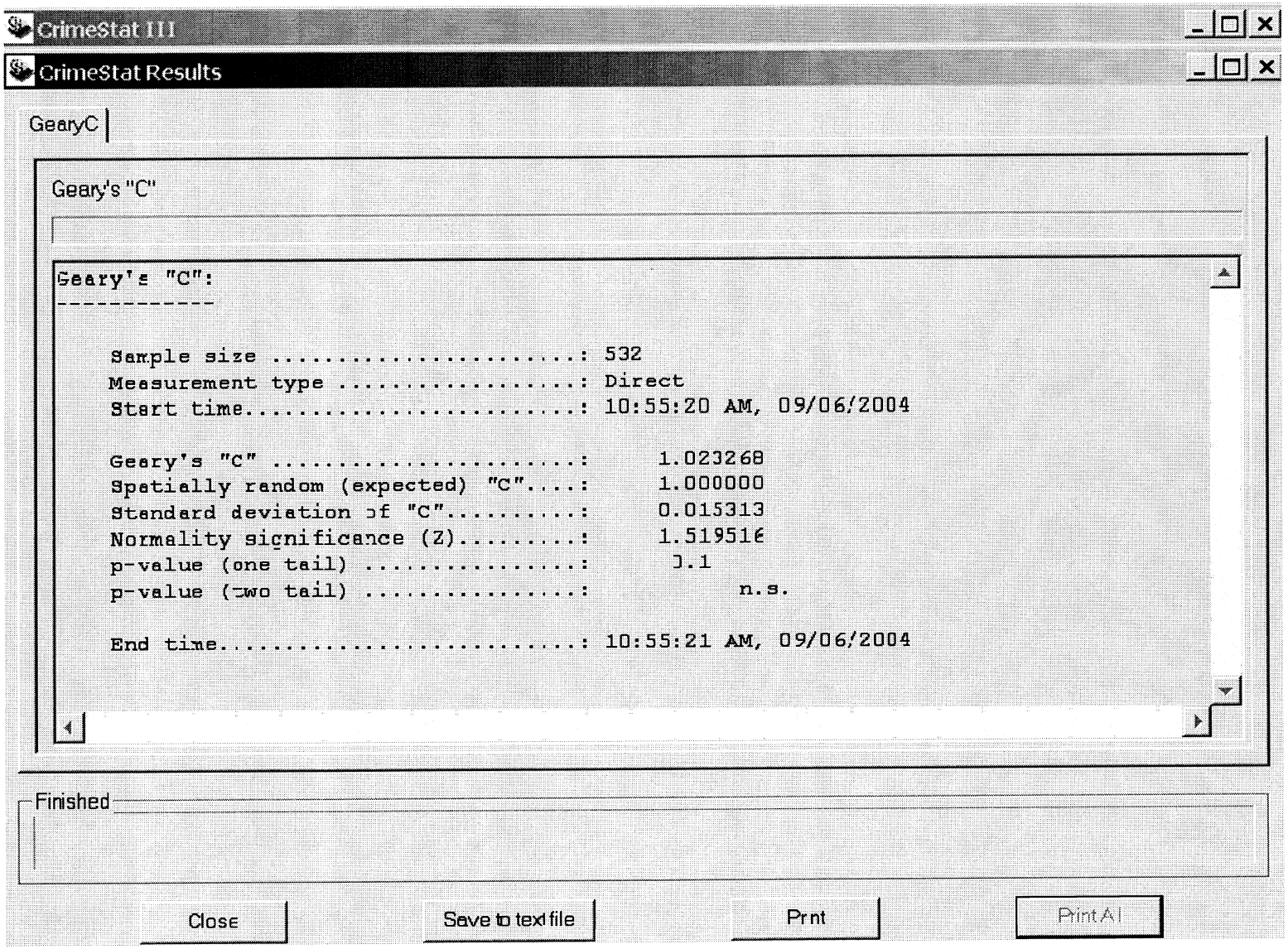
$$Z(C) = \frac{C - E(C)}{S_{E(C)}} \tag{4.37}$$

where C is the empirical value calculated from a sample, E(C) is the theoretical mean of a random distribution and $S_{E(C)}$ is the theoretical standard deviation of E(C). The usual test for C is to assume that the sample Z follows a standard normal distribution with mean of 0 and variance of 1 (normality assumption). *CrimeStat* only calculates the normality assumption though it is possible to calculate the standard error under a randomization assumption (Ripley, 1981).[9] Figure 4.32 illustrates the output.

Example 4: Testing Auto Thefts with Geary's C

Using the same data on auto thefts for Baltimore County and Baltimore City, the C value for auto thefts was 1.0355 with a Z-value of 10.68 ($p \leq .001$) while that for population was 0.924811 with a Z-value of 122.61 ($p \leq .001$). The C value of motor vehicle thefts is greater than the theoretical C of 1 and suggests *negative* spatial autocorrelation, rather than positive spatial autocorrelation. That is, the index suggests that blocks with a high

Figure 4.32: Geary's C Statistic Output



Global Moran's I and Small Distance Adjustment: Spatial Pattern of Crime in Tokyo

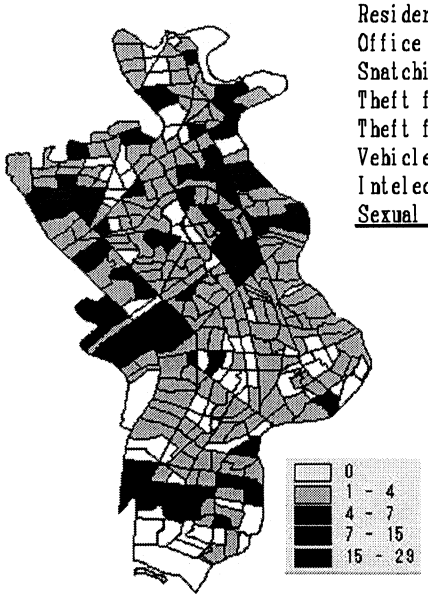
Takahito Shimada
National Research Institute of Police Science
National Police Agency, Chiba, Japan

Crimestat calculates spatial autocorrelation indicators such as Moran's I and Geary's C. These indicators can be used to compare the spatial patterns among crime types. Moran's I is calculated based on the spatial weight matrix where the weight is the inverse of the distance between two points. There is a problem that could occur for incident locations in that the weight could become very large as the distance between points become closer. In *Crimestat*, the small distance adjustment is available to solve this problem. The adjustment produces a maximum weight of 1 when the distance between points is 0.

The number of reported crimes in Tokyo increased from 1996 to 2000 although the city is generally very safe. For this analysis, 68,400 cases reported in the eastern parts of Tokyo were aggregated by census tracts (N=350). Then *Crimestat* calculated Moran's I for each crime type with and without the small distance adjustment.

The "I" value for most crime types, including burglary, theft, purse snatching, showed significantly positive autocorrelation. The results with and without the small distance adjustment were generally very close. The Pearson's correlation between the original and adjusted Moran's I is .98. Among 10 crime types, relatively strong spatial patterns were detected for car theft, sexual assaults, and residential burglary.

Spatial Patterns of Residential Burglary:
Moran's I = 0.023. z=7.58



Calculated Moran's I by Crime Types

	Original			Adjustment		
Crime Type	Moran's I	z		Moran's I	z	
Felonious Offense	0.018	4.09	**	0.003	0.96	
Violent Offense	0.030	6.27	**	0.007	3.03	**
Residential Burglary	0.055	11.21	**	0.023	7.58	**
Office Burglary	0.028	5.93	**	0.012	4.34	**
Snatching	0.031	6.48	**	0.006	2.45	*
Theft from Vender	0.030	6.38	**	0.012	4.28	**
Theft from Cars	0.081	16.08	**	0.044	13.75	**
Vehicle Theft	0.047	9.65	**	0.018	6.14	*
Intelectual Offense	0.023	4.99	**	0.003	1.79	
Sexual Assault	0.080	16.00	**	0.045	14.04	**

**: p<.01 *: p<.05

Preliminary Statistical Tests for Hotspots: Examples from London, England

Spencer Chainey
Jill Dando Institute of Crime Science
University College
London, England

Preliminary statistical tests for clustering and dispersion can provide insight into what types of patterns will be expected when the crime data is mapped. Global tests can confirm whether there is statistical evidence of clusters (i.e. hotspots) in crime data which can be mapped, rather than mapping data as a first step and struggling to accurately identify hotspots when none actually exist.

Using *CrimeStat*, four statistical tests were compared for robbery, residential burglary and vehicle crime data for the London Borough of Croydon, England. For the incident data, the standard distance deviation and nearest neighbor index were used. For crime incidents aggregated to Census block areas, Moran's I and Geary's C spatial autocorrelation indices were compared. The crime data is for the period June 1999 – May 2000.

Crime type	Number of crime records	Standard distance	NN Index	z-score (test statistic)	***Evidence of Clustering?***
Robbery	1132	3119.5 m	0.47	-34.2	***Yes***
Residential burglary	3104	3664.6 m	0.46	-57.5	***Yes***
Vehicle crime	9314	3706.2 m	0.26	-137.0	***Yes***

Crime type	**Moran's I**	**Geary's C**
All crime	0.0067	1.14
Robbery	0.0078	1.15
Residential burglary	0.014	0.99
Vehicle crime	0.0082	1.08

With the point statistics, all three crime types show evidence of clustering. Vehicle crime shows the more dispersed pattern suggesting that whilst hotspots do exist, they may be more spread out over the Croydon area than that of the other two crime types. For the two spatial autocorrelation measures, there are differences in the sensitivities of the two tests. For example, for robbery, there is evidence of global positive spatial autocorrelation (overall, Census blocks that are close together have similar values than those that are further apart). On the other hand, the Geary coefficient suggests that, at a smaller neighbourhood level, areas with a high number of robberies are surrounded by areas with a low number of robberies.

number of auto thefts are adjacent to blocks with a low number of auto thefts or with low population density. The C value of population, on the other hand, is below the theoretical C of 1 and points to positive spatial autocorrelation. Thus, Geary's C provides a different inference from Moran's I regarding the spatial distribution of the blocks.

In the example above, Moran's I indicated positive spatial autocorrelation for both auto thefts and population density. An inspection of figure 4.30 above show however, that there are little 'peaks' and 'valleys' among the blocks. Several blocks have a high number of auto thefts, but are surrounded by blocks with a low number of auto thefts.

In other words, the Moran coefficient has indicated that there is more positive spatial autocorrelation for motor vehicle thefts among the 13,101 blocks while the Geary coefficient has emphasized the irregular patterning among the blocks. The Geary index is more sensitive to local clustering (second-order effects) than the Moran index, which is better seen as measuring first-order spatial autocorrelation. This illustrates how these indices have to be used with care and cannot be generalized by themselves. Each of them emphasizes slightly different information regarding spatial autocorrelation, yet neither is sufficient by itself. They should be used as part of a larger analysis of spatial patterning.[10]

Moran Correlogram

Moran's I and Geary's C indices are summary tests of global autocorrelation. That is, they summarize all the data and don't distinguish between spatial autocorrelation for different subsets. In subsequent chapters, we will examine particular sub-sets of the data that are spatially autocorrelated, such as 'hot spots', 'cold spots' or space-time clusters.

One simple application of Moran's I is a plot of the "I" by different distance intervals (or bins). Called a *Moran Correlogram*, the plot indicates how concentrated or distributed is the spatial autocorrelation (Cliff and Haggett, 1988; Bailey and Gatrell, 1995). Essentially, a series of concentric circles is overlaid over the points and the Moran's I statistic is calculated for only those points falling within the circle. The radius of the circle changes from a small circle to a very large one. As the circle increases, the "I" calculation approaches the global value.

In *CrimeStat*, the user can specify how many distance intervals (i.e., circles) are to be calculated. The default is 10, but the user can choose any other integer value. The routine takes the maximum distance between points and divides it into the number of specified distance intervals, and then calculates the "I" value for those points falling within that radius.

Adjustment for Small Distances

If checked, small distances are adjusted so that the maximum weighting is 1 (see p. 49 above). This ensures that the "I" values for individual distances won't become excessively large or excessively small for points that are close together. The default value is no adjustment.

Simulation of Confidence Intervals

A Monte Carlo simulation can be run to estimate approximate confidence intervals around the "I" value. Each simulation inputs random data and calculates the "I" value. The distribution of the random "I" values produce an approximate confidence interval for the actual (empirical) "I". To run the simulation, specify the number of simulations to be run (e.g., 100, 1000, 10000). The default is no simulations.

Output

The output includes:

1. The sample size
2. The maximum distance
3. The bin (interval) number
4. The midpoint of the distance bin
5. The "I" value for the distance bin (I[B])

and if a simulation is run:

6. The minimum "I" value for the distance bin
7. The maximum "I" value for the distance bin
8. The 0.5 percentile for the distance bin
9. The 2.5 percentile for the distance bin
10. The 97.5 percentile for the distance bin
11. The 99.5 percentile for the distance bin.

The two pairs of percentiles (2.5 and 97.5; 0.5 and 99.5) create an approximate 5% and 1% confidence interval. The minimum and maximum "I" values create an envelope. The tabular results can be printed, saved to a text file or saved as a '.dbf' file. For the latter, specify a file name in the "Save result to" in the dialogue box. The dbf file can be imported into a spreadsheet or graphics program to make a graph.

Graphing the "I: values by Distance

A quick graph is produced that shows the "I" value on the Y-axis by the distance bin on the X-axis. Click on the "Graph" button. The graph displays the reduction in spatial autocorrelation with distance. The graph is useful for selecting the type of kernel in the Single- and Duel-kernel interpolation routines when the primary variable is weighted (see Interpolation).

Example: Moran Correlogram of 2000 Baltimore Population

I'll illustrate the Moran correlogram with the 2000 Baltimore regional population. Unlike figure 4.31 above, data by Traffic Analysis Zones (TAZ) were used. These are zones used typically for travel demand modeling (see chapter 12). The reason for using TAZ's,

however, is that data on both employment and population are available and it's possible to compare them. The TAZ data were obtained from the Baltimore Metropolitan Council, the Metropolitan Planning Organization for the Baltimore Metropolitan area.

Figure 4.33 shows a map of the 2000 Baltimore population by TAZ's. There is a higher concentration of population in the City of Baltimore, though some of the outlying TAZ's also have a large population (primarily because they are large in area). Nevertheless, the distribution of population by TAZ's falls off at a relatively slow rate from the center.

Figure 4.34 shows the Moran correlogram for the 2000 TAZ population and compares it to the maximum and minimum values from a Monte Carlo simulation of 100 runs. As seen, the "I" value at short distances of less than a mile is quite high, 0.78. As the distance between zones increase (i.e., the search circle radius gets larger), the "I" value drops off until about 8 miles whereupon it approaches the global "I" value. However, for all distance intervals, the empirical "I" value is higher than the maximum simulated "I" value with random data. In other words, it is highly unlikely that the "I" values obtained for each of the distance intervals was due to chance based on the distribution of random "I" values.

Now, let's look at the distribution of employment (figure 4.35). In this case, employment is much more concentrated in a handful of TAZ's. In most metropolitan areas, employment is usually more concentrated than population. A number of TAZ's in downtown Baltimore have a high concentration of employment as does a corridor leading northward along Charles Street. In Baltimore County, there are stretches of higher employment but, again, they tend to be limited to a handful of TAZ's. In other words, compared to the distribution of population, the distribution of employment is more clustered.

Figure 4.36 compares the Moran correlogram of employment with that of population. As seen, employment has a very high "I" value for short distances, much higher than for population. As mentioned above, the Moran I typically falls between -1.00 and +1.00, but this is not guaranteed. If the differences in values between zones is much greater than the average distance within zones, then the "I" value can exceed 1.0. In the case of figure 4.36, it approaches 3.0. Nevertheless, as the distance increases, the "I" value drops quickly and becomes lower than population for larger distance separations.

Uses and Limitations of the Moran Correlogram

In other words, the Moran correlogram provides information about the scale of spatial autocorrelation, whether it is diffuse over a larger area (e.g., as with the population example) or is more concentrated (e.g., as with the employment example). This can be useful for gauging the extent to which 'hot spots' are truly isolated concentrations of incidents or whether they are by-products of spatial clustering over a larger area. In chapter 6, we will examine a hierarchical clustering algorithm that examines a hierarchy

Figure 4.33:
Baltimore Region Population: 2000
By Traffic Analysis Zones

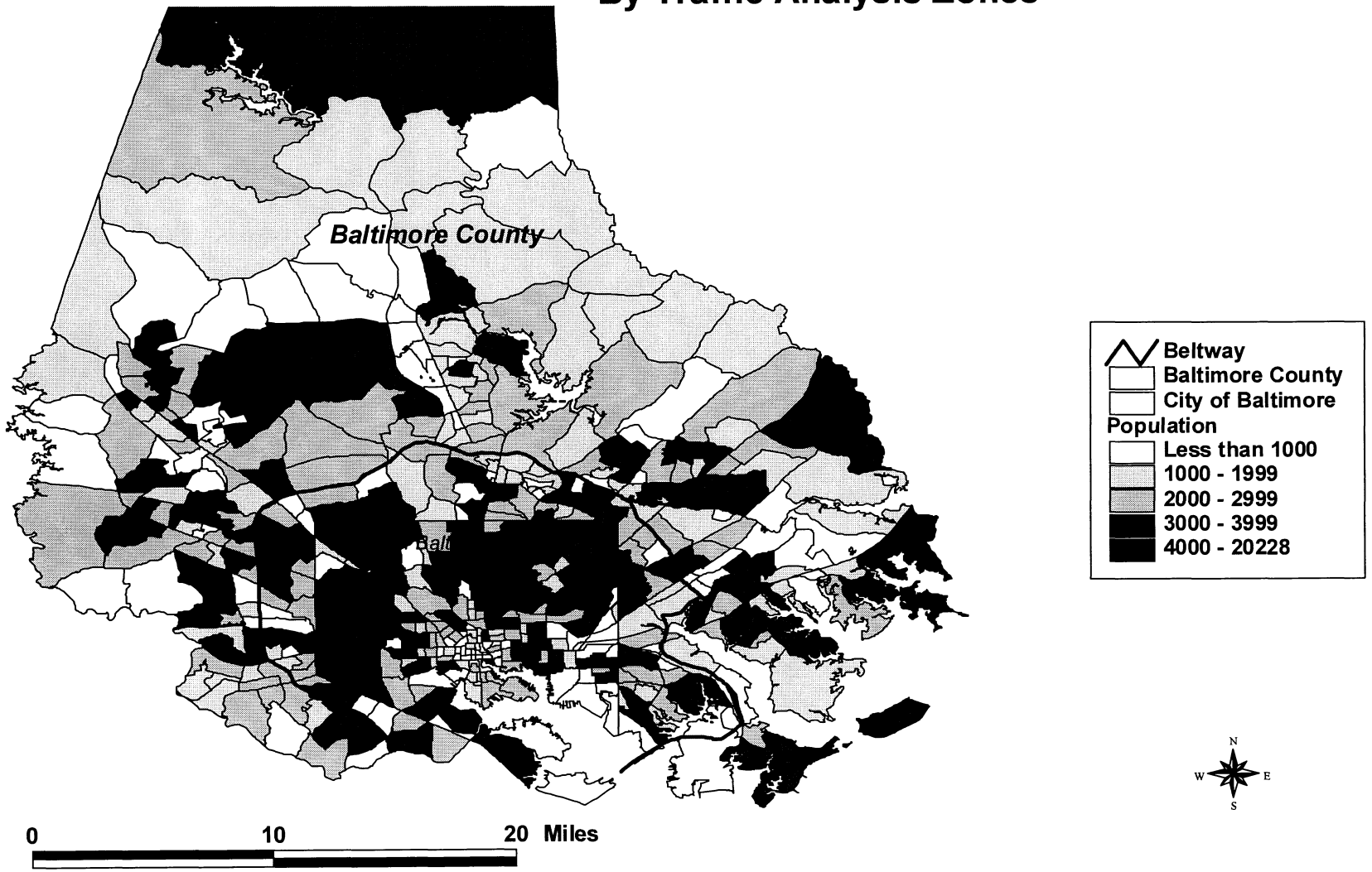


Figure 4.34:

Moran Correlogram of Baltimore Population: 2000

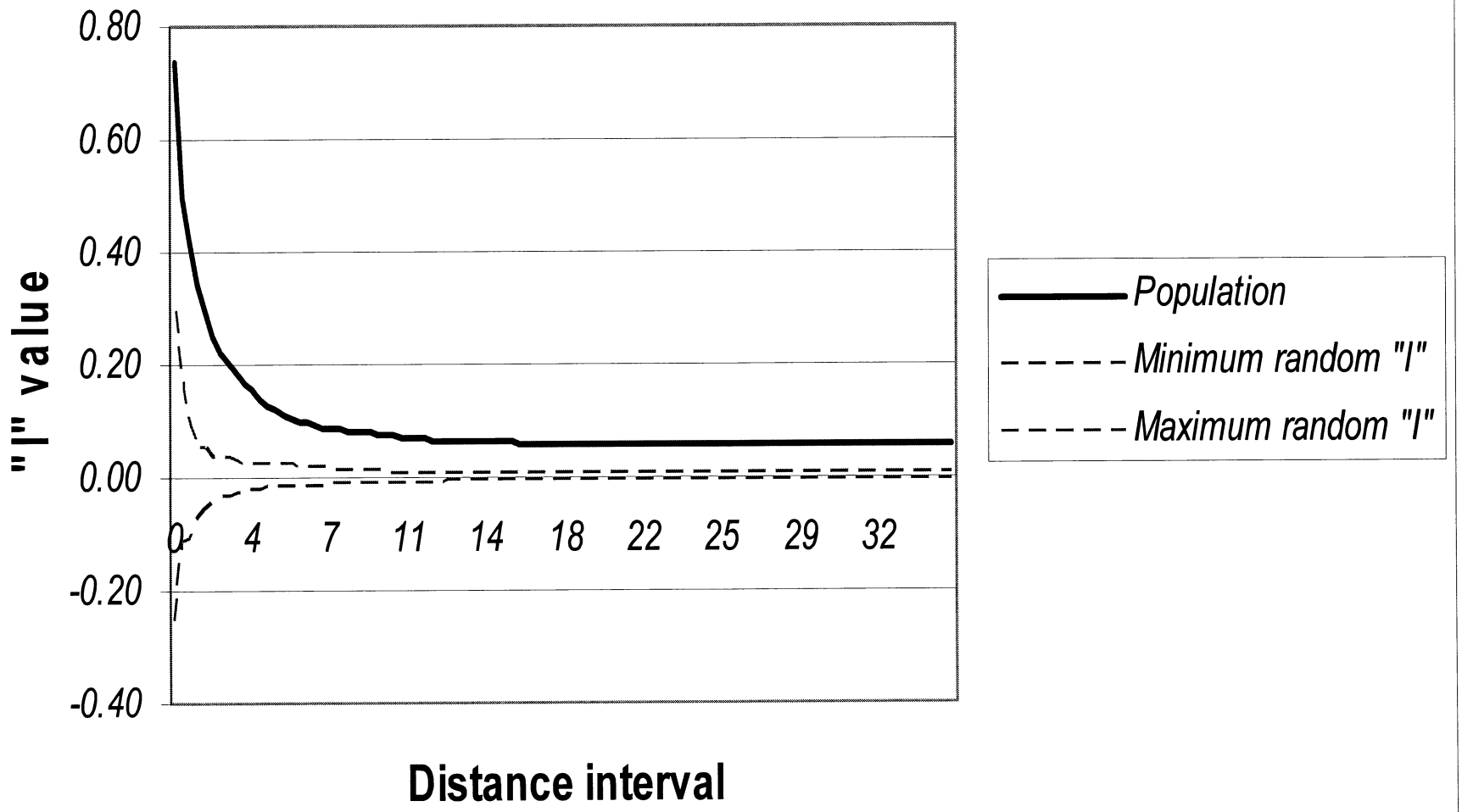


Figure 4.35:
Baltimore Region Employment: 2000
By Traffic Analysis Zones

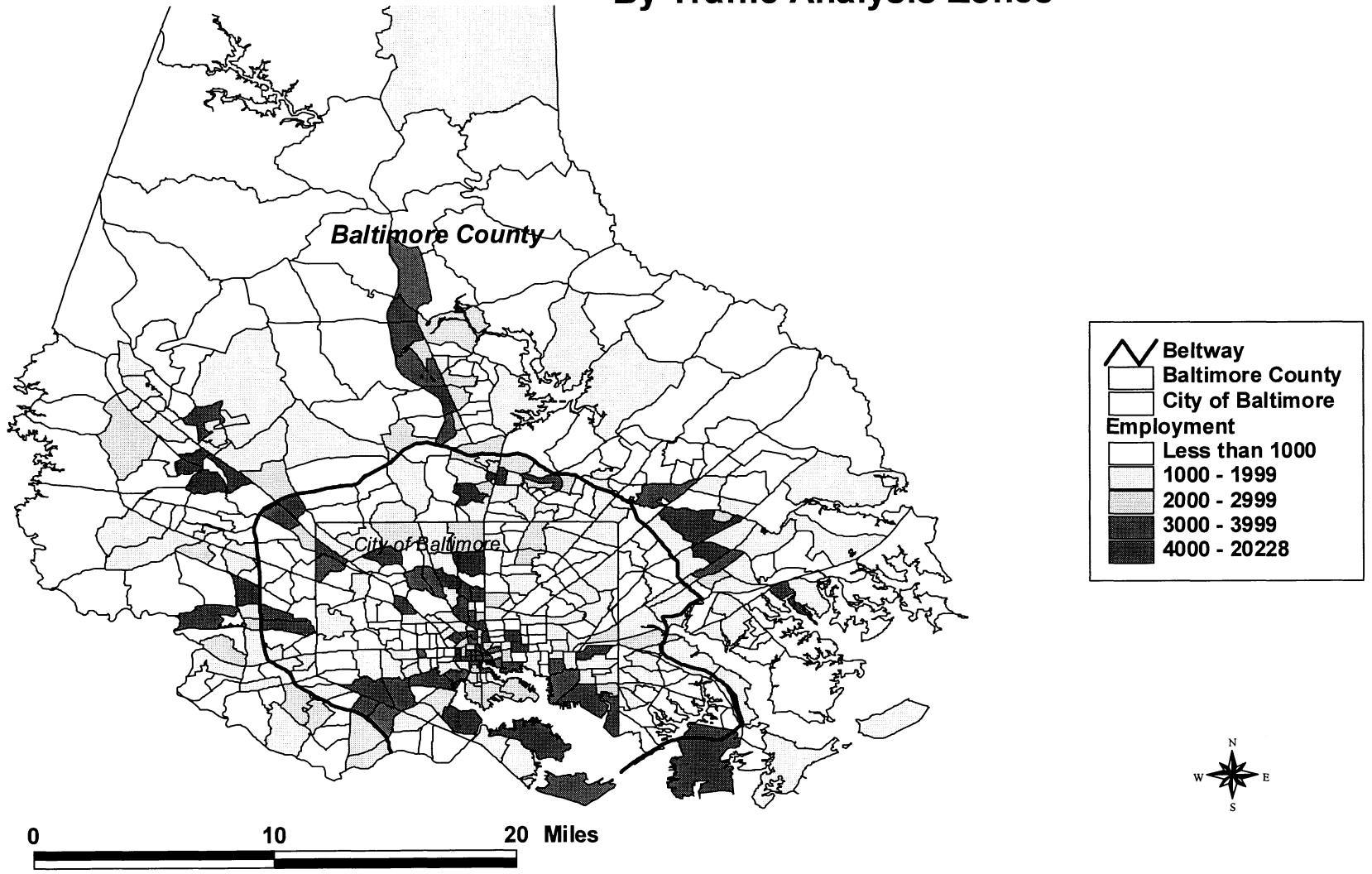
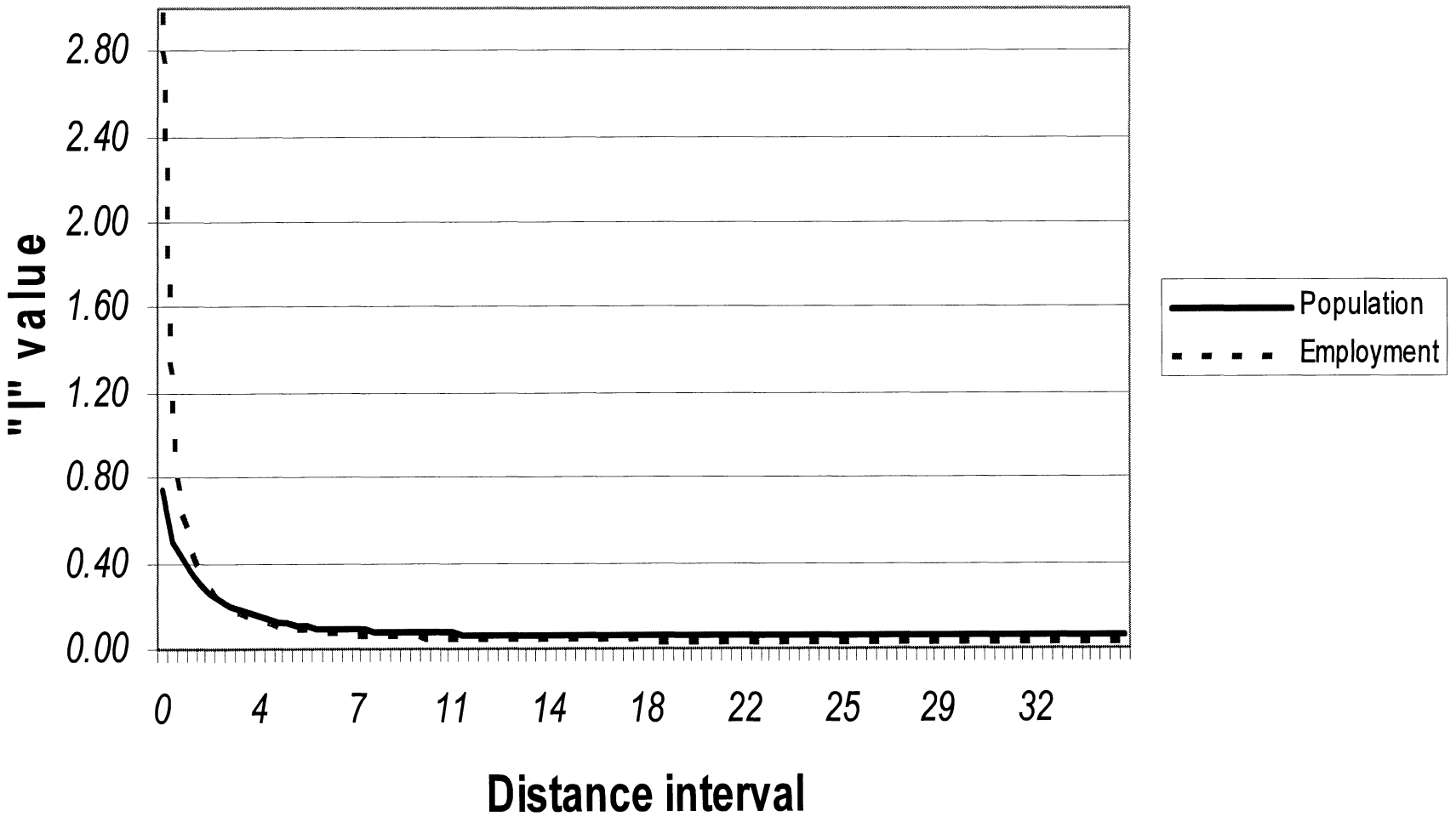


Figure 4.36:

Moran Correlogram of Baltimore Employment & Population: 2000



of clusters (e.g., first-order clusters which are within larger second-order clusters which, in turn, are within even larger third-order clusters). The Moran correlogram provides a quick snapshot of the extent of spatial autocorrelation as a function of scale.

Another use for the Moran correlogram is to estimate the type of kernel function that will be used for interpolation. In chapter 8, this methodology will be explained in detail. But, the key decision is to select a mathematical function that will interpolate data from point locations to grid cells. The shape of the Moran correlogram and the spread is a good indicator of the type of mathematical function to use.

On the other hand, like all global spatial autocorrelation statistics, the correlogram will not indicate where there is clustering or dispersion, only that it exists. For that, we'll have to examine tools that are more focused on the location of concentrations of events (or the opposite, the location of a lack of events).

To explore this further, we will next examine properties of distances between points. Chapter 5 will examine tools for measuring *second-order* effects using the properties of the distances between incident locations.

Endnotes for Chapter 4

1. Hint. There are 40 bars indicated in the status bar while a routine is running. For long runs, users can estimate the calculation time by timing how long it takes for two bars to be displayed and then multiply by 20.

2. *CrimeStat*'s implementation of the Kuhn and Kuenne algorithm is as follows (from Burt and Barber, 1996, 112-113):

   A. Let t be the number of the iteration. For the first iteration only (i.e., t=1) the weighted mean center is taken as the initial estimate of the median location, X_t and Y_t.

   B. Calculate the distance from each point, i, to the current estimate of the median location, d_{ict}, where i is a single point and ct is the current estimate of the median location during iteration t.

      a. If the coordinates are spherical, then Great Circle distances are used.

      b. If the coordinates are projected, then Euclidean distances are used.

   C. Weight each case by a weight, W_i, and calculate

      $$K_{it} = W_i \, \mathbf{e}^{-d(ict)}$$

      where **e** is the base of the natural logarithm(2.7183..) and $d_{(ict)}$ is an alternative way to write d_{ict}.

      a. If no weights are defined in the primary file, W_i is assumed to be 1.

      b. If weights are defined in the primary file, W_i takes their values.

      Note that as the distance, d_{ict}, approaches 0, then $e^{-d(ict)}$ becomes 1.

   D. Calculate a new estimate of the center of minimum distance from

      $$X^{t+1} = \frac{\Sigma K_{it} X_i}{\Sigma K_{it}} \quad \text{for } i=1...n$$

      $$Y^{t+1} = \frac{\Sigma K_{it} Y_i}{\Sigma K_{it}} \quad \text{for } i=1...n$$

where X_i and Y_i are the coordinates of point i (either lat/lon for spherical or feet or meters for projected).

E. Check to see how much change has occurred since the last iteration

$$ABS| X^{t+1} - X^t | \leq 0.000001$$

$$ABS| Y^{t+1} - Y^t | \leq 0.000001$$

a. If either the X or Y coordinates have changed by greater than 0.000001 between iterations, substitute X^{t+1} for X^t and Y^{t+1} for Y^t and repeat steps B through D.

b. If *both* the change in X and the change in Y is less than or equal to 0.000001, then the estimated X_t and Y_t coordinates are taken as the center of median distance.

3. With a weight for an observation, w_i, the squared distance is weighted and the formula becomes

$$S_{XY} = SQRT \frac{\sum w_i(d_{iMC})^2}{(\sum w_i) - 2}$$

Both summations are over all points, N.

4. Formulas for the new axes provided by Ebdon (1988) and Cromley (1992) yield standard deviational ellipses that are too small, for two different reasons. First, they produce transformed axes that are too small. If the distribution of points is random and even in all directions, ideally the standard deviational ellipse should be equal to the standard distance deviation, since $S_x = S_y$. The formula used here has this property. Since the formula for the standard distance deviation is (4.6):

$$SDD = SQRT\left[\frac{\Sigma(X_i - \bar{X})^2 + \Sigma(Y_i - \bar{Y})^2}{N-2} \right]$$

If $S_x = S_y$, then $\Sigma(X_i - \bar{X})^2 = \Sigma(Y_i - \bar{Y})^2$, therefore

$$SDD = SQRT\left[2 * \frac{\Sigma(X_i - \bar{X})^2}{N-2} \right]$$

Similarly, the formula for the transformed axes are (4.9, 4.10):

$$S_x = \text{SQRT}\left[2 * \frac{\Sigma\{(X_i - \bar{X}) \cos\theta - \Sigma(Y_i - \bar{Y}) \sin\theta\}^2}{N-2} \right]$$

$$S_y = \text{SQRT}\left[2 * \frac{\Sigma\{(X_i - \bar{X}) \sin\theta - \Sigma(Y_i - \bar{Y}) \cos\theta\}^2}{N-2} \right]$$

However, if $S_x = S_y$, then $\theta = 0$, Cos0 = 1, Sin0 = 0 and, therefore,

$$S_x = S_y = \text{SQRT}\left[2 * \frac{\Sigma(X_i - \bar{X})}{N-2} \right]$$

which is the same as for the standard distance deviation (SDD) under the same conditions. The formulas used by Ebdon (1988) and Cromley (1992) produce axes which are SQRT(2) times too small.

The second problem with the Ebdon and Cromley formulas is that they do not correct for degrees of freedom and, hence, produce too small a standard deviational ellipse. Since there are two constants in each equation, MeanX and MeanY, then there are only N-2 degrees of freedom. The cumulative effect of using transformed axes that are too small and not correcting for degrees of freedom yields a much smaller ellipse than that used here.

5. In *MapInfo*, the command is *T*able *I*mport *<Mapinfo interchange file>*. With *Atlas*GIS*, the command is *F*ile *O*pen *<boundary (*.bna) file>*. With the DOS version of *Atlas*GIS*, the *Atlas Import-Export* program has to be used to convert the '.bna' output file to an *Atlas*GIS* '.agf' file.

6. The theoretical standard deviation of "I" under the assumption of normality is (from Ebdon, 1985):

$$S_{E(I)} = \text{SQRT}\left[\frac{N^2 \Sigma_{ij} w_{ij}^2 + 3(\Sigma_{ij} w_{ij})^2 - N \Sigma_i (\Sigma_j w_{ij})^2}{(N^2 - 1)(\Sigma_{ij} w_{ij})^2} \right]$$

7. The formula for the theoretical standard deviation of "I" under the randomization assumption is (from Ebdon, 1985):

$$S_{E(I)} = \text{SQRT}\left[\frac{N\{(N^2+3-3N)\Sigma_{ij} w_{ij}^2 + 3(\Sigma_{ij} w_{ij})^2 - N\Sigma_i(\Sigma_j w_{ij})^2\} - k((N^2-N)\Sigma_{ij} w_{ij}^2 + 6(\Sigma_{ij} w_{ij})^2 - 2N(\Sigma_i(\Sigma_j w_{ij})^2)}{(N-1)(N-2)(N-3)(\Sigma_{ij} w_{ij})^2} \right]$$

8. We could have compared Moran's I for auto thefts with that of population, rather than population density. However, since the areas of blocks tend to get larger the farther the distance from the metropolitan center, the effect of testing only population is partly being minimized by the changing sizes of the blocks. Consequently, population density was used to provide a more accurate measure of population concentration. In any case, Moran's I for population is also highly significant: I = 0.00166 (Z=17.32).

9. The theoretical standard deviation for C under the normality assumption is (from Ripley, 1981):

$$S_{E(I)} = \text{SQRT}\left[\frac{(2\sum_{ij} w_{ij}^2+\sum_i (\sum_j w_{ij})^2)(N-1) - 4(\sum_{ij} w_{ij})^2}{2(N+1)(\sum_{ij} w_{ij})^2}\right]$$

10. Anselin (1992) points out that the results of the two indices are determined to a large extent by the type of weighting used. In the original formulation, where adjacent weights of 1 and 0 are used, the two indices are linearly related, though moving in opposite directions (Griffith, 1987). Thus, only adjacent zones have any impact on the index. With inverse distance weights, however, zones farther removed can influence the overall index so it is possible to have a situation whereby adjacent zones have similar values (hence, are positively autocorrelated) whereas zones farther away could have dissimilar values (hence, are negatively autocorrelated).

Chapter 5
Distance Analysis I and II

In this chapter, tools that identify characteristics of the distances between points will be described. The previous chapter provided tools for describing the general spatial distribution of crime incidents or *first-order* properties of the incident distribution (Bailey and Gattrell, 1995). First-order properties are global because they represent the dominant pattern of distribution - where it is centered, how far it spreads out, and whether there is any orientation or direction to its dispersion. *Second-order* (or *local*) properties, on the other hand, refer to sub-regional patterns or 'neighborhood' patterns within the overall distribution. If there are distinct 'hot spots' where many crime incidents cluster together, their distribution is spatially related not so much to the overall global pattern as to something unique in the sub-region or neighborhood. Thus, second-order characteristics tell something about particular environments that may concentrate crime incidents.

There are two distance analysis pages. In Distance analysis I, various second-order statistics are provided, including:

1. NN
2. Linear NN
3. Ripley
4. Assign primary points to secondary points

In Distance analysis II, there are four routines for calculating and outputing distance matrices. This chapter will discuss both sets of routines.

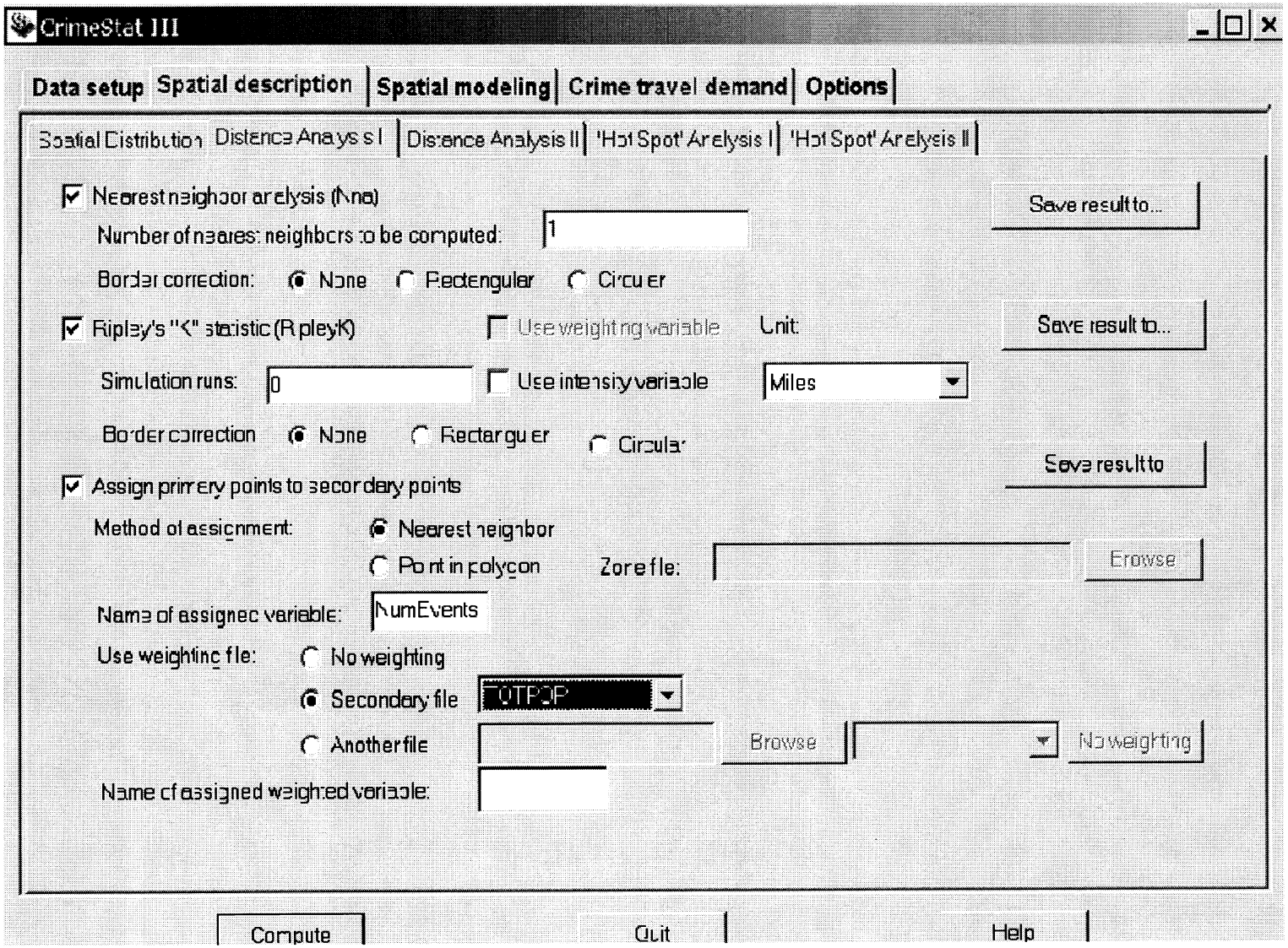
Figure 5.1 shows the Distance analysis I screen and the distance statistics on that page that are calculated by *CrimeStat*.

Nearest Neighbor Index (Nna)

One of the oldest distance statistics is the *nearest neighbor index*. It is particularly useful because it is a simple tool to understand and to calculate. It was developed by two botanists in the 1950s (Clark and Evans, 1954), primarily for field work, but it has been used in many different fields for a wide variety of problems (Cressie, 1991). It has also become the basis of many other types of distance statistics, some of which are implemented in *CrimeStat*.

The nearest neighbor index compares the distances between nearest points and distances that would be expected on the basis of chance. It is an index that is the ratio of two summary measures. First, there is the *nearest neighbor distance*. For each point (or incident location) in turn, the distance to the closest other point (nearest neighbor) is calculated and averaged over all points.

Figure 5.1: Distance Analysis I Screen



$$\text{Nearest Neighbor Distance} = d(NN) = \sum_{i=1}^{N} \left[\frac{\text{Min}(d_{ij})}{N} \right] \quad (5.1)$$

where $\text{Min}(d_{ij})$ is the distance between each point and its nearest neighbor and N is the number of points in the distribution. Thus, in *CrimeStat*, the distance from a single point to every other point is calculated and the smallest distance (the minimum) is selected. Then, the next point is taken and the distance to all other points (including the first point measured) is calculated with the nearest being selected and added to the first minimum distance. This process is repeated until all points have had their nearest neighbor selected. The total sum of the minimum distances is then divided by N, the sample size, to produce an average minimum distance.

The second summary measure is the expected nearest neighbor distance if the distribution of points is completely spatially random. This is the *mean random distance* (or the mean random nearest neighbor distance). It is defined as

$$\text{Mean Random Distance} = d(ran) = 0.5 \text{ SQRT} \left[\frac{A}{N} \right] \quad (5.2)$$

where A is the area of the region and N is the number of incidents. Since A is defined by the square of the unit of measurement (e.g., square mile, square meters, etc.), it yields a random distance measure in the same units (i.e., miles, meters, etc.).[1] If defined on the measurement parameters page by the user, *CrimeStat* will use the specified area in calculating the mean random distance. If no area measurement is provided, *CrimeStat* will take the rectangle defined by the minimum and maximum X and Y points.

The nearest neighbor index is the ratio of the observed nearest neighbor distance to the mean random distance

$$\text{Nearest Neighbor Index} = NNI = \frac{d(NN)}{d(ran)} \quad (5.3)$$

Thus, the index compares the average distance from the closest neighbor to each point with a distance that would be expected on the basis of chance. If the observed average distance is about the same as the mean random distance, then the ratio will be about 1.0. On the other hand, if the observed average distance is smaller than the mean random distance, that is, points are actually closer together than would be expected on the basis of chance, then the nearest neighbor index will be less than 1.0. This is evidence for clustering. Conversely, if the observed average distance is greater than the mean random distance, then the index will be greater than 1.0. This would be evidence for dispersion, that points are more widely dispersed than would be expected on the basis of chance.

Testing the Significance of the Nearest Neighbor Index

Some differences from 1.0 in the nearest neighbor index would be expected by chance. Clark and Evans (1954) proposed a Z-test to indicate whether the observed average nearest neighbor distance was significantly different from the mean random distance (Hammond and McCullagh, 1978; Ripley, 1981). The test is between the observed nearest neighbor distance and that expected from a random distribution and is given by

$$Z = \frac{d(NN) - d(ran)}{SE_{d(ran)}} \tag{5.4}$$

where the standard error of the mean random distance is approximately given by:

$$SE_{d(ran)} \approx \text{SQRT}\left[\frac{(4-\pi)\,A}{4\pi N^2}\right] \approx \frac{0.26136}{\text{SQRT}[\,N^2/A\,]} \tag{5.5}$$

with A being the area of region and N the number of points. There have been other suggested tests for the nearest neighbor distance as well as corrections for edge effects (see below). However, equations 5.4 and 5.5 are used most frequently to test the average nearest neighbor distance. See Cressie (1991) for details of other tests.

Calculating the statistics

Once nearest neighbor analysis has been selected, the user clicks on *Compute* to run the routine. The program outputs 10 statistics:

1. The sample size
2. The mean nearest neighbor distance
3. The standard deviation of the nearest neighbor distance
4. The minimum distance
5. The maximum distance
6. The mean random distance for both the bounding rectangle and the user input area, if provided
7. The mean dispersed distance for both the bounding rectangle and the user input area, if provided
8. The nearest neighbor index for both the bounding rectangle and the user input area, if provided
9. The standard error of the nearest neighbor index for both the maximum bounding rectangle and the user input area, if provided
10. A significance test of the nearest neighbor index (Z-test)
11. The p-values associated with a one tail and two tail significance test.

In addition, the output can be saved to a '.dbf' file, which can then be imported into spreadsheet or graphics programs.

Example 1: The nearest neighbor index for street robberies

In 1996, there were 1181 street robberies in Baltimore County. The area of the County is about 607 square miles and is specified on the measurement parameters page. *CrimeStat* returns the statistics shown in Table 5.1 with the NNA routine. The mean nearest neighbor distance was 0.116 miles while the mean nearest neighbor distance under randomness was 0.358. The nearest neighbor index (the ratio of the actual to the random nearest neighbor distance) is 0.3236. The Z-value of -44.4672 is highly significant. In other words, the distribution of the nearest neighbors of street robberies in Baltimore County is significantly smaller than what would be expected randomness.

Table 5.1
Nearest Neighbor Statistics for
1996 Street Robberies in Baltimore County
(N=1181)

Mean nearest neighbor distance:	0.11598 mi
Mean random distance based on user input area:	0.35837 mi
Nearest neighbor index:	0.3236
Standard error:	0.00545 mi
Test Statistic (Z):	-44.4672
p-value (one tail)	≤.0001
p-value (two tail)	≤.0001

It should be noted that the significance test for the nearest neighbor index is not a test for complete spatial randomness, for which it is sometimes mistaken. It is only a test whether the average nearest neighbor distance is significantly different than what would be expected on the basis of chance. In other words, it is a test of *first-order* nearest neighbor randomness.[2] There are also second-order, third-order, and so forth distributions that may or may not be significantly different from their corresponding orders under complete spatial randomness. A complete test would have to test for all those effects, what are called *K-order* effects.

Example 2: The nearest neighbor index for residential burglaries

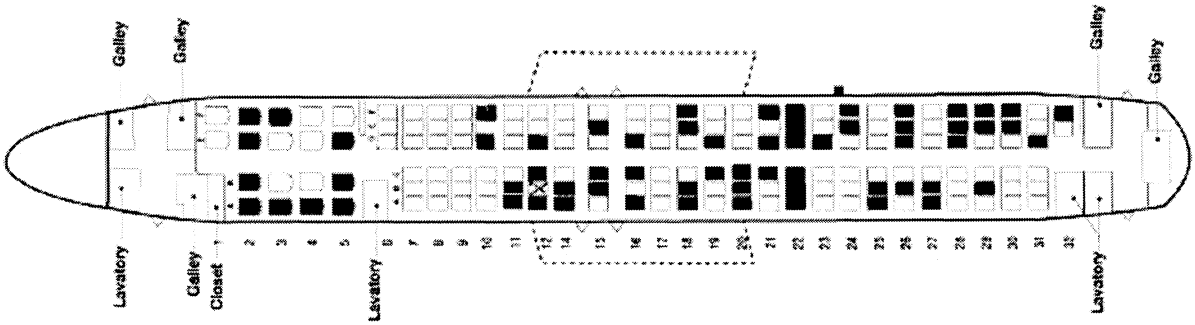
The nearest neighbor index and test can be very useful for understanding the degree of clustering of crime incidents in spite of its limitations. For example, in Baltimore County, the distribution of 6051 residential burglaries in 1996 yields the following nearest neighbor statistics (Table 5.2):

SARS and the Distribution of Passengers on an Airplane

Marta A. Guerra
Senior Staff Epidemiologist,
Centers for Disease Control and Prevention
Atlanta, GA

Illness in passengers on board airplanes occurs rather frequently, and investigations are performed to assess whether transmission to other passengers has occurred. During 2002, several passengers with Severe Acute Respiratory Syndrome (SARS) traveled to the United States by airplane while they were infectious. Since transmission of SARS can be airborne, there is concern that it could spread during an airline flight. A survey was undertaken on a flight where a confirmed SARS case was on board. Serum samples of passengers were taken to evaluate if transmission of SARS had occurred during the flight, and whether transmission is related to sitting near the SARS case.

The nearest neighbor index was used to compare the distances between the seats of passengers on this flight to distances expected on the basis of chance. A grid (7 m x 32 m) was superimposed on the airline seat configuration, and each seat was assigned an X, Y coordinate based on the width (x) and the length (y) of the airplane. In the diagram below, the seat location of the SARS index case is indicated by an X, and the passengers' seat locations are shaded in black.



Nearest Neighbor Statistics for Airline Flight with SARS Case

The nearest neighbor index of passengers' seats was 0.931 indicating that the distribution was random, not clustered. This preliminary analysis was important in order to establish that the seating arrangement of the passengers was random and independent, and that the passengers' seats were not clustered around the SARS case. Therefore, if any passengers have positive serum samples for SARS, we would be able to evaluate their locations in relation to the SARS case and assess patterns of transmission. In this survey, however, there was no evidence of transmission since none of the passengers had positive serum samples for SARS.

Table 5.2
Nearest Neighbor Statistics for
1996 Residential Burglaries in Baltimore County
(N=6051)

Mean nearest neighbor distance:	0.07134 mi
Mean random distance based on user input area:	0.16761 mi
Nearest neighbor index:	0.4256
Standard error:	0.00113 mi
Test Statistic (Z):	-85.4750
p-value (one tail)	≤.0001
P-value (two tail)	≤.0001

The distribution of residential burglaries is also highly significant. Now, suppose we want to compare the distribution of street robberies (table 5.1) with that residential burglaries (table 5.2). The significance test is not very useful for the comparison because the sample sizes are so large (1181 v. 6051); the much higher Z-value for residential burglaries indicates primarily that there was a larger sample size to test it. However, comparing the relative nearest neighbor indices can be meaningful.

$$\text{Relative Nearest Neighbor Comparison} = \frac{NNI(A)}{NNI(B)} \qquad (5.6)$$

where NNI(A) is the nearest neighbor index for one group (A) and NNI(B) is the nearest neighbor index for another group (B). Thus, comparing street robberies with residential burglaries, we have

$$\frac{NNI\ (A)}{NNI\ (B)} = \frac{NNI\ (robberies)}{NNI\ (burglaries)} = \frac{0.3057}{0.4256} = 0.7182$$

In other words, the distribution of street robberies relative to an expected random distribution appears to be more concentrated than that of burglaries relative to an expected random distribution. There is not a simple significance test of this comparison since the standard error of the joint distributions is not known.[3] But the relative index suggests that robberies are more concentrated than burglaries and, hence, are more likely to have 'hot spot' or 'hot zones' where they are particularly concentrated. This index, of course, does not prove that there are 'hot spots', but only points us towards the higher concentration of robberies relative to burglaries. In the previous chapter, it was shown that robberies had a smaller dispersion than burglaries. Here, however, the analysis is taken a step further to suggest that robberies are more concentrated than burglaries.

Use of Network Distance

In calculating the nearest neighbor index, network distance can be used to calculate the distance between points (see chapter 3). However, unless the data set is very small or you have a lot of patience, I highly recommend that you **don't** do this. Network calculations are very slow and will take a long time to complete for a large file.

K-Order Nearest Neighbors

As mentioned above, the nearest neighbor index is only an indicator of first-order spatial randomness. It compares the average distance for the nearest neighbor to an expected random distance. But what about the second nearest neighbor? Or the third nearest neighbor? Or the K^{th} nearest neighbor? *CrimeStat* constructs K-order nearest neighbor indices. On the distance analysis page, the user can specify the number of nearest neighbor indices to be calculated.

The K-order nearest neighbor routine returns four columns:

1. The order, starting from 1
2. The mean nearest neighbor distance for each order (in meters)
3. The expected nearest neighbor distance for each order (in meters)
4. The nearest neighbor index for each order

For each order, *CrimeStat* calculates the K^{th} nearest neighbor distance for each observation and then takes the average. The expected nearest neighbor distance for each order is calculated by:

$$\text{Mean Random Distance to } K^{th} \text{ nearest neighbor} = d(K_{ran}) = \frac{K\,(2K)!}{(2^K K!)^2\ \text{SQRT}\,[N/A]} \qquad (5.7)$$

where K is the order and ! is the factorial operation (e.g., 4! = 4 x 3 x 2 x 1; Thompson, 1956). The K^{th} nearest neighbor index is the ratio of the observed K^{th} nearest neighbor distance to the K^{th} mean random distance. There is not a good significance test for the K^{th} nearest neighbor index due to the non-independence of the different orders, though there have been attempts (see examples in Getis and Boots, 1978; Aplin, 1983). Consequently, *CrimeStat* does not provide a test of significance.

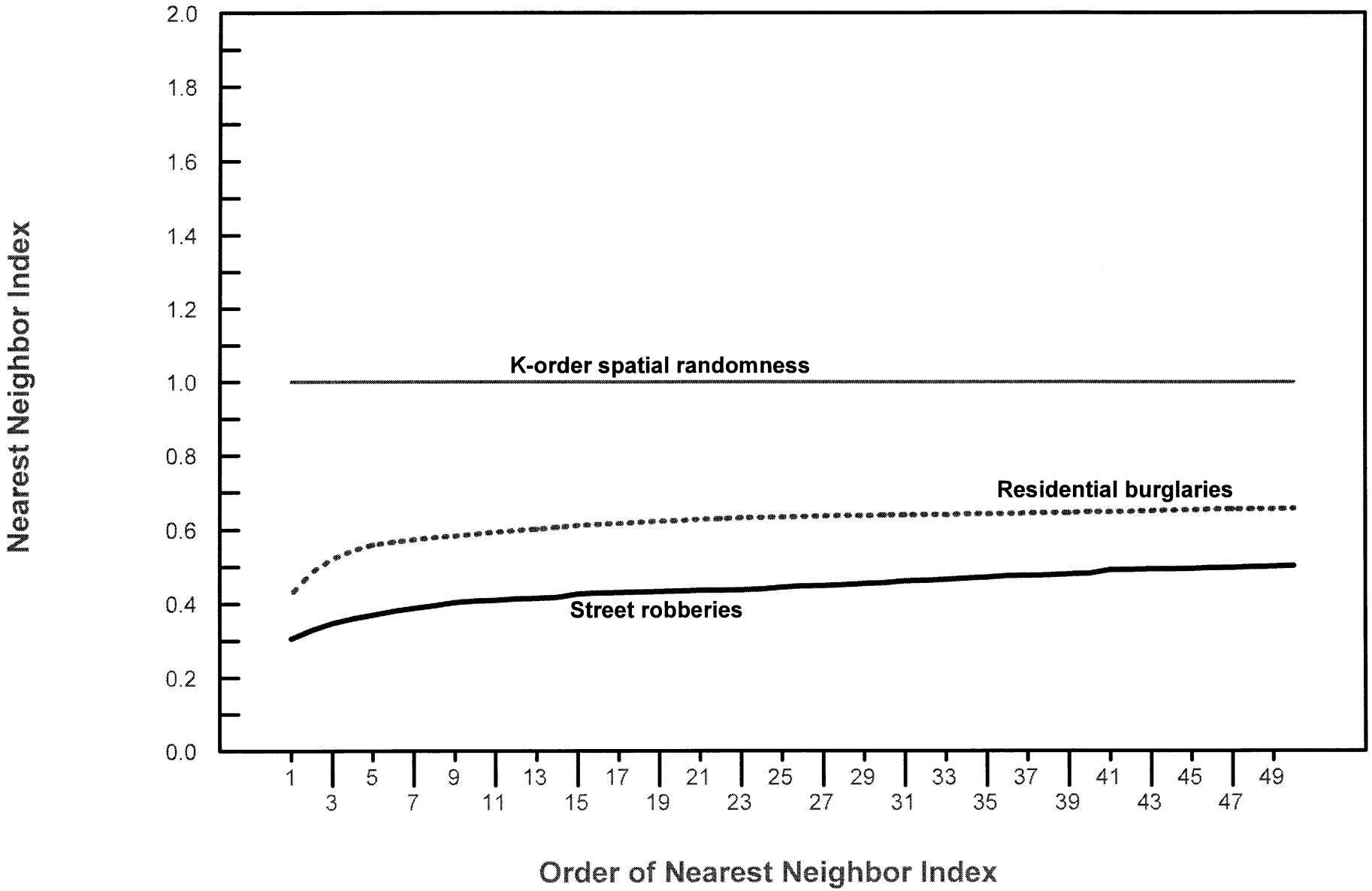
There are no restrictions on the number of nearest neighbors that can be calculated. However, since the average distance increases with higher-order nearest neighbors, the potential for bias from edge effects will also increase. It is suggested that not more than 100 nearest neighbors be calculated.[4]

Nevertheless, the K-order nearest neighbor distance and index can be useful for understanding the overall spatial distributions. Figure 5.2 compares the K-order nearest neighbor index for street robberies with that of residential burglaries. The output was

Figure 5.2

K-Order Nearest Neighbor Indices

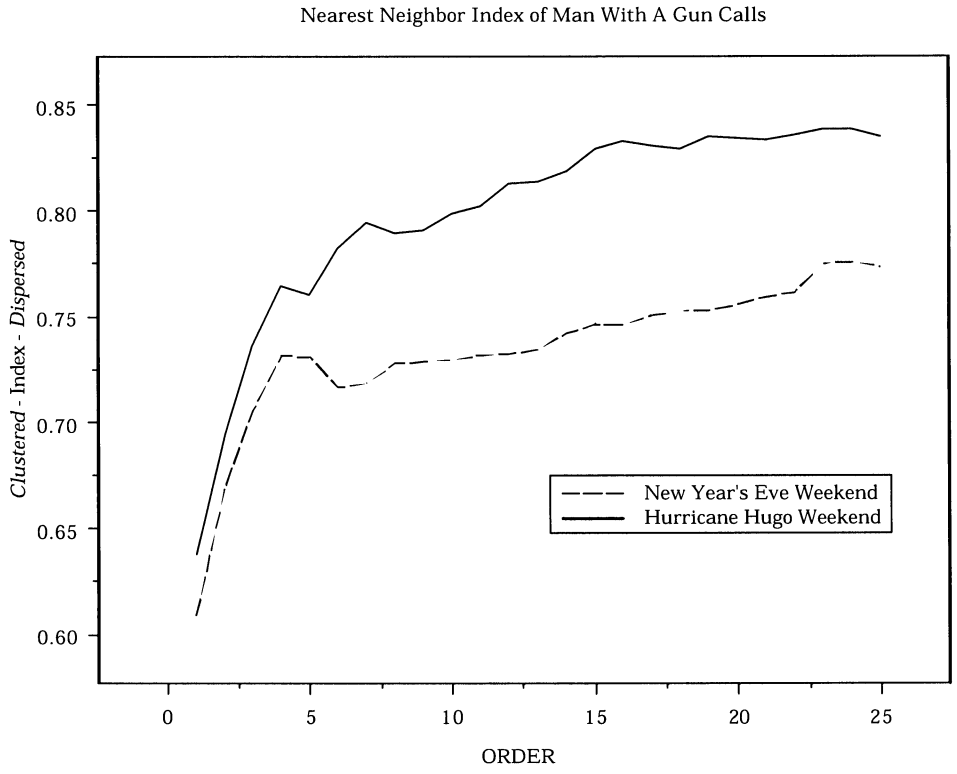
1996 Street Robberies and Residential Burglaries



Nearest Neighbor Analysis
Man With A Gun Calls
Charlotte, N.C.: 1989

James L. LeBeau
Administration of Justice
Southern Illinois University-Carbondale

A comparison was made of *Man with a Gun* calls for the weekend in which Hurricane Hugo hit the North Carolina coast (September 22 – 24) with the following New Year's Eve weekend (December 29-31, 1989). There were 146 *Man with a Gun* calls during the Hurricane Hugo weekend compared to 137 calls for New Year's Eve.



The Nearest Neighbor Index in *CrimeStat* was used to compare the distributions. From the onset, the Hurricane Hugo *Man With a Gun* locations are more dispersed than New Year's Eve. After the 5th nearest neighbor (Order 5) the differences become more pronounced

saved as a '.dbf' and was then imported into a graphics program. The graph shows the nearest neighbor indices for both robberies and burglaries up to the 50th order (i.e., the 50th nearest neighbor). The nearest neighbor index is scaled from 0 (extreme clustering) up to 1 (extreme dispersion). Since a nearest neighbor index of 1 is expected under randomness, the thin straight line at 1.0 indicates the expected K-order index. As can be seen, both street robberies and residential burglaries are much more concentrated than K-order spatial randomness. Further, robberies are more concentrated than even burglaries for each of the 50 nearest neighbors. Thus, the graph reinforces the analysis above that robberies are more concentrated than burglaries, and both are more concentrated than a random distribution.

In other words, even though there is not a good significance test for the K-order nearest neighbor index, a graph of the K-order indices (or the K-order distances) can give a picture of how clustered the distribution is as well as allow comparisons in clustering between the different types of crimes (or the same crime at two different time periods).

Graphing the K-order nearest neighbor

On the output page, there is a quick graph function that displays a curve similar to figure 5.2. This is useful for quickly examining the trends. However, a better graph is made by importing the 'dbf' file output into a spreadsheet or graphics program.

Edge Effects

It should be noted that there are potential edge effects that can bias the nearest neighbor index. An incident occurring near the border of the study area may actually have its nearest neighbor on the other side of the border. However, since there are usually no data on the distribution of incidents outside the study area, the program selects another point within the study area as the nearest neighbor of the border point. Thus, there is the potential for exaggerating the nearest neighbor distance, that is, the observed nearest neighbor distance is probably greater than what it should be and, therefore, there is an *overestimation* of the nearest neighbor distance. In other words, the incidents are probably more clustered than what has been measured (see Cressie, 1991 for details).

Nearest Neighbor Edge Corrections

The default condition is no edge correction. However, one way that the measured distance to the nearest neighbor can be corrected for possible edge effects is to assume for each observed point that there is another point just outside the border at the closest distance. If the distance from a point to the border is shorter than to its measured nearest neighbor, then the nearer theoretical point is taken as a proxy for the nearest neighbor. This correction has the effect of reducing the average neighbor distance. Since it assumes that there is always another point at the border, it probably *underestimates* the true nearest neighbor distance. The true value is probably somewhere in between the measured and the assumed nearest neighbor distance.

CrimeStat has two different edge corrections. Because *CrimeStat* is not a GIS package, it cannot locate the actual border of a study area. One would need a topological GIS package in which the distance from each point to the nearest boundary is calculated. Instead, there are two different geometric models that can be applied. The first assumes that the study area is a rectangle while the second assumes that the study area is a circle. Depending on the shape of the actual study area, one or either of these models may be appropriate.

Rectangular study area

In the rectangular adjustment, the area of the study area, A, is first calculated, either from the user input on the measurement parameters tab or from the maximum bounding rectangle defined by the minimum and maximum X/Y values (see chapter 3). If the user provides an estimate of the area, the rectangle is proportionately re-scaled so that the area of the rectangle equals A. Second, for each point, the distance to the nearest other point is calculated. This is the observed nearest neighbor distance for point i.

Third, the minimum distance to the nearest edge of the rectangle is calculated and is compared to the observed nearest neighbor distance for point i. If the observed nearest neighbor distance for point i is equal to or less than the distance to the nearest border, it is retained. On the other hand, if the observed nearest neighbor distance for point i is greater than the distance to the nearest border, the distance to the border is used as a proxy for the nearest neighbor distance of point i.

Circular study area

In the circular adjustment, first, the area of the study area is calculated, either from the user input on the measurement parameters tab (see chapter 3) or from the maximum bounding rectangle defined by the minimum and maximum X/Y values. If the user has specified a study area on the measurement parameters page, then that value is taken for A and the radius of the circle is calculated by

$$R = \text{SQRT}\,[A / \pi] \tag{5.8}$$

If the user has not specified a study area on the measurement parameters page, then A is calculated from the minimum and maximum X and Y coordinates (the bounding rectangle) and the radius of the circle is calculated with equation 5.8.

Second, for each point, the distance to the nearest other point is calculated. This is the observed nearest neighbor distance for point i. Third, for each point, i, the distance from that point to the mean center is calculated, R_i. Fourth, the minimum distance to the nearest edge of the circle is calculated using

$$R_{iC} = R - R_i \tag{5.9}$$

Fifth, for each point, i, the observed minimum distance is compared to the nearest edge of the circle, R_{ic}. If the observed nearest neighbor distance for point i is equal to or less than the distance to the nearest edge, it is retained. On the other hand, if the observed nearest neighbor distance for point i is greater than the distance to the nearest edge, the distance to the border is used as a proxy for the true nearest neighbor distance of point i.

For either correction

The average nearest neighbor distance is calculated and compared to the theoretical average nearest neighbor distance under random conditions. The indices and tests are as before (see chapter 4). Figure 5.3 below shows a graph of the K-order nearest neighbor index for the 50 nearest neighbors for 1996 motor vehicle thefts in police Precinct 11 of Baltimore County. The uncorrected nearest neighbor indices are compared with those corrected by a rectangle and a circle. As can be seen, both corrections are very similar to the uncorrected. However, they both show greater concentrations than the uncorrected index. The rectangular correction shows greater concentration than the circular because it is less compact (i.e., the average distance from the center of the geometric object to the border is slightly larger). In general, the rectangle will lead to more correction than the circle since it substitutes a greater nearest neighbor distance, on average, for a point nearer the border than to its measured nearest neighbor.

The user has to decide whether either of these corrections are meaningful or not. Depending on the shape of the study area, either correction may or may not be appropriate. If the study area is relatively rectangular, then the rectangular model may provide a good approximation. Similarly, if the study area is compact (circular), then the circular model may provide a good approximation. On the other hand, if the study area is of irregular shape, then either of these corrections may produce more distortion than the raw nearest neighbor index. One has to use these corrections with judgement. Also, in some cases, it may not make any sense to correct the measured nearest neighbor distances. In Honolulu, for example, one would not correct the measured nearest neighbor distances because there are no incidents outside the island's boundary.

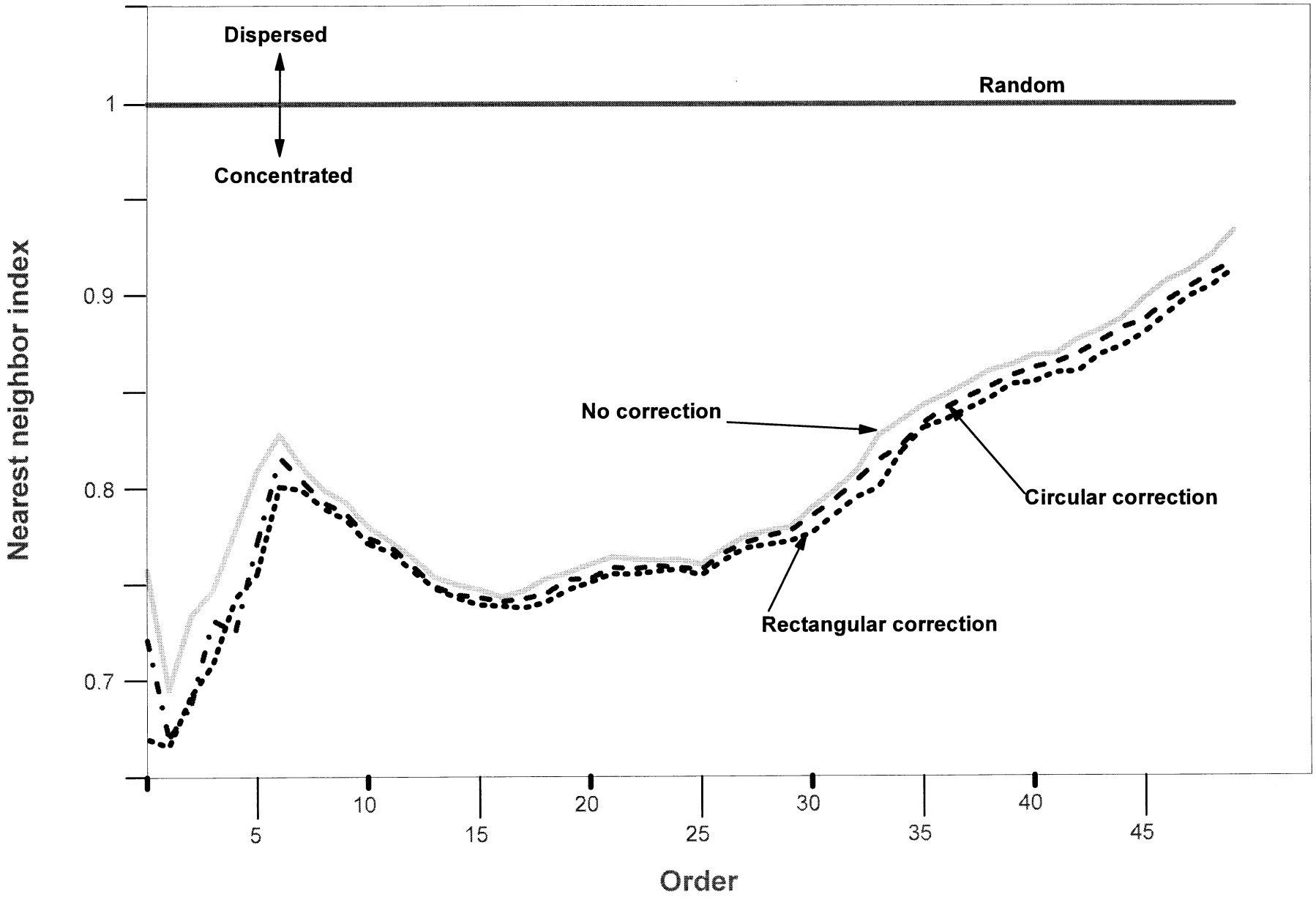
Linear Nearest Neighbor Index (Lnna)

The *linear nearest neighbor index* is a variation on the nearest neighbor routine, but one applied to a street network. All distances along this network are assumed to travel along a grid, hence indirect distances are used. Whereas the nearest neighbor routine calculates the distance between each point and its nearest neighbor using direct distances, the linear nearest neighbor routine uses indirect ('Manhattan') distances (see chapter 3). Similarly, whereas the nearest neighbor routine calculates the expected distance between neighbors in a random distribution of N points using the geographical area of the study region, the linear nearest neighbor routine uses the total length of the street network.

FIGURE 2.7

Correction of Nearest Neighbor Indices

Motor Vehicle Thefts in Precinct 11



The theory of linear nearest neighbors comes from Hammond and McCullagh (1978). The observed linear nearest neighbor distance, Ld(NN), is calculated by *CrimeStat* as the average of indirect distances between each point and its nearest neighbor. The expected linear nearest neighbor distance is given by

$$Ld(ran) = 0.5 \left[\frac{L}{N - 1}\right] \tag{5.10}$$

where L is the total length of street network and N is the sample size (Hammond and McCullagh, 1978, 279). Consequently, the linear nearest neighbor index is defined as

$$\text{Linear Nearest Neighbor Index} = LNNI = \frac{Ld(NN)}{Ld(ran)} \tag{5.11}$$

Testing the Significance of the Linear Nearest Neighbor Index

Since the theoretical standard error for the random linear nearest neighbor distance is not known, the author has constructed an approximate standard deviation for the observed linear nearest neighbor distance:

$$S_{Ld(NN)} \approx SQRT \left[\frac{\Sigma \left(Min(d_{ij}) - Ld(NN) \right)^2}{N - 1}\right] \tag{5.12}$$

where $Min(d_{ij})$ is the nearest neighbor distance for point i and Ld(NN) is the average linear nearest neighbor distance. This is the standard deviation of the linear nearest neighbor distances. The standard error is calculated by

$$SE_{Ld(NN)} = \frac{S_{Ld(NN)}}{SQRT[N]} \tag{5.13}$$

An approximate significance test can be obtained by

$$t = \frac{Ld(NN) - Ld(ran)}{SE_{Ld(NN)}} \tag{5.14}$$

where Ld(NN) is the average linear nearest neighbor distance, Ld(ran) is the expected linear nearest neighbor distance (equation 5.10), and $SE_{Ld(NN)}$ is the approximate standard error of the linear nearest neighbor distance (equation 5.13). Since the empirical standard deviation of the linear nearest neighbor is being used instead of a theoretical value, the test is a *t-test* rather than a Z-test.

Calculating the statistics

On the measurements parameters page, there are two parameters that are input, the geographical area of the study region and the length of street network. At the bottom of the page, the user must select which type of distance measurement to use, direct or indirect. If the measurement type is direct, then the nearest neighbor routine returns the standard nearest neighbor analysis (sometimes called *areal* nearest neighbor). On the other hand, if the measurement type is indirect, then the routine returns the linear nearest neighbor analysis. To calculate the linear nearest neighbor index, therefore, distance measurement must be specified as indirect and the length of the street network must be defined.

Once nearest neighbor analysis has been selected, the user clicks on *Compute* to run the routine. The *Lnna* routine outputs 9 statistics:

1. The sample size
2. The mean linear nearest neighbor distance
3. The minimum linear distance between nearest neighbors
4. The maximum linear distance between nearest neighbors
5. The mean linear random distance
6. The linear nearest neighbor index
7. The standard deviation of the linear nearest neighbor distance
8. The standard error of the linear nearest neighbor distance
9. A significance test of the nearest neighbor index (t-test)
10. The p-values associated with a one tail and two tail significance test.

Example 3: Auto thefts along two highways

The linear nearest neighbor index is useful for analyzing the distribution of crime incidents along particular streets. For example, in Baltimore County, state highway 26 in the western part and state highway 150 in the eastern part have high concentrations of motor vehicle thefts (figure 5.4). In 1996, there were 87 vehicle thefts on highway 26 and 47 on highway 150. A GIS can be used with the linear nearest neighbor index to indicate whether these incidents are greater than what would be expected on the basis of chance.

Table 5.3 presents the data. Using the GIS, we estimate that there are 3,333.54 miles of roadway segments; this number was estimated by adding up the total length of the street network in the GIS. Of all the road segments in Baltimore County, there are 241.04 miles of major arterial roads of which state highway 26 has a total length of 10.42 miles and state highway 150 has a total road length of 7.79 miles.

In 1996, there were 3,774 motor vehicle thefts in the county. If these thefts were distributed randomly, then the random expected distance between incidents would be 0.44 miles (equation 5.10). Using this estimate, table 5.3 shows the number of incidents that would be expected on each of the two state highways if the distribution were random and the ratio of the actual number of motor vehicle thefts to the expected number. As can be

Figure 5.4:

1996 Auto Thefts in Baltimore County

Incident Distribution on State Highways 26 and 150

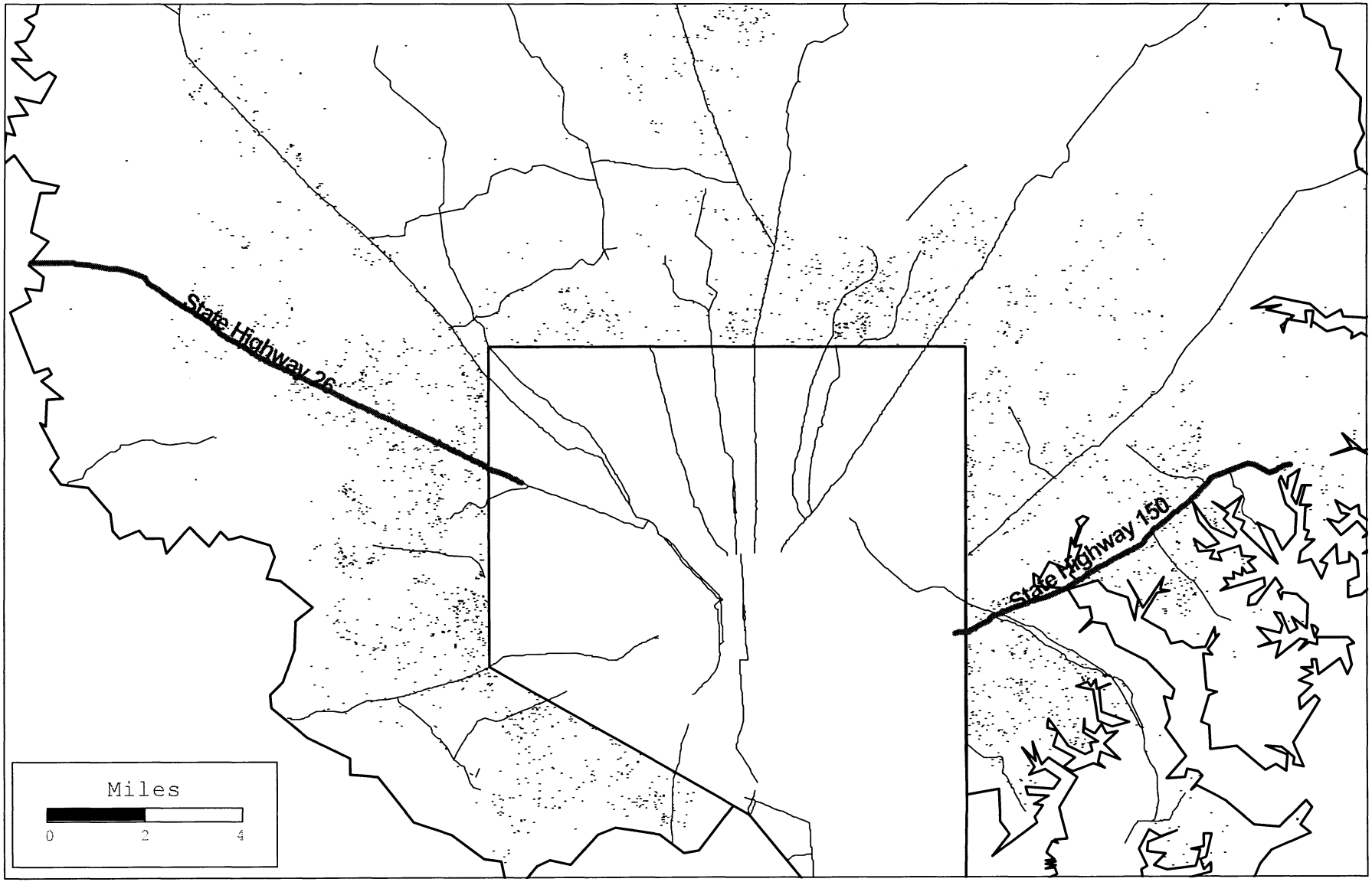


Table 5.3

Comparison of 1996 Baltimore County Auto Thefts
for Different Types of Roads
(N = 3774 Incidents)

Length of Road Segments:

Highway 26	10.42 mi
Highway 150	7.79 mi
All Major Arterials	241.04 mi
All Roads	3333.54 mi

Random Expected Distance
Between Incidents = 0.44 miles

		Proportional To Network		Proportional to Same Road		
Where Incidents Occurred	**Number of Incidents**	**Expected Number *If* Random**	**"Relative to Random" Ratio of Frequency**	**Average Linear Nearest Neighbor Distance**	**Average Random Linear Nearest Neighbor Distance**	**"Relative to Itself" Linear Nearest Neighbor Index**
Highway 26	87	11.8	7.4	0.05 mi	0.06	0.96
Highway 150	47	8.8	5.3	0.08 mi	0.08	0.94
All Major Arterials	607	272.8	2.2	0.13 mi	0.20	0.64 ($p \leq .001$)
All Roads	3774	3774.0	1.0	0.09 mi	044	0.21 ($p \leq .001$)

seen, the distribution of motor vehicle thefts is not random. On all major arterial roads, there are 2.2 times as many thefts as would be expected by a random spatial distribution. In fact, in 1996, of 28,551 road segments in Baltimore County, only 7791 (27%) had one or more motor vehicle thefts occur on them; most of these are major roads. Further, on highway 26 there were 7.4 times as much and on highway 150 there were 5.3 times as much as would be expected if the distribution was random. Clearly, these two highways had more than their share of auto thefts in 1996.

But what about the distribution of the incidents *along* each of these highways? If there were any pattern, for example, most of the incidents clustering on the western edge or in the center, then police could use that information to more efficiently deploy vehicles to respond quickly to events. On the other hand, if the distribution along these highways were no different than a random distribution, then police vehicles must be positioned in the middle, since that would minimize the distance to all occurring incidents.

Unfortunately, the results appear to be close to a random distribution. *CrimeStat* calculates that for highway 26, the average linear nearest neighbor distance is 0.05 miles which is close to the average random linear nearest neighbor distance (0.06 miles). The ratio - the linear nearest neighbor index, is 0.96 with a t-value of -0.16, which is not significantly different from chance. Similarly, for highway 150, the average linear nearest neighbor distance is 0.079 miles which, again, is almost identical to the average random linear nearest neighbor distance (0.084 miles); the nearest neighbor index is 0.94 and the t-value is -0.41 (not significant). In short, even though there was a higher concentration of vehicle thefts on these two state highways than would be expected on the basis of chance, the distribution *along* each highway is not very different than what would be expected on the basis of chance.[5]

K-Order Linear Nearest Neighbors

There is also a K-order linear nearest neighbor analysis, as with the areal nearest neighbors. The user can specify how many additional nearest neighbors are to be calculated. The linear K-order nearest neighbor routine returns four columns:

1. The order, starting from 1
2. The mean linear nearest neighbor distance for each order (in meters)
3. The expected linear nearest neighbor distance for each order (in meters)
4. The linear nearest neighbor index for each order

Since the expected linear nearest neighbor distance has not been worked out for orders higher than one, the calculation produced here is a rough approximation. It applies equation 5.10 only adjusting for the decreasing sample size, N_k, which occurs as degrees of freedom are lost for each successive order. In this sense, the index is really the k-order linear nearest neighbor distance relative to the expected linear neighbor distance for the first order. It is not a strict nearest neighbor index for orders above one.

Nevertheless, like the areal k-order nearest neighbor index, the k-order linear nearest neighbor index can provide insights into the distribution of the points, even if the first-order

is random. Figure 5.5 shows a graph of 50 linear nearest neighbors for 1996 residential burglaries and street robberies for Baltimore County. As with the areal k-order nearest neighbors (see figure 5.3) both burglaries and robberies show evidence of clustering. For both, the first nearest neighbors are closer together than a random distribution. Similarly, over the 50 orders, street robberies are more clustered than burglaries. However, measuring distance on a grid shows that for burglaries, there is only a small amount of clustering. After the fourth order neighbor, the distribution for burglaries is more dispersed than a random distribution. An interpretation of this is that there are small number of burglaries which are clustered, but the clusters are relatively dispersed. Street robberies, on the other hand, are highly clustered, up to over 30 nearest neighbors.

The linear k-order nearest neighbor distribution gives a slightly different perspective on the distribution than the areal. For one thing, the index is slightly biased as the denominator - the K-order expected linear neighbor distance, is only approximated. For another thing, the index measures distance *as if* the street follow a true grid, oriented in an east-west and north-south direction. In this sense, it may be unrealistic for many places, especially if streets traverse in diagonal patterns; in these cases, the use of indirect distance measurement will produce greater distances than what actually occur on the network. Still, the linear nearest neighbor index is an attempt to approximate travel along the street network. To the extent that a particular jurisdiction's street pattern fall in this manner, it can provide useful information.

Graphing the linear K-order nearest neighbor

On the output page, there is a quick graph function that displays a curve similar to figure 5.5 below. This is useful for quickly examining the trends.

Ripley's K Statistic

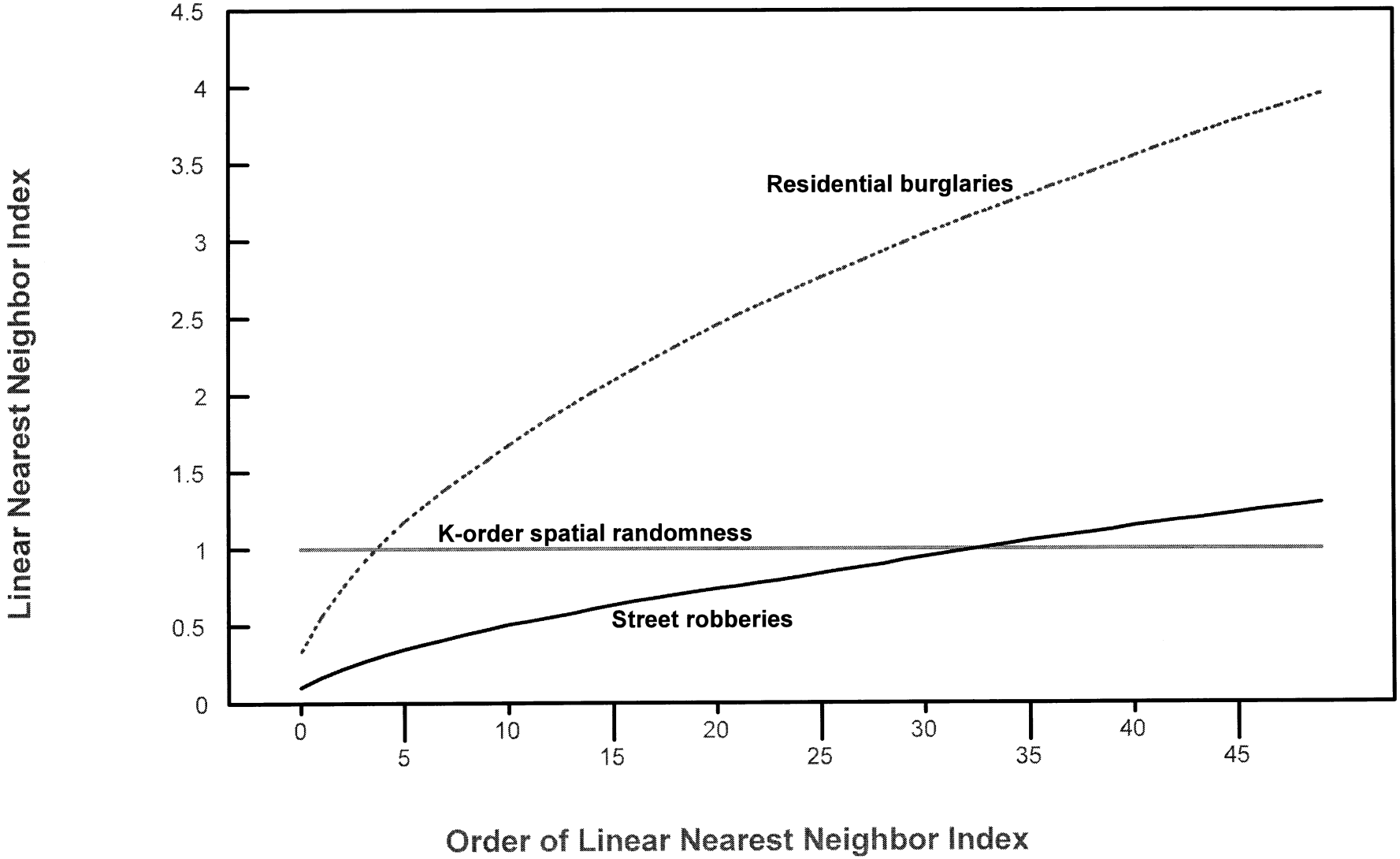
Ripley's K statistic is an index of non-randomness for different scale values (Ripley, 1976; Ripley, 1981; Bailey and Gattrell, 1995; Venables and Ripley, 1997) . In this sense, it is a 'super-order' nearest neighbor statistic, providing a test of randomness for every distance from the smallest up to some specified limit. area. It is sometimes called the *reduced second moment measure*, implying that it is designed to measure second-order trends (i.e., local clustering as opposed to a general pattern over the region). However, it is also subject to first-order effects so that it is not strictly a second-order measure.

Consider a *spatially random* distribution of N points. If circles of radius, t_s, are drawn around each point, where s is the order of radii from the smallest to the largest, and the number of other points that are found within the circle are counted and then summed over all points (allowing for duplication), then the expected number of points within that radius are

Figure 5.5

K-Order Linear Nearest Neighbor Indices

1996 Street Robberies and Residential Burglaries



$$E(\#\text{ of points within distance } d_i) = \frac{N}{A} K(t_s) \tag{5.15}$$

where N is the sample size, A is the total study area, and $K(t_s)$ is the area of a circle defined by radius, t_s. For example, if the area defined by a particular radius is one-fourth the total study area and *if* there is a spatially random distribution, on average approximately one-fourth of the cases will fall within any one circle (plus or minus a sampling error). More formally, with *complete spatial randomness* (csr), the expected number of points within distance, t_s, is

$$E(\#\text{ under csr}) = \frac{N}{A} \pi t_s^2 \tag{5.16}$$

On the other hand, if the average number of points found within a circle for a particular radius placed over each point, in turn, is greater than that found in equation 5.16, this points to clustering, that is points are, on average, closer than would be expected on the basis of chance for that radius. Conversely, if the average number of points found within a circle for a particular radius placed over each point, in turn, is less than that found in equation 5.16, this points to dispersion; that is points are, on average, farther apart than would be expected on the basis of chance for that radius. By counting the number of total numbers within a particular radius and comparing it to the number expected on the basis of complete spatial randomness, the statistic is an indicator of non-randomness.

In this sense, the K statistic is similar to the nearest neighbor distance in that it provides information about the average distance between points. However, it is more comprehensive than the nearest neighbor statistic for two reasons. First, it applies to all orders cumulatively, not just a single order. Second, it applies to all distances up to the limit of the study area because the count is conducted over successively increasing radii.

Under unconstrained conditions, K is defined as

$$K(t_s) = \frac{A}{N^2} \sum_i \sum_{i \neq j} I(t_{ij}) \tag{5.17}$$

where $I(t_{ij})$ is the number of other points, j, found within distance, t_s, summed over all points, i. That is, a circle of radius, t_s, is placed over each point, i. Then, the number of other points, j, within the circle is counted. The circle is moved to the next i and the process is repeated. Thus, the double summation points to the count of all j's for each i, over all i's. Note, the count does *not* include itself, only other points.

After this process is completed, the radius of the circle is increased, and the entire process is repeated. Typically, the radii of circles are increased in small increments so that

there are 50-100 intervals by which the statistic can be counted. In *CrimeStat*, 100 intervals (radii) are used, based on

$$t_s = \frac{R}{100} \tag{5.18}$$

where R is the radius of a circle for whose area is equal to the study area (i.e., the area entered on the measurement parameters page).

One can graph $K(t_s)$ against the distance, t_s, to reveal whether there is any clustering at certain distances or any dispersion at others (if there is clustering at some scales, then there must be dispersion at others). Such a plot is non-linear, however, typically increasing exponentially (Kaluzny et al, 1998. Consequently, $K(t_s)$ is transformed into a square root function, $L(t_s)$, to make it more linear. $L(t_s)$ is defined as:

$$L(t_s) = SQRT\left[\frac{K(t_s)}{\pi}\right] - t_s \tag{5.19}$$

That is, $K(t_s)$ is divided by π and then the square root is taken. Then the distance interval (the particular radius), t_s, is subtracted from this.[6] In practice, only the L statistic is used even though the name of the statistic K is based on the K derivation.

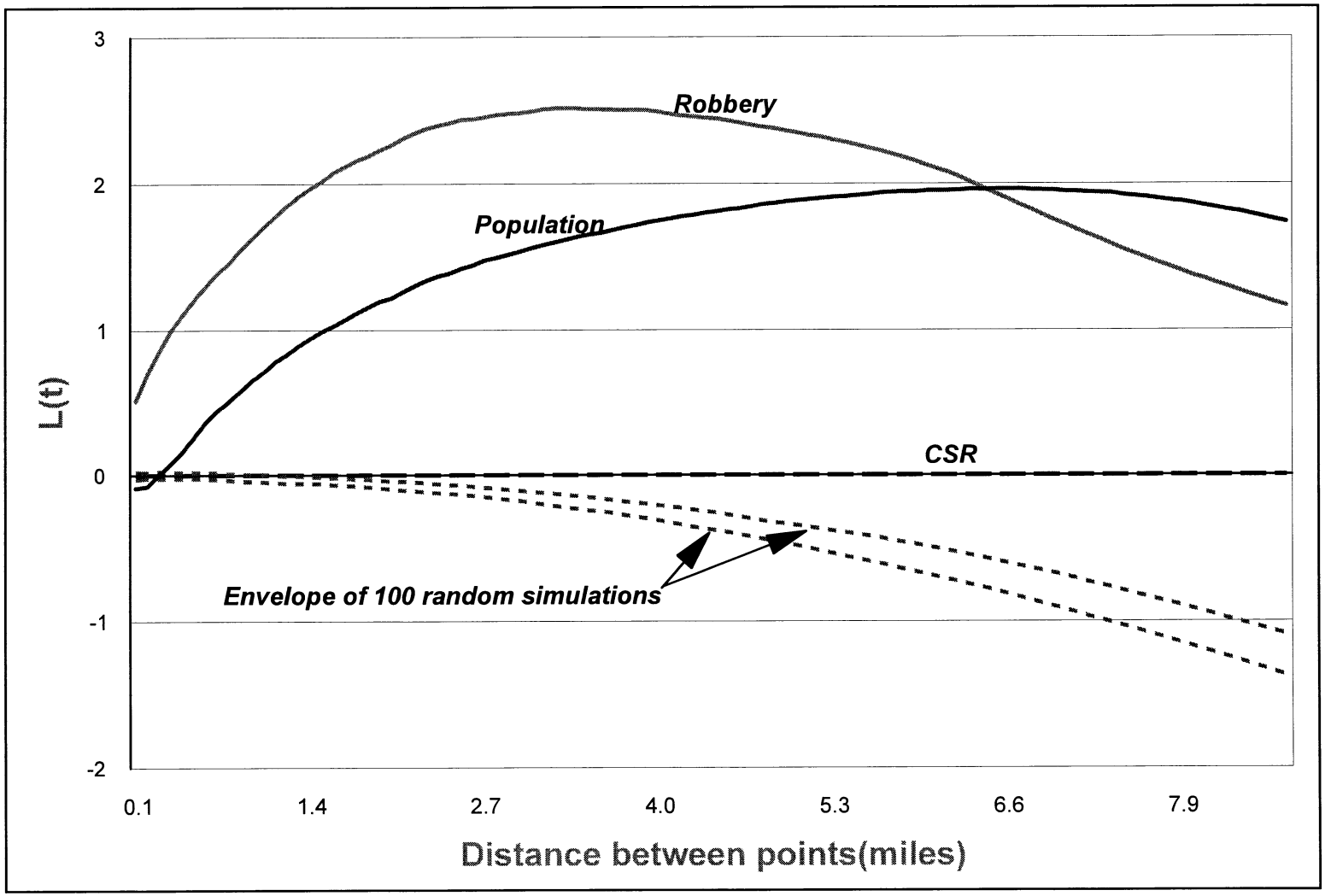
Because the $L(t_s)$ is a measure of second-order clustering, it is usually analyzed for only a short distance. In *CrimeStat III*, the distance is set at one-third the side of a square defined by the area (SQRT[A]/3).[7] Figure 5.6 shows a graph of L(t) against distance for 1996 robberies in Baltimore County. As can be seen, L(t) increases up to a distance of about 3 miles whereupon it decreases again. A "pure" random distribution, known as *complete spatial randomness* (CSR), is shown as a horizontal line at L=0.

Comparison to A Spatially Random Distribution

To understand whether an observed K distribution is different from chance, one typically uses a random distribution. Because the sampling distribution of $L(t_s)$ is not known, a simulation can be conducted by randomly assigning points to the study area. Because any one simulation might produce a clustered or dispersed pattern strictly by chance, the simulation is repeated many times, typically 100 or more. Then, for each random simulation, the L statistic is calculated for each distance interval. Finally, after all simulations have been conducted, the highest and lowest L-values are taken for each distance interval. This is called an *envelope*. Thus, by comparing the distribution of L to the random envelope, one can assess whether the particular observed pattern is likely to be different from chance.[8] In figure 5.6, the L envelope of random data is much less concentrated than that for robberies, indicating that it is highly unlikely the concentration of robberies was due to chance.

Figure 5.6:

"K" Statistic For 1996 Robberies
Compared to Random and 2000 Population Distributions
L(t) = Sqrt[K(t)/pi] - t



Specifying simulations

Because simulations can take a long time, particularly if the data sets are large, the default number of simulations is 0. However, a user can conduct simulations by writing a positive number (e.g., 10, 100, 300). If simulations are selected, *CrimeStat* will conduct the number of simulations specified by the user and will calculate the upper and lower limits for each distance interval, as well as the 0.5%, 2.5%, 5%, 95%, 97.5% and 99% intervals; these latter statistics only make sense if many simulation runs are conducted (e.g. 1000).

The way *CrimeStat* conducts the simulation is as follows. It takes the maximum bounding rectangle of the distribution, that is the rectangle formed by the maximum and minimum X and Y coordinates respectively and re-scales this (up or down) until the rectangle has an area equal to the study area (defined on the measurement parameters page). It then assigns N points, where N is the same number of points as in the incident distribution, using a uniform random number generator to this rectangle and calculates the L statistic. It then repeats the experiment for the number of specified simulations, and calculates the above statistics. For example, with 1181 robberies for 1996, the Ripley's K function calculates the empirical L statistics for 100 distance intervals and compares this to a simulation of 1181 points randomly distributed over a rectangle k times, where k is a user-defined number.

In practice, the simulation test also has biases associated with edges. Unlike the theoretical L under uniform conditions of complete spatial randomness (i.e., stretching in all directions well beyond the study area) where L is a straight horizontal line, the simulated L also declines with increasing distance separation between points. This is a function of the same type of edge bias.

Comparison to Baseline Populations

For most social distributions, such as crime incidents, randomness is not a very meaningful baseline. Most social characteristics are non-random. Consequently, to find that the amount of clustering that is occurring is greater than what would be expected on the basis of chance is not very useful for crime analysts. However, it is possible to compare the distribution of L for crime incidents with the distribution of L for various baseline characteristics, for example, for the population distribution or the distribution of employment. In almost all metropolitan areas, population is more concentrated towards the center than at the periphery; the drop-off in population density is very sharp as was shown in the last chapter. All other things being equal, one would expect more incidents towards the metropolitan center than at the periphery; consequently, the average distance between incidents will be shorter in the center than farther out. This is nothing more than a consequence of the distribution of people. However, to say something about concentrations of incidents above-and-beyond that expected by population requires us to examine the pattern of population as well as of crime incidents.

CrimeStat allows the use of intensity and weighting variables in the calculation of the K statistic. The user must define an intensity or a weight (or both in special circumstances) on the primary file page. The K routine will then use the intensity (or weight) in the

calculation of L. In Figure 5.6 above, there is an envelope produced from 100 random simulations as well as the L distribution from the 2000 population; the latter variable was obtained by taking the centroid of traffic analysis zones from the 2000 census and using population as the intensity variable. As can be seen, the amount of clustering for robberies is greater than both the random envelope as well as the distribution of population. The robbery function is higher than the population function up to about 6 miles. This indicates that robberies are more concentrated than what would be expected from the population distribution for a fairly large area.

In other words, robberies are more clustered together than even what would be expected on the basis of the population distribution and this holds for distances up to about 6 miles, whereupon the distribution of robberies is indistinguishable from a random distribution. For larger distance separations, the L function has little utility since it is usually used to understand localized spatial autocorrelation (Bailey and Gattrell, 1995).

For comparison, figure 5.7 below shows the distribution of 1996 burglaries, again compared to a random envelope and the distribution of population. We find that burglaries are more clustered than population, but less so than for robberies; the L value is higher for robberies than for burglaries for near distances but becomes more dispersed at about 3 miles; it is still more concentrated than a random distribution, however, as seen by the random envelope.. Thus, the distribution of L confirms the result that burglaries tend to be spread over a much larger geographical area in smaller clusters than street robberies, which tend to be more concentrated in large clusters. In terms of looking for 'hot spots', one would expect to find more with robberies than with burglaries.

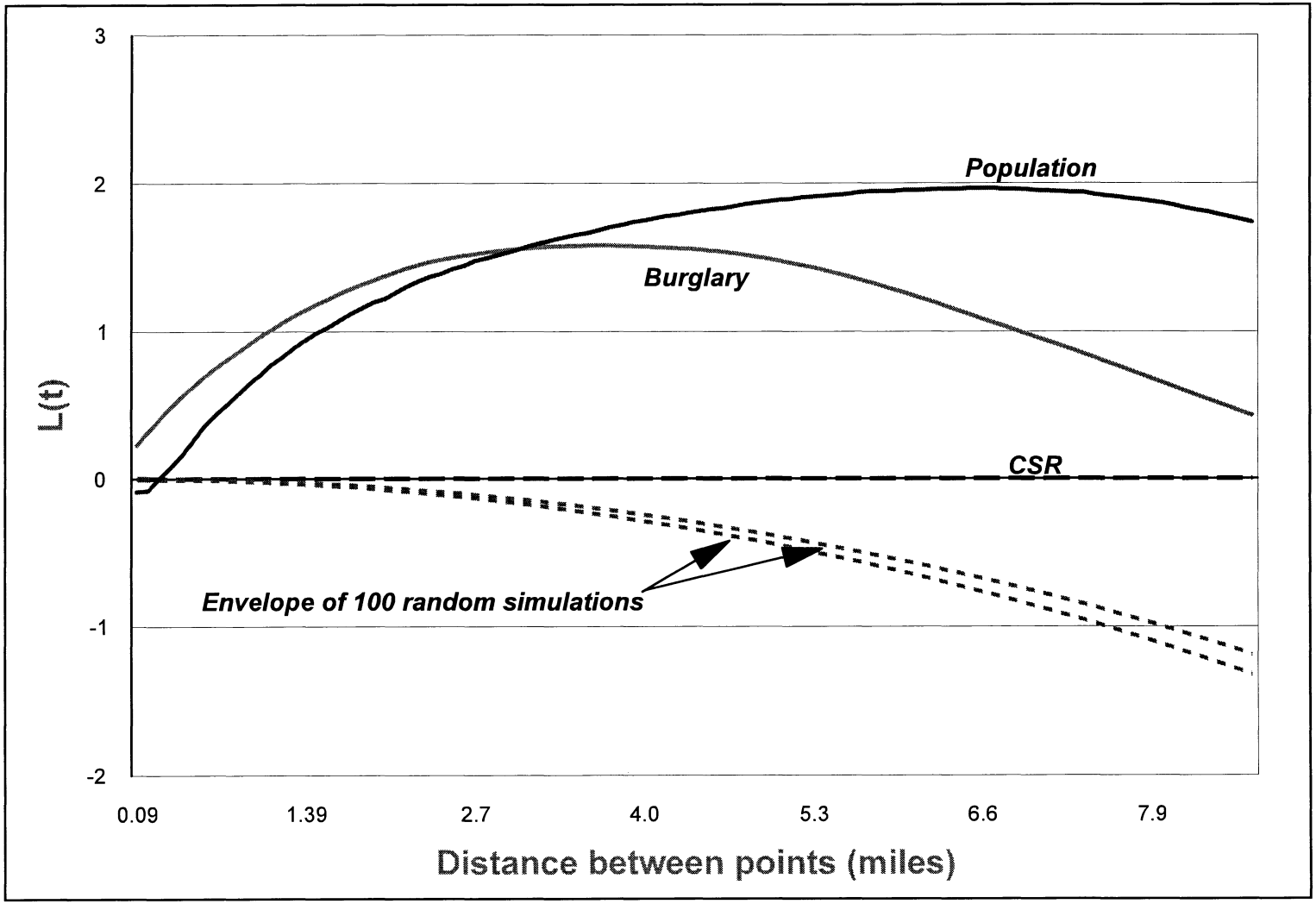
Edge Corrections for Ripley's K

The L statistic is prone to edge effects just like the nearest neighbor statistic. That is, for points located near the boundary of the study area, the number enumerated by any circle for those points will, all other things being equal, necessarily be less than points in the center of the study area because points outside the boundary are not counted. Further, the greater the distance between points that are being tested (i.e., the greater the radius of the circle placed over each point), the greater the bias. Thus, a plot of L against distance will show a declining curve as distance increases as figures 5.6 and 5.7 show.

There are various adjustments to the function to help correct the bias. One is a 'guard rail' within the study area so that points outside the guard rail, but inside the study area can only be counted for points inside the guard rail, but cannot be used for enumerating other points within a circle placed over them (that is, they can only be j's and not i's, to use the language of equation 5.17). Such an operation, however, requires manually constructing these guard rails and enumerating whether each point can be both an enumerator and a recipient or a recipient only. For complex boundaries, such as are found in most police departments, this type of operation is extremely tedious and difficult.[9]

Figure 5.7:

"K" Statistic For 1996 Burglaries
Compared to Random and 2000 Population Distributions
L(t) = Sqrt[K(t)/pi] - t



Similarly, Ripley has proposed a simple weighting to account for the proportion of the circle placed over each point that is within the study area (Venables and Ripley, 1997). Thus, equation 5.17 is re-written as:

$$K(t_s) = \frac{A}{N^2} \sum_i \sum_j W_{ij}^{-1} I(t_{ij}) \quad (5.20)$$

where W_{ij}^{-1} is the inverse of the proportion of the circumference of a circle of radius, t_s, placed over each point that is within the total study area. Thus, if a point is near the study area border, it will receive a greater weight because a smaller proportion of the circle placed over it will be within the study area. An alternative weighting scheme can be found in Marcon and Puech (2003).

In *CrimeStat*, two possible corrections are conducted. One assumes that the study area is a rectangle while the other assumes that it is a circle.

Rectangular correction

In the rectangular correction for Ripley's K, the search circle radius, R_j, is compared to the edge of an assumed rectangle with area, A, centered at the mean center. First, the area to be analyzed is defined. If the user has specified a study area on the measurement parameters page, then that value for A is taken. The maximum bounding rectangle is taken (i.e., rectangle defined by the minimum and maximum X/Y values) and proportionately re-scaled so that the area of the rectangle is equal to A. If the user does not specify an area on the measurement parameters page, then the bounding rectangle defined by the minimum and maximum X/Y values is taken for A.

Second, for each point, the minimum distance to the nearest edge of this rectangle is calculated in both the horizontal and vertical directions, $d(\min R_X)$ and $d(\min R_Y)$. Third, each of the minimum distances is compared to the search circle radius, R_j.

1. If neither the minimum distance in the X-direction - $d(\min R_X)$, nor the minimum distance in the Y-direction - $d(\min R_Y)$, are less than the search circle radius, R_j, then the circle falls entirely within the rectangle and E = 1;

2. If either the minimum distance in the X-direction - $d(\min R_X)$, or the minimum distance in the Y-direction - $d(\min R_Y)$, but NOT BOTH, are less than the search circle radius, R_j, then part of the search circle falls outside the rectangle and an adjustment is necessary. An approximate adjustment is made than is inversely proportional to the area of the search circle within the rectangle. The values of E will vary between 1 and 2 since up to one-half of the search circle could fall outside the rectangle;

3. If both the minimum distance in the X-direction - $d(\min R_X)$, and the minimum distance in the Y-direction - $d(\min R_Y)$, are less than the search circle radius, R_j,

then a greater adjustment is required since E could vary between 1 and 4 since up to three-fourth of the search circle could fall outside the rectangle.

The formulas used to calculate the rectangular weights are:

Radius does not extend beyond the rectangle:

$$W_{ij}^{-1} = k = 1 \tag{5.21}$$

Radius extends beyond one edge of the rectangle (but not two):

$$W_{ij}^{-1} = k = \{ \frac{2\pi}{\{2\pi - 2\text{Cos-1}[d(\text{min}R)/Ri]\}} \} \tag{5.22}$$

Radius extends beyond two edges of the rectangle:

$$W_{ij}^{-1} = k = \{ \frac{2\pi}{\{1.5\pi - \text{Cos-1}[d(\text{min}Rx)/Ri] - \text{Cos-1}[d(\text{min}Ry)/Ri]\}} \} \tag{5.23}$$

While intuitive, this weight, W_{ij}^{-1}, is prone to cause upward 'drift' in the K function, so a log transformation is used:

$$W'^{-1}_{ij} = \ln(W_{ij}^{-1}) + 1 \tag{5.24}$$

This has the effect of tempering the drift somewhat.[10]

Circular correction

In the circular correction for Ripley's K, the search circle radius, R_j, is compared to the edge of an assumed circle with area, A, centered at the mean center. First, the area to be analyzed is defined. If the user has specified a study area on the measurement parameters page, then that value for a is taken. The radius of the circle, R_j, is calculated by equation 5.8 above. If the user has not specified a study area on the measurement parameters page, then A is calculated from the maximum bounding rectangle and the radius of the circle is calculated by equation 5.8 above.

Second, for each point, the distance from that point to the mean center, R_j, is calculated. The nearest distance from the point to the circle's edge is given by

$$R_{jC} = R - R_j \tag{5.25}$$

Third, the search circle radius, R_j, is compared to the nearest edge of the circle, R_{iC}, and the weight will vary from 1 (point and radius totally within the study area) to 2.3834 (point is located exactly on boundary of area circle). The formulas for the circular correction are:

$$\theta = \text{Cos-1} \{(r^2 + t_c^2 - R^2) / [\, 2*r*t_c \,]\} \quad (5.26)$$

$$W_{ij}^{-1} = k = \pi / \theta \quad (5.27)$$

where *r* is the radius of the search circle, R is the radius of the circular study area, and t_c is the distance from the point to the center of the circular study area.

For either correction

During the calculation of Ripley's K, each point is multiplied by E (aside from W or I) and the K and L statistics are calculated as before (see chapter 5). The simulation of random point distributions is treated in an analogous way.

While intuitive, this weight, W_{ij}^{-1}, is prone to cause upward 'drift' in the K function, so a log transformation is used:

$$W'^{-1}_{ij} = \ln(W_{ij}^{-1}) + 1 \quad (5.28)$$

This has the effect of tempering the drift somewhat.

Figure 5.8 below shows a Ripley's K distribution for 1996 Baltimore County burglaries, with and without edge corrections. As can be seen, the uncorrected L distribution decreases and falls below the theoretical random count (complete spatial randomness, L=0) after about 7 miles whereas neither the L distribution with the rectangular correction nor the L distribution with the circular distribution do so. As expected, the rectangular distribution produces the most concentration.

Output Intermediate Results

There is a box labeled "Output intermediate results". If checked, a separate dbf file will be output that lists the intermediate calculations. The file will be called "RipleyTempOutput.dbf". There are five output fields:

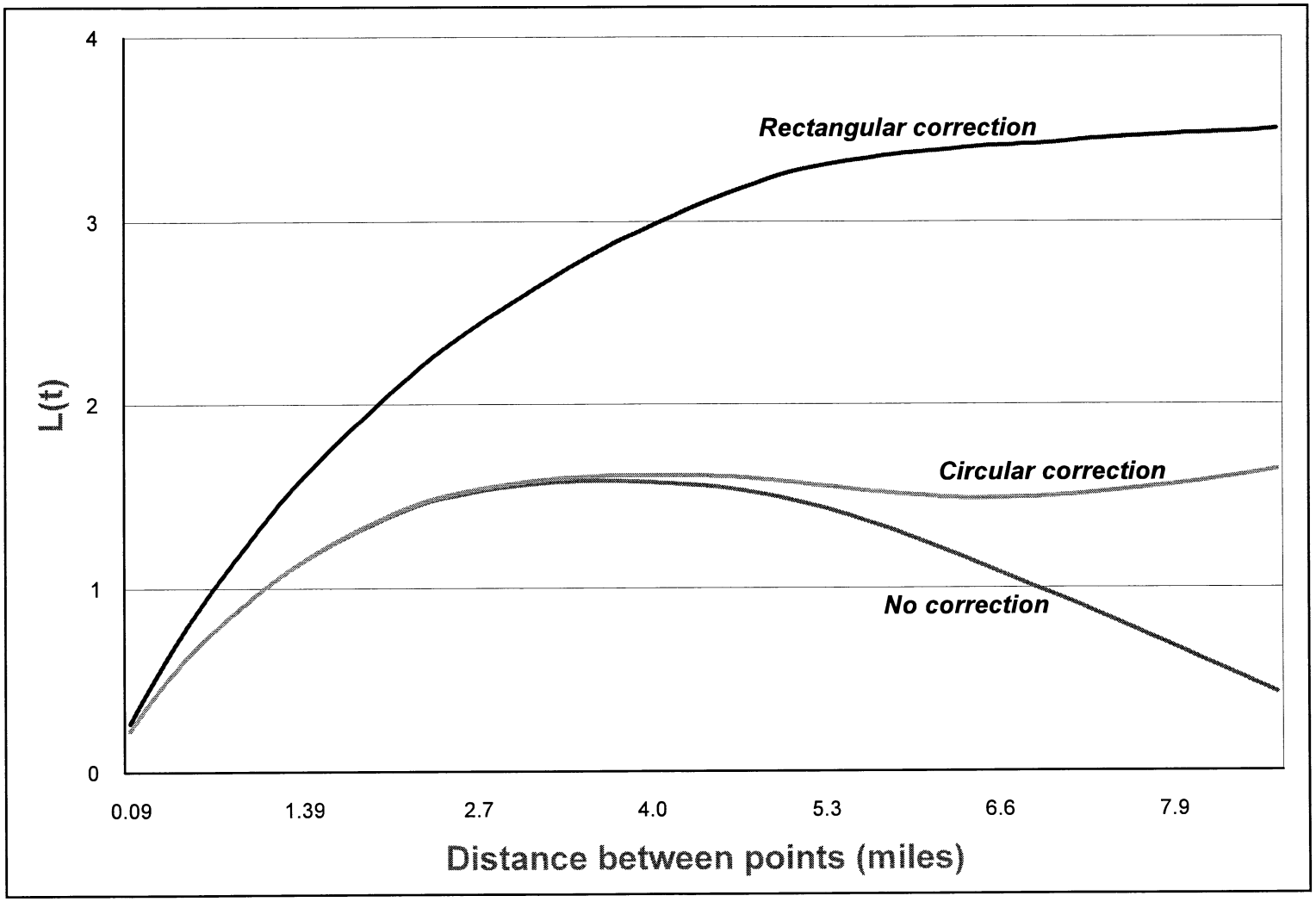
1. The point number (POINT), starting at 0 (for the first point) and proceeding to N–1 (for the Nth point)
2. The search radius in meters (SEARCHRADI)
3. The count of the number of *other* points that are within the search radius (COUNT)
4. The weight assigned, calculated from equations 5.24 or 5.28 above (WEIGHT)
5. The count times the weight (CTIMESW)

Figure 5.8:

"K" Statistic For 1996 Burglaries

With Different Types of Corrections

L(t) = Sqrt[K(t)/pi] - t

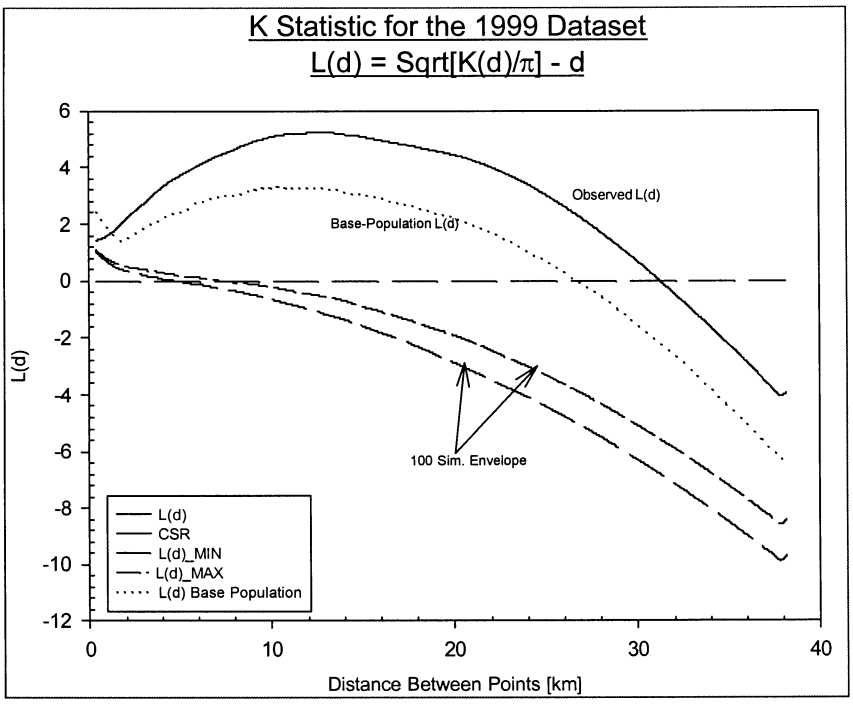


K-Function Analysis to Determine Clustering in the *Police Confrontations* Dataset in Buenos Aires Province, Argentina: 1999

Gastón Pezzuchi, Crime Analyst
Buenos Aires Province Police Force
Buenos Aires, Argentina

Sometimes crime analysts tend to produce beautiful hot spot maps without any formal evidence that clustering is indeed present in the data. One excellent and powerful tool that *CrimeStat* provides is the computation of the K function, which summarizes spatial dependence over a wide range of scales, and uses the information of all events.

We computed the K function using 1999 police confrontations data (mostly shootings) within our study area[1] and ran 100 Monte Carlo simulations in order to test for spatial randomness[2] (see figure below); the K function showed clustering up to about 30 Km. Yet, spatial randomness is not a particularly meaningful hypothesis to test considering that the "population at risk" are highly clustered. Hence we used police deployment data as a base population and calculated the K function for that data set. As can seen, the amount of clustering for the confrontation dataset is much greater than both the random envelope as well as the distribution of police officers.



[1] A years worth dataset of events occurring within a 9,500 km2 area around the Federal Capital (29 counties).

[2] Remember that Pr(L(d) > Lmax) = Pr(L(d) < Lmin) = 1 / (m + 1) where m is the number of independent simulations,

This output can be useful for examining the counts for specific points or for trying out alternative weighting schemes.

Some Cautions in Using Ripley's K

While Ripley's K is a powerful tool for analyzing spatial autocorrelation (usually clustering, rather than dispersion), like any statistic it is prone to biases. We've discussed edge biases above. But there are others. First, there is a sample size issue. The routine calculates 100 separate L(t) values, one for each distance bin. However, the precision of any one L(t) value is dependent on the sample size. With a small sample, there is insufficient data to estimate 100 independent values of L(t). While the Monte Carlo simulation partly can account for that bias, it has to be realized that the precision of the interpretation is suspect. For example, in comparing two similar distributions, say robberies and burglaries, unless the sample size is large differences for any one bin could easily be due to chance. One would need a very different type of procedure to estimate the 'standard error' of two functions with a small sample. But, I would suspect that there would be many bins for which they would be indistinguishable (shown as the two functions criss-crossing each other).

In previous versions of *CrimeStat*, there was a restriction of at least 100 data points to display the entire 100 L(t) estimates; otherwise, they were truncated. In this version, all 100 intervals are allowed for any size sample. However, there is a strict warning. Users should be very cautious in drawing conclusions about differences in the L function with small samples. Even with sample sizes greater than 100, the imprecision of any one L(t) value is considerable. Until the sample sizes get into the hundreds, precision is an issue for specific L(t) values.

A second caution has to do with the scale of the interpretation. Data sets with strong *first-order* properties (i.e., a high degree of central concentration of incidents) will exert bias on Ripley's K statistic. Thus, any data set that is correlated with human populations will most likely have a very strong 'central tendency'. Thus, there will be a high degree of concentration in the L values for even near distances. This was seen in the robbery and burglary data shown above. The K statistic was created to estimate *second-order* spatial autocorrelation, namely localized clustering. However, if the first-order effect is so dominant, then it's hard to disentangle it from a second-order effect. That is, it's often not clear whether the clustering that is observed in Ripley K is due to primary, first-order clustering or actual localized, second-order clustering. That's why it is generally wise to use the K statistic for short distance ranges and not for larger distance separations. For larger distance separations, it is almost impossible to tell whether the effect is due to the large central concentration of the population or whether there are interactions between neighborhoods at a large scale.

There are different ways to handle to problem, none of which are perfect. For example, one can estimate a first-order concentration effect and then apply Ripley's K to the residuals. Or, alternatively, one can use a baseline population to calculate a rate and test for concentration only in the rates, not the volumes of incidents. In chapters 6 and 8, there will be a discussion of using a baseline population to control for first-order effects. But, whether this

is done or not, the user should be aware of the interaction between first-order and second-order (or localized) effects.

The third caution has to do with the shape of the boundaries in interpreting the K statistic. This is particularly true when an edge correction is applied. Unless the study area was an actual rectangle, the correction may alter the interpretation compared to the uncorrected L. There are some subtle differences between the two, however, so some care should be used. The empirical L is obtained from the points within the study area, the geography of which is usually irregular. The random L, however, is calculated from a rectangle or a circle. Thus, the differences in the shape comparisons may account for some variations.

The realism of the corrected function depends on the validity of the underlying assumptions. If it is likely that there are points outside the study area, then a weighting may produce a more realistic interpretation of the L function. On the other hand, if the density of the points outside the study area is lower (e.g., if the study area is a metropolitan area, then the area outside is more likely to be suburban or rural and of low population density), then the weighting will exaggerate the function relative to what it should be. In the extreme case, if the study area is an island (e.g., Honolulu), then there are no points outside the study area and no weighting is justified. Even when weighting would be justified, the actual boundary is probably not a rectangle or a square so that the geometric correction above may distort the L function, too. In short, some understanding of the basis for weighting is necessary to produce a reasonable L function.

Assign Primary Points to Secondary Points

This routine will assign each primary point to a secondary point and then will sum by the number of primary points assigned to each secondary point. The routine is useful for summarizing data. For example, if the primary file represents the number of robberies and the secondary file represents the centroids of census tracts, then the routine will assign all robberies to a census tract and will then sum the number of robberies in each census tract. The result is a count of the number of primary points for each secondary point (zone). Other examples might be to assign students to the nearest school or to assign patients to the nearest hospital. There are many uses for summarizing data by another data reference. In the Trip Generation module (under Crime Travel Demand - see chapter 13), a model is developed for the number of crimes originating in each zone and a separate model for the number of crimes ending in each zone. The "Assign primary points to secondary points" routine is a good way to summarize the number of crimes by zones.

There are two methods for assigning the primary points to the secondary.

Nearest neighbor assignment

This routine assigns each primary point to the secondary point to which it is closest. It goes through all the primary points and sums the number assigned to each secondary point.

Thus, the logical operation is 'nearest to'. If there are two or more secondary points that are exactly equal, the assignment goes to the first one on the list.

Point-in-polygon assignment

This routine assigns each primary point to the secondary point for which it falls within its polygon (zone). The point-in-polygon assignment reads a zonal boundary file (in ArcView 'shp' format) and determines which zone each primary point falls within. In this case, the logical operation is 'belongs to'. A zone (polygon) shape file must be provided and the routine checks which secondary zone each primary point falls within.

Most GIS packages can do a point-in-polygon operation but few allow a nearest neighbor assignment. In general, the two are similar though there will be differences due to the irregular shape of zone boundaries. For example, figure 5.9 below shows an incident that is within Traffic Analysis Zone (TAZ) 0546, but is actually closer to the centroid of TAZ 0547. The characteristics associated with this incident are more likely to be associated with the characteristics of the second zone than the zone to which it belongs. The decision on which criteria to use in assigning the incident to a zone depends on how integral is the zone to which it belongs. If the zones are bounded by major arterials, then travel behavior within the zone with be defined by those arterials; in this case, it would probably be prudent to use the point-in-polygon assignment. On the other hand, if the zone boundaries are not a fundamental separation, then the nearest neighbor assignment would probably produce a better fit to the incident since the characteristics of the closer zone are liable to hold for the incident. In short, the user must decide on which theoretical basis to assign points.

Zone file

A zonal file must be provided. This is a polygon file that defines the zones to which the primary points are assigned. The zone file should be the same as the secondary file (see Secondary file). For each point in the primary file, the routine identifies which polygon (zone) it belongs to and then sums the number of points per polygon.

Name of assigned variable

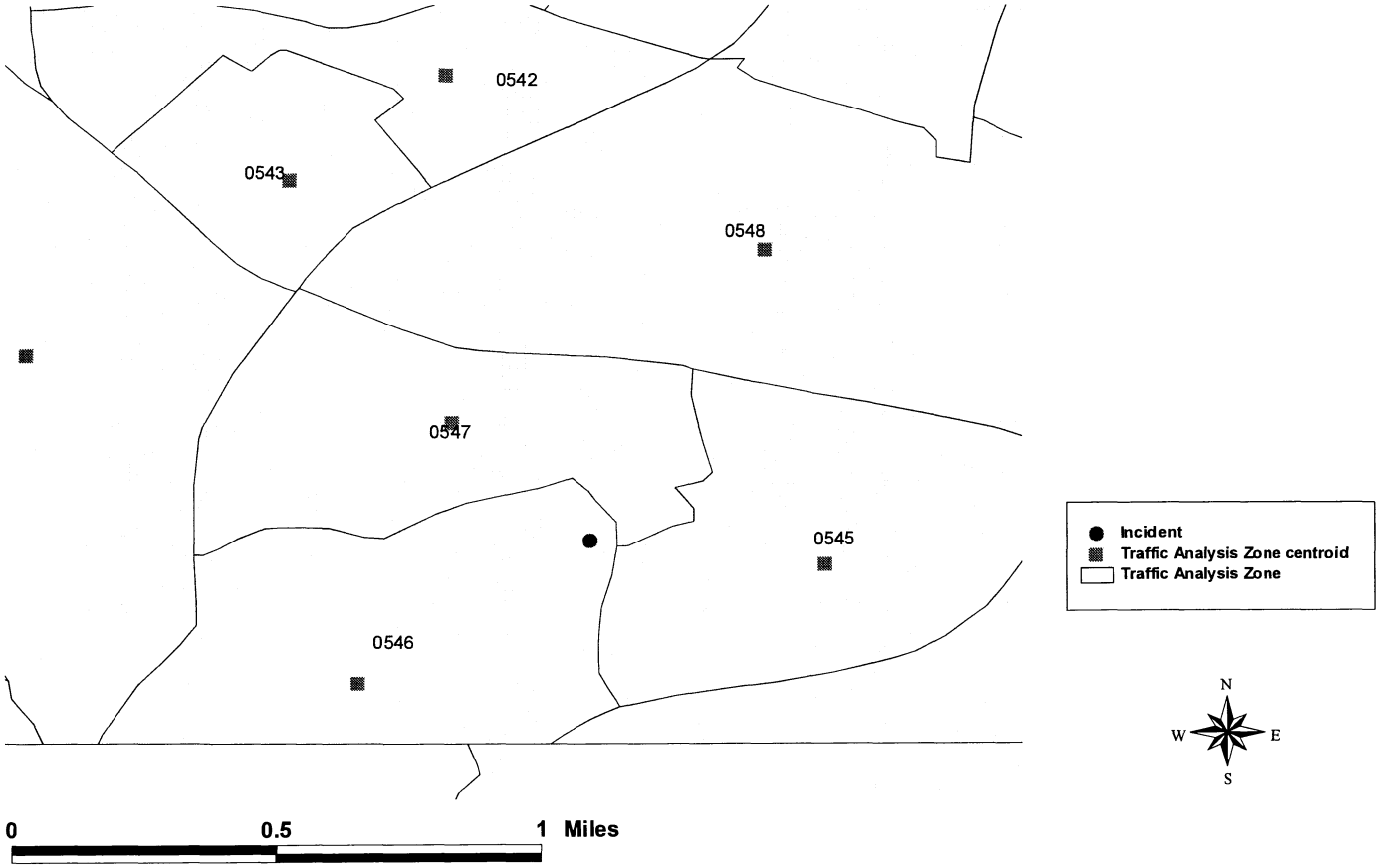
Specify the name of the summed variable. The default name is FREQ.

Use weighting file

The primary file records can be weighted by another file. This would be useful for correcting the totals from the primary file. For example, if the primary file were robbery incidents from an arrest record, the sum of this variable (i.e. the total number of robberies) may produce a biased distribution over the secondary file zones because the primary file was not a random sample of all incidents (e.g., if it came from an arrest record where the distribution of robbery arrests is not the same as the distribution of all robbery incidents).

Figure 5.9:

Incident Assignment
Point in Relation to Traffic Analysis Zone Boundaries and Centroids



The secondary file or another file can be used to adjust the summed total. The weighting variable should have a field that identifies the ratio of the true to the measured count for each zone. A value of 1 indicates that the summed value for a zone is equal to the true value; hence no adjustment is needed. A value greater than 1 indicates that the summed value needs to be adjusted upward to equal the true value. A value less than 1 indicates that the summed value needs to be adjusted downward to equal the true value.

If another file is to be used for weighting, indicate whether it is the secondary file or, if another file, the name of the other file.

Name of assigned weighted variable

For a weighted sum, specify the name of the variable. The default will be ADJFREQ.

Save result to

For both routines, the output is a 'dbf' file. Define the file name. Note: be careful about using the same name as the secondary file as the saved file will have the new variable. It is best to give it a new name.

A new variable will be added to this file that gives the number of primary points in each secondary file zone and, if weighting is used, a secondary variable will be added which has the adjusted frequency.

Example: Assigning Robberies to Zones

To illustrate the routine, table 5.4 shows the results of summarizing 1181 1997 robberies that occurred in Baltimore to 325 Traffic Analysis Zones. The two methods are compared. Only the first 30 assignments are shown. In general, they give similar results. However, there are differences due to the method. One is that the nearest neighbor method will assign points on the basis of proximity while the point-in-polygon method will not. In the case of the Baltimore County robberies, some of these were assigned to a City of Baltimore TAZ because those TAZ's were closer, rather than to a Baltimore County TAZ. Another is that if a zone is very irregular, points may be assigned to it under the point-in-polygon method which may be quite far away.

Thus, the user has to decide which method makes the most sense. If the purpose is to assign incidents to the zone which it is most likely to be related, for example, when developing a data set for zonal modeling (see chapters 12 and 13), then the nearest neighbor method may produce a better representation. The incidents are then assigned to a zone which has characteristics that probably will be related to the factors causing the incidents in the first place. On the other hand, if the object is to assign incidents on the basis of membership (e.g., assigning crimes to police precincts), then the point-in-polygon method will be the most accurate.

Table 5.4

Assigning Incidents to Zones
1181 1997 Robberies and 325 Traffic Analysis Zones

TAZ	Point-in-Polygon	Nearest Neighbor
0401	0	0
0402	0	0
0403	1	1
0404	0	0
0405	0	0
0406	0	0
0407	0	0
0408	0	0
0409	0	0
0410	0	0
0411	0	0
0412	0	0
0413	0	0
0414	1	1
0415	0	0
0416	0	0
0417	0	0
0418	0	0
0419	0	0
0420	0	0
0421	0	0
0422	0	1
0423	0	0
0424	1	0
0425	3	0
0426	2	2
0427	3	2
0428	0	0
0429	5	5
0430	0	0

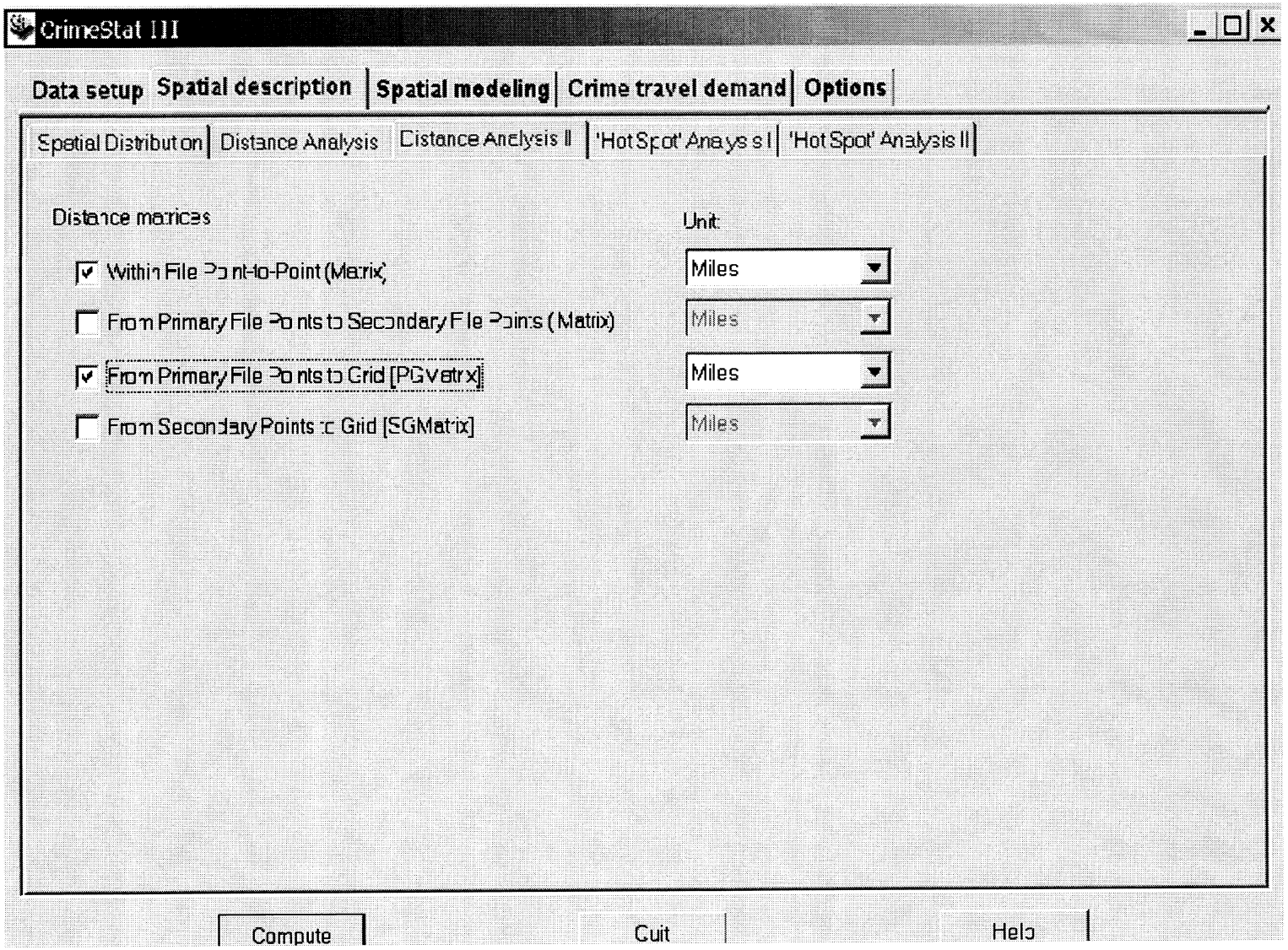
Distance Analysis II

The remaining distance analysis routines are on the Distance Analysis II page. Figure 5.10 shows the page.

Distance Matrices

CrimeStat has the capability for outputting distance matrices. There are four types of matrices that can be output.

Figure 5.10: Distance Analysis II Screen



1. First, the distance between every point in the primary file and every other point can be calculated in miles, nautical miles, feet, kilometers or meters. This is called the *Within File Point-to-Point* matrix (Matrix).

2. Second, if there is also a secondary file, *CrimeStat* can calculate the distance from every point in the primary file to every point in the secondary file, again in miles, nautical miles, feet, kilometers or meters. This is called the *From Primary File Points to Secondary File Points* matrix (Imatrix).

3. Third, if there is a reference file defined, the distance from each primary point to each grid cell can be computed. This is called the *From Primary File Points to Grid* matrix (PGMatrix).

4. Fourth, if there is also a secondary file and a reference file, the distance from each secondary point to each grid cell can be computed. This is called the *From Secondary File Points to Grid* matrix (SGMatrix).

Each of these types of matrices can be displayed or saved to an Ascii text file for import into another program. Each matrix defines incidents by the order in which they occur in the files (i.e., Record number 1 is listed as '1'; record number 2 is listed '2'; and so forth). Only a subset of each matrix is displayed on the results tab. However, there are horizontal and vertical slider bars that allow the user to scroll through the matrix. The user should move the vertical slide bar first to an approximate proportion of the matrix and click the *Go* button. The matrix will scroll through the rows of the matrix to a place which represents that proportion indicated in the slide bar. The user can then scroll across the rows with the upper slide bar.

The matrices can be used for various purposes. The *within file point-to-point matrix* can be used to examine distances between particular incidents. The *saved Ascii '.txt' matrix* can also be imported into a network program for estimating transportation routes. The *primary-to-secondary file matrix* can be used in optimization routines, for example in trying to assess optimal allocation of police cars in order to minimize response time in a police district. The distances to the grid cells can be used to compare the distances for different distributions to a central location (e.g., a police station). There are many applications where distances are the primary unit of analysis. However, the user will need other software to read the files.

Be careful in outputting distances, though, because the files will generally be very large. For example, a primary file of 1000 incidents when interpolated to 9000 grid cells (100 columns x 90 rows) will produce 9 million paired comparisons. Such a file will take a lot of disk space. For that reason, we only allow output to an Ascii text file.

This concludes the discussion of second-order properties. The next two chapters will discuss the identification of 'hot spots' with *CrimeStat*.

Endnotes for Chapter 5

1. There is also a mean random distance for a dispersed pattern, called the *mean dispersed distance* (Ebdon, 1988). It is defined as

$$d(dis) = \frac{SQRT[2]}{3^{1/4}\ SQRT[\ N/A\ ]}$$

A nearest neighbor index can be set up comparing the observed mean neighbor distance with that expected for a dispersed pattern. *CrimeStat* only provides the traditional nearest neighbor index, but it does output the mean dispersed distance.

2. Unfortunately, the term *order* when used in the context of nearest neighbor analysis has a slightly different meaning than when used as *first-order* compared to *second-order* statistics. In the nearest neighbor context, *order* really means *neighbor* whereas in the type of statistics context, *order* means the scale of the statistics, global or local. The use of the terms is historical.

3. It might be possible to test with a Monte Carlo simulation. That is, two separate random samples of 1181 'robberies' and 6051 'burglaries' respectively would be drawn. The nearest neighbor distance for each of these samples would be calculated and the ratio of the two would be taken. This experiment would be repeated many times (e.g., 1000 or more) to yield an approximate 95% confidence interval of the ratio.

4. There is not a hard-and-fast rule about how many K-order nearest neighbor distances may be calculated. Cressie (1991, p. 613) shows that error increases with increasing order and the degree of divergence from an edge-corrected measure increases over time. In a test case of 584 point locations, he shows that even after only 25 nearest neighbors, the uncorrected measure yields opposite conclusions about clustering from the corrected measures. So, as a rough approximation, orders no greater than 2.5% of the cases should be calculated.

5. Because *CrimeStat* uses indirect distance for the linear nearest neighbor index (i.e. measurement only in an horizontal or vertical direction), there is a slight distortion that can occur if the incidents are distributed in a diagonal manner, such as with State Highways 26 and 150 in Figure 5.4. The distortion is very small, however. For example, with the incidents along State Highway 26, after rotating the incident points so that they fell approximately in a horizontal orientation, the observed average linear nearest neighbor distance decreased slightly from 0.05843 miles to 0.05061 miles and the linear nearest neighbor index became 0.8354 (t=-.91; not significant). In other words, the effects of the diagonal distribution lengthened the estimate for the average linear nearest neighbor distance by about 41 feet compared to the actual distances between incidents. For a small sample size, this could be relevant, but for a larger sample it generally will be a small distortion. However, if

a more precise measure is required, then the user should rotation the distribution so that the incidents have as closely as possible a horizontal or vertical orientation. An alternative is to calculate the regular nearest neighbor distance but use a network for distance calculations (see chapter 3).

6. This form of the $L(t_s)$ is taken from Cressie (1991). In Ripley's original formulation (Ripley, 1976), distance is not subtracted from the square root function. The advantage of the Cressie formulation is that a complete random distribution will be a straight line that is parallel to the X-axis.

7. In earlier versions of *CrimeStat*, the distance was half the side of an assumed square. It has been reduced in *CrimeStat III* to emphasize the near distances to points. The statistic doesn't make much sense over a larger study region.

8. Note, that since there is not a formal test of significance, the comparison with an envelope produced from a number of simulations provides only approximate confidence about whether the distribution differs from chance or not. That is, one cannot say that the likelihood of obtaining this result by chance is less than 5%, for example.

9. The 'guard rail' concept, while frequently used, is poor methodology because it involves ignoring data near the boundary of a study area. That is, points within the guard rail are only allowed to be selected by other points and not, in turn, be allowed to select others. This has the effect of throwing out data that could be very important. It is analogous to the old, but fortunately now discarded, practice of throwing out 'outliers' in regression analysis because the outliers were somehow seen as 'not typical'. The guard rail concept is also poor policing practice since incidents occurring near a border may be very important to a police department and may require coordination with an adjacent jurisdiction. In short, use mathematical adjustments for edge corrections or, failing that, leave the data as it is.

10. The use of a log function for the weight is different than in previous versions of *CrimeStat*, both for the rectangular and circular corrections.

Chapter 6
'Hot Spot' Analysis I

In this and the next chapter, we describe seven tools for identifying clusters of crime incidents. The discussion has been divided into two chapters primarily because of the length of the discussion. This chapter discusses the concept of a hot spot and four hot spot techniques: the mode, fuzzy mode, nearest neighbor hierarchical clustering, and risk-adjusted nearest neighbor hierarchical clustering. The next chapter discusses STAC, the K-means algorithm, and Anselin's Local Moran statistics. However, the seven techniques should be seen as a continuum of approaches towards identifying hot spots.

Hot Spots

Typically called hot spots or hot spot areas, these are concentrations of incidents within a limited geographical area that appear over time. Police have learned from experience that there are particular environments that attract drug trading and crimes in larger-than-expected concentrations, so-called crime generators. Sometimes these hot spot areas are defined by particular activities (e.g., drug trading; Weisburd and Green, 1995; Sherman, Gartin and Buerger, 1989; Maltz, Gordon, and Friedman, 1989), other times by specific concentrations of land uses (e.g., skid row areas, bars, adult bookshops, itinerant hotels), and sometimes by interactions between activities and land uses, such as thefts at transit stations or bus stops (Block and Block, 1995; Levine, Wachs and Shirazi, 1986). Whatever the reasons for the concentration, they are real and are known by most police departments.

While there are some theoretical concerns about what links disparate crime incidents together into a cluster, nonetheless, the concept is very useful. Police officers patrolling a precinct can focus their attention on particular environments because they know that crime incidents will continually reappear in these places. Crime prevention units can target their efforts knowing that they will achieve a positive effect in reducing crime with limited resources (Sherman and Weisburd, 1995). In short, the concept is very useful.

Nevertheless, the concept is a perceptual construct. 'Hot spots' may not exist in reality, but could be areas where there is sufficient concentration of certain activities (in this case, crime incidents) such that they get labeled as being an area of high concentration. There is not a boundary around these incidents, but a gradient where people draw an imaginary line to indicate the location at which the hot spot starts. In reality, any variable that is measured, such as the density of crime incidents, will be continuous over an area, being higher in some parts and lower in others. Where a line is drawn in order to define a hot spot is somewhat arbitrary.

Statistical Approaches to the Measurement of 'Hot Spots'

Unfortunately, measuring a hot spot is also a complicated problem. There are literally dozens of different statistical techniques designed to identify 'hot spots' (Everitt,

1974). Many, but not all, of the techniques are typically known under the general statistical label of cluster analysis. These are statistical techniques aimed at grouping cases together into relatively coherent clusters. All of the techniques depend on optimizing various statistical criteria, but the techniques differ among themselves in their methodology as well as in the criteria used for identification. Because 'hot spots' are perceptual constructs, any technique that is used must approximate how someone would perceive an area. The techniques do this through various mathematical criteria.

Types of Cluster Analysis (Hot Spot) Methods

Several typologies of cluster analysis have been developed as cluster routines typically fall into several general categories (Everitt, 1974; Çan and Megbolugbe, 1996):

1. Point locations. This is the most intuitive type of cluster involving the number of incidents occurring at different locations. Locations with the most number of incidents are defined as 'hot spots'. CrimeStat includes two point location techniques: the Mode and Fuzzy Mode;

2. Hierarchical techniques (Sneath, 1957; McQuitty, 1960; Sokal and Sneath, 1963; King, 1967; Sokal and Michener, 1958; Ward, 1963; Hartigan, 1975) are like an inverted tree diagram in which two or more incidents are first grouped on the basis of some criteria (e.g., nearest neighbor). Then, the pairs are grouped into second-order clusters. The second-order clusters are then grouped into third-order clusters, and this process is repeated until either all incidents fall into a single cluster or else the grouping criteria fails. Thus, there is a hierarchy of clusters that can be displayed with a dendogram (an inverted tree diagram).

   Figure 6.1 shows an example of a hierarchical clustering where there are four orders (levels) of clustering; the visualization is non-spatial in order to show the linkages. In this example, all individual incidents are grouped into first-order clusters which, in turn, are grouped into second-order clusters which, in turn, are grouped into third-order clusters which all converge into a single fourth-order cluster. Many hierarchical techniques, however, do not group all incidents or all clusters into the next highest level. CrimeStat includes two hierarchical techniques: a Nearest Neighbor Hierarchical Clustering routine in this chapter and the Spatial and Temporal Analysis of Crime module (STAC) which will be discussed in chapter 7;

3. Partitioning techniques, frequently called the K-means technique, partition the incidents into a specified number of groupings, usually defined by the user (Thorndike, 1953; MacQueen, 1967; Ball and Hall, 1970; Beale, 1969). Thus, all points are assigned to one, and only one, group. Figure 6.2 shows a partitioning technique where all points are assigned to clusters and are displayed as ellipses. CrimeStat includes one partitioning technique, a K-means partitioning technique;

Figure 6.1:

Hierarchical Clustering Technique

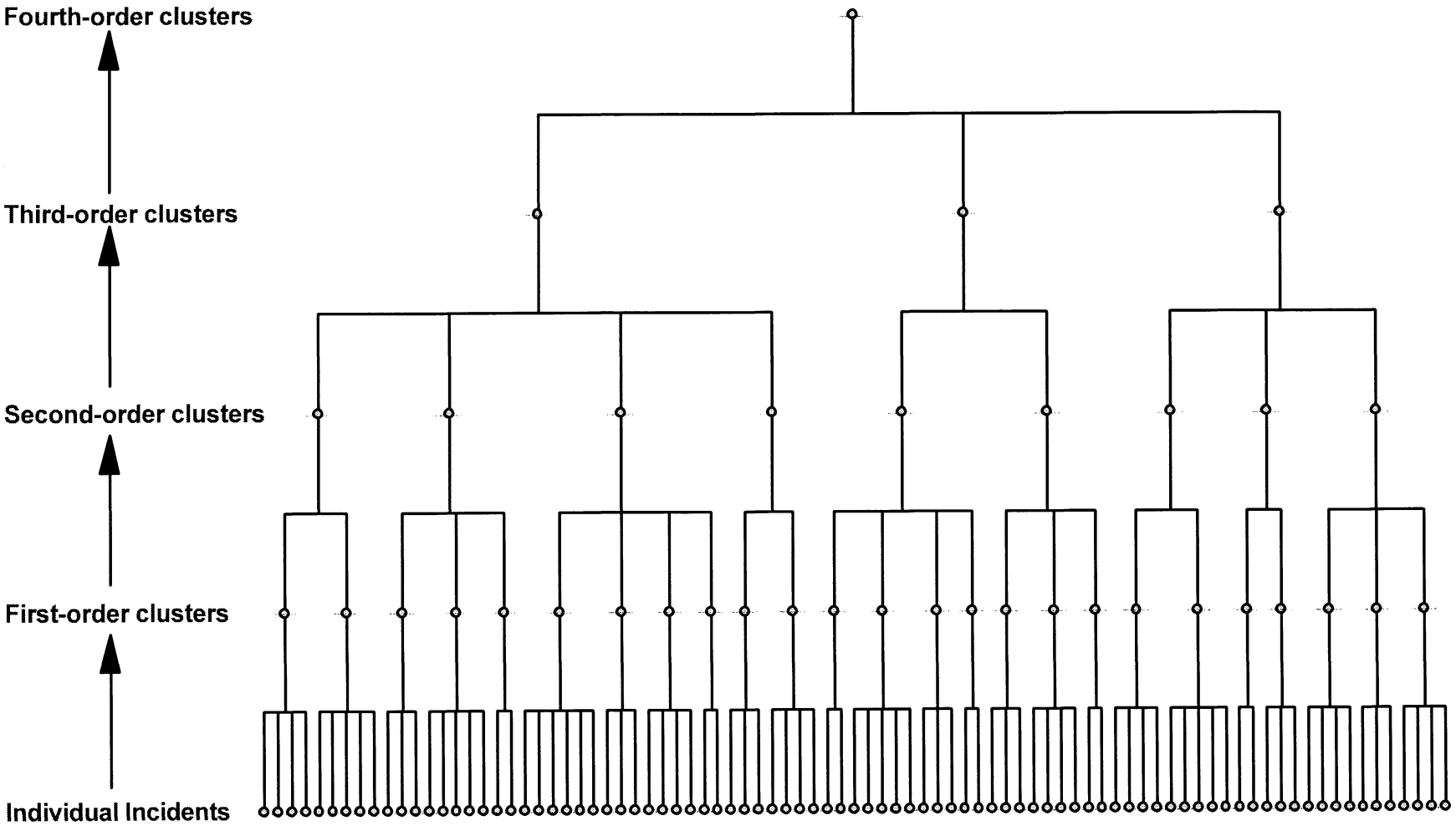
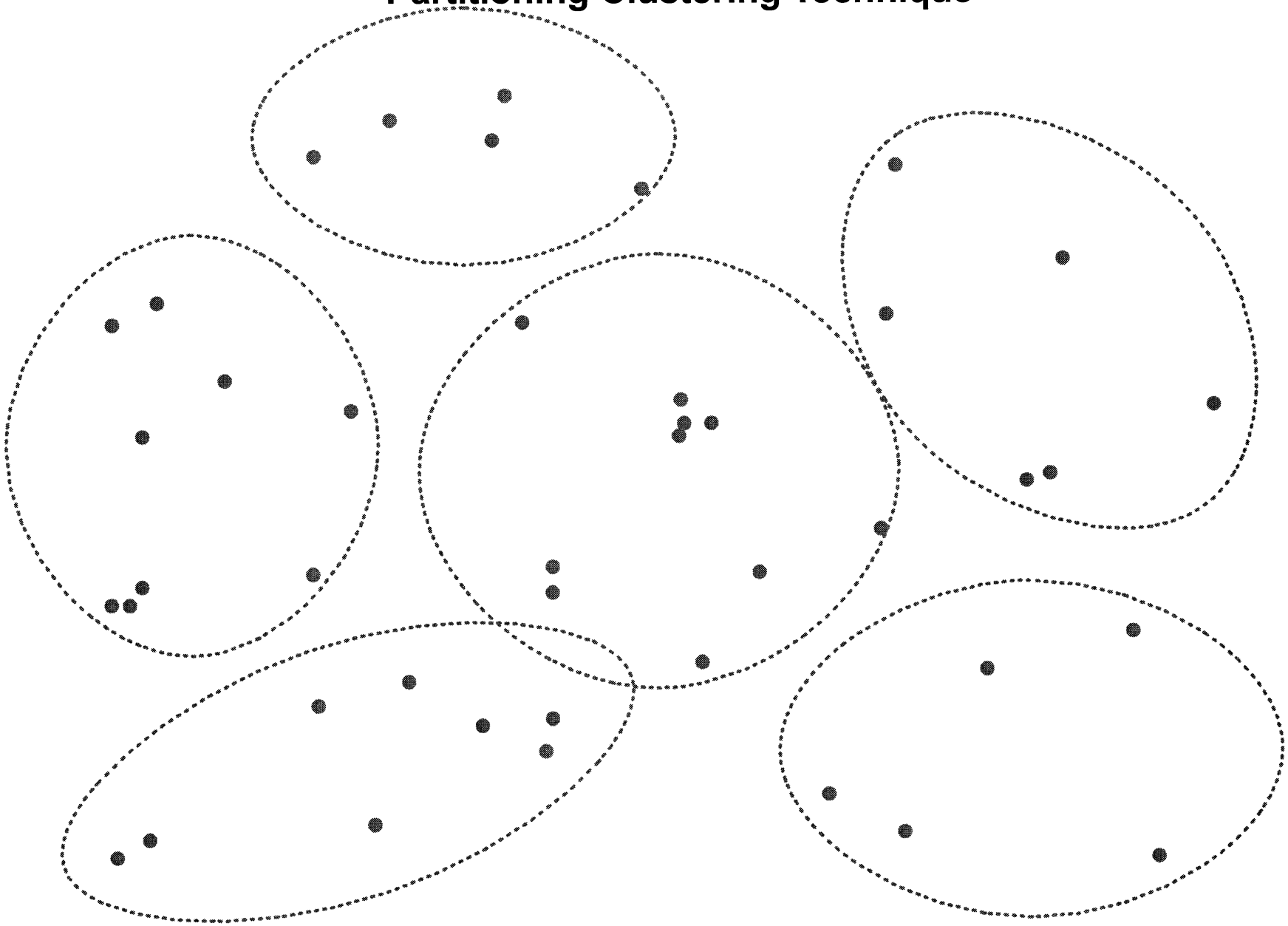


Figure 6.2:

Partitioning Clustering Technique



4. Density techniques identify clusters by searching for dense concentrations of incidents (Carmichael et al, 1968; Gitman and Levine, 1970; Cattell and Coulter, 1966; Wishart, 1969). CrimeStat has one density search algorithm using a Single Kernel Density method; this is discussed in chapter 8;

5. Clumping techniques involve the partitioning of incidents into groups or clusters, but allow overlapping membership (Jones and Jackson, 1967; Needham, 1967; Jardine and Sibson, 1968; Cole and Wishart, 1970);

6. Risk-based techniques identify clusters in relation to an underlying base 'at risk' variable, such as population, employment, or active targets (Jefferis, 1998; Kulldorff, 1997; Kulldorff and Nagarwalla, 1995). CrimeStat includes two risk-based technique - a Risk-adjusted Nearest Neighbor Hierarchical Clustering routine, discussed in this chapter, and a Duel Kernel Density method, discussed in chapter 8; and

7. Miscellaneous techniques are other methods that are less commonly used including techniques applied to zones, not incidents. CrimeStat includes Anselin's Local Moran technique for identifying neighborhood discrepancies (Anselin, 1995).

There are also hybrids between these methods. For example, the Risk-adjusted Nearest Neighbor Hierarchical Clustering routine is primarily a risk-based technique but involves elements of clumping while STAC is primarily a partitioning method but with elements of hierarchical grouping (Block and Green, 1994).

Optimization Criteria

In addition to the different types of cluster analysis, there are different criteria that distinguish techniques applied to space. Among these are:

1. The definition of a cluster - whether it is a discrete grouping or a continuous variable; whether points must belong to a cluster or whether they can be isolated; whether points can belong to multiple clusters.

2. The choice of variables in addition to the X and Y coordinates - whether weighting or intensity values are used to define similarities.

3. The measurement of similarity and distance - the type of geometry being used; whether clusters are defined by closeness or not; the types of similarity measures used.

4. The number of clusters - whether there are a fixed or variable number of clusters; whether users can define the number or not.

5. The geographical scale of the clusters - whether clusters are defined by small or larger areas; for hierarchical techniques, what level of abstraction is considered optimal.

6. The initial selection of cluster locations ('seeds') - whether they are mathematically or user defined; the specific rules used to define the initial seeds.

7. The optimization routines used to adjust the initial seeds into final locations - whether distance is being minimized or maximized; the specific algorithms used to readjust seed locations.

8. The visual display of the clusters, once extracted - whether drawn by hand or by a geometrical object (e.g., an ellipse, a convex hull); the proportion of cases represented in the visualization.

This is not the place to provide a comprehensive review of cluster techniques. Nevertheless, it should be clear that with the several types of cluster analysis and with the many criteria that can be used for any particular technique provides a large number of different techniques that could be applied to an incident data base. It should be realized that there is not a single solution to the identification of hot spots, but that different techniques will reveal different groupings and patterns among the groups. A user must be aware of this variability and must choose techniques that can complement other types of analysis. It would be very naive to expect that a single technique can reveal the existence of hot spots in a jurisdiction which are unequivocally clear. In most cases analysts are not even sure why there are hot spots in the first place and, until that is solved, it would be unreasonable to expect a mathematical or statistical routine to solve that problem.

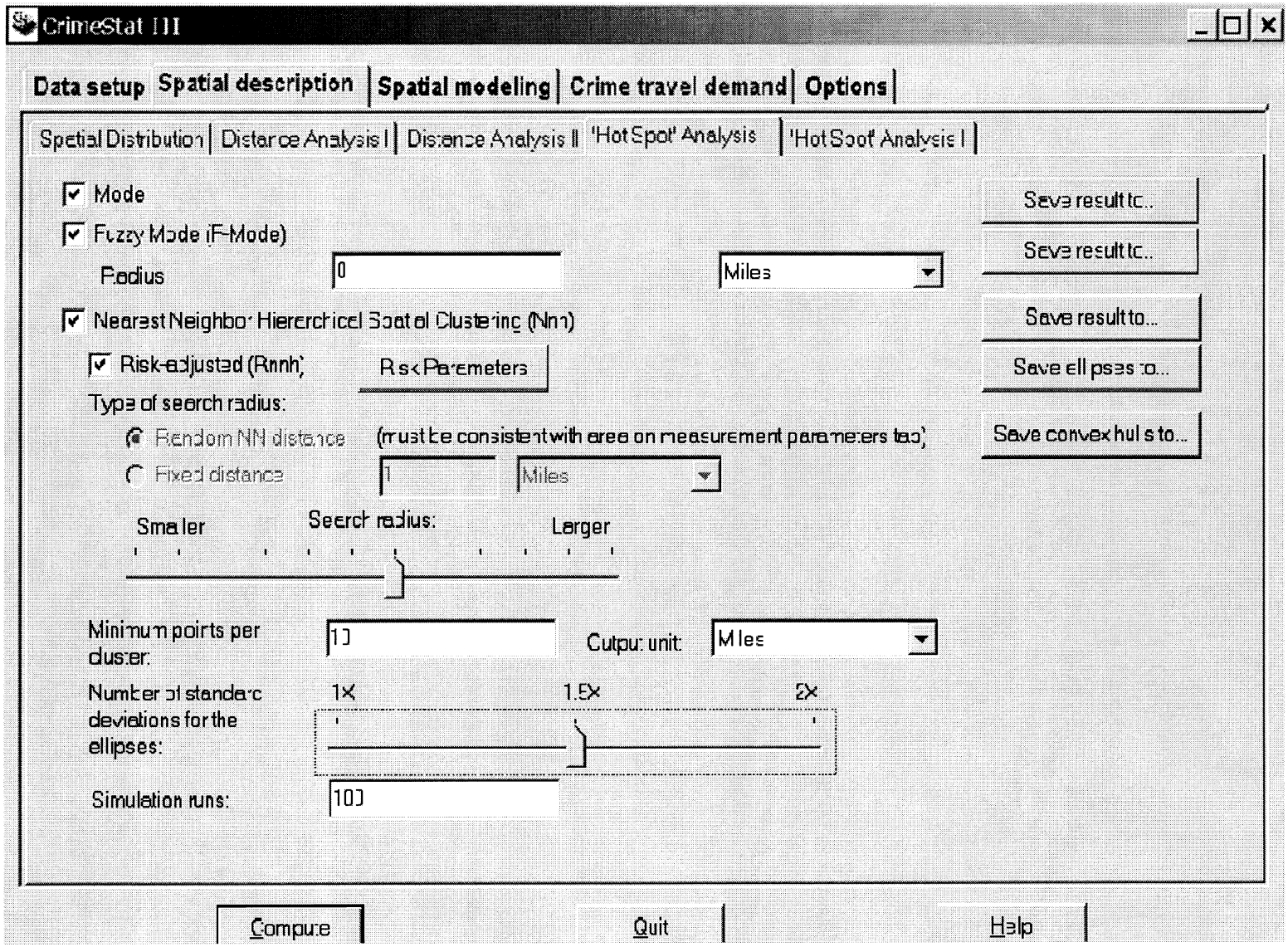
Cluster Routines in CrimeStat

Because of the variety of cluster techniques, CrimeStat includes seven techniques that cover the range of techniques that have been used:

1. The Mode
2. The Fuzzy Mode
3. Nearest neighbor hierarchical clustering
4. Risk-adjusted nearest neighbor hierarchical clustering
5. The Spatial and Temporal Analysis of Crime (STAC) module
6. K-means clustering
7. Anselin's Local Moran statistic

These are not the only techniques, of course, and analysts should use them as complements to other types of analysis. Because of the number of routines, these routines have been allocated to two different setup tabs in CrimeStat called 'Hot Spot' Analysis I and 'Hot Spot' Analysis II. However, they should be seen as one collection of similar techniques. This chapter will discuss the first four of these. Figure 6.3 shows the 'Hot Spot' Analysis I page.

Figure 6.3: 'Hot Spot' Analysis I Screen



Mode

The mode is the most intuitive type of hot spot. It is the location with the largest number of incidents. The CrimeStat Mode routine calculates the frequency of incidents occurring at each unique location (a point with a unique X and Y coordinate), sorts the list, and outputs the results in rank order from the most frequent to the least frequent.

Only locations that are represented in the primary file are identified. The routine outputs a 'dbf' file that includes four variables:

1. The rank order of the location with 1 being the location with the most incidents, 2 being the location with the next most incidents, 3 being the location with the third most incidents, and so forth until those locations that have only one incident each;

2. The frequency of incidents at the location. This is the number of incidents occurring at that location;

3. The X coordinate of the location; and

4. The Y coordinate of the location.

To illustrate, table 6.1 presents the formatted output for the ten most frequent locations for motor vehicle thefts in the Baltimore region in 1996 (the rest were ignored) and figure 6.4 maps the ten locations.[1] The map displays the locations with a round symbol, the size of which is proportional the number of incidents. Also, the number of incidents at the location is displayed. These vary from a high of 43 vehicle thefts at location number 1 to a low of 15 vehicle thefts at location number 10. In order to know what these locations represent, the user will have to overlay other GIS layers over the points. In the example, of the ten locations, eight are at shopping centers, one is the parking lot of a train station, and one is the parking lot of a large organization.

The mode is a very simple measure, but one that can be very useful. In the example, it's clear that most vehicle thefts occur at institutional settings, where there are a collection of parked vehicles. In the case of the shopping centers, the Baltimore County Police Department are aware of the number of vehicles stolen at these locations and work with the shopping center management offices to try to reduce the thefts. It also turns out that shopping centers are the most frequent locations for stolen vehicle retrievals, so it works both ways.

Fuzzy Mode

The usefulness of the mode, however, is dependent on the degree of resolution for the geo-referencing of incidents. In the case of the Baltimore vehicle thefts, thefts locations

Figure 6.4:

Ten Most Frequent Locations for Motor Vehicle Theft

Symbol Area Proportional to Frequency

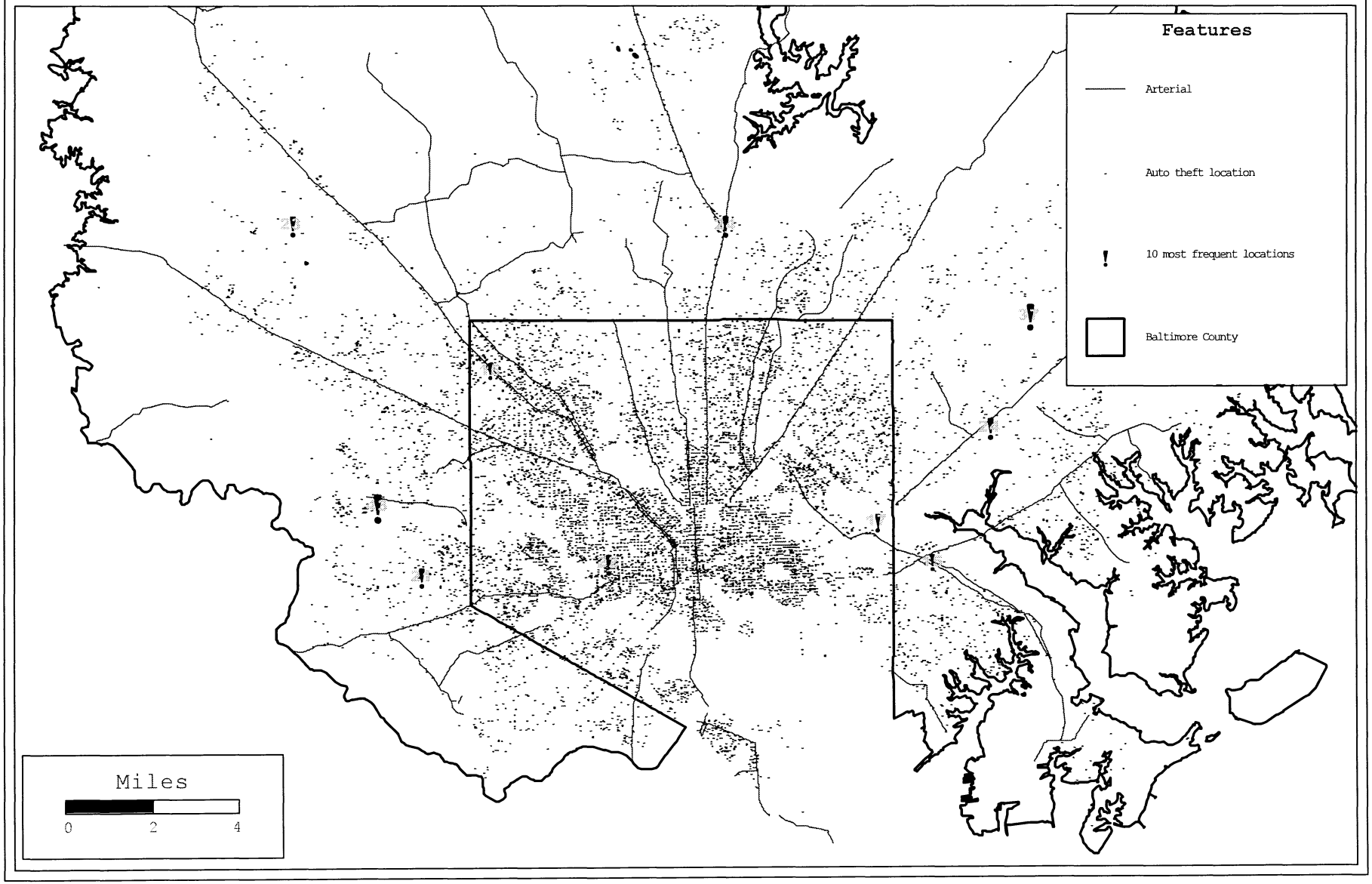


Table 6.1

Mode Output for
Most Frequent Locations for Motor Vehicle Thefts
Baltimore: 1990

Mode:

Sample size.....: 14853
Measurement type.....: Direct
Start time.....: 12:46:15 PM, 07/15/2001
End time.....: 12:50:19 PM, 07/15/2001

Displaying 45 results(s) starting from 1 (ONLY 10 SHOWN)

Rank	Freq	X	Y
1	43	-76.75070	39.31150
2	37	-76.47100	39.37410
3	24	-76.48800	39.33720
4	24	-76.60150	39.40420
5	23	-76.78770	39.40460
6	22	-76.65170	39.29270
7	21	-76.73190	39.28800
8	17	-76.53630	39.30600
9	15	-76.70260	39.35600
10	15	-76.51280	39.29270

were assigned a single point at the address. Thus, all thefts occurring at any one shopping center are assigned the same X and Y coordinates. However, there are situations when the assignment of a coordinate will not be a good indicator of the hot spot location. For example, assigning the vehicle theft location to a particular stall in a parking lot will lead to few, if any, locations coming up more than once. In this case, the mode would not be a useful statistic at all. Another example is assigning the vehicle theft location for the parking lot of a multi-building apartment complex to the address of the owner. In this case, what is a highly concentrated set of vehicle thefts become dispersed because the owners live in different buildings with different addresses.

Consequently, CrimeStat includes a second point location hot spot routine called the Fuzzy Mode. This allows the user to define a small search radius around each location to include events that occur around or near that location. For example, a user can put a 50 yard or 100 meter search radius and the routine will calculate the number of incidents that occur at each location and within a 50 yard or 100 meter radius.

The aim of the statistic is to allow the identification of locations where a number of incidents may occur, but where there may not be precision in measurement.[2] For example,

if several apartment complexes share a parking lot, any vehicle theft in the lot may be assigned to the address of the owner, rather than to the parking lot. In this case, the measurement is imprecise. Plotting the location of the vehicle thefts will make it appear that there are multiple locations, when, in fact, there is only approximately one.

Another example would be the measurement of motor vehicle crashes that all occur at a single intersection. If the measurement of the location is very precise, the crashes could be assigned to slightly different locations when, in fact, they occurred at more or less the same location. In other words, the fuzzy mode allows a flexible classification of a location where the analyst can vary slightly the area around a location.

The fuzzy mode output file is also a 'dbf' file and, like the mode, also includes four output variables:

1. The rank order of the location with 1 being the location with the most incidents, 2 being the location with the next most incidents, 3 being the location with the third most incidents, and so forth until only those locations which have only one incident each;

2. The frequency of incidents at the location. This is the number of incidents occurring at that location;

3. The X coordinate of the location; and

4. The Y coordinate of the location.

Note, that allowing a search radius around a location means that incidents are counted multiple times, one for each radius they fall within. If used carefully, the fuzzy mode can allow the identification of high incident locations more precisely than the mode routine. But, because of the multiple counting of incidents that occurs, the frequency of incidents at locations will change, compared to the mode, as well as possibly the hierarchy.

To illustrate this, figure 6.5 maps the top 13 locations for vehicle thefts identified by the fuzzy mode routine using a search radius of 100 yards. Thirteen locations are included because four were tied for number 10. The 13 locations are displayed by a magenta triangle and are compared to the 10 locations identified by the mode (blue circle). Three of the locations identified by the fuzzy mode routine are at the same approximate locations as that identified by the mode, but the remaining eight locations are clustered at a place not identified by the mode.

Figure 6.6 zooms in to display the eight clustered locations. This is a small regional mall within Baltimore city that has a subway station, a Maryland state motor vehicle administration office, and a parole/probation office. There are multiple parking lots located within the mall. Within this space, approximately 29 vehicle thefts occurred in 1996.

Figure 6.5:

Most Frequent Small Zones for Motor Vehicle Theft

Search Radius of 100 Yards

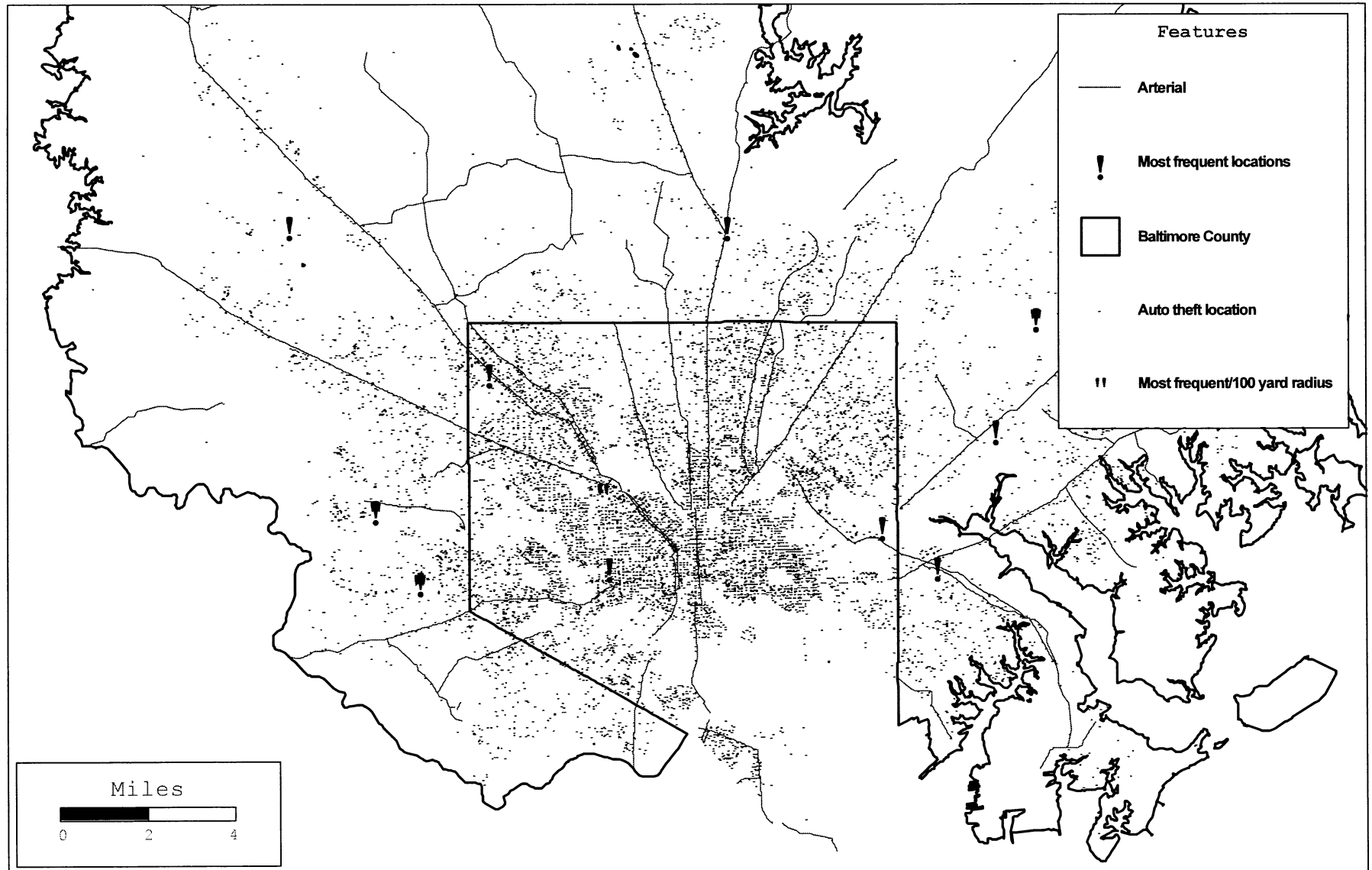


Figure 6.6:

Most Frequent Small Zones for Motor Vehicle Theft

Search Radius of 100 Yards

The fuzzy mode has identified a general location where there are multiple sub-locations in which vehicle thefts occur.

In other words, the fuzzy mode allows the identification of small hot spot areas, rather than exact locations. But, because all points within the user-defined search area are counted, points are counted multiple times. Thus, any one location may not have a sufficient number of incidents to be grouped in the 'top 10' by itself, but, because it is close to other locations that have incidents occurring, it may be elevated to the 'top 10' due to its adjacency to these other incident locations.

Still, the user must be careful in the analysis. By changing the search radius, the number of incidents counted for any one location changes as well as it's order in the hierarchy. For example, when a quarter mile search radius was used, all top locations occurred within a short distance of each other (not shown).

Nearest Neighbor Hierarchical Clustering (Nnh)

The nearest neighbor hierarchical clustering (Nnh) routine in CrimeStat identifies groups of incidents that are spatially close. It is a hierarchical clustering routine that clusters points together on the basis of a criteria. The clustering is repeated until either all points are grouped into a single cluster or else the clustering criteria fails. Hierarchical clustering methods are among the oldest cluster routines (Everitt, 1974; King, 1967; Systat, 2000). Among the clustering criteria that have been used are the nearest neighbor method (Johnson, 1967; D'andrade. 1978), farthest neighbor, the centroid method (King, 1967), median clusters (Gowers, 1967), group averages (Sokal and Michener, 1958), and minimum error (Ward, 1967).

The CrimeStat Nnh routine uses a method that defines a threshold distance and compares the threshold to the distances for all pairs of points. Only points that are closer to one or more other points than the threshold distance are selected for clustering. In addition, the user can specify a minimum number of points to be included in a cluster. Only points that fit both criteria - closer than the threshold and belonging to a group having the minimum number of points, are clustered at the first level (first-order clusters).

The routine then conducts subsequent clustering to produce a hierarchy of clusters. The first-order clusters are themselves clustered into second-order clusters. Again, only clusters that are spatially closer than a threshold distance (calculated anew for the second level) are included. The second-order clusters, in turn, are clustered into third-order clusters, and this re-clustering process is continued until either all clusters converge into a single cluster or, more likely, the clustering criteria fails.

Criteria 1: Threshold Distance

The first criteria in identifying clusters is whether points are closer than a specified threshold distance. There are two choices in selecting the threshold distance: 1) a random nearest neighbor distance (the default) and 2) a fixed distance.

Random nearest neighbor distance

The default choice to use the expected random nearest neighbor distance for first-order nearest neighbors. The user specifies a one-tailed confidence interval around the random expected nearest neighbor distance. The t-value corresponding to this probability level, t, is selected from the Student's t-distribution under the assumption that the degrees of freedom are at least 120.[3]

This selection is controlled by a slide bar under the routine (see Figure 6.3). From chapter 5, the mean random distance was defined as

$$\text{Mean Random Distance} = d(ran) = 0.5\ \text{SQRT}\left[\frac{A}{N}\right] \quad \text{repeat} \tag{5.2}$$

where A is the area of the region and N is the number of incidents. The confidence interval around that distance is defined as

$$\begin{aligned}\text{Confidence Interval for Mean Random Distance} &= \text{Mean Random Distance} \pm t^* SE_{d(ran)} \\ &= 0.5\ \text{SQRT}\left[\frac{A}{N}\right] \pm t\left[\frac{0.26136}{\text{SQRT}[N^2/A]}\right]\end{aligned} \tag{6.1}$$

where A is the area of the region, N is the number of incidents, t is the t-value associated with a probability level in the Student's t-distribution.

The lower limit of this confidence interval is

$$\text{Lower Limit of Confidence Interval for Mean Random Distance} = 0.5\ \text{SQRT}\left[\frac{A}{N}\right] - t\left[\frac{0.26136}{\text{SQRT}[N^2/A]}\right] \tag{6.2}$$

and the upper limit of this confidence interval is

$$\text{Upper Limit of Confidence Interval for Mean Random Distance} = 0.5\ \text{SQRT}\left[\frac{A}{N}\right] + t\left[\frac{0.26136}{\text{SQRT}[N^2/A]}\right] \tag{6.3}$$

The confidence interval defines a probability for the distance between any pair of points. For example, for a specific one-tailed probability, p, fewer than p% of the incidents would have nearest neighbor distances smaller than this selected limit if the distribution was spatially random. If the data were spatially random and if the mean random distance is selected as the threshold criteria (the default position on the slide bar), approximately 50% of the pairs will be closer than this distance. For randomly distributed data, if a $p \leq .05$ level is taken for t (two steps to the left of the default or the fifth in from the left), then only about 5% of the pairs would be closer than the threshold distance. Similarly, if a $p \leq .75$ level is taken for t (one step to the right of the default or the fifth in from the right), then about 75% of the pairs would be closer than the threshold distance.

In other words, the threshold distance is a probability level for selecting any two points (a pair) on the basis of a chance distribution. The slide bar has 12 levels and is associated with a probability level for a t-distribution from a sample of 120 or larger. From the left, the p-values are approximately (Table 6.2):

Table 6.2

Approximate Probability Values Associated with Threshold Scale Bar

Scale Bar Position	Probability	Description
1	0.00001	Far left point of slide bar
2	0.0001	Second from left
3	0.001	Third from left
4	0.01	Fourth from left
5	0.05	Fifth from left
6	0.1	Sixth from left
7	0.5	Sixth from right (default value)
8	0.75	Fifth from right
9	0.9	Fourth from right
10	0.95	Third from right
11	0.99	Second from right
12	0.999	Far right point of slide bar

Taking a broader conception of this, if there is a spatially random distribution, then for all distances between pairs of points, of which there are

$$\frac{N(N-1)}{2}$$

combinations, fewer than p% of the pairs will be shorter than this threshold distance.

This does not mean, however, that the probability of finding a cluster is equal to this probability. It only indicates the probability of selecting two points (a pair) on the basis of a

chance distribution. If additional points are to be included in the cluster, then the probability of obtaining the cluster will be less. Thus, the probability of selecting three points or four points or more points on the basis of chance will be much smaller.

Note that it is very important that area be defined correctly for this routine to work. If the user defines the area on the measurement parameters page (see chapter 3), the Nnh routine uses that value to calculate the threshold distance. If the user does not define the area on the measurement parameters page, the routine calculates the area from the minimum and maximum X/Y values (the bounding rectangle). In either case, the routine will be able to calculate a threshold distance and run the routine.

However, if the area units are defined incorrectly on the measurement parameters page, then the routine will certainly calculate the threshold distance wrongly. For example, if data are in feet but the area on the measurement parameters page are defined in square miles, most likely the routine will not find any points that are farther apart the threshold distance since that distance is defined in miles. In other words, it is essential that the area units be consistent with the data for the routine to properly work.

Fixed distance

The second choice in selecting a threshold distance is to choose a fixed distance (in miles, nautical miles, feet, kilometers, or meters). The uses checks the "Fixed distance" box and selects a threshold distance. The main advantage in this method is that the search radius can be specified exactly. This is useful for comparing the number of clusters for different distributions (e.g., the number of robbery hot spots compared to burglary hot spots using a search radius of 0.5 miles). The main disadvantage of this method is that the choice of a threshold is subjective. The larger the distance that is selected, the greater the likelihood that clusters will be found by chance. Of course, this can be tested using a Monte Carlo simulation (see below).

Criteria 2: Minimum Number of Points

Whatever method is used for selecting a threshold distance, a second criteria is the minimum number of points that are required for each cluster. This criteria is used to reduce the number of very small clusters. With large data sets, hundreds, if not thousands, of clusters can be found if only points are selected by being closer than a threshold distance. To minimize numerous very small clusters as well as reduce the likelihood that clusters could be found by chance, the user can set a minimum number restriction. The default is 10. By decreasing this number, more clusters are produced; conversely, by increasing this number, fewer clusters are produced. The routine will only include points in the final clustering that are part of groups (or clusters) in which the minimum number is found.

First-order clustering

Using these criteria, CrimeStat constructs a first-order clustering of the points.[4] For each first-order cluster, the center of minimum distance is output as the cluster center, which can be saved as a '.dbf' file.

Second and higher-order clusters

The first-order clusters are then tested for second-order clustering. The procedure is similar to first-order clustering except that the cluster centers are now treated as 'points' which themselves are clustered.[5] The process is repeated until no further clustering can be conducted, either all sub-clusters converge into a single cluster, or the threshold distance criteria fails, or there are fewer than four seeds in the higher-order cluster.

Visualizing the Cluster Output

To identify the approximate cluster location, CrimeStat III allows the cluster to be output as either an ellipse, a convex hull, or both.

Ellipse output

A standard deviational ellipse is calculated for each cluster (see chapter 4 for definition). The user can choose between 1X (the default), 1.5X, and 2X. Typically, one standard deviation will cover more than 50% of the cases, one and a half standard deviations will cover more than 90% of the cases, and two standard deviations will cover more than 99% of the cases, although the exact percentage will depend on the distribution. The user specifies the number of standard deviations to save as ellipses in ArcView '.shp', MapInfo '.mif' or Atlas*GIS '.bna' formats.

In general, use a 1X standard deviational ellipse since 1.5X and 2X standard deviations can create an exaggerated view of the underlying cluster. The ellipse, after all, is an abstraction from the points in the cluster that may be arranged in an irregular manner. On the other hand, for a regional view, a 1X standard deviational ellipse may not be very visible. The user has to balance the need to accurately display the cluster compared to making it easier for a viewer to understand its location.

Convex hull output

A convex hull is calculated for each cluster (see chapter 4 for definition). The convex hull draws a polygon around the points in the cluster. It is a literal definition of the cluster, as opposed to the ellipse which is an abstraction. The convex hull can be saved in ArcView '.shp', MapInfo '.mif' or Atlas*GIS '.bna' formats.

Ellipse or convex hulls

In previous versions of CrimeStat, only ellipses were used for cluster graphical output. With the addition of a convex hull, the user can visualize the cluster in two different ways. There are advantages and disadvantages of each approach. The convex hull has the advantage of being a polygon that corresponds exactly to the cluster. For neighborhood level analysis, it is probably preferable to the ellipse, which is an abstraction. On the other hand, any convex hull is based on a sample (e.g., this year's robberies compared to last year's robberies) and like any sample will vary from one instance to

another. It may not capture all the space associated with the hot spot. The shape also is often un-intuitive, following the outline of the incidents. The ellipse, on the other hand, is more general and will usually be more stable from year to year. It usually looks better on a map or at least users seem to understand it better; it is a more familiar graphical object than an irregular polygon. The biggest disadvantage to an ellipse is that it forces a certain shape on the data, whether there are incidents in every part of it or not. So, in extreme cases, one finds ellipses that go outside of study area boundaries or extend into reservoirs or lakes or other features which are logically impossible.

In short, the user needs to balance the generality and visual familiarity of an ellipse with the limits of the actual hot spot. Probably for a small scale, regional perspective, the ellipses are more adequate and are preferable since a viewer can quickly see where the hot spots are located. For detailed neighborhood-level work, however, the convex hull is probably better since it shows where the incidents actually occurred.

Guidelines for Selecting Parameters

In the Nnh routine, the user has to define three parameters - the threshold distance, the minimum number of points, and the visual output of the hot spots. For a fixed threshold distance, the user has to choose something that is meaningful. For crime incidents, probably the threshold distance should not be more than 0.5 miles and, preferably, smaller.

If the random nearest neighbor distance is used as a threshold, the p-value is selected with a likelihood slider bar (see figure 6.3). This bar indicates a range of p-values from 0.00001 (i.e., the likelihood of obtaining a pair by chance is 0.001%) to 0.999 (i.e., the likelihood of obtaining a pair by chance is 99.9%). The slider bar actually controls the value of t in equation 6.3, which varies from -3.719 to +3.090. The smaller the t-value, the smaller the threshold distance. With smaller threshold distances, fewer clusters are extracted, which are typically smaller (although not always).

If only pairs of points were being grouped, then the threshold distance would be critical. Thus, if the default $p \leq .5$ value is selected, then about half the pairs would be selected by chance if the data were truly random. However, since there are a minimum number of points that are required, the likelihood of finding a cluster with the minimum number of points is much smaller. The higher the minimum number that is required, the smaller the likelihood of obtaining a cluster by chance.

Therefore, one can think of the slide bar as a filter for grouping points. One can make the filter smaller (moving the slide bar to the left) or larger (moving the slide bar to the right). There will be some effect on the final number of clusters, but the likelihood of obtaining a cluster by chance will be generally low. Statistically, there is more certainty with small threshold distances than with larger ones using this technique. Thus, a user must trade off the number of clusters and the size of an area that defines a cluster with the likelihood that the result could be due to chance.

This choice will depend on the needs of the user. For interventions around particular locations, the use of a small threshold distance may actually be appropriate; some of the ellipses seen in figure 6.7 below cover only a couple of street segments. These define micro-neighborhoods or almost pure hot spot locations. On the other hand, for a patrol route, for example, a cluster the size of several neighborhoods might be more appropriate. A patrol car would need to cover a sizeable area and having a larger area to target might be more appropriate than a 'micro' environment. However, there will be less precision with a larger cluster size covering this type of area.

A second criterion is the minimum number of points that are required to define a cluster. If a cluster does not have this minimum number, CrimeStat will ignore the seed location. Without this criteria, the Nnh routine could identify clusters of two or three incidents each. A hot spot of this size is usually not very useful. Consequently, the user should increase the number to ensure that the identified cluster represents a meaningful number of cases. The default value is 10, but the user can type in any other value.

The user may have to experiment with several runs to get a solution that appears right. As a rule of thumb, start with the default settings. If there appears to be too many clusters, tighten up the criteria by selecting a lower probability for grouping a pair by chance (i.e., shifting the threshold distance to the left) or increasing the minimum number of points required to be defined as a cluster (e.g., from 10 to 20). On the other hand, if there appears to be too few clusters, loosen the criteria by selecting a higher probability for grouping pairs by chance (i.e., shifting the threshold distance to the right) or decreasing the minimum number of points in a cluster (e.g., from 10 to 5). Then, once an appropriate solution has been found, the user can fine tune the results by slight changes.

In general, the minimum number of points criteria is more critical for the number of clusters than the threshold distance, though the latter can also influence the results. For example, with the 1996 Baltimore County robbery data set (N=1181 incidents), a minimum of 26 and a maximum of 28 clusters were found by changing the threshold distance from the minimum p-value ($p \leq 0.00001$) to the maximum p-value ($p \leq 0.999$). On the other hand, changing the minimum number of points per clusters from 10 to 20 reduced the number of clusters found (with the default threshold distance) from 26 to 11.

The third criterion is the visual display of the clusters. The convex hull is literal; it will draw a polygon around the points in the cluster. The ellipse, on the other hand, requires a decision by the user on the number of standard deviations to be displayed. The choices are 1X (the default), 1.5X and 2X standard deviations. Typically, one standard deviation will cover more than 50% of the cases, one and a half standard deviations will cover more than 90% of the cases, and two standard deviations will cover more than 99% of the cases, although the exact percentage will depend on the distribution.

In general, use a one standard deviational ellipse since 1.5X and 2X standard deviations can create an exaggerated view of the underlying cluster. On the other hand, for a regional view, a one standard deviational ellipse may not be very visible. The user has to balance the need to accurately display the cluster compared to making it easier for a viewer to understand its location.

Nnh Output Files

The Nnh routine has six outputs. First, for each cluster that is identified, the hierarchical order and the cluster number. Second, for each cluster that is calculated, CrimeStat calculates the mean center of the cluster. Only 45 of the seed locations are displayed on the screen. The user can scroll down or across by adjusting the horizontal and vertical slider bars and clicking on the Go button. This can be saved as a '.dbf' file. Third, the standard deviational ellipses of the clusters is shown, whether the graphical output is an ellipse or a convex hull. The size of the ellipses are determined by the number of standard deviations to be calculated (see above). Fourth, the number of points in the cluster. Fifth, the area of the ellipse and, sixth, the density of the cluster (number of points divided by area).

The ellipses and convex hulls can be saved in ArcView '.shp', MapInfo '.mif' or Atlas*GIS '.bna' formats. Because there are also orders of clusters (i.e., first-order, second-order, etc.), there is a naming convention that distinguishes the order.

For the ellipses, the convention is

Nnh<O><username>

where O is the order number and username is a name provide by the user. Thus,

Nnh1robbery

are the first-order clusters for a file called 'robbery' and

Nnh2NightBurglaries

are the second-order clusters for a file called 'NightBurglaries'. Within files, clusters are named

Nnh<O>Ell<N><username>

whereO is the order number, N is the ellipse number and username is the user-defined name of the file. Thus,

Nnh1Ell10robbery

is the tenth ellipse within the first-order clusters for the file 'robbery' while

Nnh2Ell1NightBurglaries

is the first ellipse within the second-order clusters for the file 'NightBurglaries'.

For the convex hulls, the name will be output with a 'CNNH1' prefix for the first-order clusters, a 'CNNH2' prefix for the second-order clusters, and a 'CNNH3' prefix for the third-order clusters. Higher-order clusters will index only the number.

In other words, names of files and features can get complicated. The easiest way to understand this, therefore, is to import the file into one of the GIS packages and display it.

Example 1: Nearest neighbor hierarchical clustering of burglaries

The Nnh routine was applied to the Baltimore County 1996 burglary data (n=6,051 incidents). A default one-tailed probability level of .05 (or 5%) was selected and each cluster was required to contain a minimum of 10 points (the default). CrimeStat returned 122 first-order clusters, 15 second-order clusters and two third–order clusters. Figure 6.7 shows the first-order clusters displayed as 1x standard deviational ellipses. Since the criteria for clustering is the lower limit of the mean random distance, the distances involved are very small, as can be seen. Note, the standard deviational ellipse is defined by the points in the cluster and includes approximately 50% of the points. Thus, the clusters actually extend a little beyond the ellipses.

Figure 6.8 shows the 20 second-order clusters (dashed lines) and the two third-order clusters (double lines). As seen, they cover much larger areas than the first-order clusters. Finally, figure 6.9 shows a part of east Baltimore County where there are 29 first-order clusters (solid line), five second-order clusters (dashed lines), and one third-order cluster (double line). The street network is presented to indicate the scale. Most first-order clusters cover an area the size of a small neighborhood while the second-order clusters cover larger neighborhoods.

To illustrate how the convex hull produces a different visualization, figure 6.10 shows the same clusters as in figure 6.9 but the clusters are displayed as convex hulls rather than ellipses. As seen, the convex hulls are irregular in shape and more limited in geographical spread; they show only the incidents that are clusters. The second-order and third-order clusters are also more defined. From a policing viewpoint, this is probably more useful in that it shows where the hot spot incidents are actually located. As mentioned above, the polygons created by the convex hulls are irregular and are, therefore, less familiar to most people. Consequently, for presentations of crime patterns at a regional level or even neighborhood-level for non-specialists, the ellipses may convey better where the hot spots are located.

Advantages of Hierarchical Clustering

There are four advantages to this technique. First, it can identify small geographical environments where there are concentrated incidents. This can be useful for specific targeting, either by police deployment or community intervention. There are clearly micro-environments that generate crime incidents (Levine, Wachs and Shirazi, 1986; Maltz, Gordon and Friedman, 1989). The technique tends to identify these small environments because the lower limit of the mean random distance is used to group the

Figure 6.7:

First-Order Baltimore County Burglary 'Hot Spots': Ellipses

Using Nearest Neighbor Hierarchical Clustering Method

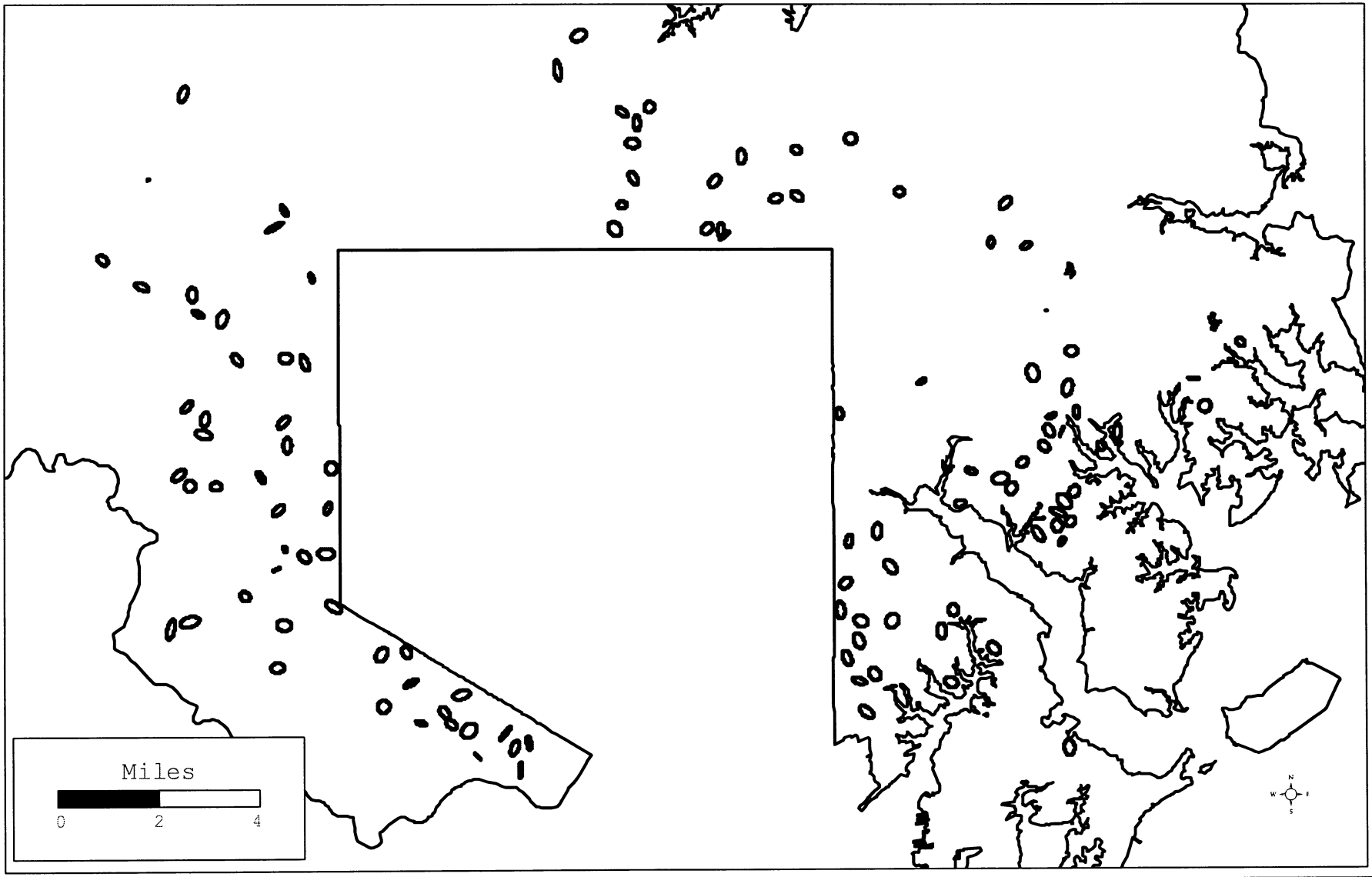


Figure 6.8:

Second- and Third-Order Burglary 'Hot Spots': Ellipses

Using Nearest Neighbor Hierarchical Clustering Method

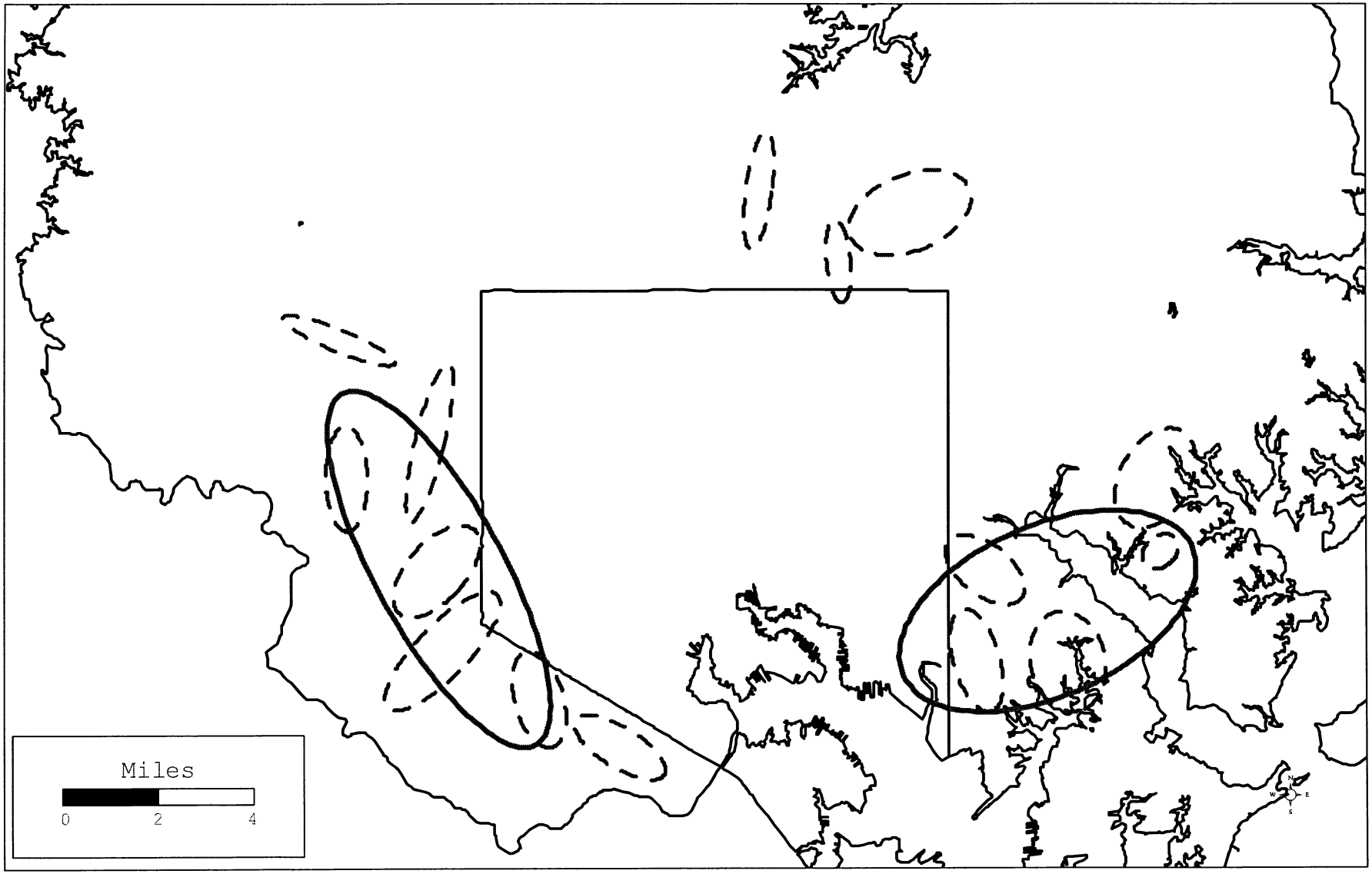


Figure 6.9:

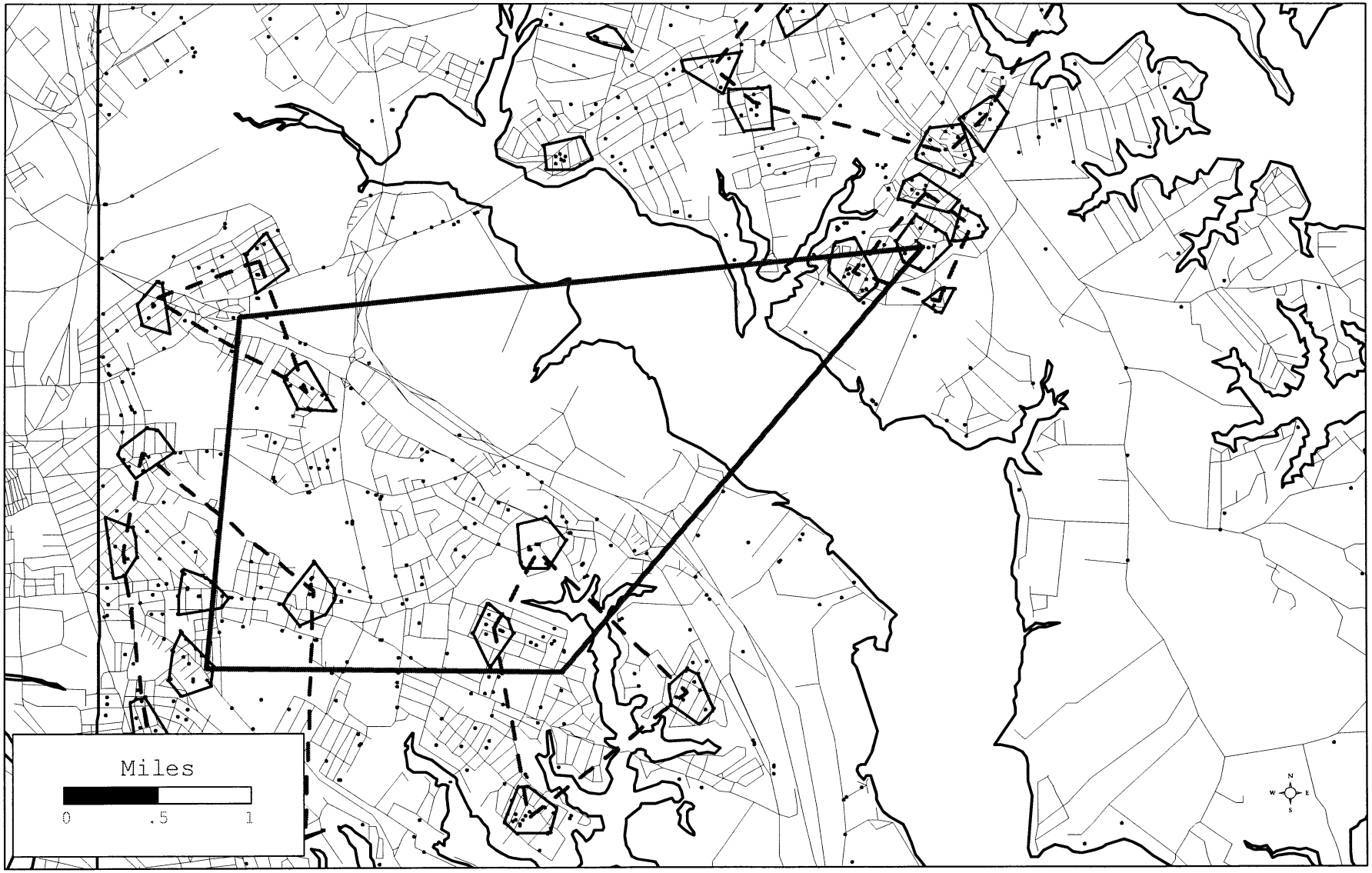
First, Second- and Third-Order Burglary 'Hot Spots': Ellipses

Using Nearest Neighbor Hierarchical Clustering Method

Figure 6.10:

First, Second- and Third-Order Burglary 'Hot Spots': Convex Hulls

Using Nearest Neighbor Hierarchical Clustering Method



clusters. The user can, of course, control the size of the grouping area by loosening or tightening either the threshold distance or the minimum number of required points. Thus, the sizes of the clusters can be adjusted to fit particular groupings of points.

Second, the technique can be applied to any entire data set, such as for Baltimore County and Baltimore City, and need not only be applied to smaller geographical areas, such as precincts. This increases the ease of use for analysts and can facilitate comparisons between different areas without having to limit arbitrarily the data set.

Third, the linkages between several small clusters can be seen through the second- and higher-order clusters. Frequently, 'hot spots' are located near other 'hot spots' which, in turn, are located near other 'hot spots'. As we've seen from the maps of robbery, burglary and motor vehicle thefts in Baltimore County, there are large areas within the County that have a lot of incidents. Within these large areas, there are smaller hot spots and within some of those hot spots, there are even small ones. In other words, there are different scales to the clustering of points - different geographical levels, if you will, and the hierarchical clustering technique can identify these levels.

Fourth, each of the levels imply different policing strategies. For the smallest level, officers can intervene effectively in small neighborhoods, as discussed above. Second-order clusters, on the other hand, are more appropriate as patrol areas; these areas are larger than first-order clusters, but include several first-order clusters within them. If third- or higher-order clusters are identified, these are generally areas with very high concentrations of crime incidents over a fairly large section of the jurisdiction. The areas start to approximate precinct sizes and need to be thought of in terms of an integrated management strategy - police deployment, crime prevention, community involvement, and long-range planning. Thus, the hierarchical technique allows different security strategies to be adopted and provides a coherent way of approaching these communities.

Simulating Statistical Significance

Testing the significance of clusters from the Nnh routine is difficult. Conceptually, using the random nearest neighbor distance for the threshold distance defines the probability that two points could be grouped together on the basis of chance; the test is for the confidence interval around the first-order nearest neighbor distance for a random distribution. If the probability level is p%, then approximately p% of all pairs of points would be found under a random distribution. Under this situation, we would know whether the number of clusters (pairs) that were found were significantly greater than would be expected on the basis of chance.

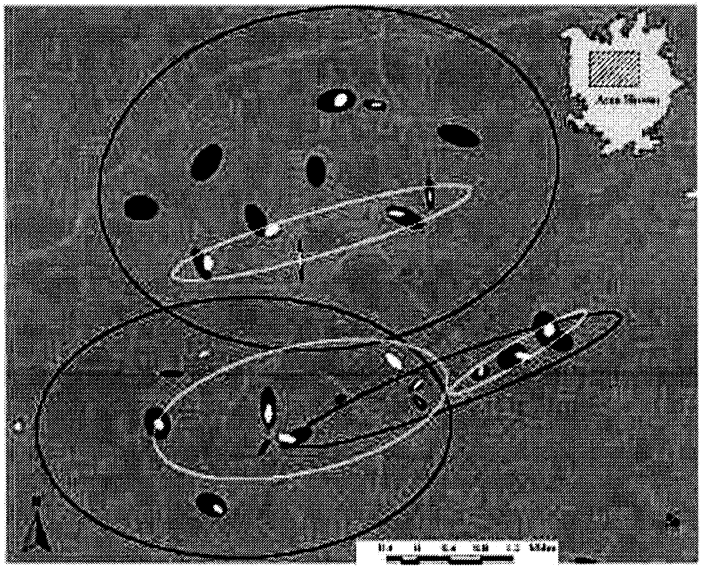
The problem is, however, that the routine is not just clustering pairs of points, but clustering as many points as possible that fall within the threshold distance. Further, the additional requirement is added that there be a minimum number of points, with the minimum defined by the user. The probability distribution for this situation is not known. Consequently, there is a necessity to resort to a Monte Carlo simulation of randomness under the conditions of the Nnh test (Dwass, 1957; Barnard, 1963).

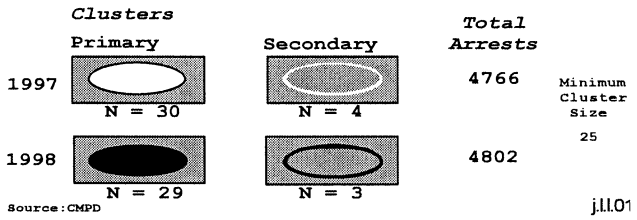
Visualizing Change in Drug Arrest Hot Spots Using Nearest Neighbor Hierarchical Clustering: Charlotte, N.C. 1997 - 98

James L. LeBeau
Administration of Justice
Southern Illinois University at Carbondale

Stephen Schnebly
Criminology & Criminal Justice
University of Missouri – St Louis

The *CrimeStat* Nearest Neighbor Hierarchical clustering routine and GIS were used for defining, comparing, analyzing, and visualizing changes in drug arrest clusters between 1997 and 1998. Using a minimum cluster size of 25 arrests some of the emerging patterns or relationships include: 1) the overlapping of secondary clusters, but those emerging during 1998 were much larger, especially in the north because of new primary clusters; 2) many primary clusters during 1997 remaining static or increasing in area during 1998; and 3) the disappearing of some 1997 primary clusters during 1998, with new clusters emerging close by implying displacement.



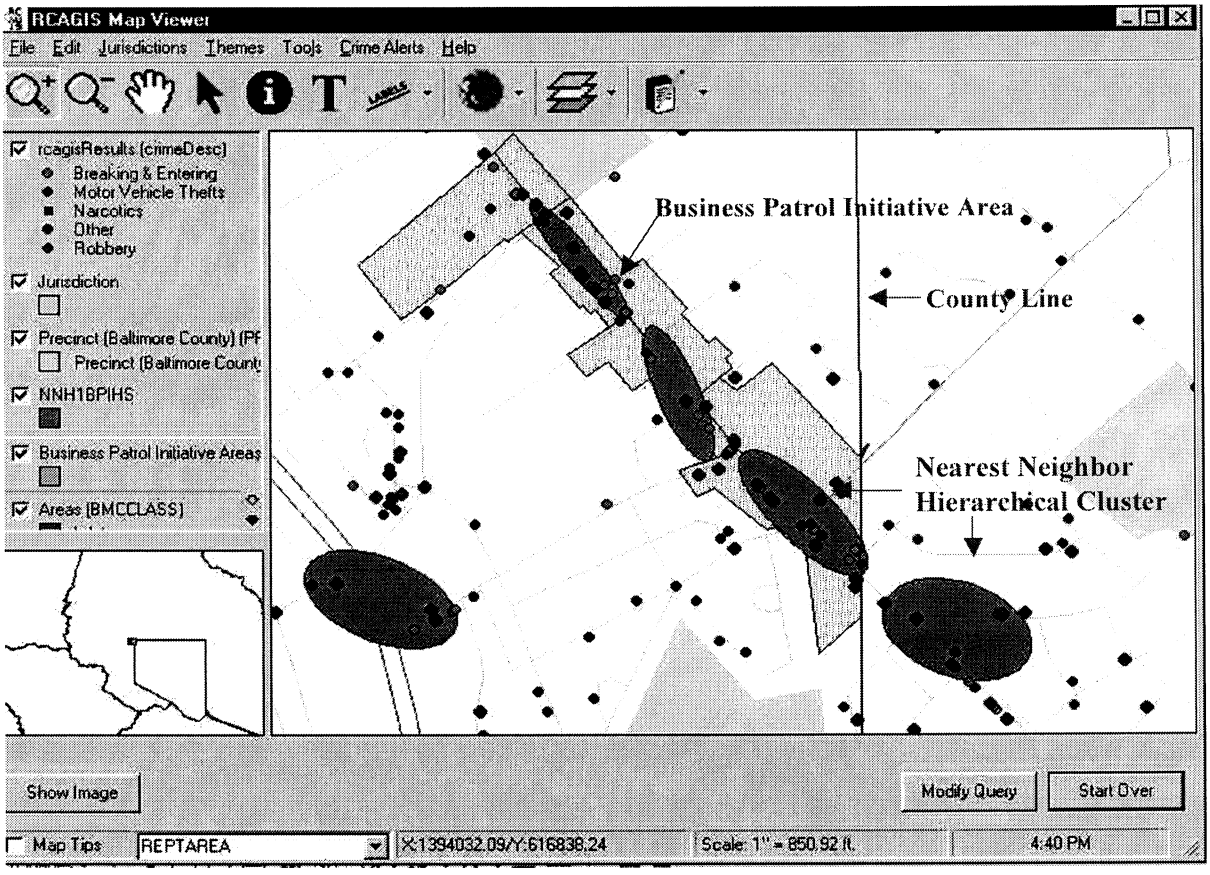


Using Nearest Neighbor Hierarchical Clustering to Identify High Crime Areas Along Commercial Corridors

Philip R. Canter
Baltimore County Police Department
Towson, Maryland

Robberies in Baltimore County had increased by 45% between 1990 and 199, and by 1997, were the highest on record. In 1997, 73% of all reported robberies in Baltimore County were occurring in commercial areas. The department wanted to target commercial districts with intensive patrol and outreach programs. These high crime commercial districts were identified as Business Patrol Initiative (BPI) areas. A total of 40 police officers working two 8-hour shifts were assigned to BPI areas. Robberies in the BPI areas declined by 26.7% during the first year of the program and another 13.8% one year following the BPI program.

Police analysts used *CrimeStat*'s Nearest Neighbor Hierarchical clustering (Nnh) method to identify high crime areas along commercial corridors. The Nnh routine was very effective in identifying commercial areas having the highest concentration of crime. The clustering also demonstrated that commercial crime was not restricted to county borders; rather, crime crossed municipal boundaries into neighboring jurisdictions. A neighboring jurisdiction was shown the crime cluster map, leading to their decision to implement a similar BPI program.



CrimeStat includes a Monte Carlo simulation routine that produces approximate confidence intervals for the first-order Nnh clusters that has been run; second- and higher-order clusters are not simulated since their structure depends on the first-order clusters. Essentially, the routine assigns N cases randomly to a rectangle with the same area as the defined study area, A, and evaluates the number of clusters according to the defined parameters (i.e., threshold distance and minimum number of points). It repeats this test K times, where K is defined by the user (e.g., 100, 1,000, 10,000). By running the simulation many times, the user can assess approximate confidence intervals for the particular first-order Nnh.

The output includes five columns and twelve rows:

Columns:

1. The percentile,
2. The number of first-order clusters found for that percentile,
3. The area of the cluster for that percentile,
4. The number of points in the cluster for that percentile, and
5. The density of points (per unit area) for that percentile.

Rows:

1. The minimum (smallest) value obtained,
2. 0.5^{th} percentile,
3. 1^{st} percentile,
4. 2.5^{th} percentile,
5. 5^{th} percentile,
6. 10^{th} percentile,
7. 90^{th} percentile,
8. 95^{th} percentile,
9. 97.5^{th} percentile,
10. 99^{th} percentile,
11. 99.5^{th} percentile, and
12. The maximum (largest) value obtained.

The manner in which percentiles are calculated are as follows. First, over all simulation runs (e.g., 1000), the routine calculates the number of first-order clusters obtained for each run, sorts them in order, and defines the percentiles for the list. Thus, the minimum is the fewest number of clusters obtained over all runs, the 0.5 percentile is the lowest half of a percent for the number of clusters obtained over all runs, and so forth until the maximum number of clusters obtained over all runs. The routine does not calculate second- or higher-order clusters since those are dependent on the first order clustering. Second, within each run, the routine calculates the number of points per cluster, the area of each ellipse, and the density of each ellipse. Then, it groups all clusters together, over all runs, and sorts them into a list. The percentiles for individual clusters are then calculated. Note that the points refer to the cluster whereas the area and density refer to the ellipses, which is a geometrical abstraction from the cluster.

Table 6.3 presents an example. An Nnh run was conducted on the Baltimore robbery data base (N=1181 incidents) using the default threshold distance ($p \leq .5$ for grouping a pair by chance) and a minimum number of points of at least five for each cluster. Then, 1000 Monte Carlo runs were conducted with simulated data. For the actual data, the Nnh routine identified 69 first-order clusters and 7 second-order clusters. Table 6.3 presents the parameters for the first ten first-order clusters.

In examining a simulation, one has to select percentiles as choice points. In this example, we use the 95th percentile. That is, we are willing to accept a one-tailed Type I error of only 5% since we are only interested in finding a greater number of clusters than by chance. For the simulation, let's look at each column in turn. Column 2 presents the number of clusters found in each simulation. Over the 1000 runs, there was a minimum of one cluster found (for at least one simulation) and a maximum of 7 clusters found (for at least one simulation). That is, running 1000 simulations of randomly assigned data only yielded between 1 and 7 clusters using the parameters defined in the particular Nnh run. The 95th percentile was 3. It is highly unlikely that the 69 first-order clusters that were identified would have been due to chance. That is, we would have expected at most three of them to have been due to chance. It appears that the robbery data is significantly clustered, though we have only tested significance through a random simulation.

Column 3 shows the areas of clusters that were found over the 1000 runs. For the individual clusters, the simulation showed a range from about 0.04 to 0.38. The 95th percentile was 0.31. In the actual Nnh, the area of clusters varied between 0.05 and 0.27, indicating that all first-order clusters were smaller than the smallest value found in the simulation. In other words, the real clusters are more compact than random clusters even though the random clusters are subject to the same threshold distance as the real data. This is not always true, but, in this case, it is.

Column 4 presents the number of points found per cluster. In the simulations, the numbers varied between 5 and 9 points per cluster. The 95th percentile was 7. With the actual data, the number of points varied between 5 and 40. Thus, some of the clusters could have been due to chance, at least in terms of the number of points per cluster. Analyzing the distribution (not shown), 27 of the 69 clusters had 7 or fewer points. In other words, about 39% had only as many points as might be expected on the basis of a chance distribution. Putting it another way, about 40% of the clusters had more points than would be expected on the basis of chance 95% of the time.

Finally, column 5 presents the density of points found per cluster. Since the output unit is squared miles, density is the number of points per square mile. The simulation presents a range from 15.6 points per square mile to 156.1 points per square mile. The 95th percentile was 73.4 points per square mile. The actual Nnh, on the other hand, finds a range of densities from 27.1 points per square mile to a very high number (11071821 points per square mile). Again, there is overlap between the actual clusters and what might be expected on the basis of chance; 26 out of 69 clusters have densities that are lower than the 95th percentile found in the simulation. Again, about 38% have densities are not different than would be expected on the basis of chance.

Table 6.3

Simulated Confidence Intervals for Nnh Routine
Baltimore County Robberies: N=1181

Nearest Neighbor Hierarchical Clustering:

Sample size.....:	1181
Likelihood of grouping pair of points by chance....:	0.50000 (50.000%)
Z-value for confidence interval.....:	0.000
Measurement type.....:	Direct
Output units.....:	Miles, Squared Miles, Points per Squared Miles
Clusters found.....:	76
Simulation runs.....:	1000

Displaying ellipse(s) starting from 1

Order	Cluster	Mean X	Mean Y	Rotation	X-Axis	Y-Axis	Area	Points	Density
1	1	-76.44927	39.31455	77.09164	0.28303	0.09636	0.08568	40	466.828013
1	2	-76.60219	39.40050	11.98132	0.11540	0.27452	0.09952	33	331.580616
1	3	-76.44601	39.30490	16.66988	0.21907	0.16239	0.11176	25	223.684859
1	4	-76.78123	39.36088	25.36983	0.27643	0.14530	0.12618	29	229.826284
1	5	-76.73103	39.34319	67.71617	0.19445	0.16058	0.09810	29	295.628310
1	6	-76.72945	39.28910	79.88383	0.16428	0.25957	0.13396	29	216.476166
1	7	-76.51486	39.25986	87.32563	0.19148	0.29428	0.17703	27	152.520725
1	8	-76.45374	39.32106	54.57635	0.15150	0.18261	0.08692	7	80.538112
1	9	-76.75368	39.31132	89.56994	0.19748	0.22914	0.14216	22	154.753006
1	10	-76.71641	39.29139	10.43857	0.15048	0.16879	0.07980	14	175.444372

...etc.

Distribution of the number of clusters found in simulation (percentile):

Percentile	Clusters	Area	Points	Density
min	1	0.03845	5	15.615111
0.5	1	0.04922	6	16.608967
1.0	1	0.05603	6	17.162252
2.5	1	0.06901	6	18.570113
5.0	1	0.08243	6	19.468353
10.0	1	0.10045	6	21.256559
90.0	2	0.28706	7	61.173748
95.0	3	0.31074	7	73.463654
97.5	3	0.32442	7	87.550868
99.0	4	0.35279	8	115.460337
99.5	5	0.36489	8	122.625375
max	7	0.38424	9	156.056837

In other words, the simulation suggests that around 60% of the clusters are real with the other 40% being no different than might be expected on the basis of chance. There are far more clusters found in the actual Nnh than would be expected on the basis of chance and they are more compact than would be expected. On the other hand, only about half have densities that are higher than would be expected on the basis of chance.

It should be clear that testing the significance of a cluster analysis is complex. In the example, some of the criteria chosen were definitely different than a chance distribution (as evidenced by the simulation) while other criteria were not very different. In this case, the user would be wise to re-run the Nnh and simulation under tighter conditions, either lowering the threshold distance or increasing the minimum number of points per cluster. With experimentation, it is frequently possible to obtain a solution in which all the criteria are greater than would be expected on the basis of chance.

Limitation to Hierarchical Clustering

There are also limitations to the technique, some technical and others theoretical. First, the method only clusters incidents (points); a weighting or intensity variable will have no effect. Second, the size of the grouping area is dependent on the sample size when the confidence interval around the mean random distance is used as the threshold distance criteria (see equation. 4.2). For crime distributions that have many incidents (e.g., burglary), the threshold distance will be a lot smaller than distributions that have fewer incidents (e.g., robbery). In theory, a hot spot is dependent on an environment, not the number of incidents. Thus, that approach does not produce a consistent definition of a hot spot area. Using a fixed distance for the threshold distance can partly overcome this. However, the fixed distance needs to be tested for randomness using the Monte Carlo simulation.

Third, there is a certain arbitrariness in the technique due to the minimum points rule. This implicitly requires the user to define a meaningful cluster size, whether the number of points are 5, 10, 15 or whatever. To some extent, this is how patterns are defined by human beings; with one or two incidents in a small area, people don't perceive any pattern. As soon as the number of incidents increases, say to 10 or more, people perceive the pattern. This is not a statistical way for defining regularity, but it is a human way. However, it can lead to arbitrariness since two different users may interpret the size of a hot spot differently. Similarly, the selectivity of the p-value, vis-a-via the Student's t-distribution, can allow variability between users.

In short, the technique does produce a constant result, but one subject to manipulation by users. Hierarchical techniques are, of course, not the only clustering procedures to allow users to adjust the parameters; in fact, almost all the cluster techniques have this property. But it is a statistical weakness in that it involves subjectivity and is not necessarily consistently applied across users.

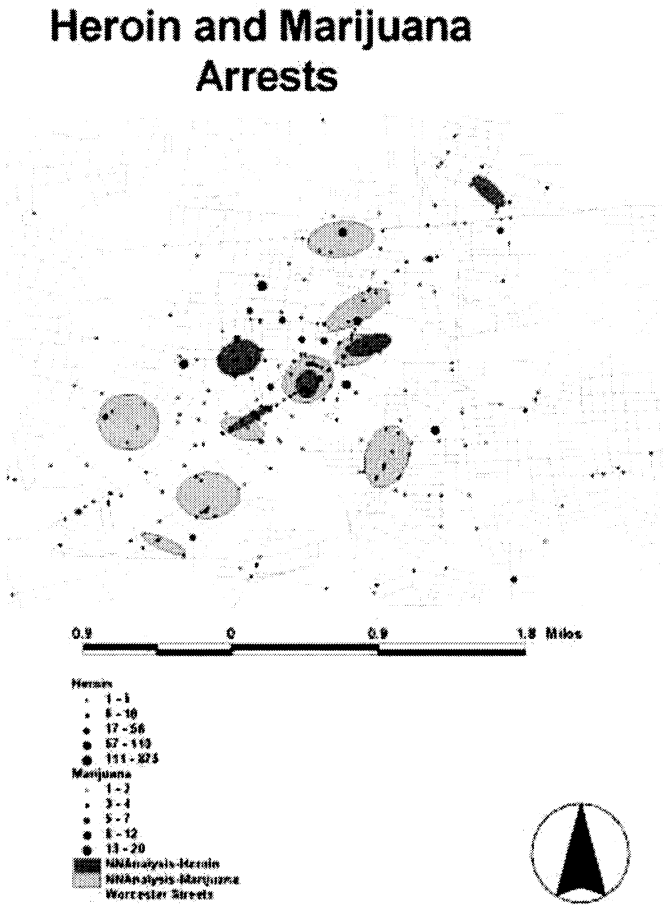
Finally, there is no theory or rationale behind the clusters. They are empirical derivatives of a procedures. Again, many clustering techniques are empirical groupings and also do not have any explanatory theory. However, if one is looking for a substantive

Arrest Locations as a Means for Directing Resources

Daniel Bibel
Massachusetts State Police
Crime Reporting Unit
Framingham, Massachusetts

The Massachusetts State Police is collecting incident addresses as part of its state-level implementation of the FBI's National Incident Based Reporting System (NIBRS). They intend to develop a regional and statewide crime mapping and analysis program. As an example of the type of analysis that can be done with the enhanced NIBRS database, the State Police's Crime Reporting Unit analyzed year 2000 drug arrests for one city in the Commonwealth, focusing on arrests for possession of heroin and marijuana. The arrest locations were plotted, with the size of points proportionate to the amount of drugs seized. A nearest neighbor clustering analysis was done of the data. It indicates that, while there is some small amount of overlap, the arrest locations for the two drug types are generally different.

This type of analysis can be very useful for smaller police agencies that do not have the resources to conduct their own analysis of crime data. It may also prove useful for crime problems with cross-jurisdictional boundaries.

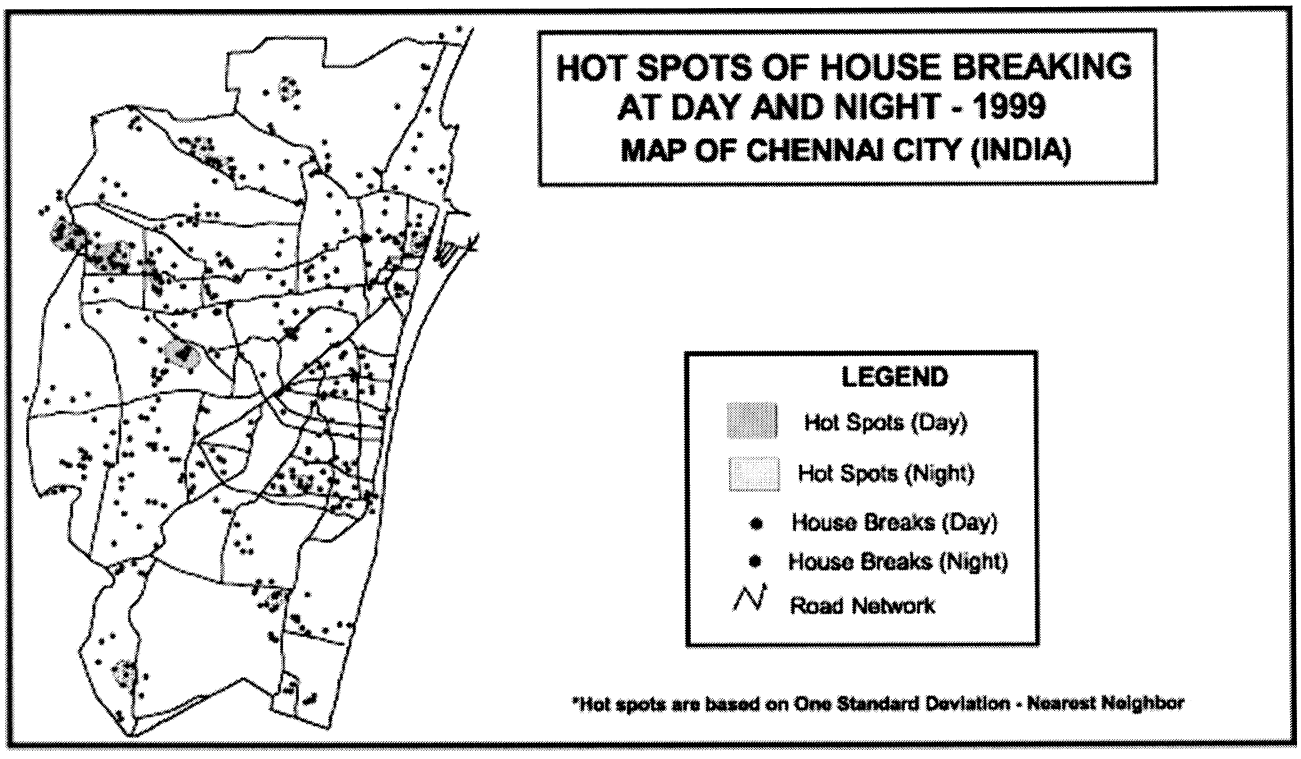


Use of *CrimeStat* in Crime Mapping in India: An Application for Chennai City Policing

Jaishankar Karuppannan
Department of Criminology & Criminal Justice
Manonmaniam Sundaranar University
Tamil Nadu, India

The present study was done as an implementation of GIS technology in Chennai (Madras), India. In the present study hotspot analysis was done with the help of *CrimeStat*. We converted the output to *Arcview* shape files.

When hotspot analysis examined changes over a period of time, the change seemed to be significant. There exists not only a change in the location of the hotspots, but also in their areal extent. The numbers of hotspots also differ over time. The map shows hotspots for residential burglary for both day and night. The hot spots for daytime house break-ins are confined to a smaller area in the west of the city, whereas the hot spots for nighttime residential break-ins are seen in all parts of the city. In particular, the Posh area of Anna Nagar is more prone to daytime burglaries. In this area, a higher proportion of couples work, which appears to make the homes in this neighborhood more open for burglaries.



hot spot defined by a unique constellation of land uses, activities, and targets, the technique does not provide any insight into why the clusters are occurring or why they could be related. I will return to this point at the end of the next chapter, but it should be remembered that these are empirical groupings, not necessarily substantive ones.

Risk-Adjusted Nearest Neighbor Hierarchical Clustering

CrimeStat also includes a risk-adjusted nearest neighbor hierarchical clustering routine (Rnnh), which is a variation on the Nnh routine discussed above. It combines the hierarchical clustering capabilities of the Nnh routine with kernel density interpolation techniques, that are discussed in chapter 8.

The Nnh routine identifies clusters of points that are close together. That is, it will identify groups of points that are closer together than a threshold distance and in which the minimum number of points is greater than a user-defined value. Many of these clusters, however, are due to a high concentration of persons in the vicinity. That is, because the population is not arranged randomly over a plane, but is, instead, highly concentrated in population centers, there is a higher likelihood of incidents happening (whatever they are) simply due to the higher population concentration. In the above examples, many of the clusters for Baltimore burglaries or vehicle thefts were due primarily to a high concentration of households and vehicles in the center of the metropolitan area. In fact, one would normally expect a higher concentration of incidents in the center since there are more persons residing in the center and, certainly, more persons being concentrated there during the daytime through employment, shopping, cultural attendance, and other urban activities.

For many police purposes, the concentration of incidents is of sufficient interest in itself. Police have to intervene at high incidence locations irrespective of whether there is also a larger population at those locations. The demands for policing and responding to community emergency needs is population sensitive since there are more demands where there are more persons. From a service viewpoint, the concentration of incidents is what is important.

But for other purposes, the concentration of incidents relative to the baseline population is of interest. Crime prevention activities, for example, are aimed at reducing the number of crimes that occur for every area in which they are applied. For these purposes, the rate of decrease in the number of crimes is the prime focus. Similarly, after-school programs are aimed at neighborhoods where there is a high risk of crime, whether or not there is also a large population. In other words, for many purposes, the risk of crime or other types of incidents is of paramount importance, rather than the volume (i.e., absolute amount) of crime by itself. If the aim is to assess where there are high risk clusters, then the Nnh routine is not appropriate.

CrimeStat includes a Risk-adjusted Nearest Neighbor Hierarchical Clustering routine (or Rnnh) that defines clusters of points that are closer than what would be expected on the basis of a baseline population. It does this by dynamically adjusting the threshold distance in the Nnh routine according to the distribution of a second, baseline

variable. Unlike the Nnh routine where the threshold distance is constant throughout the study area (i.e., it is used to pair point irrespective of where they are within the area), the Rnnh routine adjusts the threshold distance according to what would be expected on the basis of the baseline variable. It is a risk measure, rather than a volume measure.

Dynamic Adjustment of the Threshold Distance

To understand how this works, think of a simple example. In a typical metropolitan area, there are more people living towards the center than in the periphery. There are topographical and social factors that might modify this (e.g., an ocean, a mountain range, a lake), but in general population densities are much higher in the center than in the suburbs. In the next chapter, we will examine the distribution of population and how it affects incidence of crime over an entire metropolitan area. If a different baseline variable were selected than population, for example, employment, one would generally find even higher concentrations since central city employment tends to be very high relative to suburban employment. Thus, if population or employment (or another variable that is correlated with population density) is taken as the baseline, then one would expect more people and, hence, more incidents occurring in the center rather than the periphery. In other words, all other things being equal, there should be more robberies, more burglaries, more homicides, more vehicle thefts, and more of any other type of event in the center than in the periphery of an urban area. This is just a by-product of urban societies.

Using this idea to cluster incidents together, then, intuitively, the threshold distance must be adjusted for the varying population densities. In the center, the threshold must be short since one would expect there to be more persons. Conversely, in the periphery - the far suburbs, the threshold distance must be a lot longer since there are far fewer persons per unit of area. In other words, dynamic adjustment of the threshold grouping distance means changing the distance inversely proportional to the population density of the location; in the center, a high density means a short threshold distance and in the periphery, a low density means a larger threshold distance.

Kernel Adjustment of the Threshold Distance

To implement this logic, CrimeStat overlays a standard grid and uses an interpolation algorithm, based on the kernel density method, to estimate the expected number of incidents per grid cell if the actual incident file was distributed according to the baseline variable. The next chapter discusses in detail the kernel density method and the reader should be familiar with the method before attempting to use the Rnnh routine. If not, the author highly recommends that Chapter 8 be read before reading the rest of this section.

Steps in the Rnnh Routine

The Rnnh routine works as follows:

1. Both a primary and secondary file are required. The primary file are the basic incidents (e.g., robberies) while the secondary file is the baseline

variable (e.g., population of zones; all crimes as a baseline; or another baseline variable). If the baseline variable are zones, the user must define both the X and Y coordinates as well as the variable assigned to the zone (e.g., population); the latter will typically be an intensity or weight variable (see Chapter 3).

2. A grid is defined in the reference file tab of the data setup section (see Chapter 3). The Rnnh routine takes the lower-left and upper-right limits of the grid, but uses a standard number of columns (50).

3. The area of the study is defined in the measurement parameters tab of the data setup section (see Chapter 3). If no area is defined, the routine uses the area of the entire grid.

4. The user checks the Risk-adjusted box under the Nnh routine. The risk variable is estimated with the parameters defined in the Risk Parameters box. These are the kernel parameters. Without going into detail, the user must define:

   A. The method of interpolation, which is the type of kernel used: normal, uniform, quartic, triangular, or negative exponential. The normal distribution is the default.

   B. The choice of bandwidth, whether a fixed or adaptive (variable) bandwidth is used. For a fixed bandwidth, the user must define the size of the interval (e.g., 2 miles). For an adaptive bandwidth, the user must define the minimum sample size to be included in the circle that defines the bandwidth. The default is an adaptive bandwidth with a minimum sample size of 100 incidents.

   C. The output units, which are points per unit of area: squared miles, squared nautical miles, squared feet, squared kilometers, or squared meters. The default is squared miles.

   D. Also, if an intensity or weight variable is used (e.g., the centroids of zones with population being an intensity variable), the intensity or weight box should be checked (be careful about checking both if there are both an intensity and a weight variable).

   Consult Chapter 8 for more detail about these parameters.

5. Once the baseline variable (the secondary file) is interpolated to the grid using the above parameters, it is converted into absolute densities (points per grid cell) and re-scaled to the same sample size as the primary incident file. This has the effect of making the interpolation of the baseline variable the same sample size as the incident variable. For example, if there are 1000 incidents in the primary file, the interpolation of the secondary file will be re-scaled so that all grid cells add to 1000 points, irrespective of how many units

the secondary variable actually represented. This creates a distribution for the primary file (the incidents) that is proportional to the secondary file (the baseline variable) if the primary file had the same distribution as the secondary file. It is then possible to compare the actual distribution of the incident variable with the expected distribution if it was similar to the baseline variable.

6. Once the risk parameters have been defined, the selection of parameters is similar to the Nnh routine with one exception.

   A. The threshold probabilities are selected with the scale bar. The probabilities are identical to those in Table 6.2.

   B. However, for each grid cell, a unique threshold distance is defined using formulas similar to 6.1 and 6.2. The difference is, however, that the formulas are applied to each grid cell with a unique distance for each grid cell (formulas 6.5-6.8):

$$\text{Mean Random Distance of Grid Cell } i = d(ran) = 0.5\ \text{SQRT}\left[\frac{A_i}{N_i}\right] \tag{6.5}$$

where A_i is the area of the grid cell and N_i is the estimated number of points from the kernel density interpolation. Thus, each grid cell has its own unique expected number of points, N_i, its own unique area, A_i (though, in general, all grid cells will have approximately equal areas), and, consequently, its own unique threshold distance.

$$\text{Confidence Interval for Mean Random Distance of Grid Cell } i = \text{Mean Random Distance of grid cell } i \pm t^* SE_{d(ran)}$$

$$= 0.5\ \text{SQRT}\left[\frac{A_i}{N_i}\right] \pm t\left[\frac{0.26136}{\text{SQRT}[N_i^2/A_i]}\right] \tag{6.6}$$

where the Mean Random Distance of Grid Cell i, A_i and Ni are as defined above, t is the t-value associated with a probability level in the Student's t-distribution (defined by the scale bar)

The lower limit of this confidence interval is

$$\text{Lower Limit of Confidence Interval for Mean Random Distance of Grid Cell } i = 0.5\ \text{SQRT}\left[\frac{A_i}{N_i}\right] - t\left[\frac{0.26136}{\text{SQRT}\left[N_i^2/A_i\right]}\right] \quad (6.7)$$

and the upper limit of this confidence interval is

$$\text{Upper Limit of Confidence Interval for Mean Random Distance} = 0.5\ \text{SQRT}\left[\frac{A_i}{N_i}\right] + t\left[\frac{0.26136}{\text{SQRT}\left[N_i^2/A_i\right]}\right] \quad (6.8)$$

C. In addition, the user defines a minimum sample size for each cluster, as with the Nnh routine.

6. The actual incident points are then identified by the grid cell that they fall within and the unique threshold distance (and confidence interval) for that grid cell. For each pair of points that are compared for distance, there is, however, asymmetry. The unique threshold distance for point A will not necessarily be the same as that for point B. The Rnnh routine, therefore, requires the distance between each pair of points to be the shorter of the two distances between the points.

7. Once pairs of points are selected, the Rnnh routine proceeds in the same way as the Nnh routine.

In other words, points are clustered together according to two criteria. First, they must be closer than a threshold distance. However, the threshold distance varies over the study area and is inversely proportional to the baseline variable. Only points that are closer together than would be expected on the basis of the baseline variable are selected for grouping. Second, clusters are required to have a minimum number of points with the minimum being defined by the user. The result are clusters that are more concentrated than would be expected, not just from chance but, from the distribution of the baseline variable. These are high risk clusters.

Area must be defined correctly

Note that it is very important that area be defined correctly for this routine to work. If the user defines the area on the measurement parameters page (see chapter 3), the Rnnh routine uses that value to calculate the area of each grid cell and, in turn, the grid-specific threshold distance. If the user does not define the area on the measurement parameters

page, the routine calculates the total area from the minimum and maximum X/Y values (the bounding rectangle) and uses that value to calculate the area of each grid cell and, in turn, the grid-specific threshold distance. In either case, the routine will be able to calculate a threshold distance for each grid cell and run the routine.

However, if the area units are defined incorrectly on the measurement parameters page, then the routine will certainly calculate the grid cell-specific threshold distances wrongly. For example, if data are in feet but the area on the measurement parameters page are defined in square miles, most likely the routine will not find any points that are farther apart than any of the grid cell threshold distances since each distance will be defined in miles. In other words, it is essential that the area units be consistent with the data for the routine to properly work.

Use kernel bandwidths that produce stable estimates

Another concern is that the bandwidth for the baseline variable be defined as to produce a stable density estimate of the variable. Be careful about choosing a very small bandwidth. This could have the effect of creating clusters at the edges of the study area or very large clusters in low population density areas. For example, in low population density areas, there will probably be fewer persons or events than in more built-up areas. This will have the effect on the Rnnh calculation of producing a very large matching distance. Points that are quite far apart could be artificially grouped together, producing a very large cluster. Using a larger bandwidth will produce a more stable average.

Example 2: Simulated Rnnh Clustering

To illustrate the logic of the Rnnh routine, a simulated example is presented. Twenty-seven points were assigned to three groups in the Baltimore metropolitan region (Figure 6.11). The 27 points were grouped in a similar pattern, but one was placed in the center of the metropolitan region (near downtown Baltimore) while the other two were placed in less populated areas. The Nnh and Rnnh routines were compared with these data. One would expect the Nnh routine to cluster the 27 points into three groups whereas the Rnnh routine should cluster only 18 of the points into two groups. The reason for the lack of a third group is that one would expect a high number of incidents in the center; consequently, it is not high relative to the underlying baseline population. Figures 6.12 and 6.13 show exactly this solution.

In other words, the Nnh routine clusters points together irrespective of the distribution of the baseline population whereas the Rnnh routine clusters points together relative to the baseline population.

Rnnh Output Files

The output files are similar to the Nnh routine. The Rnnh routine has three outputs. First, final seed locations of each cluster and the parameters of the selected standard deviational ellipse are calculated for each cluster. These can be output to a '.dbf'

Figure 6.11:

Incidents in Relation to Population: Baltimore Region 1990

Incident Location and Persons Per Square Mile

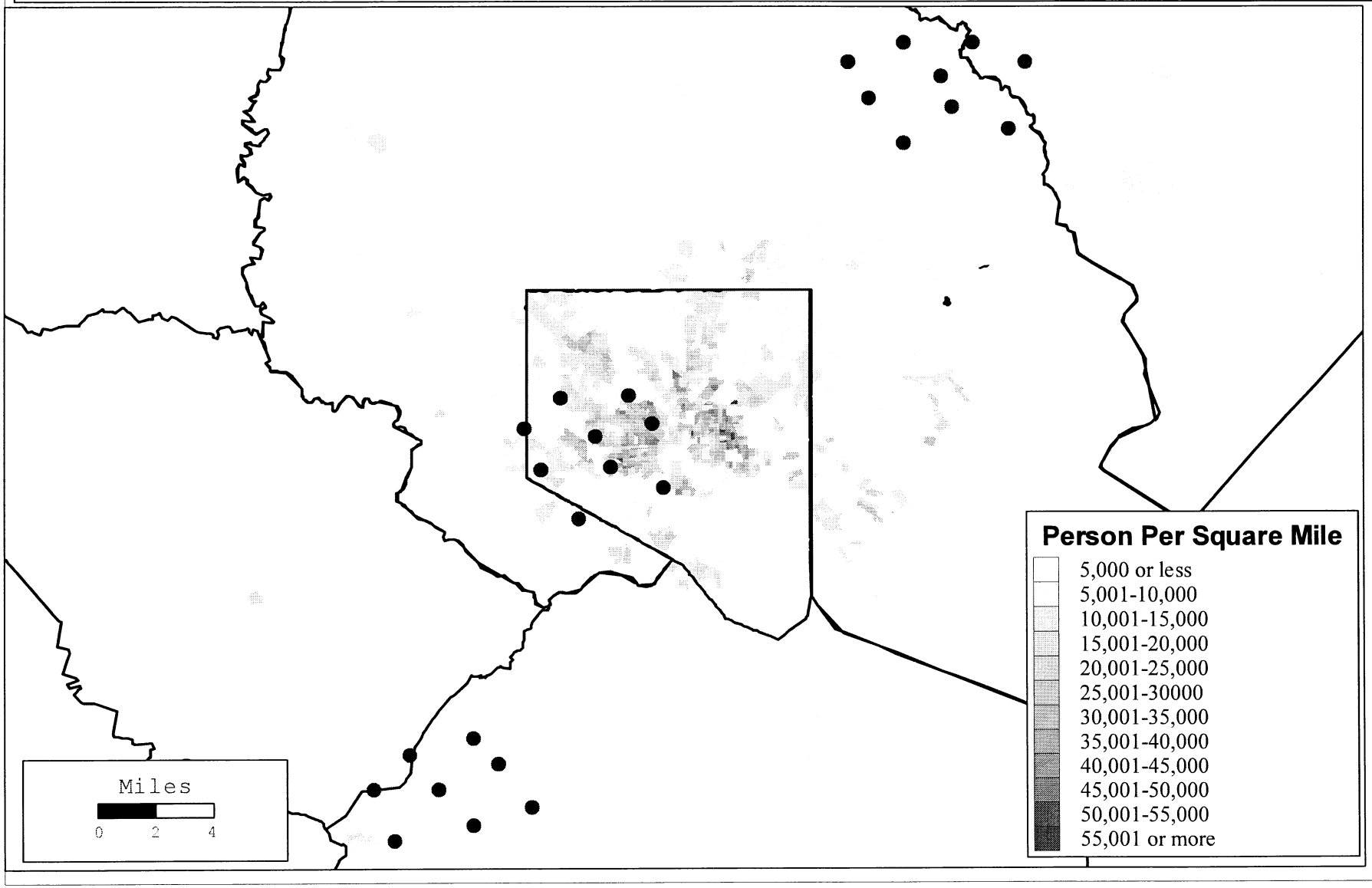


Figure 6.12:

Nearest Neighbor Clustering of Incidents

Nnh Clusters and Incident Locations

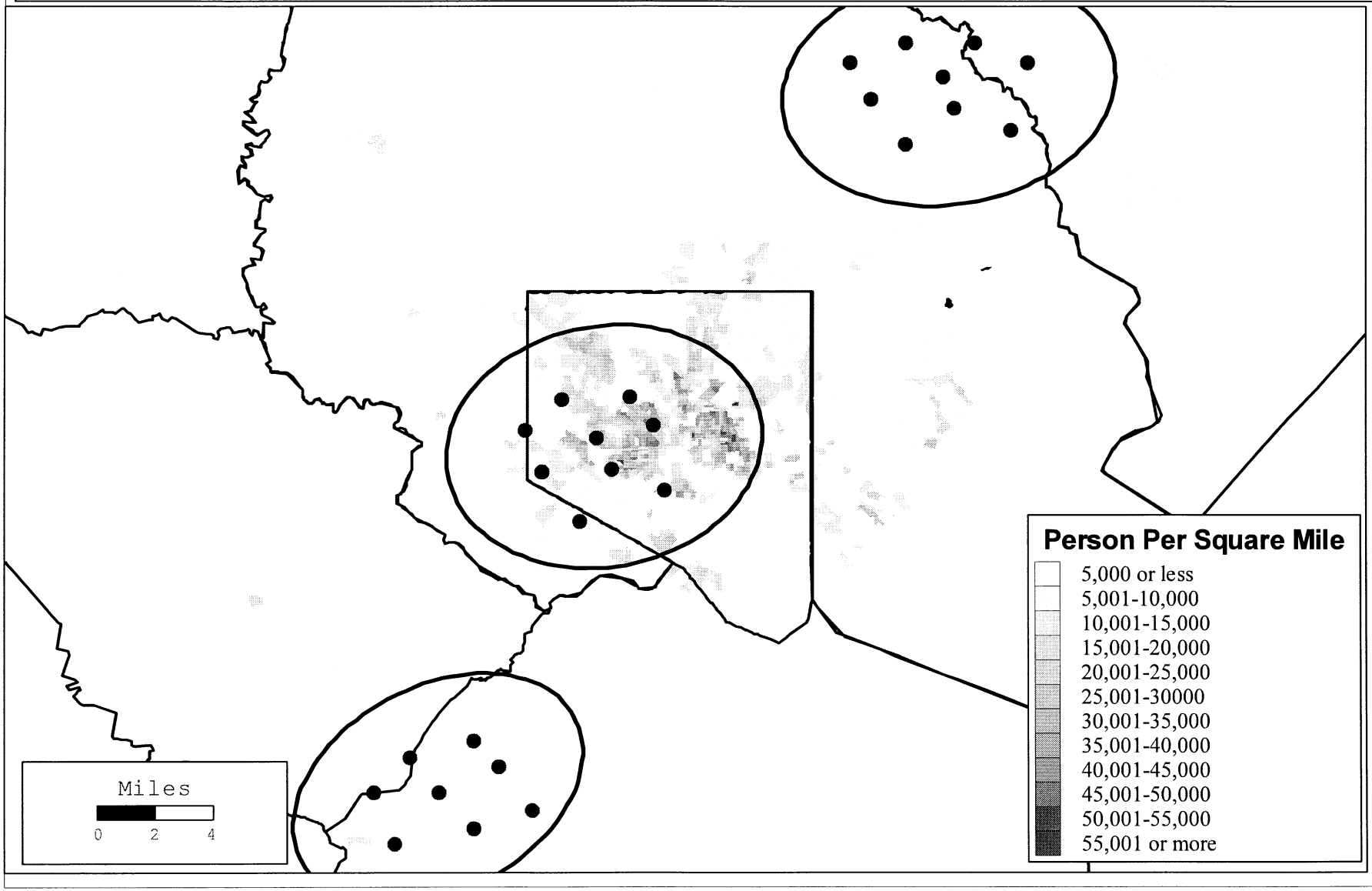
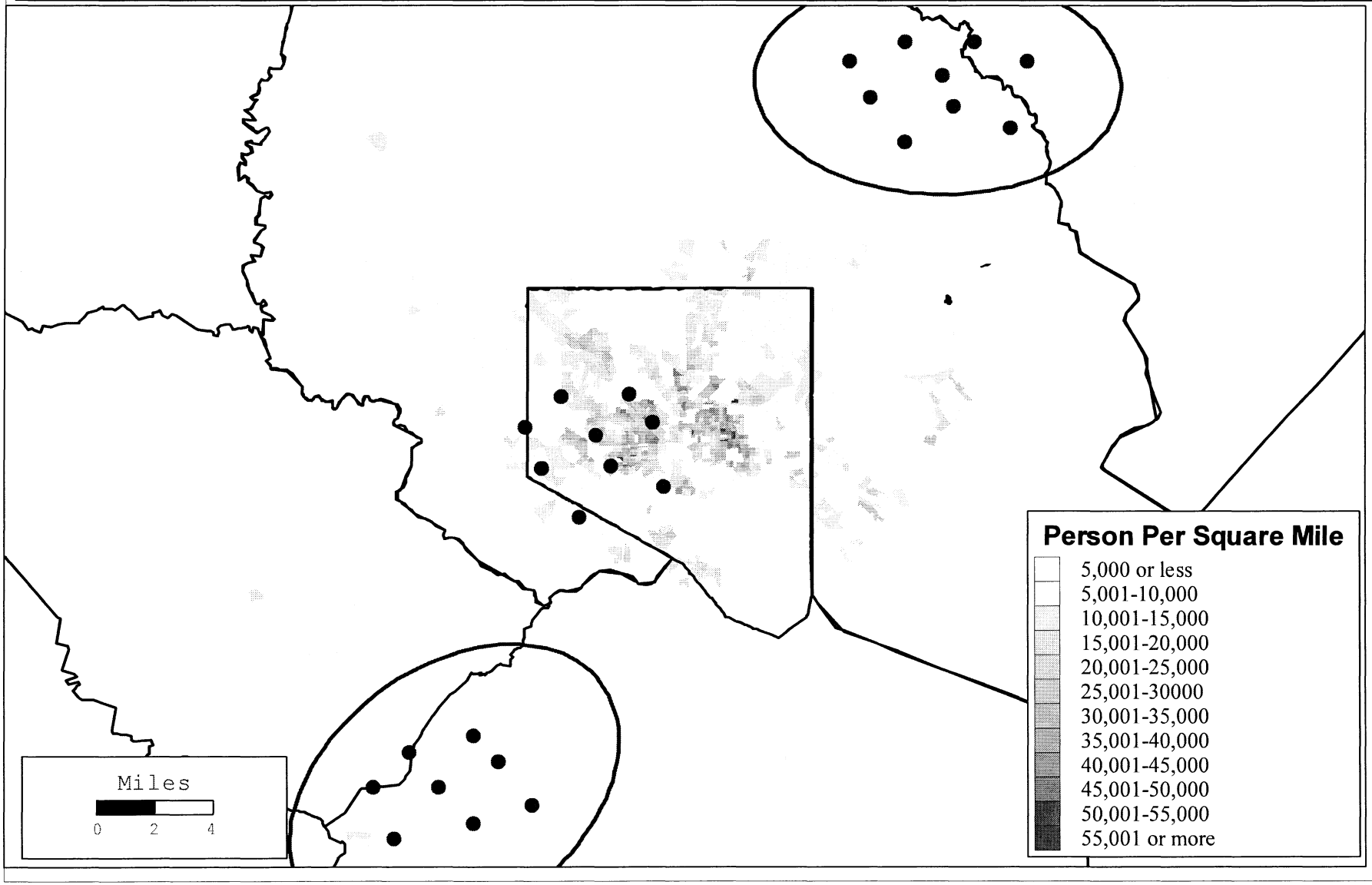


Figure 6.13:

Risk-Adjusted Nearest Neighbor Clustering of Incidents Relative to Population

Rnnh Clusters and Incident Locations



file or saved as a text ('.txt') file. Only 45 of the seed locations are displayed on the screen. The user can scroll down or across by adjusting the horizontal and vertical slider bars and clicking on the Go button.

Second, for each order that is calculated, CrimeStat calculates the mean center of the cluster. This can be saved as a '.dbf' file. Third, either standard deviational ellipses or convex hulls of the clusters can be saved in ArcView '.shp', MapInfo '.mif' or Atlas*GIS '.bna' formats. Again, the convex hulls display polygons around the incidents whereas the ellipses are determined by the number of standard deviations to be calculated (see above). In general, use a 1X standard deviational ellipse since 1.5X or 2X standard deviations can create an exaggerated view of the underlying cluster. On the other hand, for a regional view, a one standard deviational ellipse may not be very visible. The user has to balance the need to accurately display the cluster compared to making it easier for a viewer to understand its location.

Because there are also orders of clusters (i.e., first-order, second-order, etc.), there is a naming convention that distinguishes the order.

For the ellipses, the convention is

Rnnh<O><username>

where O is the order number and username is a name provide by the user. Thus,

Rnnh1robbery

are the first-order clusters for a file called 'robbery' and

Rnnh2burglary

are the second-order clusters for a file called 'burglary'. Within files, clusters are named

Rnnh<O>Ell<N><username>

whereO is the order number, N is the cluster number and username is the user-defined name of the file. Thus,

Rnnh1Ell10robbery

is the tenth cluster within the first-order clusters for the file 'robbery' while

Rnnh2Ell1burglary

is the first cluster within the second-order clusters for the file 'burglary'.

For the convex hulls, the cluster numbers are the same as the ellipses but the prefix name is output with a 'CRNNH1' prefix for the first-order clusters, a 'CRNNH2' prefix for

the second-order clusters, and a 'CRNNH3' prefix for the third-order clusters. Higher-order clusters will index only the number.

Example 3: Rnnh Clustering of Vehicle Thefts

A second example is the clustering of 1996 Baltimore vehicle thefts relative to the 1990 population of census block groups. The test is for clusters of vehicle thefts that are more concentrated than would be expected on the basis of the population distribution.[6] Using the default threshold probabilities and a minimum sample size per cluster of 25, the Rnnh routine identified five first-order and one second-order cluster (Figure 6.14); the incidents are not shown. As seen, there are only five clusters, most of which are peripheral to the downtown area.

Compare this distribution with the results of the Nnh on the same data, using the same parameters (Figure 6.15). The Nnh found 28 first-order clusters and two second-order clusters. As expected, they are more concentrated in the center. Note that there are far fewer clusters identified in the Rnnh routine than in the Nnh. Many of the clusters in the Nnh routine are due to a higher concentration of population. Once this is normalized, one finds that there are only a few areas of very high risk for vehicle theft. In other words, the Rnnh routine identifies areas of high risk for vehicle theft whereas the Nnh routine identifies areas of high volume for vehicle theft.

Simulating Statistical Significance

Because the sampling distribution of the clustering method is not known, the Rnnh routine allows Monte Carlo simulations to approximate confidence intervals, similar to the Nnh routine (Dwass, 1957; Barnard, 1963). The output is identical to the Nnh routine. Essentially, it produces approximate confidence intervals for the number of first-order clusters, the area of clusters, the number of points in each cluster, and the density of each cluster. Second- and higher-order clusters are not simulated since their structure depends on the first-order clusters. The user can see whether the first-order cluster structure is different than that which is produced by a random distribution. See the notes above under Nnh for more details. Table 6.4 shows the output for 1996 Baltimore County robberies with the default search threshold and a minimum sample size of 20 incidents.

The results also show those obtained from 1000 Monte Carlo simulations. There were seven first-order clusters and one second-order cluster. Looking at the Monte Carlo simulations, the two most critical parameters are the number of first-order clusters found and the density of the clusters. In the simulation, the minimum number of clusters found under these conditions (i.e., with the default threshold distance and a minimum sample size of 20 incidents) was one while the maximum number was five. The 95th percentile was four incidents. Since the Rnnh routine produced seven first-order clusters, the routine has identified more clusters than would normally be expected on the basis of chance. Looking at the density estimates from the simulation, the maximum density was 11.913282 and the 95th percentile was 7.365212. Since all seven first-order clusters had densities higher than

Figure 6.14:

1996 Metropolitan Baltimore Vehicle Theft

Risk-Adjusted Nearest Neighbor Clusters

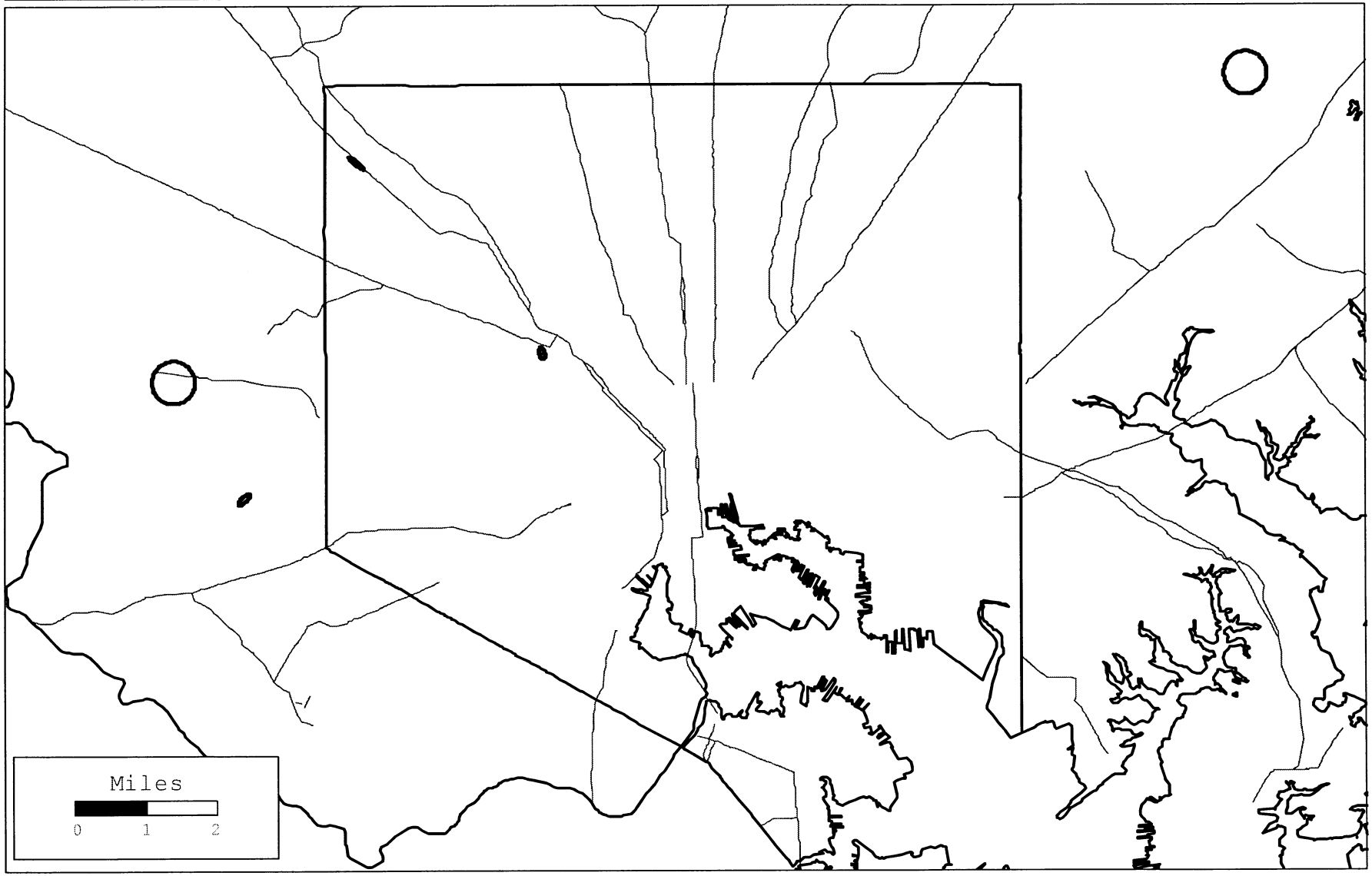


Figure 6.15:

1996 Metropolitan Baltimore Vehicle Theft

Nearest Neighbor Clusters

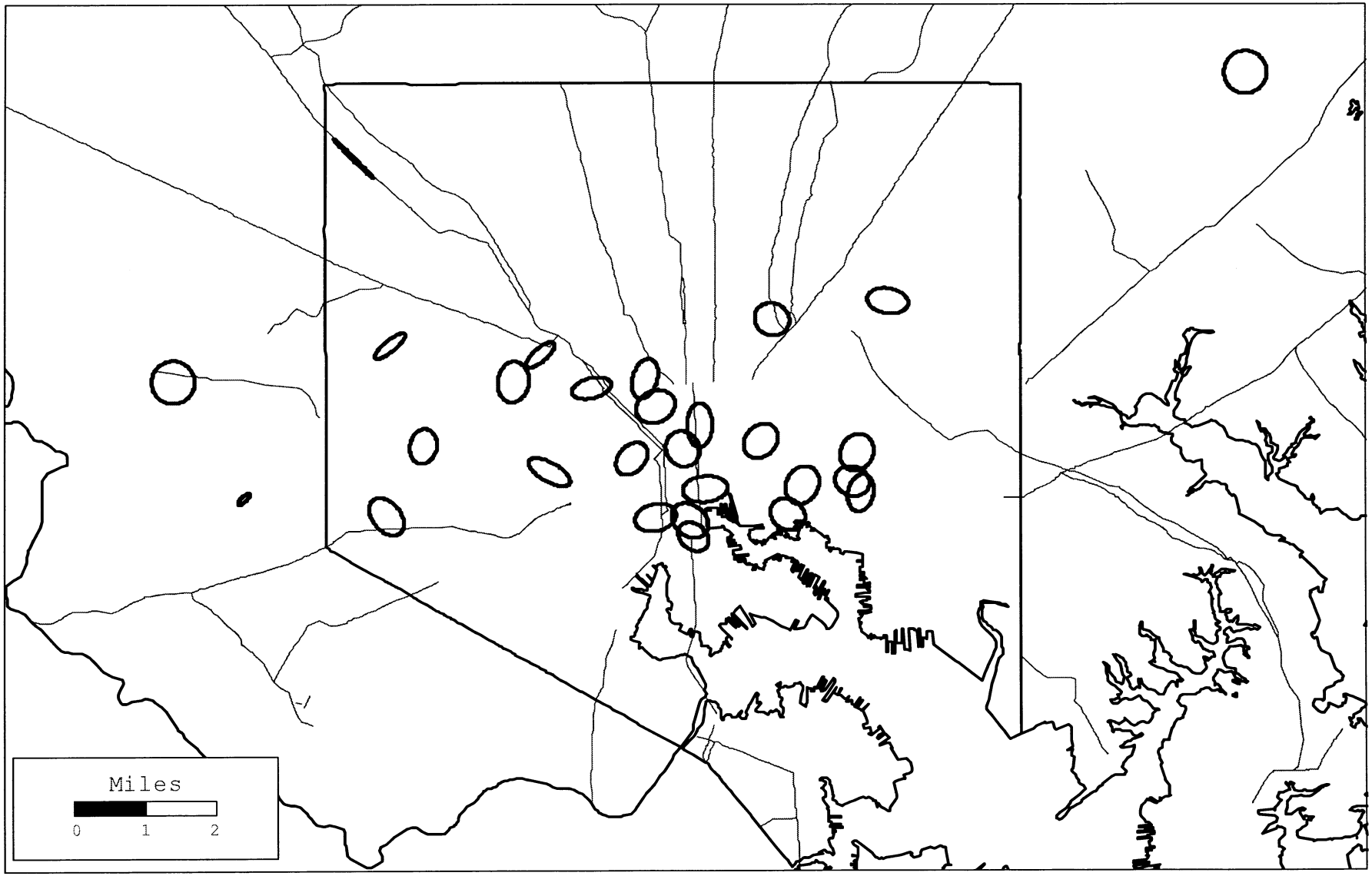


Table 6.4

Risk-adjusted Clustering of Baltimore County Robberies: 1996

```
Risk-Adjusted Nearest Neighbor Hierarchical Clustering:
-------------------------------------------------------

    Sample size..................: 1181
    Likelihood of grouping
      pair of points by chance...: 0.50000 (50.000%)
    Z-value for confidence
      interval...................: 0.000
    Measurement type.............: Direct
    Output units.................: Miles, Squared Miles, Points per Squared Miles
    Clusters found...............: 8
    Simulation runs..............: 1000

    Displaying 8 ellipse(s) starting from 1
```

Order	Cluster	Mean X	Mean Y	Rotation	X-Axis	Y-Axis	Area	Points	Density
1	1	-76.44973	39.31523	73.89169	0.19429	0.09230	0.05634	31	550.251866
1	2	-76.60194	39.40076	4.40641	0.12272	0.12929	0.04984	23	461.446220
1	3	-76.78279	39.36184	62.61813	0.24605	0.15511	0.11990	26	216.852324
1	4	-76.73157	39.34387	4.30498	0.08916	0.07321	0.02051	24	1170.341418
1	5	-76.44539	39.30523	13.63299	0.19639	0.11154	0.06882	20	290.622622
1	6	-76.75368	39.31132	89.56994	0.19748	0.22914	0.14216	22	154.753006
1	7	-76.73132	39.28897	11.83419	0.09359	0.18312	0.05384	21	390.033756
2	1	-76.74984	39.32650	66.40941	4.19556	1.63703	21.57723	4	0.185381

```
    Distribution of the number of clusters found in simulation (percentile):
```

Percentile	Clusters	Area	Points	Density
min	1	1.67880	20	1.648432
0.5	1	2.36257	20	1.874836
1.0	1	2.51219	20	1.996056
2.5	1	2.67031	20	2.208136
5.0	1	2.98150	20	2.372246
95.0	4	13.57660	50	7.365212
97.5	4	13.95390	53	7.932653
99.0	5	14.34076	56	8.643887
99.5	5	14.60388	58	9.595312
max	5	15.41259	67	11.913282

the 95th percentile, the density of these clusters is greater than what would normally be expected on the basis of chance. In other words, the routine has identified more clusters and higher density clusters than would be expected on the basis of chance.

Guidelines for Selecting Parameters

The guidelines for selecting parameters in the Rnnh routine are similar to the Nnh except the user must also model the baseline variable using a kernel density interpolation. The process is a little like tuning a shortwave radio, adjusting the dial until the signal is

detected. We suggest that the user first develop a good density model for the baseline variable (see Chapter 8). The user has to develop a trade-off between identify areas of high and low population concentration to produce an estimate that is statistical reliable (stable).

There are two types of 'fine tuning' that have to go on. First, the 'background' variation has to be tuned (the baseline 'at risk' variable). This is done through the kernel density interpolation. If too narrow a bandwidth is selected, the density surface will have numerous undulations with small 'peaks' and 'valleys'; this could produce unreal and unstable risk estimates. A grid cell with a very small density value could produce an extremely large threshold distance whereas a grid cell with a very low density could produce an extremely small threshold distance. Conversely, if too large a bandwidth is selected, the density surface will not differentiate very well and each grid cell will have, more or less, the same threshold distance. In this case, the Rnnh routine would yield a result not very different from the Nnh routine.

Second, there is tuning of the clusters themselves through the threshold adjustment and minimum size criteria. If a large threshold probability is selected, too many incidents may be grouped; conversely, if a small threshold probability is selected, the result may be too restrictive. Similarly, if a small minimum sample size for clusters is used, there could be too many clusters whereas the opposite will happen if a large minimum sample size is chosen (i.e., zero clusters). The user must experiment with both these types of adjustment to produce a sensible cluster solution that captures the areas of high risk, but no more.

Limitations of the Technique

There are some technical limitations that the Rnnh routine shares with the Nnh routine. First, the method only clusters incidents (points); a weighting or intensity variable will have no effect. Second, the size of the grouping area is dependent on the sample size if the confidence interval around the mean random distance is used as the threshold distance criteria. However, since the threshold distance is adjusted dynamically, this has less effect than in the Nnh since it is now a relative comparison rather than an absolute distance.

Third, there is arbitrariness in the technique due to the minimum points rule. Different users could define the minimum differently, which could lead to different conclusions about the location of high risk clusters. Finally, unique to the Rnnh, the method requires both an incident file (the primary file) and a baseline file (the secondary file. It cannot work on calculated rates (e.g., incidents per capita by zones). For the latter, the user should look at techniques such as the SatScan method (Kulldorff, 1997).

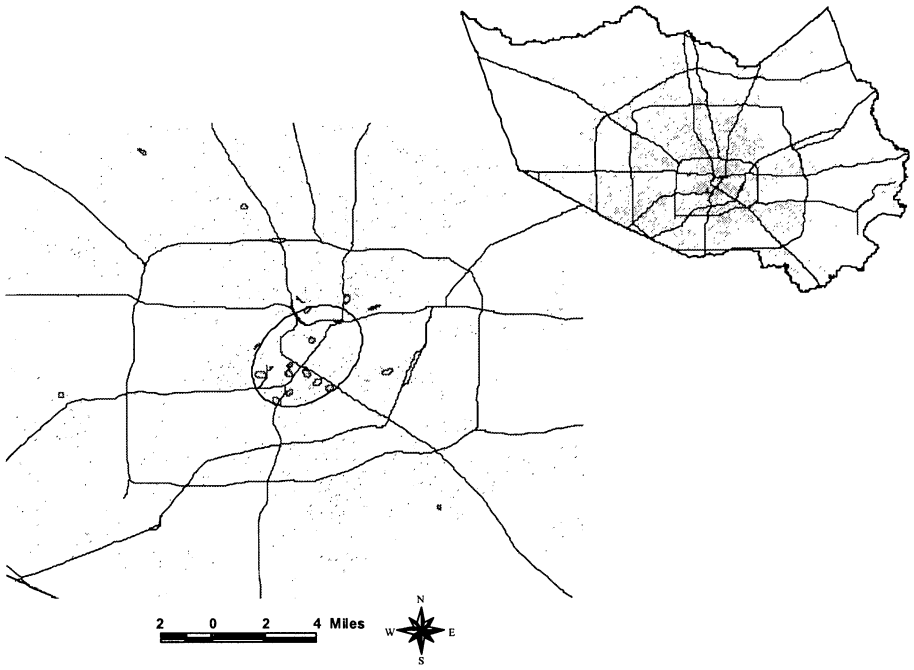
Nevertheless, the Rnnh routine is a useful technique for identifying clusters that are more concentrated than would be expected on the basis of the population distribution.

Risk Adjusted Nearest Neighbor Hierarchical Clustering of Tuberculosis Cases in Harris County, Texas: 1995 to 1998

Matthew L. Stone, MPH
Center for Health Policy Studies
University of Texas-Houston, School of Public Health-Houston, Texas

Data was collected from an ongoing, population-based, active surveillance and molecular epidemiology study of tuberculosis cases reported to the City of Houston Tuberculosis Control Office from October 1995 to September 1998. During this time, 1774 cases of tuberculosis were reported and 1480 of those who participated in this study were successfully geocoded.

CrimeStat was used to make an initial survey of potential hot spot areas of tuberculosis cases where more focused TB control efforts could be implemented. Given a .05 level of significance for grouping a pair of points by chance and a minimum of five cases per cluster, 24 first-order clusters and one second-order cluster were detected after adjusting for the underlying population. Most first-order clusters were detected in the center of Harris County, including the metropolitan downtown area. By adjusting for the underlying population, the clusters identify areas with higher than average TB incidence. Some of these clusters are homeless shelters as many homeless persons are particularly prone to TB.



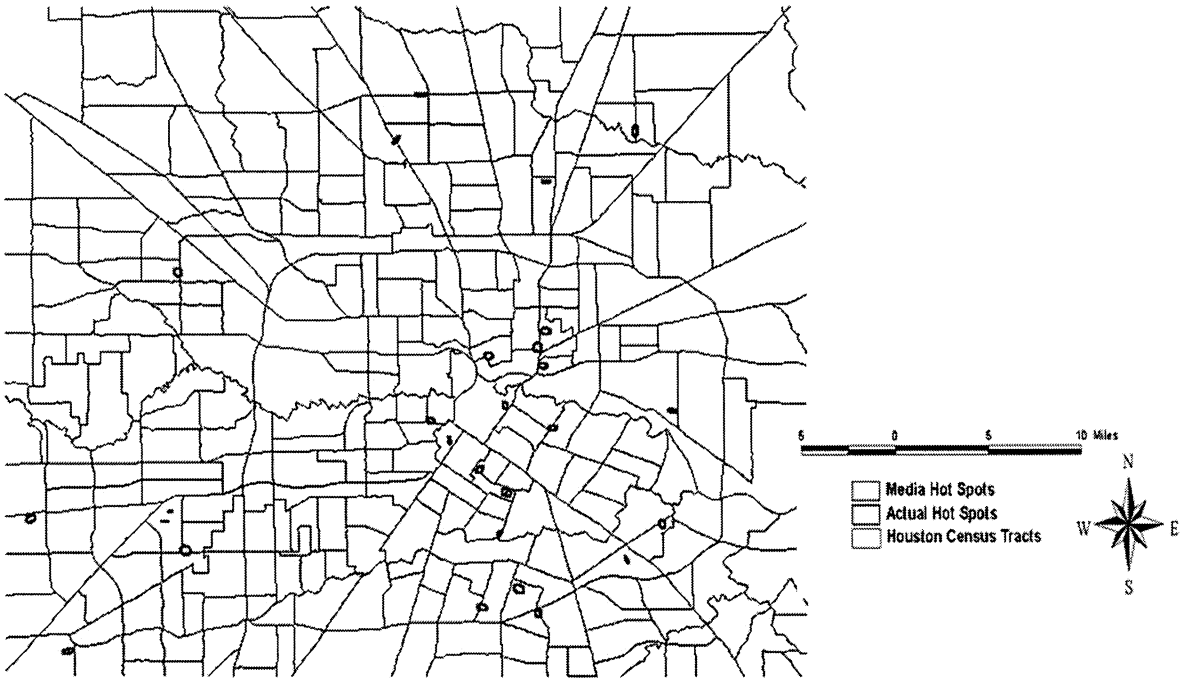
Using Risk Adjusted Nearest Neighbor Hierarchical Clustering to Compare Actual and Media Hotspots of Homicide

Derek J. Paulsen
Department of Criminal Justice and Police Studies
Eastern Kentucky University

Crimestat offers an excellent method for determining risk adjusted hot spots of crime incidents within a jurisdiction. Risk-adjusted nearest neighbor hierarchical spatial clustering (Rnnh) is a spatial clustering routine that groups points together based on both proximity to other points and the distribution of a baseline variable. In this example two different Rnnh analyses were conducted and compared for homicides in Houston, Texas. The first involves homicide incident locations adjusted for the population of each census tract, while the second involves incidents that were covered in the newspaper adjusted for the homicide rate of each census tract. The purpose of this analysis is to determine if there are differences in the spatial clustering of actual homicide incidents and those that are covered in the newspaper.

The preferences for the analysis were the same for both Rnnh analyses. For the primary file (homicide incidents & incidents covered in the newspaper) the pair probability search radius was set at .01, with a minimum of 10 points per cluster. For the secondary file (population & homicide rate), a quartic kernel density interpolation was used with an adaptive bandwidth and a minimum sample size of 100. Importantly, the analysis showed that media hot spots and actual hot spots do not coincide. Media coverage showed homicides to be concentrated in different areas than they are actually concentrated.

Actual Homicide Hot Spots vs. Media Coverage Hot Spots in Houston Texas



Endnotes for Chapter 6

1. The output in table 6.1 has been formatted. CrimeStat only outputs an Ascii file. In this case, the Ascii file was pasted into Word Perfect®, the word processing program used for this manual, and was then formatted so that the underscore was consistent with the title words and the columns lined up.

2. In the statistical literature, this type of statistic is known as a spatial scan with a fixed circular window (Kulldorff, 1997; Kulldorff and Nagarwalla, 1995). However, our emphasis here is on defining approximate point locations where there is either measurement error or very small locational differences. In this sense, the term 'fuzzy' is more similar to the classification literature where imprecise boundaries exist and an incident can belong to two or more groups (Bezdek, 1981; McBratney and deGruijter, 1992; Xie and Beni, 1991).

3. This is the next highest degree of freedom in the Student's t-table below infinity.

4. The particular steps are as follows:

   A. All distances between pairs of points are calculated, using either direct or indirect distance as defined on the measurements parameters page. The matrix is assumed to be symmetrical, that is the distance between A and B is assumed to be identical to the distance between B and A.

   B. The mean expected random distance is calculated using formula 5.2 and the threshold distance (the confidence interval for the corresponding t) is calculated using formulas 6.2 and 6.3 depending on whether it is a lower or upper confidence interval. The particular interval is selected by user on the slide bar.

   C. All distance pairs smaller than the threshold distance are selected for clustering.

   D. For each incident point, the number of distances to other points that are smaller than the threshold distance are counted and placed in a reduced matrix. Any incident point which does not have another point within the threshold distance is not clustered. Any distance that is greater than the threshold distance is not considered for clustering.

   E. All points in the reduced matrix are sorted in descending order of the number of distances to other points shorter than the threshold distance, and the incident point with the largest number of below threshold distances is selected for the initial seed of the first cluster.

   F. All other incidents that are within the threshold distance of the initial seed point are selected for cluster 1.

G. The number of points within the cluster are counted. If the number is equal to or greater than the minimum specified, then the cluster is kept. If the number is less than the minimum specified, then the cluster is dropped.

H. For those clusters that are kept, the center of minimum distance is calculated for each to identify the cluster center.

I. The clustered points are removed from further clustering.

J. Of the remaining points, the incident point with the largest number of distances to other points shorter than the threshold distance is selected for the initial seed the second cluster.

K. All other points which are within the threshold distance of the first cluster seed point are selected for cluster 2.

L. The mean center of these selected points is calculated to identify the cluster center.

M. These points are removed from further clustering.

N. Steps J through M are repeated for all remaining points in the reduced matrix until no more points are remaining in the reduced matrix or until there are fewer than the specified minimum number of points for those remaining in the reduced matrix.

5. The steps are as follows:

A. Using the same p-values selected in the first-order, the mean random expected distance is calculated. However, the sample size is the number of first-order clusters identified, not the original number of points. Thus, the threshold distance is calculated by

$$\text{Confidence Interval for Second-order Mean Random Distance} = 0.5\ \text{SQRT}\left[\frac{A}{M}\right] +/- \ t\left[\frac{0.26136}{\text{SQRT}\left[M^2/A\right]}\right] \quad \text{repeat} \qquad (6.1)$$

where A is the area of the region and M is the number of first-order clusters identified during first-order clustering (i.e., not N). Thus, there is a different threshold distance for the second-order clustering. The t-value specified in the first-order clustering is maintained for second- and higher-order clustering.

B. All distances between first-order cluster centers are calculated and only those that are smaller than the second-order threshold distance are selected for

second-order clustering.

C. If there are no distances between first-order cluster centers that are smaller than the second-order threshold distance, then the clustering process ends.

D. If there are distances between first-order cluster centers that are smaller than the second-order threshold distance, then the steps specified in endnote 3 are repeated to produce second-order clusters. A minimum of four first-order clusters is required to allow a second- or higher-order cluster.

E. If there are second-order clusters, then this process is repeated to either extract third-order clusters or to end the clustering process if no distances between second-order cluster centers are smaller than the (new) third-order threshold distance or if there are fewer than four new seeds in the cluster.

F. The process is repeated until no further clustering can be conducted, either all sub-clusters converge into a single cluster or the threshold distance criteria fails or there are fewer than four seeds in the higher-order cluster

6. It is not an exact risk test since we are comparing 1996 vehicle thefts with 1990 population. It is an approximate risk test.

Chapter 7
'Hot Spot' Analysis II

This chapter continues the discussion of hot spots. Three additional routines are discussed: ICJIA's STAC routine (discussed by Richard and Carolyn Block), the K-means routine, and Anselin's Local Moran. Figure 7.1 displays the 'Hot Spot' Analysis II page. The first of these routines, the Spatial and Temporal Analysis of Crime (STAC), was developed by the Illinois Criminal Justice Information Authority and integrated into *CrimeStat* in version 2. The second routine - K-means, is a partitioning technique. The third technique - Anselin's Local Moran, is a zonal hot spot method. We'll start first with STAC, and who better to explain it than the authors of the routine, Richard and Carolyn Block.

Spatial and Temporal Analysis of Crime (*STAC*)

by

Richard Block
Professor of Sociology
Criminal Justice
Loyola University
Chicago, IL

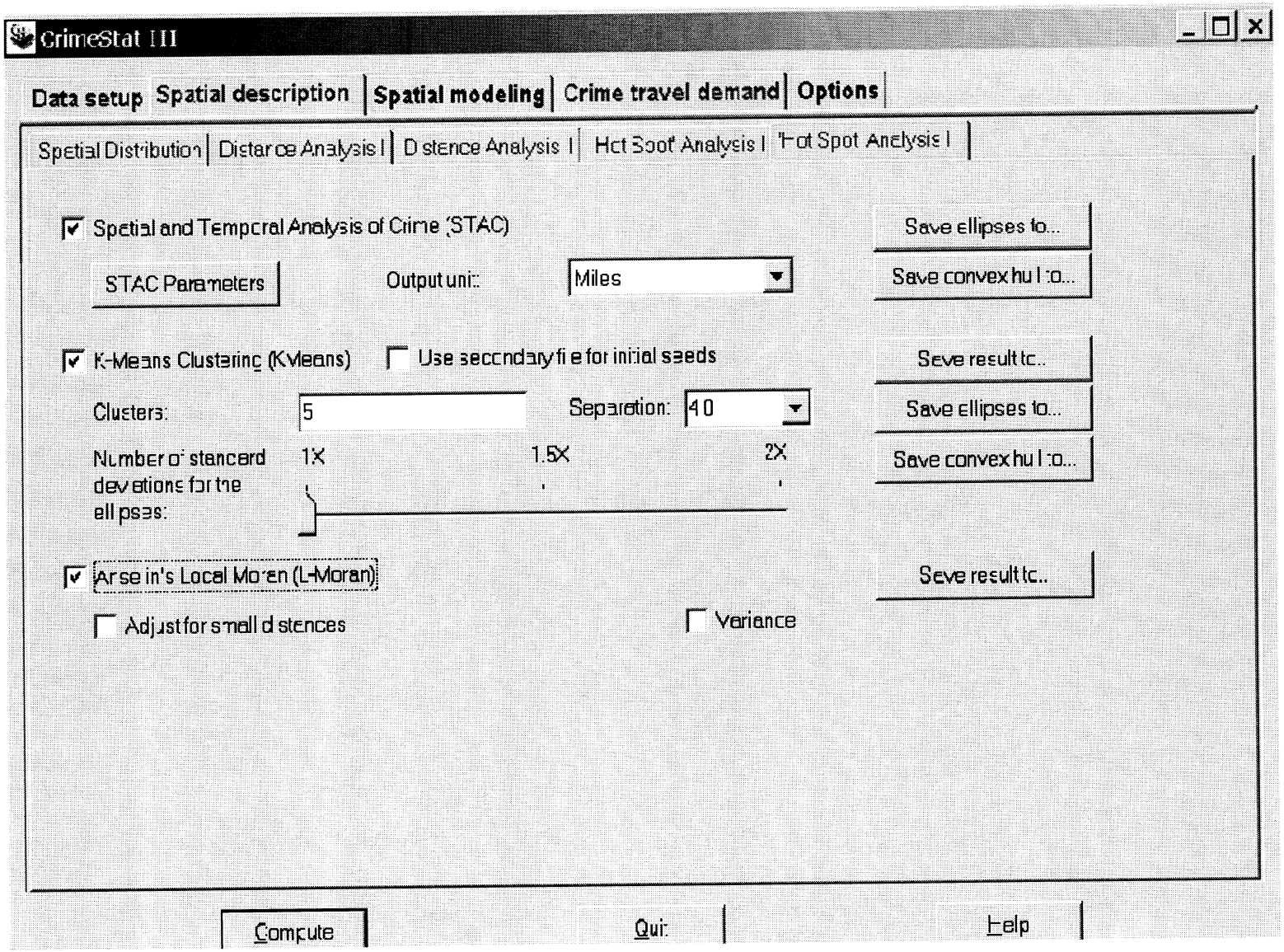
Carolyn Rebecca Block
Senior Research Analyst
Illinois Criminal Justice Information Authority
Chicago, IL

The amount of information available in an automated pin map can be enormous. When geographic information systems were first introduced into policing, there were few ways to summarize the huge reservoir of mapped information that was suddenly available. In 1989, police departments in Illinois asked the Illinois Criminal Justice Information Authority to develop a technique to identify Hot Spot Areas (the densest clusters of points on a map). The result was STAC, the first crime hot spot program.[1] Through the years, "bells and whistles" have been added to STAC, but the algorithm has remained essentially the same. STAC is a quick, visual, easy-to-use program for identifying Hot Spot Areas.

The STAC Hot Spot Area routine in *CrimeStat* searches for and identifies the densest clusters of incidents based on the scatter of points on the map. The STAC Hot Spot Area routine creates areal units from point data. It identifies the major concentrations of points for a given distribution. It then represents each dense area by the

STAC is a scan-type clustering algorithm in which a circle is repeatedly laid over a grid and the number of points within the circle are counted (Openshaw, Charlton, Wymer and Craft, 1987; Openshaw, Craft, Charlton, and Birch, 1988; Turnbull, Iwano, Burnett, Howe, and Clark, 1990; Kuldorff, 1995). It, thus, shares with those other scan routines the property of multiple tests, but it differs in that the overlapping clusters are combined into larger cluster until there are no longer any overlapping circles. Thus, STAC clusters can be of differing sizes. The routine, therefore, combines some elements of partitioning clustering (the search circles) with hierarchical clustering (the aggregating of smaller clusters into larger clusters).

Figure 7.1: 'Hot Spot' Analysis II Screen



The STAC Hot Spot Area routine in *CrimeStat* searches for and identifies the densest clusters of incidents based on the scatter of points on the map. The STAC Hot Spot Area routine creates areal units from point data. It identifies the major concentrations of points for a given distribution. It then represents each dense area by either a standard deviational ellipse or a convex hull, or both (see chapter 4). The boundaries of the ellipses or convex hulls can easily be displayed as mapped layers by standard GIS software.

STAC is not constrained by artificial or political boundaries, such as police beats or census tracts. This is important, because clusters of events and places (such as drug markets, gang territories, high violence taverns, or graffiti) do not necessarily stop at the border of a police beat. Also, shading over an entire area may make it seem that the whole neighborhood is high-crime (or low-crime), even though the area may contain only one or two dense pockets of crime. Therefore, area-shaded maps could be misleading. In contrast, STAC Hot Spot Areas are based on the actual clusters of events or places on the map.

STAC is designed to help the crime analyst summarize a vast amount of geographic information so that practical policy-related issues can be addressed, such as resource allocation, crime analysis, beat definition, tactical and investigation decisions, or development of intervention strategies. An immediate concern of a law enforcement user of automated pin maps is the identification of areas that contain especially dense clusters of events. These pockets of crime demand police attention and could indicate different things for different crimes. For instance, a grouping of Criminal Damage to Property offenses could indicate gang activity. If motor vehicle thefts consistently cluster in one section of town, it could point to the need to change patrol patterns and procedures.

To take an example, Figure 7.2 shows the location of the seven densest Hot Spot Areas of street robbery in 1999 in Chicago. Four of the seven span the boundaries of police districts and two cover only a small part of a larger district. In a shaded area map, these dense clusters of robbery might be not easily identifiable. An area that is really dense might appear to be low-crime because it is divided by an arbitrary boundary. Using a shaded areal map aggregating the data within each district would give a general idea of the distribution of crime over the entire map, but it would not tell exactly where the clusters of crime are located.

For example, figure 7.3 zooms in on Hot Spot Area 4 (the northernmost Hot Spot Area in Figure 7.2). Hot Spot Area 4 covers parts of two districts (shown by a pink boundary line in figure 7.2) There are also four beats (shown by blue boundary lines). The shaded map indicates many incidents in beat 2311, but few in beats 2312, and 2313.[2] The incident distribution indicates that while few incidents occurred overall in 2312 and 2313, most of the incidents that did occur were near to beat 2311. Incidents in beat 2311 mainly occurred on its eastern boundary. Portions of the beat were relatively free from street robbery. The Hot Spot Area identifies this clustering that spans beats and districts. Hot Spot Areas that overlap beat and district boundaries might indicate to patrol officers in these neighboring areas that they should coordinate their efforts in combating crime.

Figure 7.2: STAC Hot Spots for 1999 Street Robberies

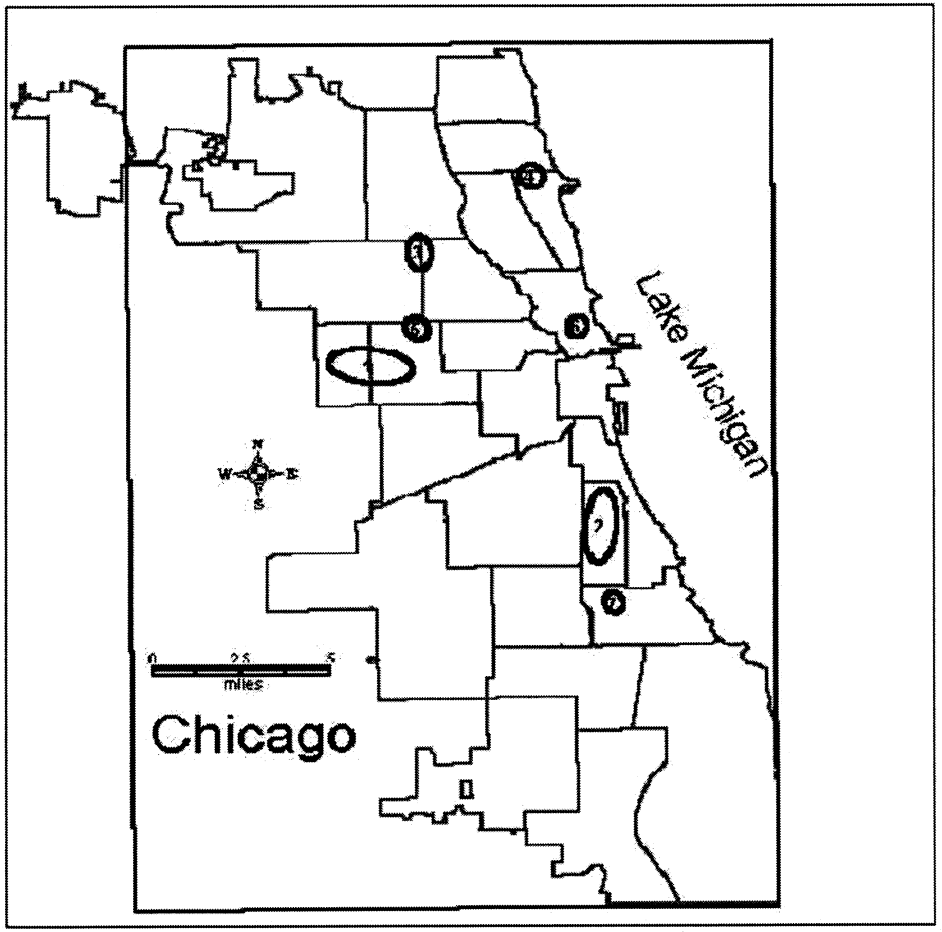
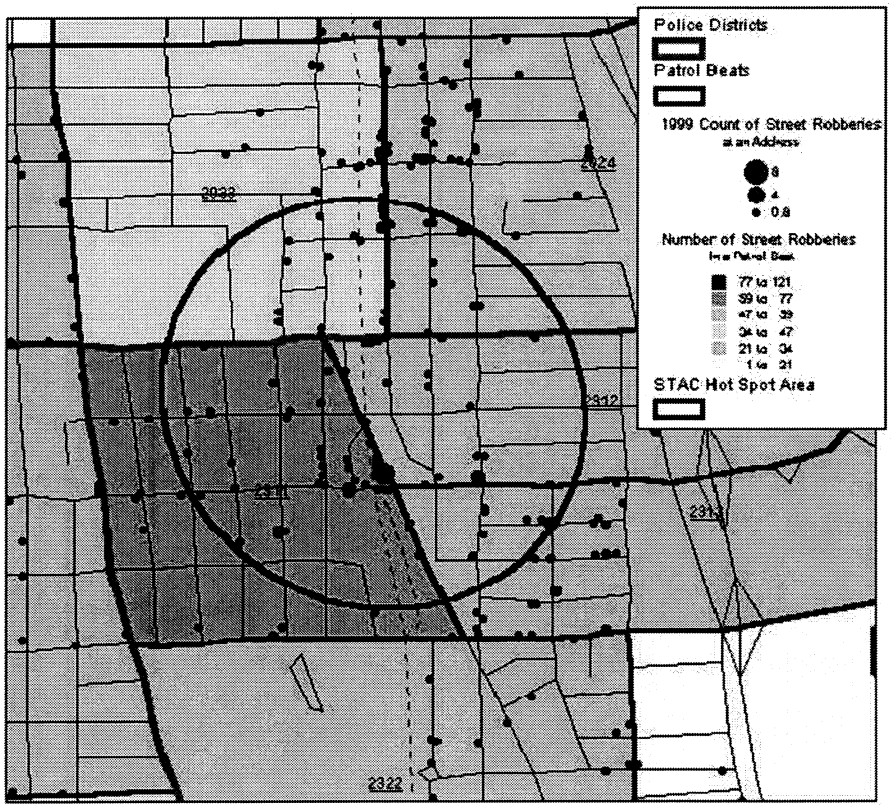
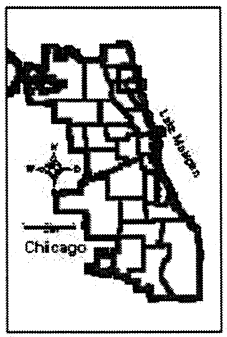


Figure 7.3: STAC 1999 Street Robbery Hot Spot Area 4



Location of 1999 Street Robberies
Chicago: Mid Northside



Source: Chicago Police Department

How STAC Identifies Hot Spot Areas

The following procedures identifies hot spots in STAC. The program implements a search algorithm, looking for Hot Spot Areas.

1. STAC lays out a 20 x 20 grid structure (triangular or rectangular, defined by the user) on the plane defined by the area boundary (defined by the user).

2. STAC places a circle on every node of the grid, with a radius equal to 1.414 (the square root of 2) times the specified search radius. Thus, the circles overlap.

3. STAC counts the number of points falling within each circle, and ranks the circles in descending order.

4. For a maximum of 25 circles, STAC records all circles with at least two data points along with the number of points within each circle. The X and Y coordinates of any node with at least two incidents within the search radius are recorded, along with the number of data points found for each node.

5. These circles are then ranked according to the number of points and the top 25 search areas are selected.

6. If a point belongs to two different circles, the points within the circles are combined. This process is repeated until there are no overlapping circles. This routine avoids the problem of data points belonging to more than one cluster, and the additional problem of different cluster arrangements being possible with the same points. The result is called Hot Clusters.

7. Using the data points in each Hot Cluster, for each cluster the program can calculate the best-fitting standard deviational ellipse or convex hull (see chapter 4). These are called *Hot Spot Areas*. Because the standard deviational ellipse is a statistical summary of the Hot Cluster points, it may not contain every Hot Cluster point. It also may contain points that are not in the Hot Cluster. On the other hand, the convex hulls will create a polygon around all points in the cluster.

The user can specify different search radii and re-run the routine. Given the same area boundary, different search radii will often produce slightly different numbers of Hot Clusters. A search radius that is either too large or too small may fail to produce any. Experience and experimentation are needed to determine the most useful search radii.

Steps in Using *STAC*

STAC is available on the Hot Spot Analysis II tab under Spatial Description (see figure 7.1). A brief summary of the steps is as follows:

1. *STAC* requires a primary file and a reference file (see chapter 3). Optionally, STAC requires the reference file area (on the measurement parameters tab) if simulation runs are requested. Note: while *STAC* runs quite quickly, it runs more quickly with a Euclidean coordinate system such as UTM or State Plane. For example, an analysis of 13,000 street robberies in Chicago ran in less than two seconds on a 800 mhz PC with projected coordinates (Euclidean), while it took longer with spherical coordinates (latitude/longitude).

2. Define the reference file (see chapter 3). While *CrimeStat* does not include a data base manager or query system, a user can carry out analysis of different areas of a jurisdiction by using the boundaries of several reference areas. For example, define all of Chicago as a reference area and define each of the twenty-five police districts as additional reference areas. Hot Spot Areas can be identified for the city as a whole and for each district. In other words, the same incident file may be used for analysis of different map areas by using multiple reference files.

3. Define the search radius. Generally, a two-stage analysis is best. Start with a larger search radius and then analyze Hot Spot Areas with a smaller search radius. A search radius of more than one mile may not yield useful results in an area the size of Chicago (320 square miles).

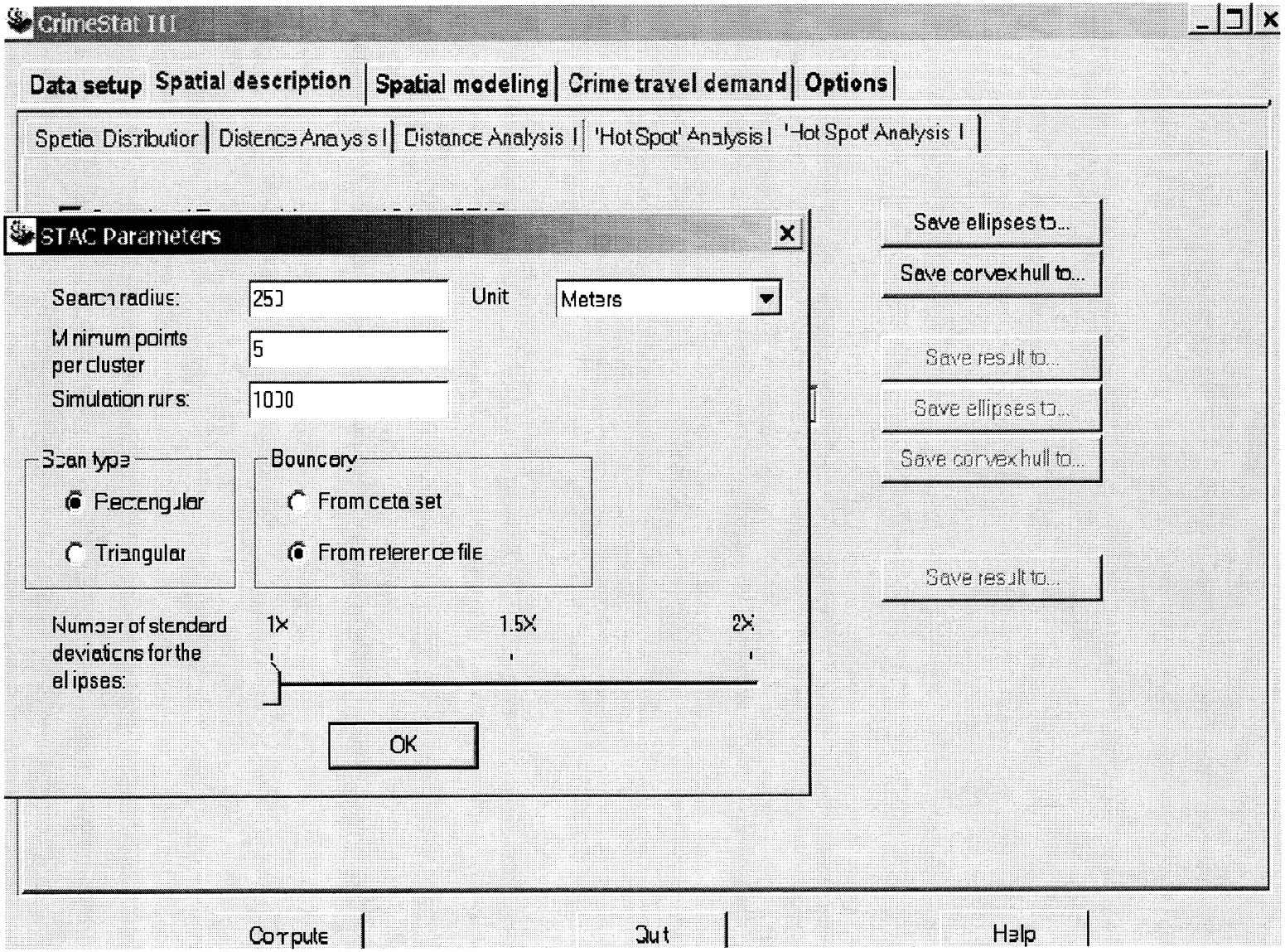
4. Set the output units to miles or kilometers.

5. Specify the file output name for the ellipses or convex hulls.

6. Click on the *STAC* parameters button.

The object of *STAC* is to identify hot spots and display them with ellipses or convex hulls. Its key function is visual. Save the ellipses or hulls in the form most appropriate for the system (e.g., *ArcView*, *Atlas*, *MapInfo*). Because the ellipses or convex hulls are generated as polygons, they can be used for selections, queries, or thematic maps in the GIS. In addition to the ellipses and convex hulls, a table is output with all the information on density and location for each ellipse. It can be saved to a 'dbf' file, which can then be read by any spreadsheet program. The ellipses and convex hulls are numbered in the same order as the printed output.

Figure 7.4: STAC Parameters Setup



STAC Parameters

The two most important parameters for running STAC are the boundary of the study area (reference area) and the search radius. A detailed discussion of the parameters follows. Figure 7.4 shows the *STAC* parameters screen.

Search Radius

1. The search radius is the key setting in *STAC*. In general, the larger the search radius, the more incidents that will be included in each Hot Cluster and the larger the ellipse that will be displayed. Smaller search radii generally result in more ellipses of a smaller size. A good strategy is to initially use a larger radius and then re-analyze areas that are 'hot' with a smaller radius. In Chicago, we have found that a 750 meter radius is appropriate for the city as a whole and a 200 meter search radius for one of the 25 districts. It will be necessary to experiment to determine an appropriate search radius.

Units

2. Specify the units for the search radius. The default is miles and the default search radius is 0.5 miles. Be careful about using larger search radii. In Chicago, a search radius larger than one mile generates ellipses that are too large to be of any tactical or planning use. Other good choices are 750 meters or 0.25 miles.

Minimum Points Per Cluster

3. Specify the minimum number of points to be included in a Hot Cluster. The limit for the minimum points in a Hot Cluster is two. We usually use a minimum of 10.

Boundary

4. Select the reference file to be used for the analysis. The user can choose the boundary from the data set (i.e., the minimum and maximum X/Y values) or from the reference boundary. In our opinion, the choice of the reference boundary is best. If the data set is used to define the reference boundary, the smallest rectangle that encompasses all incident will be used.

Scan Type

5. Select the scan type for the grid. Choose Rectangular if the analysis area has a mostly grided street pattern. Chose Triangular if the analysis area generally has an irregular street pattern.

Graphical output files

6. Select whether the graphical output will be displayed as standard deviational ellipse or as convex hulls, or both (see chapter 4). For ellipses, select the number of standard deviations for the ellipses. One (1X), 1.5X, and 2X standard deviations can be selected. One standard deviational ellipses should be sufficient for most analysis. While one standard deviational ellipses rarely overlap, 1.5X and 2X two standard deviational ellipses often do. A larger ellipse will include more of the Hot Cluster points; a small ellipse will produce a more focused Hot Cluster identification. The user will have to work out a balance between defining a cluster precisely compared to making it so large as to be unclear where one starts and another ends.

Simulation Runs

7. Specify whether any simulation runs are to be made. To test the significance of *STAC* clusters, it is necessary to run a Monte Carlo simulation (Dwass, 1957; Barnard, 1963). *CrimeStat* includes a Monte Carlo simulation routine that produces approximate confidence intervals for the particular STAC model that has been run. The difference between the density of incidents in *STAC* ellipses in a spatially random data set and the *STAC* ellipses in the actual data set is a test of the strength of the clustering detected by *STAC*. Essentially, the Monte Carlo simulation assigns N cases randomly to a rectangle with the same area as the defined study area as specified on the Measurement Parameters tab and evaluates the number of clusters according to the defined parameters (i.e., search radius). It repeats this test K times, where K is defined by the user (e.g., 100, 1,000, 10,000). By running the simulation many times, the user can assess approximate confidence intervals for the particular number of clusters and density of clusters. The default is zero simulation runs because the simulation run option usually increases the calculation time considerably. If a simulation run is selected, the user should identify the area of the study region on the Measurement Parameters tab. It is better to use the jurisidictional area rather than the reference area if the jurisdiction is irregularly shaped.

Output

Ellipses or convex hulls

The ellipses are output with a prefix of "St" before the output file name while the convex hulls are output with a prefix of "Cst" before the output file name. These objects can easily be incorporated into a GIS system. *ArcView* shape files can be opened as themes. STAC graphic files also can be added as a *MapInfo* layer using the Universal Translator Tool. *MapInfo* Mif/Mid files must be imported using the command table—>import. Both *MapInfo* and *ArcView* files are polygons and can be used for queries, thematics, and selections.

Printed Output

Table 7.1 shows the printed output. Note that the printed output does not include the file name. Be sure to record the file name and the reference file (if any that is used).

1. The first section of the output documents parameter settings and file size. Sample size indicates the number of points in the file specified in the setup.

2. Measurement Type indicates the type of distance measurement, direct or Indirect (Manhattan).

3. Scan Type indicates a rectangular or triangular grid specified in the setup.

4. Input Unit indicates the units of the coordinates specified in the setup, degrees (if latitude/longitude) or meters or feet (if projected).

5. Output Units indicate the unit of density and length specified in the setup for the output and ellipses. Output Units are generally, miles or kilometers.

6. Search Radius is the units specified in the setup. In Figure 7.2 above, this is meters.

7. Boundary identifies the coordinates of the lower left and upper right corner of the study area.

8. Points inside the boundary count the number of points within the reference file. This may be fewer than the number of points in the total file when a smaller area is being used for analysis (see above).

9. Simulation Runs indicates the number of runs, if any specified in the setup.

10. Finally, STAC printed output provides summary statistics for each Hot Spot Area.

    A. Cluster— an identification number for each ellipse. This corresponds to their order in a table view in *ArcView*, or the browser in *MapInfo*.

    B. Mean X and Mean Y - Coordinates of the mean center of the ellipse.

    C. Rotation- the degrees the ellipse is rotated (0 is horizontal; 90 is vertical).

    D. X-axis and Y-axis - the length (in the selected output units) of the x and y axis. In the example, the length of the x axis of ellipse 1 is 1.04768 miles.

Table 7.1
Printed Output for STAC

```
Spatial and Temporal Analysis of Crime:
---------------------------------------

   Sample size ..........:  1181
   Measurement type .....:  Direct
   Scan type..... .......:  Rectangular
   Input units .... .....:  Degrees
   Output units ... .....:  Miles, Squared Miles, Points per Squared Miles
   Standard Deviations ...:  1
   Search radius.........:  804.672000
   Boundary..............:  -76.83302,39.23274 to -76.38390,39.59103
   Points inside boundary.:  1179
   Simulation runs .....:  1000

                                                                    Ellipse
Cluster  Mean X   Mean Y    Rotation  X-Axis   Y-Axis  Area  Points  Density
-------  -------  -------   --------  -------  ------  ----  ------  -------
 1     -76.44915  39.31484  89.41867  1.04768  0.25053 0.82460  106  128.546688
 2     -76.73681  39.28658  69.91502  0.22142  0.88202 0.61354   63  102.682109
 3     -76.57098  39.38499  37.10812  0.34793  0.82213 0.89863   61   67.880882
 4     -76.77129  39.35987  11.26360  0.94336  0.26216 0.77695   61   78.511958
 5     -76.51830  39.26019   8.37773  0.43717  0.25497 0.35017   43   22.796997
 6     -76.60231  39.40086  14.84392  0.17969  0.29466 0.16634   36   16.423811
 7     -76.73087  39.34246  41.07812  0.31007  0.25885 0.25215   35   38.806566
 8     -76.75451  39.31110  74.78196  0.19154  0.31572 0.18998   24   26.326405

   Distribution of the number of clusters found in simulation (percentile):

   Percentile Clusters           Area       Points          Density
   ---------- -------- -------------- ------------ ----------------
          min       12        0.01113            5         4.673554
          0.5       13        0.02389            5         4.924993
          1.0       13        0.03587            5         4.977644
          2.5       14        0.05081            5         5.236646
          5.0       14        0.06177            5         5.505124
         95.0       19        1.24974           14        82.281060
         97.5       19        1.39923           16       101.053102
         99.0       20        1.58861           17       140.078387
         99.5       20        1.67065           19       209.279368
          max       20        2.08665           23       449.401912
```

E. Area - the area of the ellipse in square units. Ellipses are ordered according to their size. In the example, Ellipse 1 is 0.8246 square miles.

F. Points - the number of points in the Hot Cluster. In the example, there are 61 points in cluster 3.

G. Cluster Density - the number of points per square unit. The largest cluster is not necessarily the densest. In this example, cluster eight is the smallest, but its density is higher than two other clusters.

The best way to print or save *CrimeStat* printed output is to place the cursor inside the output window and *Select all,* then copy and paste the selection into a word processing document in landscape mode.

Make sure to adequately annotate the file, especially the type of incidents, the reference boundary, and the name of the output file. This can be very important for future reference.

For Old *STAC* Users

In general, *STAC* has retained all the functionality and speed of previous versions. The ellipses will look somewhat different than previous versions, because a more widely accepted method for calculating standard deviational ellipses has been used. *STAC for DOS* used a 1x standard deviation ellipse. Analysts who want results similar to *STAC* for DOS should set standard deviations to 1.

The *CrimeStat* version of *STAC* has the following improvements over STAC for DOS:

1. *STAC* no longer requires the use of a special ASCII data file. The data file can be any of those available in *CrimeStat.*

2. Any projection can be used, including latitude/longitude. Files are not converted into a Euclidean projection.

3. We have not found a limit on the number of points that can be analyzed with the *CrimeStat* version of STAC. Therefore, a small radius can now be used over large areas.

4. *STAC* can generate Shape files for ArcView or Mif/Mid files for MapInfo. Both are polygons-not points.

5. It is easier for the user to specify the number of standard deviations for an ellipse (1X, 1.5X, or 2X).

6. Convex hull output has been added.

7. The user can run *STAC* on a spatially random data set to get an estimate of the degree of clustering detected by *STAC* in the incident data.

8. The study area boundary (reference file) can be generated from the data set (we would not suggest doing this since it will be difficult to compare distributions).

Example 1: A STAC Analysis of 1999 Chicago Street Robberies

STAC Hot Spot Areas were calculated for all street (or sidewalk or alley) robberies occurring in Chicago in 1999 (n=13,009).[3] There were 13,007 within the search boundary. The search radius was set for 750 meters (approximately ½ mile), and the ellipses were set to one standard deviation. Ten was the minimum number of incidents per cluster.

In figure 7.2 (shown earlier), STAC detected seven ellipses. The areas of the seven ellipses ranged from 5 square kilometers to 0.7 square kilometers, and the number of incidents in an ellipse ranged from 760 to 153. The smallest ellipse (number 7 in figure 7.2) was the densest, 222 robberies per square kilometer. Of the 13,007 incidents, 2,375 were in a cluster. Therefore, 18 percent of all of Chicago's street robberies in 1999 occurred in 6% of its 233 square mile area.

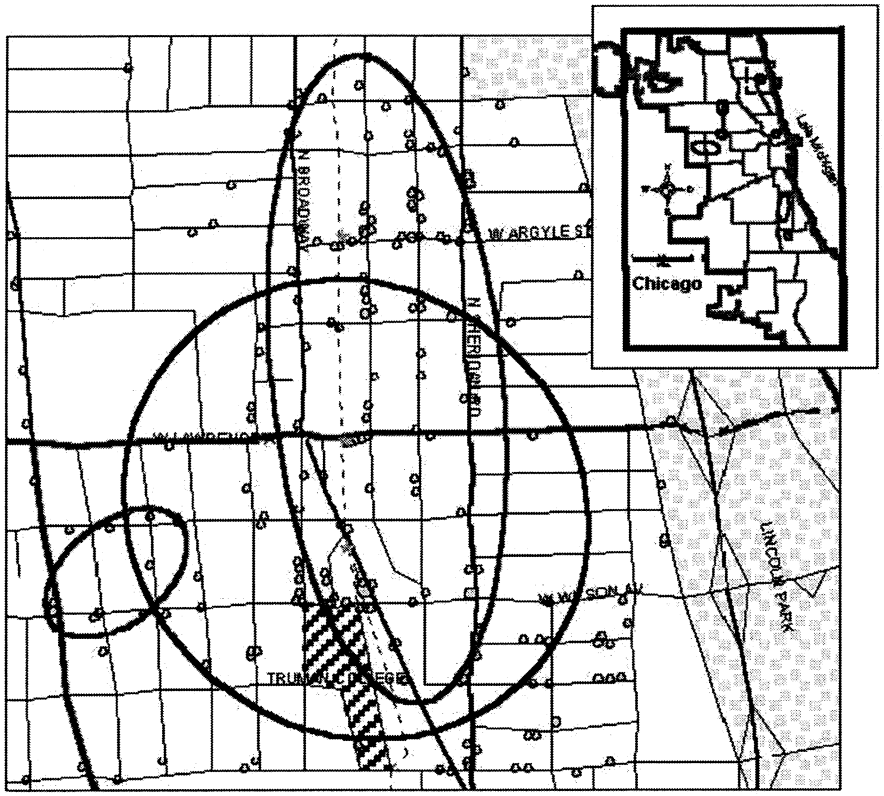
To map the results, the ellipse boundaries were imported into *MapInfo* as a mif/mid file and overlaid on a map of police districts. The large blue rectangle in figure 7.2 designates the search boundary (reference file). O'Hare Airport was excluded because exact geo-coding is not possible for the few street robberies that occurred there. At a city-wide scale, the map is interesting, but is mainly useful for confirming what is already known. Ellipse 1, on the west side, has had a high level of violence for many years. Ellipses 2 and 6 are centered on areas where high rise public housing projects are gradually being abandoned. Overall, these ellipses are not very useful for tactical purposes. However, they point out that four Hot Spot Areas cross District boundaries, and that the large number of street robberies in these areas might be lost in separate district reports.

A Neighborhood STAC Analysis

The presence of Ellipse 4 (the northernmost ellipse in figure 7.2) might be unexpected to many Chicagoans. The mid-Northside, near the Lake Michigan, is generally considered to be a relatively affluent and safe neighborhood. However, the neighborhood around Ellipse 4 has had a high level of crime for many years. It was an entertainment center in the Roaring Twenties, and several institutions of that era remain. Today it is an area with multiple, often conflicting, uses. A more detailed analysis of the neighborhood with the help of STAC may point to specific areas that need increased patrol or prevention activities.

The second step of STAC analysis was to define a focused search boundary area around Ellipse 4. This was done easily by creating a new map layer in MapInfo and drawing a rectangle around the desired study area. Clicking on the study area gave the required *CrimeStat* reference boundary maximum and minimum coordinates. Using this more focused boundary, STAC was run a second time with a 200 meter search radius and

Figure 7.5: STAC Hot Spots for Northeast Side Street Robberies



Northeast Side
STAC Hot Spot Analysis
Street Robbery 1999

Source: Chicago Police Department

EL Stations
Main Surface Streets
STAC Northside Elliipses 200 Meter
Police Districts
Count of Street Robberies
8
4
0.8

the same file of 13,009 cases. The search boundary (reference file) now contained 442 incidents. STAC detected three ellipses that contained 231 incidents. The STAC ellipses were then imported into *MapInfo* and mapped (Figure 7.5).

As the area covered by a map grows smaller, detailed information about crime patterns and the community can be added. In this map, the STAC ellipses were overlain with the address locations of incidents (sized according to the number occurring at each location) and streets.[4] Much of the area is relatively crime-free. The most frequent locations for street robbery do not coincide with main streets. Street robbery incidents tend to cluster near rapid transit stations and the blocks immediately surrounding them. For example, Argyle Street, between Broadway and Sheridan, is the site of "New China Town." It is an an area with a number of street robberies and is a destination area for "Northsiders" who want an inexpensive Chinese or Vietnamese meal.

There is a particularly risky area in the neighborhood of Broadway and Wilson adjacent to Truman Community College. In a previous analysis of the Bronx, Fordham University was shown to be a similar attractor for robbery incidents. Colleges supply good targets for street robbery. Also, authority for security is split between the college and the city police. The area around Broadway and Wilson has been risky for many years. Ninety years ago, it was the northern terminus of rapid transit, and the site of several very inexpensive hotels, two of which still exist. Today the area has several pawn shops and currency exchanges. There is an ATM located in the EL station. The area looks dangerous and dirty. Finally, the area has many blind corners and alleys that could serve as sites for robbery; this is unusual for Chicago. The census block that includes the northwest corner of Broadway and Wilson ranked fifth among Chicago's 21,000 census blocks in number of street robberies in 1999.

Changes need to be made to reduce the risk of street robbery in this area. Mapping identifies a problem with street robberies, but to investigate possible changes it is necessary to go beyond mapping. Aside from changes in patrol practices, what physical changes might aid in crime reduction? The campus has very little parking. The administration assumes that students take public transportation, but many do not. A secure parking garage that could serve both the elevated station and the school could be constructed (vacant land is available). In addition, increased police patrol in the area between the school and the el station could be implemented.

Advantages of STAC

STAC has a number of advantages as a clustering algorithm:

1. STAC can analyze a very large number of cases quickly. It is very fast using a Euclidean projection such as UTM or State Plane, and not quite as fast using spherical coordinates (latitude/longitude).

2. The STAC user controls the approximate size of the ellipses (search radius), the minimum number of points per ellipse, and the study area. These

features allow for a broad search for Hot Spot Areas over an entire city and a second search concentrating on a smaller area and deriving focused Hot Spot Areas for local tactical use.

3. STAC and Heirarchical Clustering are complimentary. Heirarchical Clustering first derives small ellipses and then aggregates to larger ones. The recommended STAC procedure is to first derive large scale ellipses and then to analyze these for tactical use.

4. The visual display of STAC ellipses or convex hulls is quite intuitive.

5. Hot spots need not be limited to a single kind of crime, place or even. For example, ellipses of drug crime can be overlain on those for burglary. Some causal factors are also analyzable with STAC ellipses. For example, ellipses of street robbery can be compared to those for liquor licenses.

9. STAC combines features of a hierarchical and partitioning search methods and adapts itself to the size of the clusters.

10. Unlike the Nnh routine, which has a constant threshold (search radius), STAC can create clusters of unequal size because overlapping clusters are combined until there is no overlap.

Limitations of STAC

There are also some limitations to using STAC:

1. The distribution of incidents within clusters is not necessarily uniform. The user should be careful not to assume that it is. A mapped theme of the Mode routine (see chapter 6) according to number of incidents or the single kernel density interpolation (see chapter 8) overlaid with STAC ellipses are good ways to overcome this problem (figure 7.5 above and figure 7.6 below).

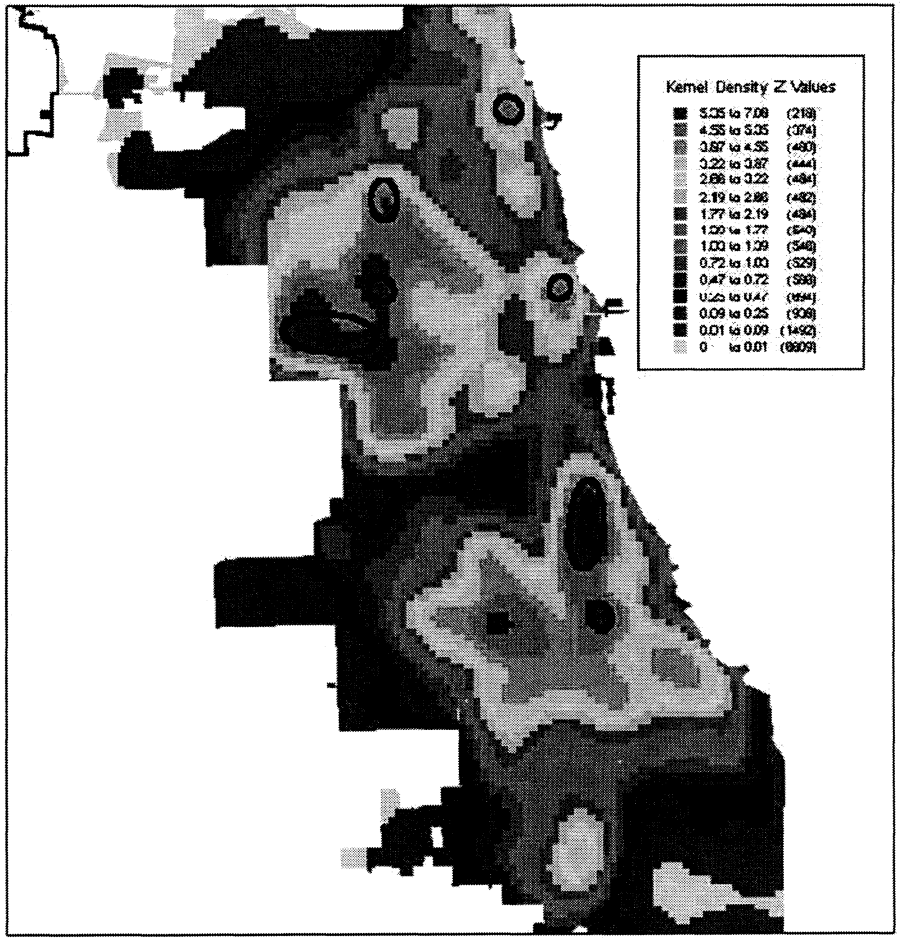
2. STAC is based on the distribution of data points. Neither land use nor risk factors is accounted for. It is up to the analyst to identify the characteristics that make a Hot Spot 'hot'.

3. Small changes in the STAC study area boundary can result in quite different depictions of the ellipses. This is true of any clustering routine. Retaining the same reference file over repeated analyses alleviates this problem. The analysis should also be documented for the analysis parameters.

Nevertheless, if used carefully, STAC is a powerful tool for detecting clusters and can allow an analyst to experiment with varying search radii and reference boundaries.

Next, the K-means clustering routine is examined.

Figure 7.6: STAC Robbery Hot Spots and Kernel Density Estimation



Chicago Street Robbery 1999 :
Comparison of STAC and
Single Kernel Density Estimation

K-Means Partitioning Clustering

The *K-means* clustering routine (Kmeans) is a partitioning procedure where the data are grouped into K groups defined by the user. A specified number of seed locations, K, are defined by the user (Fisher, 1958; MacQueen, 1967; Aldenderfer and Blashfield, 1984; Systat, 2000). The routine tries to find the best positioning of the K centers and then assigns each point to the center that is nearest. Like the Nnh routine, the Kmeans assigns points to one, and only one, cluster. However, unlike the nearest neighbor hierarchical (Nnh) procedure, all points are assigned to clusters. Thus, there is no hierarchy in the routine, that is there are no second- and higher-order clusters.

The technique is useful when a user want to control the grouping. For example, if there are 10 precincts in a jurisdiction, an analyst might want to identify the 10 most compact clusters, one for precinct. Alternatively, if a previous analysis has shown there were 24 clusters, then an analyst could check whether the clusters have shifted over time by also asking for 24 clusters. By definition, the technique is somewhat arbitrary since the user defines how many clusters are to be expected. Whether a cluster could be a 'hot spot' or not would depend on the extent to which a user wanted to replicate 'hot spots' or not.

The theory of the K-means procedure is relatively straightforward. The implementation is more complicated. K-means represents an attempt to define an optimal number of K locations where the sum of the distance from every point to each of the K centers is minimized. It is a variation of the old location theory paradigm of how to locate K facilities (e.g., police stations, hospitals, shopping centers) given the distribution of population (Haggett, Cliff, and Frey, 1977). That is, how does one identify *supply* locations in relation to *demand* locations. In theory, solving this question is an empirical solution, what is frequently called *global optimization*. One tries every combination of K objects where K is a subset of the total population of incidents (or people), N, and measures the distance from every incident point to every one of the K locations. The particular combination which gives the minimal sum of all distances (or all squared distances) is considered the best solution. In practice, however, solving this is computationally almost impossible, particularly if N is large. For example, with 6000 incidents grouped into 20 partitions (clusters), one cannot solve this with any normal computer since there are

$$\frac{6000!}{20!\,5980!} = 1.456 \times 10^{57}$$

combinations. No computer can solve that number and few spreadsheets can calculate the factorial of N greater than about 127.[5] In other words, it is almost impossible to solve computationally.

Practically, therefore, the different implementations of the K-means routine all make initial guesses about the K locations and then optimize the seating of this location in relation to the nearby points. This is called *local optimization*. Unfortunately, each K-means routine has a different way to define the initial locations so that two K-means

procedures will usually not produce the same results, even if *K* is identical (Everitt, 1974; Systat, Inc., 1994).

CrimeStat K-means Routine

The K-means routine in *CrimeStat* also makes an initial guess about the *K* locations and then optimizes the distribution locally. The procedure that is adopted makes initial estimates about location of the K clusters (seeds), assigns all points to its nearest seed location, re-calculates a center for each cluster which becomes a new seed, and then repeats the procedure all over again. The procedure stops when there are very few changes to the cluster composition.[6]

The default K-means clustering routine follows an algorithm for grouping all point locations into one, and only one, of these K groups. There are two general steps: 1) the identification of an initial guess (seed) for the location of the K clusters, and 2) local optimization which assigns each point to the nearest of the K clusters. A grid is overlaid on the data set and the number of points falling within each grid cell is counted. The grid cell with the most points is the initial first cluster. Then, the second initial cluster is the grid cell with the next most points that is separated by at least:

$$\text{Separation} = t * 0.5 * \text{SQRT}\left[\frac{A}{N}\right] \quad (7.1)$$

where t is the Student's t-value for the .01 significance level (2.358), A is the area of the region, and N is the sample size. A third initial cluster is then selected which is the grid cell with the third most points and is separated from the first two grid cells by at least the separation factor defined above. This process is repeated until all K initial seed locations are chosen.

The algorithm then conducts *local optimization*. It assigns each point to the nearest of the K seed locations to form an initial cluster. For each of the initial clusters, it calculates the center of minimum distance and then re-assigns all points to the nearest cluster, based on the distance to the center of minimum distance. It repeats this process until no points change clusters. To increase the flexibility of the routine, the grid that is overlaid on the data points is re-sized to accommodate different cluster structures, increasing or decreasing in size to try to find the K clusters. After iterating through different grid sizes, the code makes sure that the final seeds are from the "best" grid or the grid that produces the most clusters. Finally, for each cluster, the routine calculates a standard deviational ellipse and optionally can output the results graphically as either standard deviational ellipses or a convex hulls.

Control over Initial Selection of Clusters

Changing the separation between clusters

One problem with this approach is that in highly concentrated distributions, such as with most crime incidents in a metropolitan area, the separation between clusters may not be sufficiently large to detect clusters farther away from the concentration; the algorithm will tend to sub-divide concentrated groupings of incidents into multiple clusters rather than seek clusters that are less concentrated and, usually, farther away. To increase the flexibility of the routine, *CrimeStat* allows the user to modify the initial selection of clusters since this has a large effect on the final grouping (Everett, 1974). There are two ways the initial selection of cluster centers can be modified. The user can increase or decrease the separation factor. Formula 7.1 is still used to separate each of the initial clusters, but the user can either select a t-value from 1 to 10 from the drop down menu or write in any number for the separation, including fractions, to increase or decrease the separation between the initial clusters. The default is set at 4.

Figure 7.7 shows a simulation of eight clusters, four of which have higher concentrations than the other two. Two partitions of the data set into eight groups are shown, one using a separation of 4 (dashed green ellipses) and one with a separation of 15 (solid blue ellipses). As seen, the partition with the larger separation captures the eight clusters better. With the smaller separation, the routine will tend to sub-divide more concentrated clusters because that reduces the distance of each point from the cluster center. Depending on the purpose of the partitioning, a greater or lesser separation may be desired.

Selecting the initial seed locations

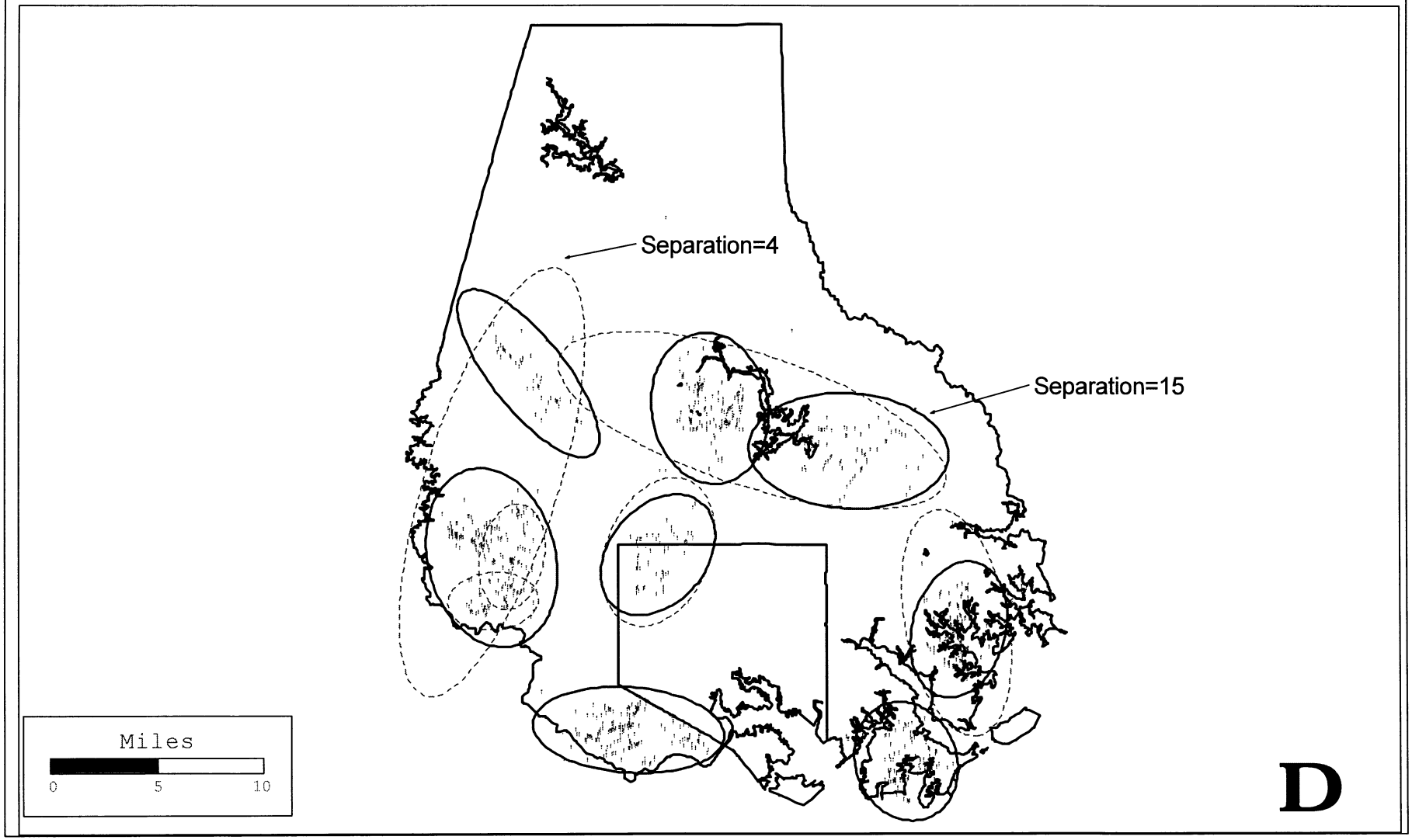
Alternatively, the initial clusters can be modified to allow the user to define the actual locations for the initial cluster centers. This approach was used by Friedman and Rubin (1967) and Ball and Hall (1970). In *CrimeStat*, the user-defined locations are entered with the secondary file which lists the location of the initial clusters. The routine reads the secondary file and uses the number of points in the file for K and the X/Y coordinates of each point as the initial seed locations. It then proceeds in the same way with local optimization. When eight points that were approximately in the middle of the eight clusters in figure 7.7 were input as the secondary file, the K-means routine immediately identified the eight clusters (results not shown). Again, depending on the purpose the user can test a particular clustering by requiring the routine to consider that model, at least for the initial seed location. The routine will conduct local optimization for the rest of the clustering, as in the above method.

The K-means output is similar for both routines. It includes the parameters for the standard deviational ellipse of each cluster in the table. In addition, graphically one can output each cluster as a standard deviational ellipse or as a convex hull (see chapter 4). The convex hull draws a polygon around all the points in a cluster. Hence it is a literal

Figure 7.7:

Separated Data and K-Means Solution

K=8 Paritions with Two Separations of Initial Seed Locations



description of the extent of the cluster. The ellipse, on the other hand, is an abstraction for the cluster. Typically, one standard deviation will cover more than 50% of the cases, one and a half standard deviations will cover more than 90% of the cases, and two standard deviations will cover more than 99% of the cases, although the exact percentage will depend on the distribution. In general, use a 1X standard deviational ellipse since 1.5X and 2X standard deviations can create an exaggerated view of the underlying cluster. The ellipse, after all, is an abstraction from the points in the cluster which may be arranged in an irregular manner. On the other hand, for a regional view, a convex hull or a one standard deviational ellipse may not be very visible. The user has to balance the need to accurately display the cluster compared to making it easier for a viewer to understand its location.

Mean squared error

In addition, the output for each cluster lists two additional statistics:

$$\text{Sum of squares of cluster C} = SSE_C = \sum_{i=1}^{N_C} \{[(X_{iC} - MeanX_C]^2 + [Y_{iC} - MeanY_C]^2\} \quad (7.2)$$

$$\text{Mean squared error of cluster C} = MSE_C = SSE_C / (N_C - 1) \quad (7.3)$$

where X_{iC} is the X value of a point that belongs to cluster C, Y_{iC} is the Y value of a point that belongs to cluster C, $MeanX_C$ is the mean X value of cluster C (i.e., of only those points belonging to C), $MeanY_C$ is the mean Y value of cluster C, and N_C is the number of points in cluster C. There is also a total sum of squares and a total mean square error which is summed over all clusters

$$\text{Total Sum of Squares} = \sum_C SSE_C \quad (7.4)$$

$$\text{Total Mean Squared Error} = \sum_C SSE_C/(N-K-1) \quad (7.5)$$

where SSE_C is the sum of squares for cluster C, N is the total sample size, and K is the number of clusters. The sum of squares is the squared deviations of each cluster point from the center of minimum distance while the mean squared error is the average of the squared deviations for each cluster.

The sum of squares (or sum of squared errors) is frequently used as a criteria for identifying 'goodness of fit' (Everett, 1974; Aldenderfer and Blashfield, 1984; Gersho and Gray, 1992). In general, for a given number of clusters, K, those with a smaller sum of squares and, correspondingly, smaller mean square error are better defined than clusters with a larger sum of squares and larger mean squared error. Similarly, a K-means

solution that produces a smaller overall sum of squares is a tighter grouping than a grouping that produces a larger overall sum of squares.

But, there can be exceptions. If there are points which are 'outliers', that is which don't obviously fall into one cluster or another, re-assigning them to one or another cluster can distort the sum of squares statistics. Also, in highly concentrated distributions, such as with crime incidents, a smaller sum of squares criteria can be obtained by splitting the concentrations rather than clustering less central and less dense groups of incidents (such as in figure 7.7); the results, while minimizing the sum of squared errors from the cluster centers, will be less desirable because the peripheral clusters are ignored. Thus, these statistics are presented for the user's information only. In assigning points to clusters, *CrimeStat* still uses the distance to the nearest seed location, rather than a solution that minimizes the sum of squared distances.

Visualizing the Cluster

Finally, the K-means clustering routine (Kmeans) outputs clusters graphically as either ellipses or convex hulls, similar to the other clustering routines. For the ellipses, the user can choose between 1X, 1.5X, and 2X standard deviations to display the ellipses. The graphical ellipses are output with the prefix 'KM' before the file name. It should be noted, however, that the ellipses are an abstraction of the cluster. The clusters are *not* necessarily arranged in ellipses. They are for visualization purposes only. For the convex hull, the routine draws a polygon around the points in each cluster. The graphical convex hulls are output with the prefix "CKM" before the file name.

K-means Output Files

The naming system for the K-means outputs is simpler than the Nnh routine since there are no higher-order clusters. Each file is named

Km<*username*>	[for the ellipse]
Ckm<*username*>	[for the convex hull]

where *username* is the name of the file provided by the user. Within the file, each cluster is named

KmEll<N><*username*>	[for the ellipse]
CkmHull<N><*username*>	[for the convex hull]

where *N* is the cluster number and *username* is the name of the file provided by the user. For example,

KmEll3robbery

is the third ellipse for the file called 'robbery' and

CkmHull12burglary

is the 12th convex hull for the file called 'burglary'.

For the ellipses, a slide-bar allows ellipses to be defined for 1X, 1.5X, and 2X standard deviations and can be output in *ArcView* '.shp', *MapInfo* '.mif' or *Atlas*GIS* '.bna' formats. The convex hulls, on the other hand, draw a polygon around the clustered points.

Example 2: K-means Clustering of Street Robberies

In *CrimeStat*, the user specifies the number of groups to sub-divide the data. Using the 1996 robbery incidents for Baltimore County, the data were partitioned into 10 groups with the K-means routine (figure 7.8). As can be seen, the clusters tend to fall along the border with Baltimore City. But there are three more dispersed clusters, one concentrated in the central eastern part of the county and two north of the border with the City. Because these clusters are very large, a finer mesh clustering was conducting by partitioning the data into 31 clusters (figure 7.9). Thirty-five clusters were requested but the routine only found 31 seed location. Consequently, it outputted 31 clusters, which are displayed as ellipses. Though the ellipses are still larger than those produced by the nearest neighbor hierarchical procedure (see figure 6.7 in chapter 6), there is some congruency; clusters identified by the nearest neighbor procedure have corresponding ellipses using the K-means procedure.

Figure 7.10 shows a section of southwest Baltimore County with four full clusters and three partial clusters visible, displayed as convex hulls. Looking at the distribution, several clusters make intuitive sense while a couple of others do not. For example, two clusters highlight a concentration along a major arterial (U.S. Highway 40). Similarly, the cluster in the middle right appears to capture incidents along two arterial roads. However, the other three full clusters do not appear to capture meaningful patterns and appear somewhat arbitrary.

Other uses of the K-means algorithm are possible. One problem that affects most police departments is the need to allocate personnel throughout a city in a balanced and fair way. Too often, some police precincts or districts are overburdened with Calls for Service whereas others have more moderate demand. The issue of re-drawing or re-assigning police boundaries in order to re-establish balance is a continual one for police departments. The K-means algorithm can help in defining this balance, though there are many other factors that will affect particular boundaries. The number of groupings, K, can be chosen based on the number of police districts that exist or that are desired. The locations of division or precinct stations can be entered in a secondary file in order to define the initial 'seed' locations. The K-means routine can then be run to assign all incidents to each of the K groups. The analyst can vary the location of the initial seeds or, even, the number of groups in order to explore different arrangements in space. Once an agreed upon solution is found, it is easy to then re-assign police beats to fit the new arrangement.

Figure 7.8:

Baltimore County Robbery 'Hot Spots'

Using K-Means Routine with K=10 Clusters

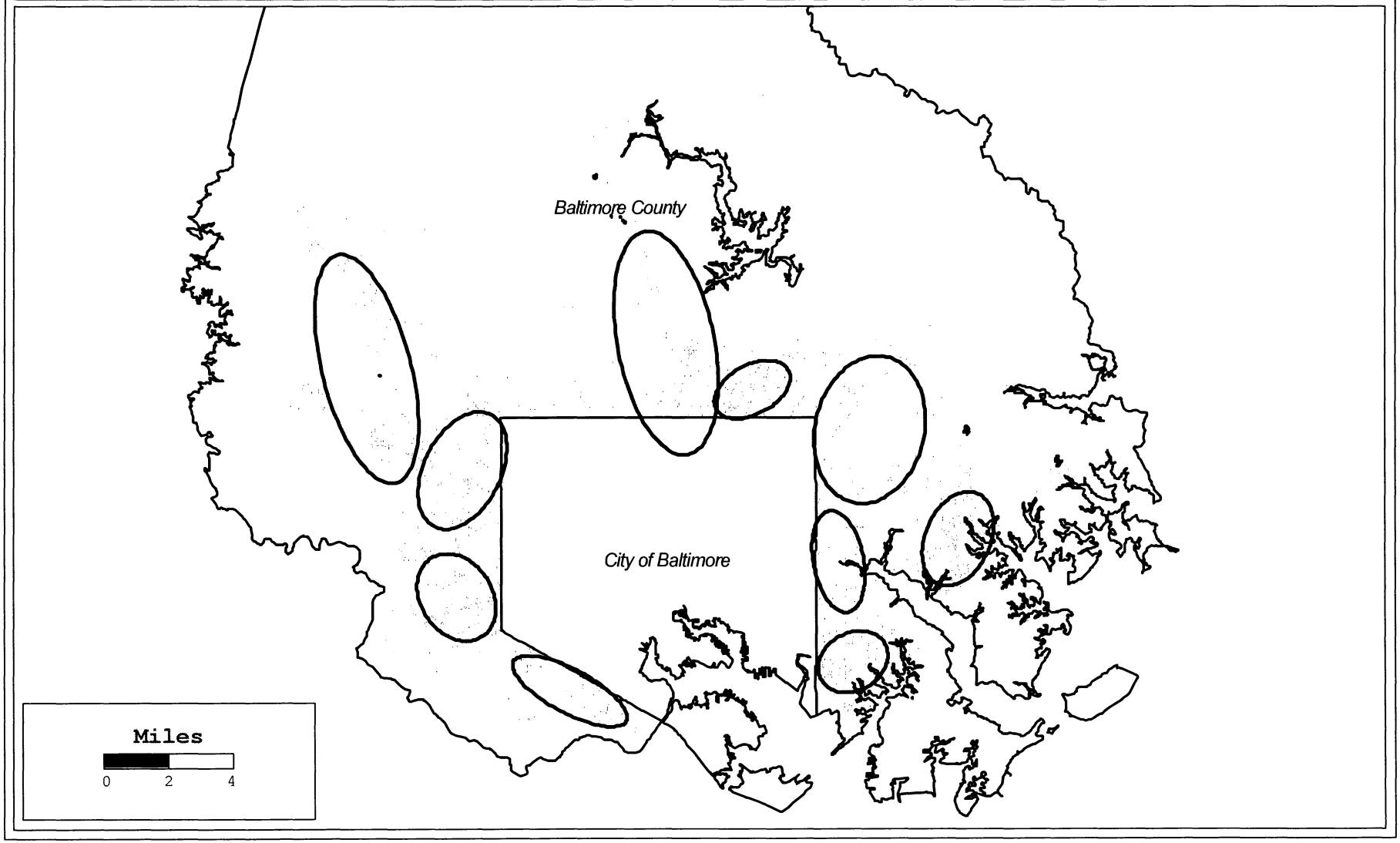


Figure 7.9:

Baltimore County Robbery 'Hot Spots'

Using K-Means Routine with K=31 Clusters

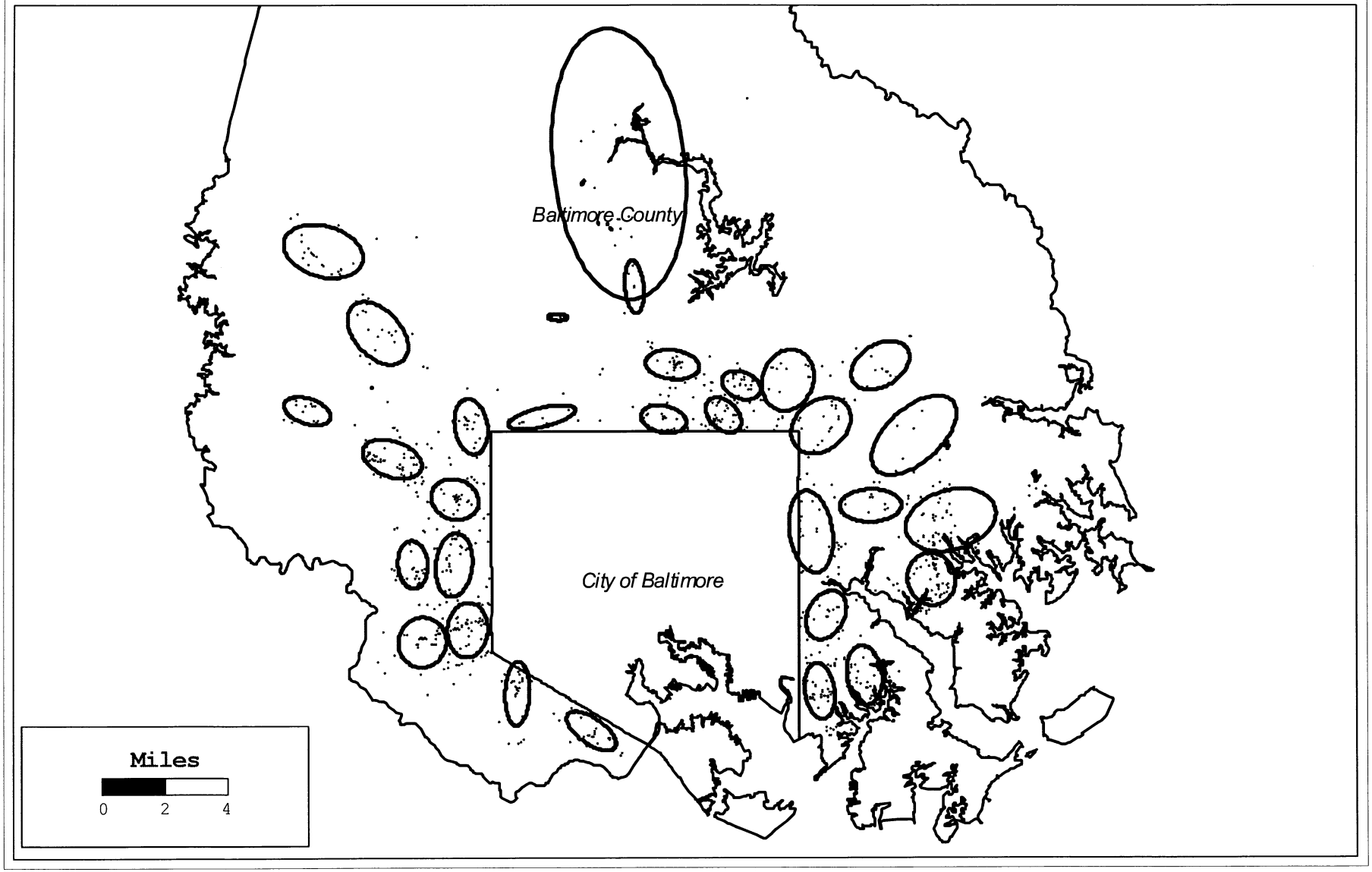


Figure 7.10:

Southwest Baltimore County Robbery 'Hot Spots'

Using K-Means Routine with K=31 Clusters

Advantages and Disadvantages of the K-means Procedure

In short, the K-means procedure will divide the data into the number of groups specified by the user. Whether these groups make any sense or not will depend on how carefully the user has selected clusters. Choosing too many will lead to defining patterns that don't really exist whereas choosing too few will lead to poor differentiation among neighborhoods that are distinctly different.

It is this choice that is both a strength of the technique as well as a weakness. The K-means procedure provides a great deal of control for the user and can be used as an exploratory tool to identify possible 'hot spots'. Whereas the nearest neighbor hierarchical method produces a solution based on geographical proximity with most clusters being very small, the K-means can allow the user to control the size of the clusters. In terms of policing, the K-means is better suited for defining larger geographical areas than the nearest neighbor method, perhaps more appropriate for a patrol area than for a particular 'hot spot'. Again, if carefully used, the K-means gives the user the ability to 'fine tune' a particular model of 'hot spots', adjusting the size of the clusters (vis-a-via the number of clusters selected) in order to fit a particular pattern which is known.

Yet it is this same flexible characteristic that makes the technique potentially difficult to use and prone to misuse. Since the technique will divide the data set into *K* groups, there is no assumption that these *K* groups represent real 'hot spots' or not. A user cannot just arbitrarily put in a number and expect it to produce meaningful results. A more extensive discussion of this issue can be found in Murray and Grubesic (2002). Grubesic and Murray (2001) present some newer approaches in the K-means methodology.

The technique is, therefore, better seen as both an exploratory tool as well as a tool for refining a 'hot spot' search. If the user has a good idea of where there should be 'hot spots', based on community experience and the reports of beat officers, then the technique can be used to see if the incidents actually correspond to the perception. It also can help identify 'hot spots' which have not been perceived or identified by officers. Alternatively, it can identify 'hot spots' that don't really exist and which are merely by-products of the statistical procedure. Experience and sensitivity are needed to know whether an identified 'hot spot' is real or not.

Anselin's Local Moran Statistic (LMoran)

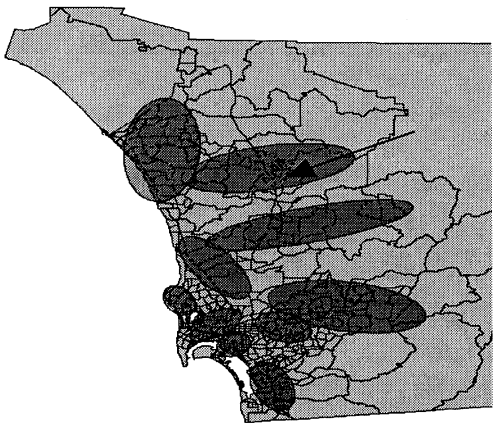
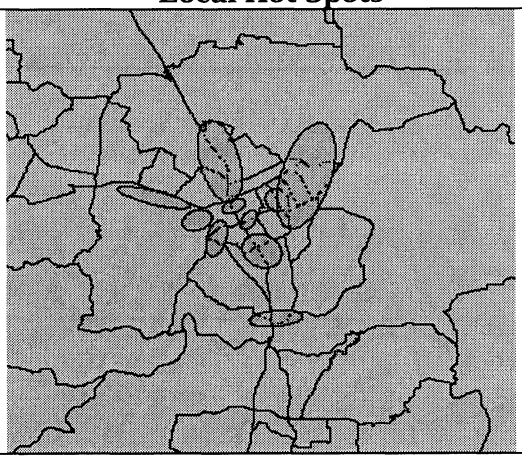
The last 'hot spot' technique in *CrimeStat* is a zonal technique called the *Anselin's Local Moran* statistic and was developed by Luc Anselin (1995). Unlike the nearest neighbor hierarchical and K-means procedures, the local Moran statistic requires data to be aggregated by zones, such as census block groups, zip codes, police reporting areas or other aggregations. The procedure applies Moran's I statistic to individual zones, allowing them to be identified as similar or different to their nearby pattern.

K-Means Clustering as an Alternative Measure of Urban Accessibility

Richard J. Crepeau
Department of Geography and Planning
Appalachian State University
Boone, NC

The relationship between land use and the transportation system is an important issue. Many planners recognize that transportation policies, practices and outcomes affect changes in land use, and vice versa, but there is disagreement as to how best to describe this phenomenon. Traditional methods include measures of accessibility via a matrix of zones (tracts, traffic analysis zones, etc.). However, there are limits to the way interaction and accessibility is described with such discrete units.

Through the use of K-Means clustering, an alternate measure of accessibility can be calculated. Rather than relying on census geography, the left map shows ten retail clusters in San Diego County (1995) as calculated by *CrimeStat*'s K-Means clustering technique (using 1x standard deviational ellipse). The retail hot spots were calculated using a geocoded point file of retail establishments in the county. These clusters are not bound by census geography and allow a more realistic appraisal about the attractiveness of specific regions within the county. An analyst can then determine if residential location within a hot spot has an effect on travel patterns, or if there is a relationship between proximity to a hot spot and travel behavior. While this example illustrates a measure of regional retail attractiveness, the flexibility of *CrimeStat* allows an analyst to evaluate these relationships on a local level, thus allowing a scope of inquiry from regional to local accessibility (as shown in right map, which uses the same parameters as the left figure, but limiting its sample to retail in a sub-region of San Diego County noted by the arrow).

Regional Hot Spots	Local Hot Spots
	

Hot Spot Verification in Auto Theft Recoveries

Bryan Hill
Glendale Police Department
Glendale, AZ

We use *CrimeStat* as a verification tool to help isolate clusters of activity when one application or method does not appear to completely identify a problem. The following example utilizes several *CrimeStat* statistical functions to verify a recovery pattern for auto thefts in the City of Glendale (AZ). The recovery data included recovery locations for the past 6 months in the City of Glendale which were geocoded with a county-wide street centerline file using *ArcView*.

First, a spatial density "grid" was created using *Spatial Analyst* with a grid cell size of 300 feet and a search radius of 0.75 miles for the 307 recovery locations. We then created a graduated color legend, using standard deviation as the classification type and the value for the legend being the *CrimeStat* "Z" field that is calculated.

In the map, the K-means (red ellipses), Nnh (green ellipses) and *Spatial Analyst* grid (red-yellow grid cells) all showed that the area was a high density or clustering of stolen vehicle recoveries. Although this information was not new, it did help verify our conclusion and aided in organizing a response

The basic concept is that of a *local indicator of spatial association* (*LISA*) and has been discussed by a number of researchers (Mantel, 1967; Getis, 1991; Anselin, 1995). For example, Anselin (1995) defines this as any statistic that satisfies two requirements:

1. The *LISA* for each observation indicates the extent to which there is significant spatial clustering of similar values around that observation; and

2. The sum of the *LISA*s for all observations is proportional to the global indicator of spatial association.

$$L_i = f(Y_i, Y_{Ji}) \tag{7.6}$$

where L_i is the local indicator, Y_i is the value of an intensity variable at location i, and Y_{Ji} are the values observed in the neighborhood J_i of i.

In other words, a *LISA* is an indicator of the extent to which the value of an observation is similar or different from its neighboring observations. This requires two conditions. First, that each observation has a variable value that can be assigned to it (i.e., an intensity or a weight) in addition to its X and Y coordinates. For crime incidents, this means data that are aggregated into zones (e.g., number of incidents by census tracts, zip codes, or police reporting districts). Second, the *neighborhood* has to be defined. This could be either adjacent zones or all other zones negatively weighted by the distance from the observation zone.

Once these are defined, the *LISA* indicates the value of the observation zone in relation to its neighborhood. Thus, in neighborhoods where there are 'high' intensity values, the *LISA* indicates whether a particular observation is similar (i.e., also 'high') or different (i.e., low) and, conversely, in neighborhoods where there are 'low' intensity values, the *LISA* indicates whether a particular observation is similar (i.e., also 'low') or different (i.e., 'high'). That is, the *LISA* is an indicator of similarity, not absolute value of the intensity variable.

Formal Definition of Local Moran Statistic

The I_i statistic

Anselin (1995) has applied the concept to a number of spatial autocorrelation statistics. The most commonly used, which is included in *CrimeStat*, is Anselin's Local Moran statistic, I_i, the use of Moran's I statistic as a *LISA*. The definition of I_i is (from Getis and Ord, 1996):

$$I_i = \frac{(Z_i - \bar{Z})}{S_Z^2} * \sum_{j=1}^{N} [\, W_{ij} * (Z_j - \bar{Z}) \,] \tag{7.7}$$

where $\bar{Z}$ is the mean intensity over all observations, Z_i is the intensity of observation i, Z_j is intensity for all other observations, j (where $j \neq i$), S_Z^2 is the variance over all observations, and W_{ij} is a distance weight for the interaction between observations i and j. Note, the first term refers only to observation i, while the second term is the sum of the weighted values for all other observations (but not including i itself).

Distance weights

The weights, W_{ij}, can be either an indicator of the adjacency of a zone to the observation zone (i.e., '1' if adjacent; 0 if not adjacent) or a distance-based weight which decreases with distance between zones i and j. Adjacency indices are useful for defining near neighborhoods; the adjacent zones have full weight while all other zones have no weight. Distance weights, on the other hand, are useful for defining spatial interaction; zones which are farther away can have an influence on an observation zone, although one that is much less. *CrimeStat* uses distance weights, in two forms.

First, there is a traditional distance decay function:

$$W_{ij} = \frac{1}{d_{ij}} \tag{7.8}$$

where d_{ij} is the distance between the observation zone, i, and another zone, j. Thus, a zone which is two miles away has half the weight of a zone that is one mile away.

Small distance adjustment

Second, there is an adjustment for small distances. Depending on the distance scale used (miles, kilometers, meters), the weight index becomes problematic when the distance falls below 1 (i.e., below 1 mile, 1 kilometer); the weight then increases as the distance decreases, going to infinity for $d_{ij} = 0$. To correct for this, *CrimeStat* includes an adjustment for small distances so that the maximum weight can be never be greater than 1.0 (see chapter 4). The adjustment scales distances to one mile. When the small distance adjustment is turned on, the minimal distance is scaled automatically to be one mile. The formula used is

$$W_{ij} = \frac{\text{one mile}}{\text{one mile} + d_{ij}} \tag{7.9}$$

in whichever units are specified.

Similarity or dissimilarity

An exact test of significance has not been worked out because the distribution of the statistic is not known. The expected value of I_i and the variance of I_i are somewhat complicated (see endnote 7 for the formulas).[7] Instead, high positive or high negative standardized scores of I_i, $Z(I_i)$, are taken as indicators of similarity or dissimilarity. A high *positive* standardized score indicates the spatial clustering of similar values (either high or low) while a high *negative* standardized score indicates a clustering of dissimilar values (high relative to a neighborhood that is low or, conversely, low relative to a neighborhood that is high). The higher the standardized score, the more the observation is similar (positive) or dissimilar (negative) to its neighbors.

In other words, the Local Moran statistic is a good indicator of either 'hot spots' or 'cold spots', zones which are different from their neighborhood. 'Hot spots' would be seen where the number of incidents in a zone is much higher than in the nearby zones. 'Cold spots' would be seen where the number of incidents in a zone is much lower than in the nearby zones. The Local Moran statistic indicates whether the zone is similar or dissimilar to its neighbors. A user must then look at the absolute value of the zone (i.e., the number of incidents in the zone) to see whether it is a 'hot spot' or a 'cold spot'.

For each observation, *CrimeStat* calculates the Local Moran statistic and the expected value of the Local Moran. *If* the *variance* box is checked, the program will also calculate the variance and the standardized Z-value of the Local Moran. The default is for the variance not to be calculated because the calculations are very intense and may take a long time. Therefore, a user should test how long it takes to calculate variances for a small sample on a particular computer before running the variance routine on a large sample.

Example 3: Local Moran Statistics for Auto Thefts

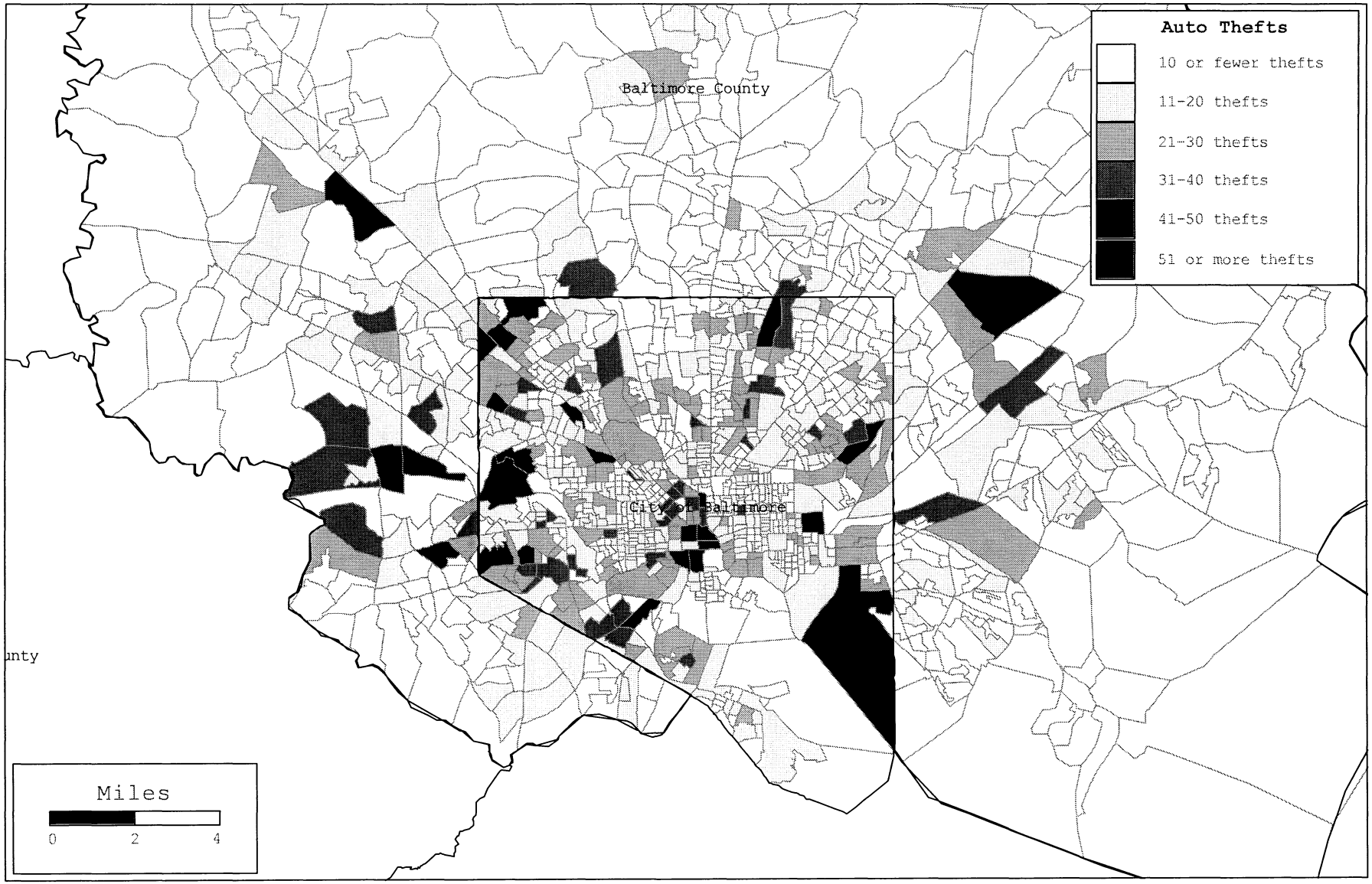
Using data on 14,853 motor vehicle thefts for 1996 in both Baltimore County and Baltimore City, the number of incidents occurring in each of 1,349 census block groups was calculated with a GIS (Figure 7.11). As seen, the pattern shows a higher concentration towards the center of the metropolitan area, as would be expected, but that the pattern is not completely uniform. There are many block groups within the City of Baltimore with very low number of auto thefts and there are a number of block groups within the County with a very high number.

Using these data, *CrimeStat* calculated the Local Moran statistic with the variance box being checked and the small distance adjustment being used. The range of I_i values varied from -37.26 to +180.14 with a mean of 5.20. The pseudo-standardized Local Moran 'Z' varied from -12.71 to 50.12 with a mean of 1.61. Figure 7.12 maps the distribution. Because a negative I_i value indicates dissimilarity, these values have been drawn in red, compared to blue for a positive I_i value. As seen, in both the City of Baltimore and the County of Baltimore, there are block groups with large negative I_i values, indicating that they differ from their surrounding block groups. For example, in the central part of Baltimore City, there is a small area of about eight block groups with low numbers of auto

Figure 7.11:

1996 Motor Vehicle Thefts

Number of Auto Thefts Per Block Group



thefts, compared to the surrounding block groups. These form a 'cold spot'. Consequently, they appear in dark tones in figure 7.12 indicating that they have high I_i values (i.e., negative autocorrelation). Similarly, there are several block groups on the western side of the County which have relatively high numbers of auto thefts compared to the surrounding block groups. They form a 'hot spot'. Consequently, they also appear in dark tones in figure 7.12 because this indicates negative spatial autocorrelation, having values that are dissimilar to the surrounding blocks.

Another use of Anselin's Local Moran statistic is to identify 'outliers', zones that are very different from their neighbors. In this case, zones with a high negative I value (e.g., with an I smaller than two standard deviations below the mean, -2) are indicative of outliers. They either have a high number of incidents whereas their neighbors have a low number or, the opposite, a low number of incidents amidst zones with a high number of incidents. Identifying the outliers can focus on zones which are unique (and which should be studied) or, in multivariate analysis, on zones which need to be statistically treated different in order to minimize a large modeling error (e.g., creating a dummy variable for the extreme outliers in a regression model).

In short, the Local Moran statistic can be a useful tool for identifying zones which are dissimilar from their neighborhood. It is the only statistic that is in *CrimeStat* that demonstrates dissimilarity. The other 'hot spot' tools will only identify areas with high concentrations. To use the Local Moran statistic, however, requires that the data be summarized into zones in order to produce the necessary intensity value. Given that most crime incident databases will list individual events without intensities, this will entail additional work by a law enforcement agency.

Some Thoughts on the Concept of 'Hot Spots'

Advantages

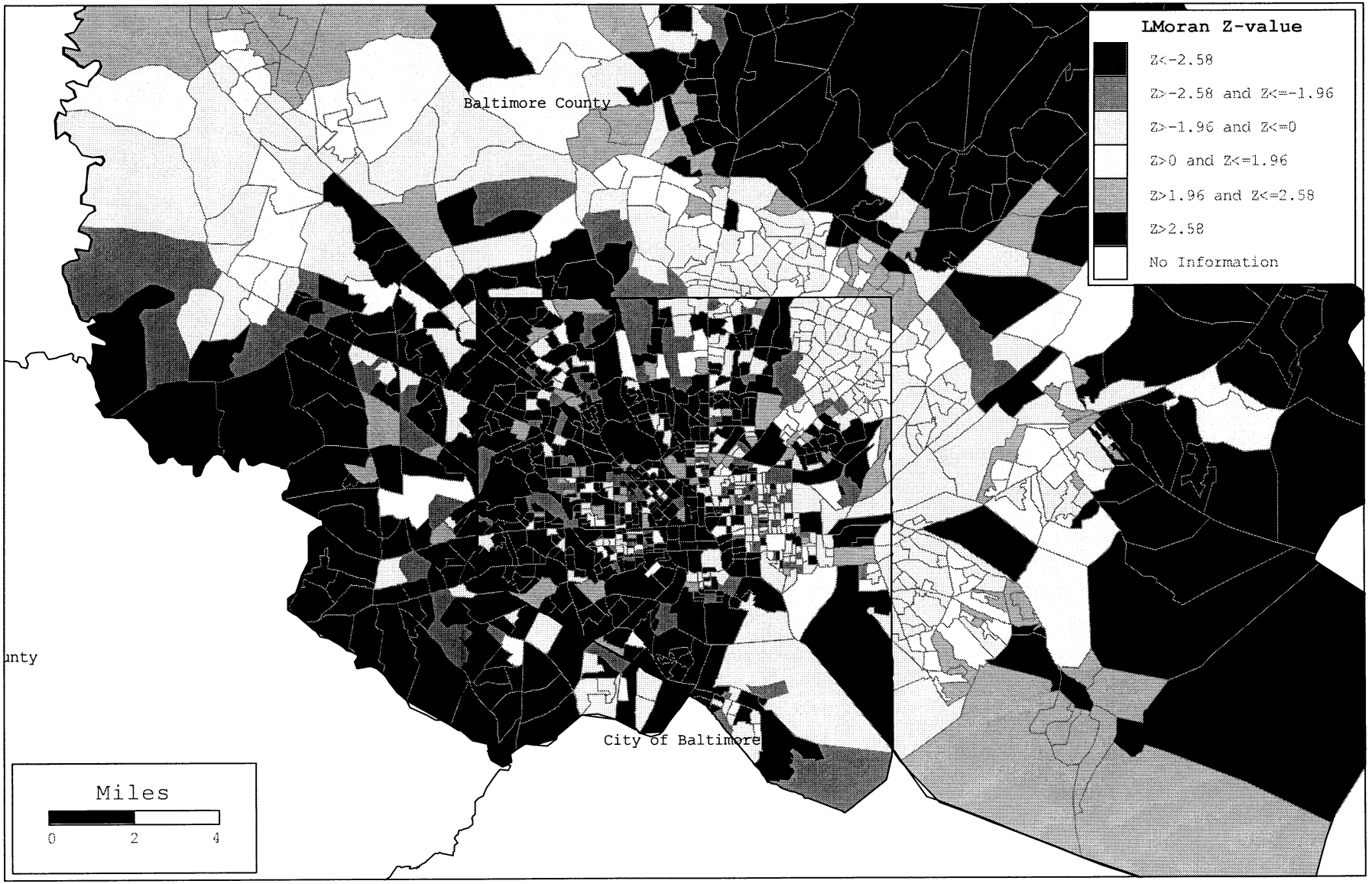
The seven techniques discussed in this and the last chapter have both advantages and disadvantages. Among the advantages are that they attempt to isolate areas of high concentration (or low concentration in the case of the Local Moran statistic) of incidents and can, therefore, help law enforcement agencies focus their resources on these areas. One of the powerful uses of a 'hot spot' concept is that it is focused. It can provide new information about locations that police officers or community workers may not recognize (Rengert, 1995). Given that most police departments are understaffed, a strategy that prioritizes intervention is very appealing. The 'hot spot' concept is imminently practical.

Another advantage to the identification of 'hot spots' is that the techniques systematically implement an algorithm. In this sense, they minimize bias on the part of officers and analysts since the technique operates somewhat independently of preconceptions. As has been mentioned, however, these techniques are not totally without human judgement since the user must make decisions on the number of 'hot spots' and the size of the search radius, choices that can allow different users to come to different

Figure 7.12:

Local Spatial Autocorrelation of 1996 Vehicle Thefts

Local Moran Z-Value of Block Groups

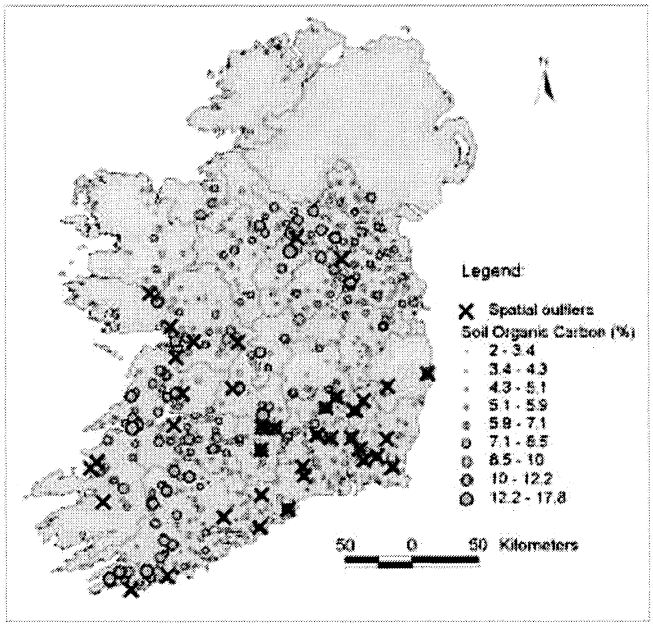


Using Local Moran's I to Detect Spatial Outliers in Soil Organic Carbon Concentrations in Ireland

Chaosheng Zhang[1] David McGrath[2]
Lecturer in GIS Research Officer
[1] Department of Geography, National University of Ireland, Galway, Ireland
[2] Teagasc, Johnstown Castle Research Centre, Wexford, Ireland

One objective in the study of soil organic carbon concentrations is to produce a reliable spatial distribution map. A geostatistical variogram analysis was applied to study the spatial structure of soils in Ireland for the purpose of carrying out a spatial interpolation with the Kriging method. The variogram looks at similarities in organic carbon concentrations as a function of distance. In the analysis, a relatively poor variogram was observed, and one of the main reasons was the existence of spatial outliers. Spatial outliers make the variogram curve erratic and hard to interpret, and impair the quality of the spatial distribution map.

CrimeStat was used to identify the spatial outliers. The parameter of the standardized Anselin's Local Moran's I (z) was used. When $z < -1.96$, the sample was defined as a spatial outlier. Out of 678 soil samples, a total of 39 samples were detected as spatial outliers, and excluded in the spatial structure calculation. As a consequence, the variogram curve was significantly improved. This improvement made the final spatial distribution map more reliable and trustable.



Spatial outliers are clearly different from the majority of samples nearby. Compared with the samples nearby, high value spatial outliers are found in the southeastern part, and low value spatial outliers are located in the western and northern parts of the country.

conclusions. There is probably no way to get around subjectivity since law enforcement personnel may not use a result unless it partly confirms what they already know. But, by implementing an algorithm, it forces users to at least go through the steps systematically.

A third advantage is that these techniques are visual, particularly when used with a GIS. The mode and fuzzy mode routines output the results as a dbf file, which can be displayed in a GIS as a proportional circle. The Nnh, Rnnh, Stac, and Kmeans routines can output the results directly as graphical objects, either as standard deviational ellipses or convex hulls; these can be displayed directly in a GIS. The Local Moran technique can be adapted for thematic mapping (as Figure 7.12 demonstrates). Visual information can help crime analysts and officers to understand the distribution of crime in an areas, a necessary step in planning a successful intervention. We should never underestimate the importance of visualization in any analysis.

Limitations

However, there are also some distinct limitations to the concept of a 'hot spot', some technical and some theoretical. The choice involved in a user making a decision on how strict or how loose to create clusters allows the potential for subjectivity, as has been mentioned. In this sense, isolating clusters (or 'hot spots') can be as much an art as it is a science. There are limits to this, however. As the sample size goes up, there is less difference in the result that can be produced by adjusting the parameters. For example, with 6,000 or more cases, there is very little difference between using the 0.1 significance level in the nearest neighbor clustering routine and the 0.001 significance level.[8] Thus, the subjectivity of the user is more important for smaller samples than larger ones.

A second problem with the 'hot spot' concept is that it is usually applied to the volume of incidents and not to the underlying risk. Clusters (or 'hot spots') are defined by a high concentration of incidents within a small geographical area, that is on the volume of incidents within an area. This is an implicit *density* measure - the number of incidents per unit of area (e.g., incidents per square mile). But higher density can also be a function of a higher population at risk.

For some policing policies, this is fine. For example, beat officers will necessarily concentrate on high incident density neighborhoods because so much of their activity revolves around those neighborhoods. From a viewpoint of providing concentrated policing, the density or volume of incidents is a good index for assigning police officers (Sherman and Weisburd, 1995). From the viewpoint of ancillary security services, such as access to emergency medical services, neighborhood watch organizations, or residential burglar alarm retail outlets, areas with higher concentrations of incidents may be a good focal point for organizing these services.

But for other law enforcement policies, a density index is not a good one. From the viewpoint of crime prevention, for example, high incident volume areas are not necessarily unsafe and that effective preventive intervention will not necessarily lead to reduction in crime. It may be far more effective to target high risk areas rather than high volume

areas. In high risk areas, there are special circumstances which expose the population to higher-than-expected levels of crime, perhaps particular concentrations of activities (e.g., drug trading) or particular land uses that encourage crime (e.g., skid row areas) or particular concentrations of criminal activities (e.g., gangs). A prevention strategy will want to focus on those special factors and try to reduce them.

Risk, which is defined as the number of incidents relative to the number of potential victims/targets, is only loosely correlated with the volume of incidents. Yet, 'hot spots' are usually defined by volume, rather than risk. The risk-adjusted hierarchical nearest neighbor clustering routine, discussed in chapter 6, is the only tool among these that identifies risk, rather than volume. It is clear that more tools will be needed to examine hot spot locations that are more at risk.

The final problem with the 'hot spot' concept is more theoretical. Namely, given a concentration of incidents, how do we explain it? To identify a concentration is one thing. To know how to intervene is another. It is imperative that the analyst discover some of the underlying causes that link the events together in a systematic way. Otherwise, all that is left is an empirical description without any concept of the underlying causes. For one thing, the concentration could be random or haphazard; it could have happened one time, but never again. For another, it could be due to the concentration of the population *at risk*, as discussed above. Finally, the concentration could be circumstantial and not be related to anything inherent about the location.

The point here is that an empirical description of a location where crime incidents are concentrated is only a first step in defining a real 'hot spot'. It is an *apparent* 'hot spot'. Unless the underlying vector (cause) is discovered, it will be difficult to provide adequate intervention. The causes could be environmental (e.g., concentrations of land uses that attract attackers and victims) or behavioral (e.g., concentrations of gangs). The most one can do is try to increase the concentration of police officers. This is expensive, of course, and can only be done for limited periods. Eventually, if the underlying vector is not dealt with, incidents will continue and will overwhelm the additional police enforcement. In other words, ultimately, reducing crime around a 'hot spot' will need to involve many other policies than simply police enforcement, such as community involvement, gang intervention, land use modification, job creation, the expansion of services, and other community-based interventions. In this sense, the identification of an empirical 'hot spot' is frequently only a window into a much deeper problem that will involve more than targeted enforcement.

Endnotes for Chapter 7

1. STAC is an abbreviation for Spatial and Temporal Analysis of Crime. The temporal section of the program was superceded by several other programs and was not updated for the millennium. Because many law enforcement users refer to STAC ellipses, we have retained that name.

2. The first two digits of a beat number designate the District.

3. The Chicago Police Department made available the incidents in this analysis to Richard Block for the evaluation of the Chicago Alternative Police Strategy (CAPS).

4. In general a designated main surface street occurs every mile on Chicago's grid, and there are eight blocks to the mile. In this map, Lawrence and Ashland are main Grid streets. In this area, there are also several diagonal main streets that either follow the lake shore or old Indian trails.

5. The total number of ways for selecting K distinct combinations of N incidents, irrespective of order, is (Burt and Barber, 1996, 155):

$$\frac{N!}{K!\,(N-K)!}$$

6. The steps are as follows:

Global Selection of Initial Seed Locations

A. A 100 x 100 grid is overlaid on the point distribution; the dimensions of the grid are defined by the minimum and maximum X and Y coordinates.

B. A separation distance is defined, which is

$$\text{Separation} = t * 0.5\ \text{SQRT}\left[\frac{A}{N}\right]$$

where t is the Student's t-value for the .01 significance level (2.358), A is the area of the region, and N is the sample size. The separation distance was calculated to prevent adjacent cells from being selected as seeds.

C. For each grid cell, the number of incidents found are counted and then sorted in descending order.

D. The cell with the highest number of incidents found is the initial seed for cluster 1.

E. The cell with the next highest number of incidents is temporarily selected. If the distance between that cell and the seed 1 location is *equal to or greater than* the separation distance, this cell becomes initial seed 2.

F. If the distance is less than the separation distance, the cell is dropped and the routine proceeds to the cell with the next highest number of incidents.

G. This procedure is repeated until *K initial seeds* have been located thereby selecting the remaining cell with the highest number of incidents and calculating its distance to all prior seeds. If the distance is equal to or greater than the separation distance, then the cell is selected as a seed. If the distance is less than the separation distance, then the cell is dropped as a seed candidate. Thus, it is possible that *K* initial seeds cannot be identified because of the inability to locate *K* locations greater than the threshold distance. In this case, *CrimeStat* keeps the number it has located and prints out a message to this effect.

Local Optimization of Seed Locations

H. After the *K* initial seeds have been selected, all points are assigned to the nearest initial seed location. These are the initial cluster groupings.

I. For each initial cluster grouping in turn, the center of minimum distance is calculated. These are the second seed locations.

J. All points are assigned to the nearest second seed location.

K. For each new cluster grouping in turn, the center of minimum distance is calculated. These are third seed locations.

L. Steps J and K are repeated until no more points change cluster groupings. These are the final seed locations and cluster groupings.

7. The formulas are as follows as follows. The expected value of the Local Moran is:

$$E(I_i) = \frac{-\sum_{j=1}^{N} W_{ij}}{N - 1}$$

where W_{ij} is a distance weight for the interaction between observations i and j (either an adjacency index or a weight decreasing with distance). The variance of the Local Moran is defined in three steps:

A. First, define b_2.

$$b_2 = \sum \{ \frac{(X_i - \bar{X})^4}{N} \} \; / \; [\sum \{ \frac{(X_i - \bar{X})^2}{N} \}]^2$$

This is the fourth moment around the mean divided by the squared second moment around the mean.

B. Second, define $2w_{i(kh)}$:

$$2w_{i(kh)} = \sum \sum W_{ik} W_{ih} \qquad \text{where } k \neq i \text{ and } h \neq i$$

This term is twice the sum of the cross-products of all weights for i with themselves, using k and h to avoid the use of identical subscripts. Since each pair of observations, i and j, has its own specific weight, a cross-product of weights are two weights multiplied by each other (where $i \neq j$) and the sum of these cross-products is twice the sum of all possible interactions irrespective of order (i.e., $W_{ij} = W_{ji}$). Because the weight of an observation with itself is zero (i.e., $W_{ii} = 0$), all terms can be included in the summation.

C. Third, define the variance, standard deviation, and an approximate (pseudo) standardized score of I_i:

$$\text{Var}(I_i) = \frac{(\sum w_{ij}^2)*(n - b_2)}{(n-1)} + \frac{2w_{i(kh)}(2b_2 - n)}{(n-1)(n-2)} + \frac{(\sum w_{ij})^2}{(n-1)^2}$$

$$S(I_i) = \sqrt{[\text{Var}(I_i)]}$$

$$Z(I_i) = [I_i - E(I_i)] / S(I_i)$$

8. On one test of 6,051 burglaries with a minimum cluster size requirement of 10 incidents, for example, we obtained 100 first-order clusters, 9 second-order clusters, and no third-order clusters by using a 0.1 significance level for the nearest neighbor hierarchical clustering routine. When the significance level was reduced to 0.001, the number of clusters extracted was 97 first-order clusters, 8 second-order clusters, and no third-order clusters.

CrimeStat III

Part III: Spatial Modeling

Chapter 8
Kernel Density Interpolation

In this chapter, we discuss tools aimed at interpolating incidents, using the kernel density approach. *Interpolation* is a technique for generalizing incident locations to an entire area. Whereas the spatial distribution and hot spot statistics provide statistical summaries for the data incidents themselves, interpolation techniques generalize those data incidents to the entire region. In particular, they provide *density* estimates for all parts of a region (i.e., at any location). The density estimate is an intensity variable, a Z-value, that is estimated at a particular location. Consequently, it can be displayed by either surface maps or contour maps that show the intensity at all locations.

There are many interpolation techniques, such as Kriging, trend surfaces, local regression models (e.g., Loess, splines), and Dirichlet tessellations (Anselin, 1992; Cleveland, Grosse and Shyu, 1993; Venables and Ripley, 1997). Most of these require a variable that is being estimated as a function of location. However, *kernel density estimation* is an interpolation technique that is appropriate for individual point locations (Silverman, 1986; Härdle, 1991; Bailey and Gatrell, 1995; Burt and Barber, 1996; Bowman and Azalini, 1997).

Kernel Density Estimation

Kernel density estimation involves placing a symmetrical surface over each point, evaluating the distance from the point to a reference location based on a mathematical function, and summing the value of all the surfaces for that reference location. This procedure is repeated for all reference locations. It is a technique that was developed in the late 1950s as an alternative method for estimating the density of a histogram (Rosenblatt, 1956; Whittle, 1958; Parzen, 1962). A histogram is a graphic representation of a frequency distribution. A continuous variable is divided into intervals of size, s (the interval or bin width), and the number of cases in each interval (bin) are counted and displayed as block diagrams. The histogram is assumed to represent a smooth, underlying distribution (a density function). However, in order to estimate a smooth density function from the histogram, traditionally researchers have linked adjacent variable intervals by connecting the midpoints of the intervals with a series of lines (Figure 8.1).

Unfortunately, doing this causes three statistical problems (Bowman and Azalini, 1997):

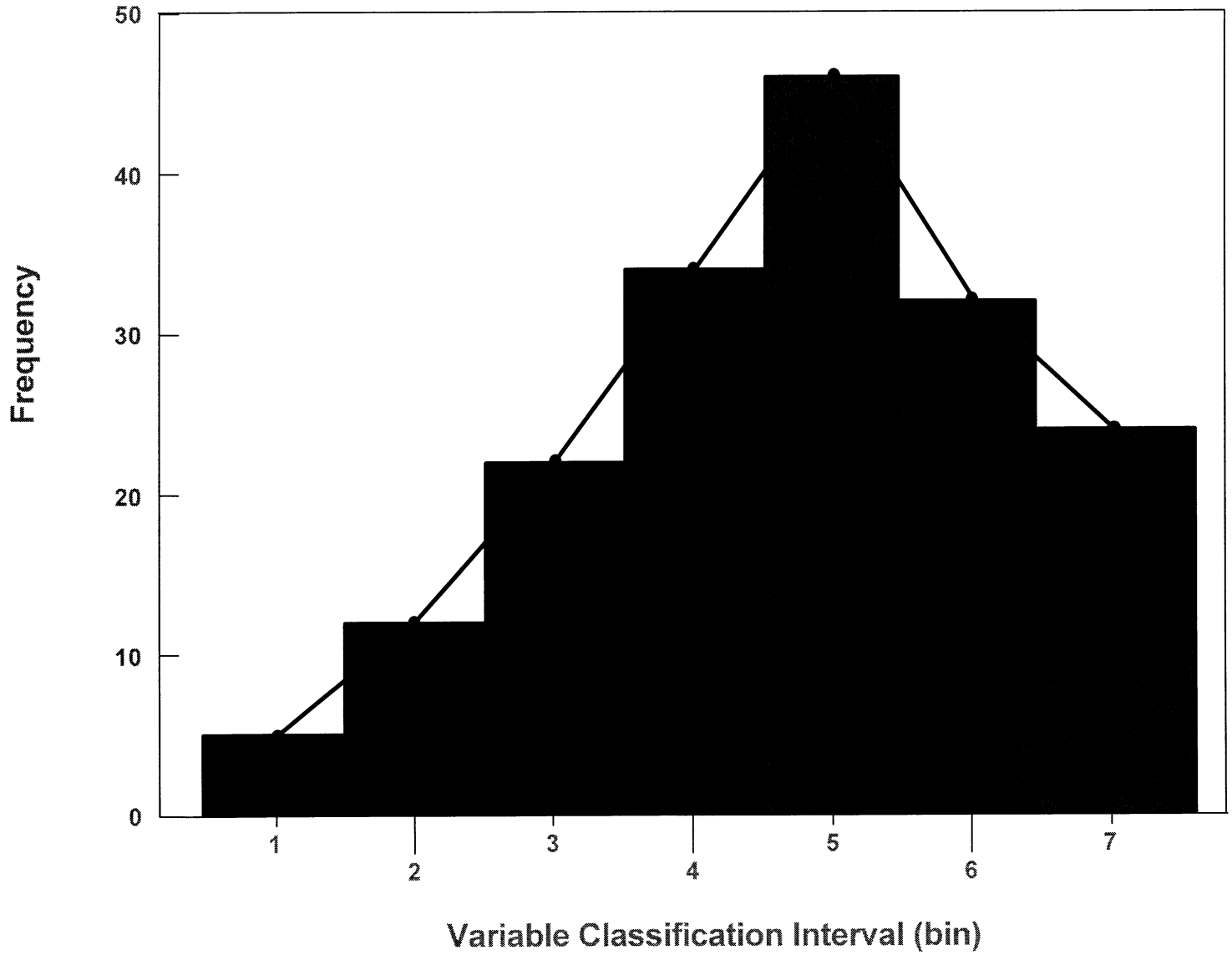
1. Information is discarded because all cases within an interval are assigned to the midpoint. The wider the interval, the greater the information loss.

2. The technique of connecting the midpoints leads to a discontinuous and not smooth density function even though the underlying density function is assumed to be smooth. To compensate for this, researchers will reduce the width of the interval. Thus, the density function becomes smoother with

Figure 8.1:

Constructing A Density Estimate From A Histogram

Method of Connecting Midpoints



smaller interval widths, although still not very smooth. Further, there are limits to this technique as the sample size decreases when the bin width gets smaller, eventually becoming too small to produce reliable estimates.

3. The technique is dependent on an arbitrarily defined interval size (bin width). By making the interval wider, the estimator becomes cruder and, conversely, by making the interval narrower, the estimator becomes finer. However, the underlying density distribution is assumed to be smooth and continuous and not dependent on the interval size of a histogram.

To handle this problem, Rosenblatt (1956), Whittle (1958) and Parzen (1962) developed the kernel density method in order to avoid the first two of these difficulties; the bin width issue still remains. What they did was to place a smooth *kernel function*, rather than a block, over each point and sum the functions for each location on the scale. Figure 8.2 illustrates the process with five point locations. As seen, over each location, a symmetrical kernel function is placed; by symmetrical is meant that is falls off with distance from each point at an equal rate in both directions around each point. In this case, it is a normal distribution, but other types of symmetrical distribution have been used. The underlying density distribution is estimated by summing the individual kernel functions at *all* locations to produce a smooth cumulative density function. Notice that the functions are summed at every point along the scale and not just at the point locations. The advantages of this are that, first, each point contributes equally to the density surface and, second, the resulting density function is continuous at all points along the scale.

The third problem mentioned above, interval size, still remains since the width of the kernel function can be varied. In the kernel density literature, this is called *bandwidth* and refers essentially to the width of the kernel. Figure 8.3 shows a kernel with a narrow bandwidth placed over the same five points while figure 8.4 shows a kernel with a wider bandwidth placed over the points. Clearly, the smoothness of the resulting density function is a consequence of the bandwidth size.

There are a number of different kernel functions that have been used, aside from the normal distribution, such as a triangular function (Burt and Barber, 1996) or a quartic function (Bailey and Gatrell, 1995). Figure 8.5 illustrates a quartic function. But the normal is the most commonly used (Kelsall and Diggle, 1995a).

The ***normal distribution*** function has the following functional form:

$$g(x_j) = \sum \left\{ [W_i * I_i] * \frac{1}{h^2 * 2\pi} * e^{-\left[\frac{d_{ij}^2}{2*h^2}\right]} \right\} \qquad (8.1)$$

where d_{ij} is the distance between an incident location and any reference point in the region, h is the standard deviation of the normal distribution (the bandwidth), W_i is a weight at the point location and I_i is an intensity at the point location. This function extends to infinity in all directions and, thus, will be applied to any location in the region.

Figure 8.2:

Kernel Density Estimate

Summing of Normal Kernel Functions for 5 Points

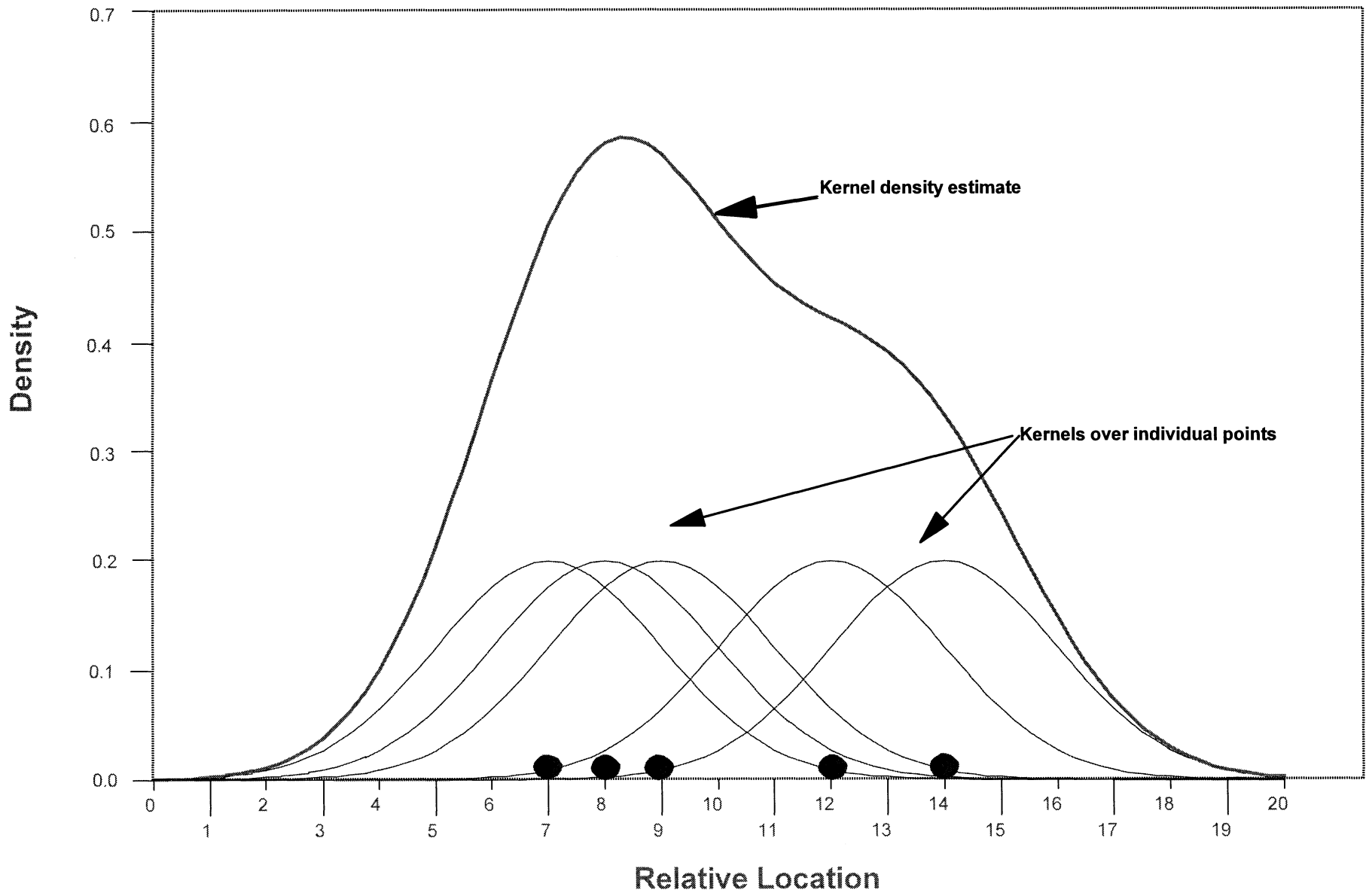


Figure 8.3:

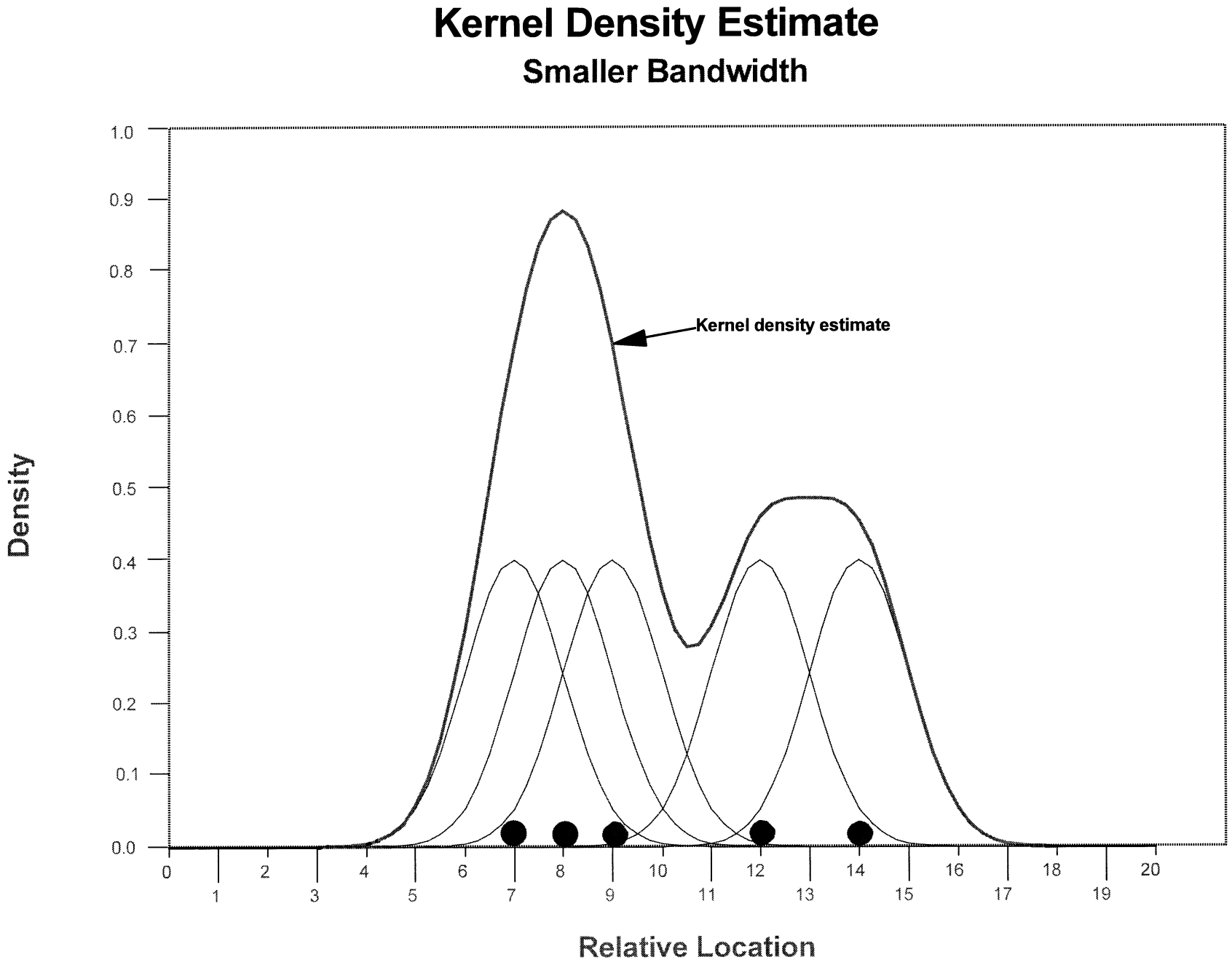


Figure 8.4:

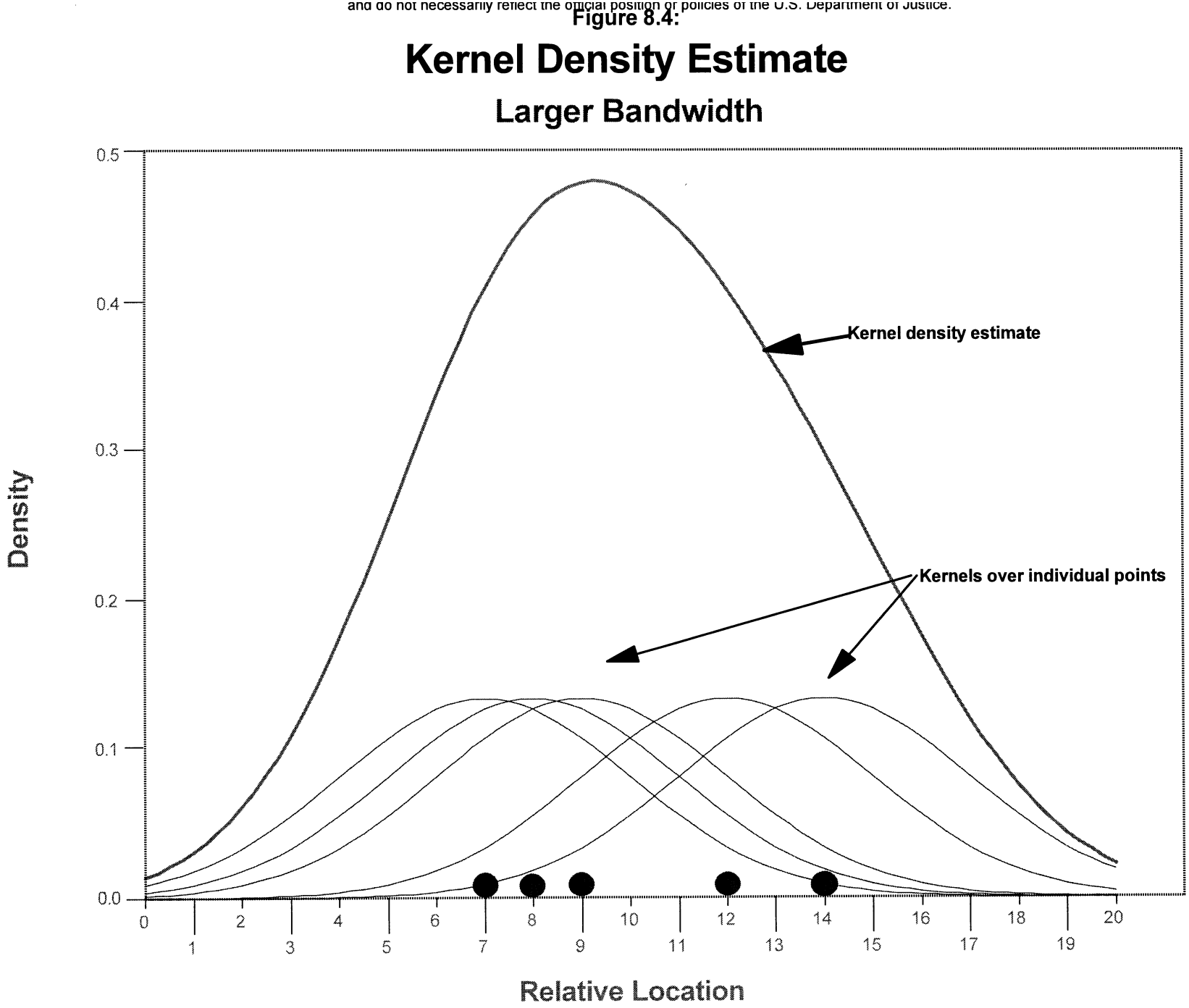
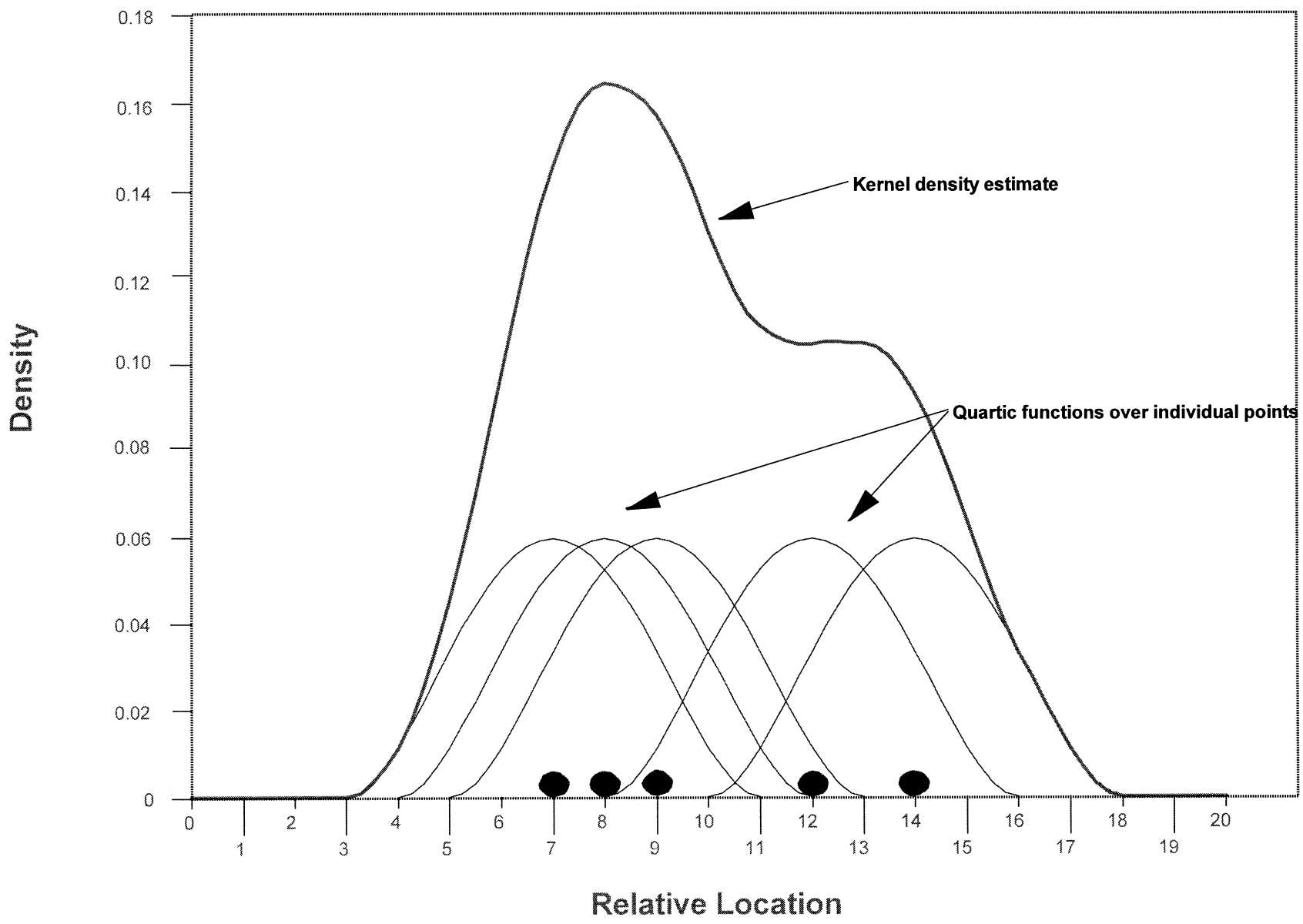


Figure 8.5:



In *CrimeStat*, there are four alternative kernel functions that can be used, all of which have a circumscribed radius (unlike the normal distribution). The ***quartic*** function is applied to a limited area around each incident point defined by the radius, h. It falls off gradually with distance until the radius is reached. Its functional form is:

1. Outside the specified radius, h:

$$g(x_j) = 0 \tag{8.2}$$

2. Within the specified radius, h:

$$g(x_j) = \sum \left\{ [W_i * I_i] * \left[\frac{3}{h^2 * \pi} \right] * \left[1 - \frac{d_{ij}^2}{h^2} \right]^2 \right\} \tag{8.3}$$

where d_{ij} is the distance between an incident location and any reference point in the region, h is the radius of the search area (the bandwidth), W_i is a weight at the point location and I_i is an intensity at the point location.

The ***triangular*** (or conical) distribution falls off evenly with distance, in a linear relationship. Compared to the quartic function, it falls off more rapidly. It also has a circumscribed radius and is, therefore, applied to a limited area around each incident point, h. Its functional form is:

1. Outside the specified radius, h:

$$g(x_j) = 0 \tag{8.4}$$

2. Within the specified radius, h:

$$g(x_j) = \sum [K - K/h] * d_{ij} \tag{8.5}$$

where K is a constant. In *CrimeStat*, the constant K is initially set to 0.25 and then re-scaled to ensure that either the densities or probabilities sum to their appropriate values (i.e., N for densities and 1.00 for probabilities).

The ***negative exponential*** (or peaked) distribution falls off very rapidly with distance up to the circumscribed radius. Its functional form is:

1. Outside the specified radius, h:

$$g(x_j) = 0 \tag{8.6}$$

2. Within the specified radius, h:

$$g(x_j) = \sum A * e^{-K * dij} \tag{8.7}$$

where A is a constant and K is an exponent. In *CrimeStat*'s implementation, K is set to 3 while A is initially set to 1 and then re-scaled to ensure that either the densities or probabilities sum to their appropriate values (i.e., N for densities and 1.00 for probabilities).

Finally, the ***uniform*** distribution weights all points within the circle equally. Its functional form is:

1. Outside the specified radius, h:

$$g(x_j) = 0 \quad (8.8)$$

2. Within the specified radius, h:

$$g(x_j) = \Sigma K \quad (8.9)$$

where K is a constant. Initially, K is set to 0.1 but then re-scaled to ensure that either the densities or probabilities sum to their appropriate values (i.e., N for densities and 1.00 for probabilities).

Kernel Parameters

The user can select these five different kernel functions to interpolate the data to the grid cells. They produce subtle differences in the shape of the interpolated surface or contour. The normal distribution weighs all points in the study area, though near points are weighted more highly than distant points. The other four techniques use a circumscribed circle around the grid cell. The uniform distribution weighs all points within the circle equally. The quartic function weighs near points more than far points, but the fall off is gradual. The triangular function weighs near points more than far points within the circle, but the fall off is more rapid. Finally, the negative exponential weighs near point much more highly than far points within the circle.

The use of any of one of these depends on how much the user wants to weigh near points relative to far points. Using a kernel function which has a big difference in the weights of near versus far points (e.g., the negative exponential or the triangular) tends to produce finer variations within the surface than functions which are weight more evenly (e.g., the normal distribution, the quartic, or the uniform); these latter ones tend to *smooth* the distribution more.

Shape and size of the bandwidth

However, Silverman (1986) has argued that it does not make that much difference as long as the kernel is symmetrical. There are also edge effects that can occur and there have been different proposed solutions to this problem (Venables and Ripley, 1997).

There have also been variations of the size of the of bandwidth with various formulas and criteria (Silverman, 1986; Härdle, 1991; Venables and Ripley, 1997). Generally, bandwidth choice fall into either fixed or adaptive (variable) choices (Kelsall and Diggle, 1995a; Bailey and Gatrell, 1995). *CrimeStat* follows this distinction, which will be explained below.

Another suggestion is to use the Moran correlogram, which was discussed in chapter 4, to estimate the shape of the weighting function (Cliff and Haggett, 1988; Bailey and Gattrell, 1995). This would be appropriate for variables that have *weights*, such as population or employment. The Moran correlogram displays the degree of spatial autocorrelation as a function of distance. Whether the autocorrelation falls off quickly or more slowly can be used to select an approximate kernel function (e.g., a negative exponential function falls off quickly whereas a quartic function falls off very slowly). The bandwidth could also be selected by the distance at which the Moran correlogram levels off (i.e., approaches the global I value). This would lead to an estimate that minimizes spatial autocorrelation in the data set. It would be good for capturing major trends in the data, but would not be good for identifying local clusters (hot spots) since the bandwidth distance would incorporate most of a metropolitan area.

Three-dimensional kernels

The kernel function can be expanded to more than two dimensions (Härdle, 1991; Bailey and Gatrell, 1995; Burt and Barber, 1996; Bowman and Azalini, 1997). Figure 8.6 shows a three-dimensional normal distribution placed over each of five points with the resulting density surface being a sum of all five individual surfaces. Thus, the method is particularly appropriate for geographical data, such as crime incident locations. The method has also been developed to relate two or more variables together by applying a kernel estimate to each variable in turn and then dividing one by the other to produce a three-dimensional estimate of *risk* (Kelsall and Diggle, 1995a; Bowman and Azalini, 1997).

Significance testing of density estimates is more complicated. Current techniques tend to focus on simulating surfaces under spatially random assumptions (Bowman and Azaline, 1997; Kelsall and Diggle, 1995b). Because of the still experimental nature of the testing, *CrimeStat* does not include any testing of density estimates in this version.

CrimeStat Kernel Density Methods

CrimeStat has two interpolation techniques, both based on the kernel density technique. The first applies to a single variable, while the second to the relationship between two variables. Both routines have a number of options. Figure 8.7 shows the interpolation page in *CrimeStat*. Users indicate their choices by clicking on the tab and menu items. For either technique, it is necessary to have a reference file, which is usually a grid placed over the study region (see chapter 3). The reference file represents the region to which the kernel estimate will be generalized (figure 8.8).

Figure 8.6:

Kernel Density Surfaces

Summing of Normal Kernel Surfaces for 5 Points

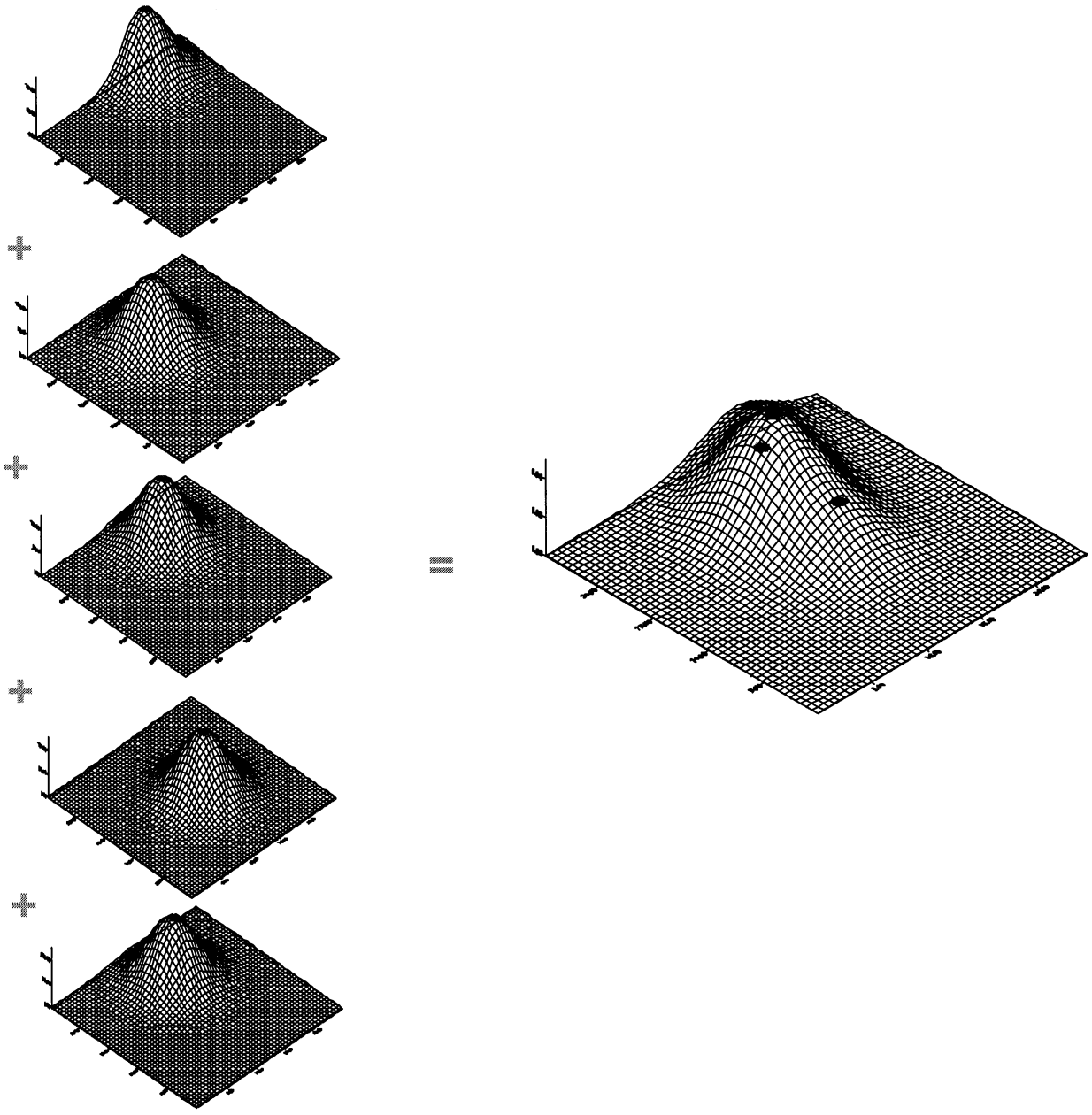


Figure 8.7: Interpolation Screen

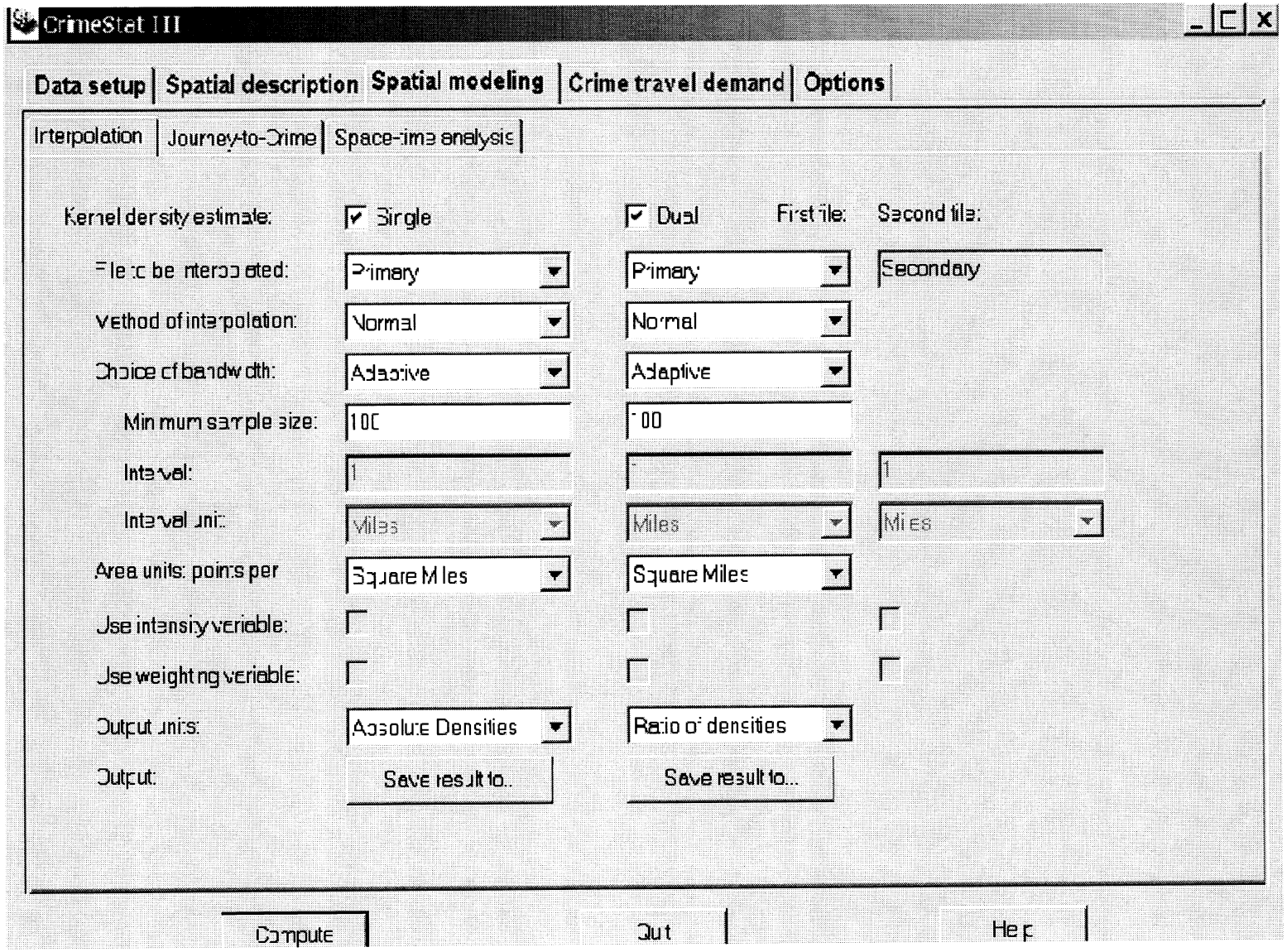
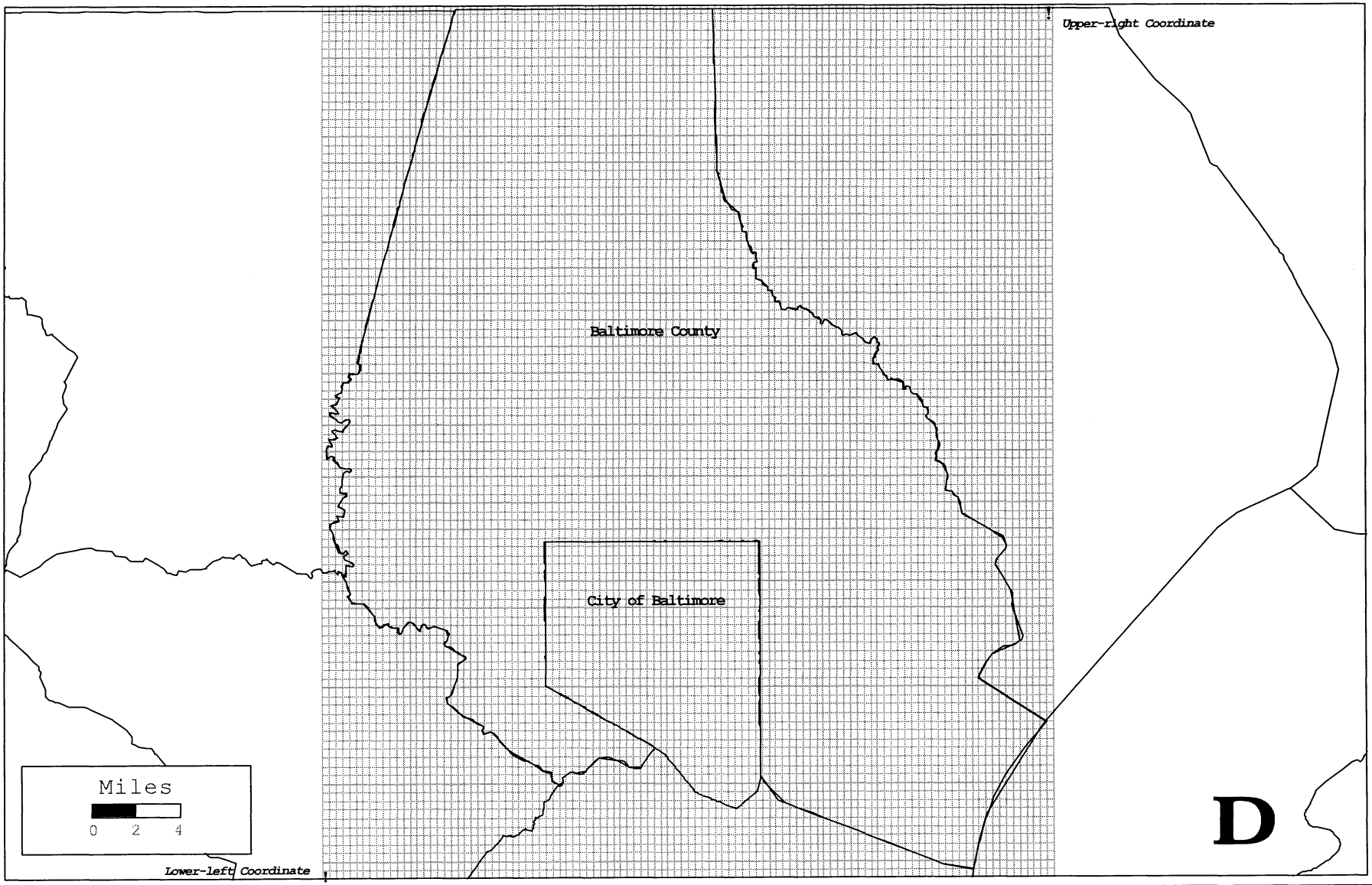


Figure 8.8:

Grid Cell Structure for Baltimore Region

108 Width x 100 Height Grid Cells



Single Density Estimates

The single kernel density routine in *CrimeStat* is applied to a distribution of point locations, such as crime incidents. It can be used with either a primary file or a secondary file; the primary file is the default. For example, the primary file can be the location of motor vehicle thefts. The points can also have a weighting or an associated intensity variable (or both). For example, the points could represent the location of police stations while the weights (or intensities) represent the number of calls for service. Again, the user must be careful in having both a weighting variable and an intensity variable as the routine will use both variables in calculating densities; this could lead to double weighting.

Having defined the file on the primary (or secondary) file tabs, the user indicates the routine by checking the 'Single' box. Also, it is necessary to define a reference file, either an existing file or one generated by *CrimeStat* (see chapter 3). There are other parameters that must be defined.

File to be Interpolated

The user must indicate whether the primary file or the secondary file (if used) is to be interpolated.

Method of Interpolation

The user must indicate the method of interpolation. Five types of kernel density estimators are used:

1. Normal distribution (bell; default)
2. Uniform (flat) distribution
3. Quartic (spherical) distribution
4. Triangular (conical) distribution
5. Negative exponential (peaked) distribution

In our experience, there are advantages to each. The normal distribution produces an estimate over the entire region whereas the other four produce estimates only for the circumscribed bandwidth radius. If the distribution of points is sparse towards the outer parts of the region, then the four circumscribed functions will not produce estimates for those areas, whereas the normal will. Conversely, the normal distribution can cause some edge effects to occur (e.g., spikes at the edge of the reference grid), particularly if there are many points near one of the boundaries of the study area. The four circumscribed functions will produce less of a problem at the edges, although they still can produce some spikes. Within the four circumscribed functions, the uniform and quartic tend to smooth the data more whereas the triangular and negative exponential tend to emphasize 'peaks' and 'valleys'. The differences between these different kernel functions are small, however. The user should probably start with the default normal function and adjust accordingly to how the surface or contour looks.

Choice of Bandwidth

The user must indicate how bandwidths are to be defined. There are two types of bandwidth for the single kernel density routine, fixed interval or adaptive interval.

Fixed interval

With a fixed bandwidth, the user must specify the interval to be used and the units of measurement (squared miles, squared nautical miles, squared feet, squared kilometers, or squared meters). Depending on the type of kernel estimate used, this interval has a slightly different meaning. For the normal kernel function, the bandwidth is the standard deviation of the normal distribution. For the uniform, quartic, triangular, or negative exponential kernels, the bandwidth is the radius of the search area to be interpolated.

There are few guidelines for choosing a particular bandwidth other than by visual inspection (Venables and Ripley, 1997). Some have argued that the bandwidth be no larger than the finest resolution that is desired and others have argued for a variation on random nearest neighbor distances (see Spencer Chainey application later in this chapter). Others have argued for particular sizes (Silverman, 1986; Härdle, 1991; Kadafar, 1996; Farewell, 1999; Talbot, Kulldorff, Forand, and Haley, 2000).[1] There does not seem to be consensus on this issue. Consequently, *CrimeStat* leaves the definition up to the user.

Typically, a narrower bandwidth interval will lead to a finer mesh density estimate with all the little peaks and valleys. A larger bandwidth interval, on the other hand, will lead to a smoother distribution and, therefore, less variability between areas. While smaller bandwidths show greater differentiation among areas (e.g., between 'hot spot' and 'low spot' zones), one has to keep in mind the statistical precision of the estimate. If the sample size is not very large, then a smaller bandwidth will lead to more imprecision in the estimates; the peaks and valleys may be nothing more than random variation. On the other hand, if the sample size is large, then a finer density estimate can be produced. In general, it is a good idea to experiment with different fixed intervals to see which results make the most sense.

Adaptive interval

An adaptive bandwidth adjusts the bandwidth interval so that a minimum number of points are found. This has the advantage of providing constant precision of the estimate over the entire region. Thus, in areas that have a high concentration of points, the bandwidth is narrow whereas in areas where the concentration of points is more sparse, the bandwidth will be larger. This is the default bandwidth choice in *CrimeStat* since we believe that consistency in statistical precision is paramount. The degree of precision is generally dependent on the sample size of the bandwidth interval. The default is a minimum of 100 points within the bandwidth radius. The user can make the estimate more fine grained by choosing a smaller number of points (e.g., 25) or more smooth by choosing a larger number of points (e.g., 200). Again, experimentation is necessary to see which results make the most sense.

Output Units

The user must indicate the measurement units for the density estimate in points per squared miles, squared nautical miles, squared feet, squared kilometers, or squared meters. The default is points per square mile.

Intensity or Weighting Variables

If an intensity or weighting variable is used, these boxes must be checked. Be careful about using both an intensity and a weighting variable to avoid 'double weighting'.

Density Calculations

The user must indicate the type of output for the density estimates. There are three types of calculation that can be conducted with the kernel density routine. The calculations are applied to each reference cell:

1. The kernel estimates can be calculated as *absolute density* estimates using formulas 8.1-8.9, depending on what type of kernel function is used. The estimates at each reference cell are re-scaled so that the sum of the densities over all reference grids equals the total number of incidents; this is the default value.

2. The kernel estimates can be calculated as *relative density* estimates. These divide the absolute densities by the area of the grid cell. It has the advantage of interpreting the density in terms that are familiar. Thus, instead of a density estimate represented by points per grid cell, the relative density will convert this to points per, say, square mile.

3. The densities can be converted into *probabilities* by dividing the density at any one cell by the total number of incidents.

Since the three types of calculation are directly interrelated, the output surface will not differ in its variability. The choice would depend on whether the calculations are used to estimate absolute densities, relative densities, or probabilities. For comparisons between different types of crime or between the same type of crime and different time periods, usually absolute densities are the unit of choice (i.e., incidents per grid cell). However, to express the output as a probability, that is, the likelihood that an incident would occur at any one location, then outputing the results as probabilities would make more sense. For display purposes, however, it makes no difference as both look the same.

Output Files

Finally, the results can be displayed in an output table or can be output into two formats: 1) Raster grid formats for display in a surface mapping program- *Surfer for Windows* '.dat' format (Golden Software, 1994) or *ArcView Spatial Analyst* 'asc' format

(ESRI, 1998); or 2) Polygon grids in *ArcView* '.shp', *MapInfo* '.mif' or *Atlas*GIS* '.bna' formats.[2] However, all but *Surfer for Windows* require that the reference grid be created by *CrimeStat*.

Example 1: Kernel Density Estimate of Street Robberies

An example can illustrate the use of the single kernel density routine. Figure 8.9 shows a *Surfer for Windows* output of the 1180 street robberies for 1996 in Baltimore County. The reference grid was generated by *CrimeStat* and had 100 columns and 108 rows. Thus, the routine calculated the distance between each of the 10,800 reference cells and the 1180 robbery incident locations, evaluated the kernel function for each measured distance, and summed the results for each reference cell. The normal distribution kernel function was selected for the kernel estimator and an adaptive bandwidth with a minimum sample size of 100 was chosen as the parameters.

There are three views in the figure: 1) a map view showing the location of the incidents; 2) a surface view showing a three-dimensional interpolation of robbery density; and 3) a contour view showing contours of high robbery density. The surface and contour views provide different perspectives. The surface shows the peaks very clearly and the relative density of the peaks. As can be seen, the peak for robberies on the eastern part of the County is much higher than the two peaks in the central and western parts of the County. The contour view can show where these peaks are located; it is difficult to identify location clearly from a three-dimensional surface map. Highways and streets could be overlaid on top of the contour view to identify more precisely where these peaks are located.

Figure 8.10 shows an *ArcViewSpatial Analyst* map of the robbery density with the robbery incident locations overlaid on top of the density contours. Here, we can see quite clearly that there are three strong concentrations of incidents, one spreading over a distance of several miles on the west side, one on northern border between Baltimore City and Baltimore County, and one on the east side; there is also one smaller peak in the southeast corner of the County.

From a statistical perspective, the kernel estimate is a better 'hot spot' identifier than the cluster analysis routines discussed in chapter 6. Cluster routines group incidents into clusters and distinguish between incidents which belong to the cluster and those which do not belong. Depending on which mathematical algorithms are used, different clustering routines will return differing allocations of incidents to clusters. The kernel estimate, on the other hand, is a continuous surface; the densities are calculated at *all* locations; thus, the user can visually inspect the variability in density and decide what to call a 'hot spot' without having to define arbitrarily where to cut-off the 'hot spot' zone.

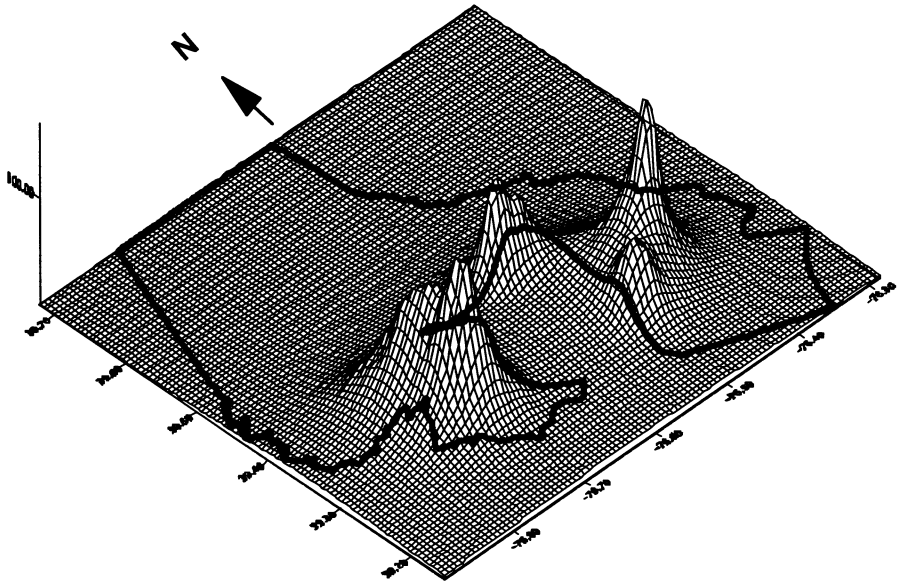
Going back to the *Surfer for Windows* output, figure 8.11 shows the effects of varying the bandwidth parameters. There are three fixed bandwidth intervals (0.5, 1, and 2 miles respectively) and there are two adaptive bandwidth intervals (a minimum of 25 and 100 points respectively). As can be seen, the fineness of the interpolation is affected by

Figure 8.9:

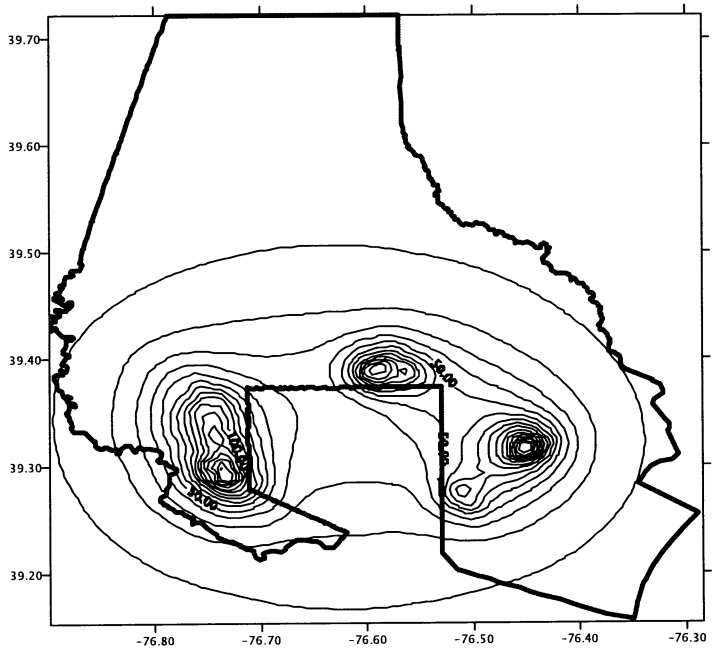
Baltimore County Robberies: 1996-97

Kernel Density Interpolation

Surface View



Contour View



Ground Level View

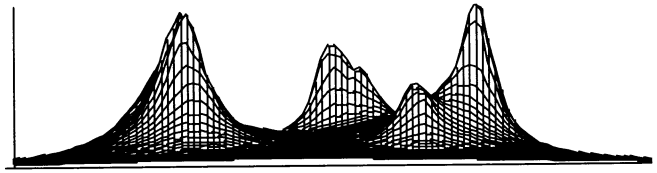
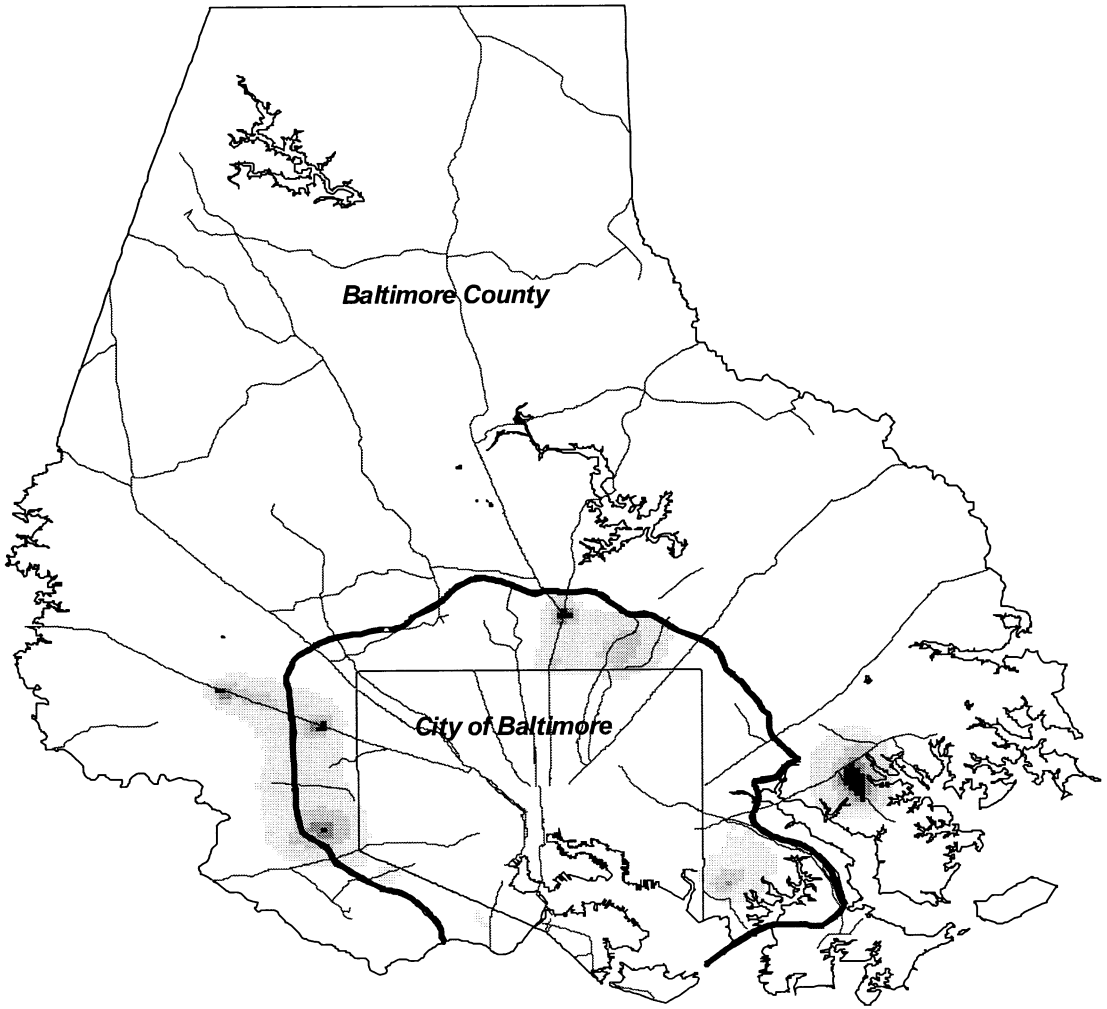
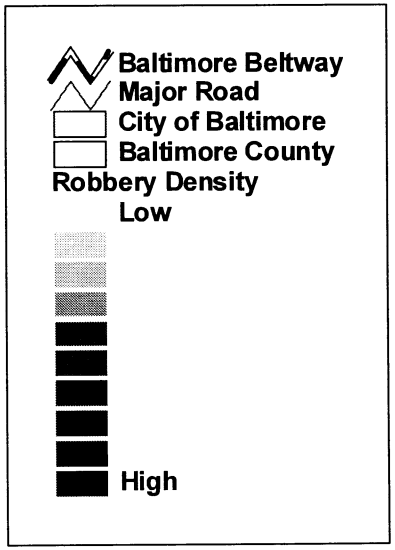


Figure 8.10
Baltimore County Street Robberies: 1996
Kernel Density Estimate





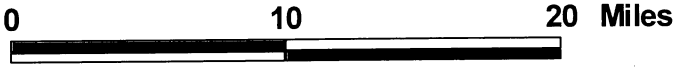
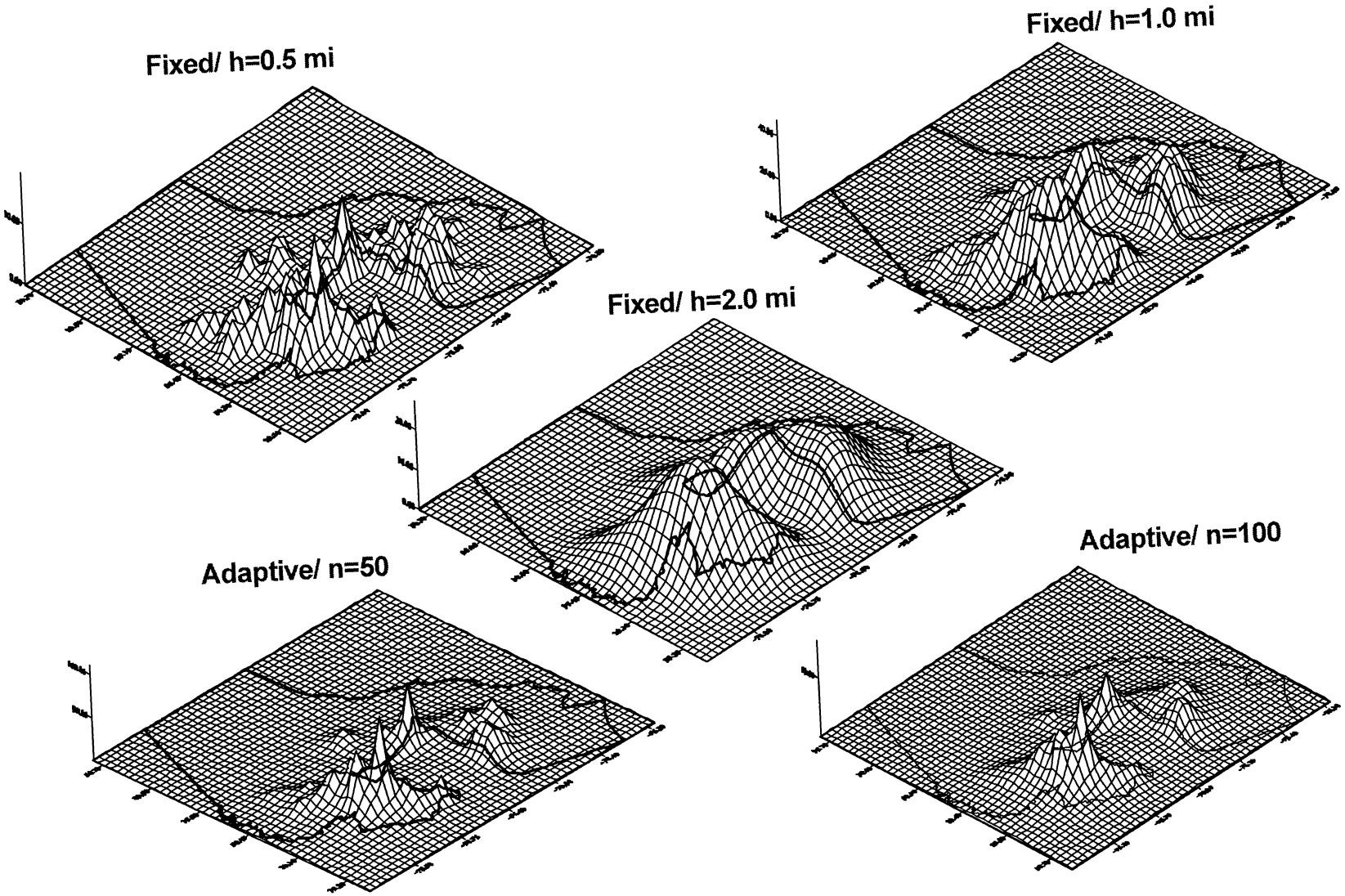


Figure 8.11:

Interpolation of Baltimore County Auto Thefts: 1996

Different Smoothing Parameters



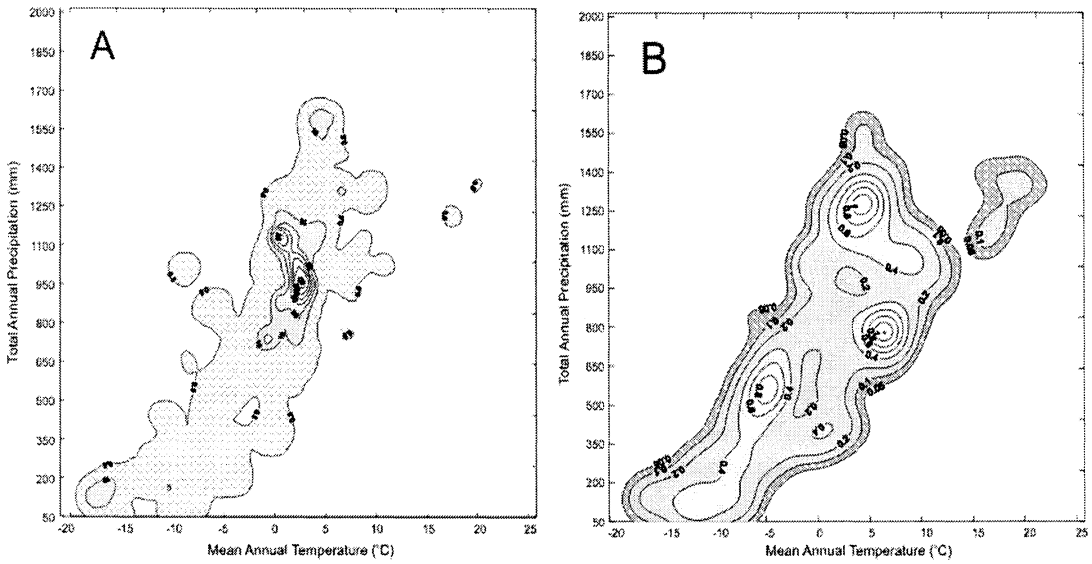
Kernel Density Interpolation to Estimate Sampling Bias in the Climatic Response of *Sphagnum* Spores in North America

Mike Sawada
Laboratory for Applied Geomatics and GIS Science
University of Ottawa, Department of Geography, Canada

Sphagnum moss, the dominant species of bogs, thrives under certain ranges of temperature and precipitation. *Sphagnum* releases spores for reproduction and these are transported, often long distances, by wind and water. Thus, the presence of a spore in the fossil record may not indicate nearby *Sphagnum* plants. However, spores should be most numerous near *Sphagnum* plants. Over time, these spores and pollen from other plants accumulate in lake and bog sediments and leave a fossil record of vegetation history.

We wanted to use the amount of fossil *Sphagnum* spores in different parts of North America to infer past climates. To do so, we had to first show that *Sphagnum* spores are most abundant in climates where *Sphagnum* plants thrive and secondly, that this center of abundance is not biased sampling because of under sampling in parts of climate space. First, we developed a *Sphagnum* spore response surface showing the relative abundance of spores along the axes of temperature and precipitation (Fig. A).

CrimeStat was used in the second stage to develop a kernel density surface using a quartic kernel for 3007 sample sites within climate space (Fig. B). These were smoothed and visualized in *Surfer*. The surface showed that the intensity of points is higher in regions surrounding the response maximum. This gave us confidence that the *Sphagnum* response was real since other parts of climate space are well sampled but unlikely to produce high spore proportions. This fact allowed climate inferences to be made within the fossil record for past time periods using the amount of *Sphagnum* spores present.



Figures modified from Gajewski, Viau, Sawada et al. 2001. Global Biogeochemical Cycles,

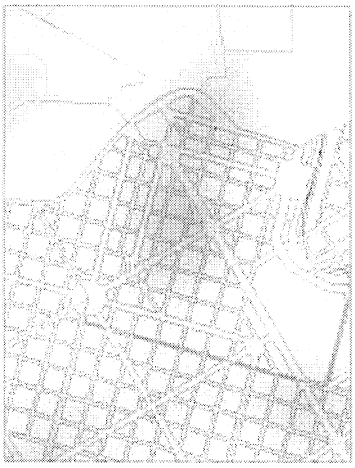
Describing Crime Spatial Patterns By Time of Day

Renato Assunção, Cláudio Beato, Bráulio Silva
CRISP, Universidade Federal de Minas Gerais, Brazil

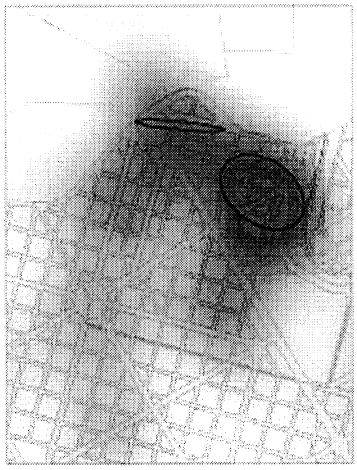
We used the kernel density estimate to visualize time trends for crime occurrences on a typical weekday. We found markedly different spatial distributions depending on the time, with the amount of crime varying and the hot spots, identified by the ellipses, appearing in different places.

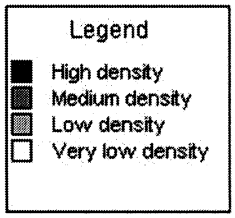
The analysis used 1114 weekday robberies from 1995 to 2000 in downtown Belo Horizonte. Breaking the data into hours, we used the normal kernel, a fixed bandwidth of 450 meters and outputted densities option (points per square unit of area). Note that the latter option could be useful if one is interested only in the hot spot locations, and not in the distribution during the day. To make the ellipses, we used the nearest neighbor hierarchical spatial clustering technique with a minimum of 35 incidents. We output the results to *MapInfo*, keeping the same scale for all maps. Four of them are shown below.

9:00 AM

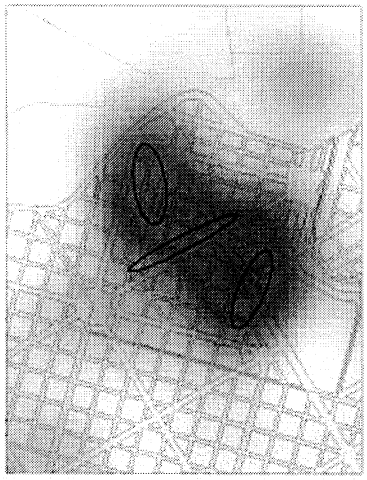


1:00 PM

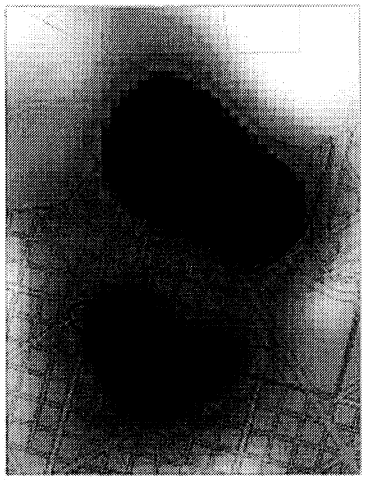




7:00 PM



11:00 PM



Using Kernel Density Smoothing and Linking to *ArcView*: Examples from London, England

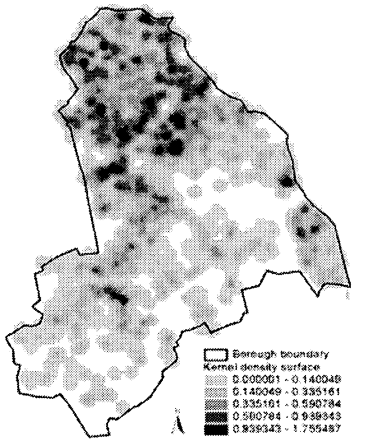
Spencer Chainey
Jill Dando Institute of Crime Science
University College
London, England

CrimeStat offers an effective method for creating kernel density surfaces. The example below uses residential burglary incidents in the London Borough of Croydon, England for the period June 1999 – May 2000 (N=3104). The single kernel routine was used to produce a kernel density surface representing the distribution of residential burglary.

The kernel function used was the quartic, which is favoured by most crime mappers as it applies added weight to crimes closer to the centre of the bandwidth. Rather than choosing an arbitary interval it is useful to use the mean nearest neighbour distance for different orders of K, which can be calculated by *CrimeStat* as part of a nearest neighbour analysis. For the Croydon data, an interval of 269 metres was chosen, which relates to a mean nearest neighbour distance at a K-order of 13. The output units were densities in square kilometres and was output to *ArcView*.

Kernel density estimation is a particularly useful method as it helps to precisely identify the location, spatial extent and intensity of crime hotspots. It is also visually attractive, so helping to invoke further enquiry and the reasoning behind why crime and disorder is concentrated. The density surface that is created can reflect the distribution of incidents against the natural geography of the area of interest, including representing the natural boundaries, such as reservoirs and lakes, or an alignment that follows a particular street in which there is a high concentration of offending. The method is also less subjective if clear guidelines are followed for the setting of parameters.

Residential burglary hotspots (by volume)
in the London Borough of Croydon, England.

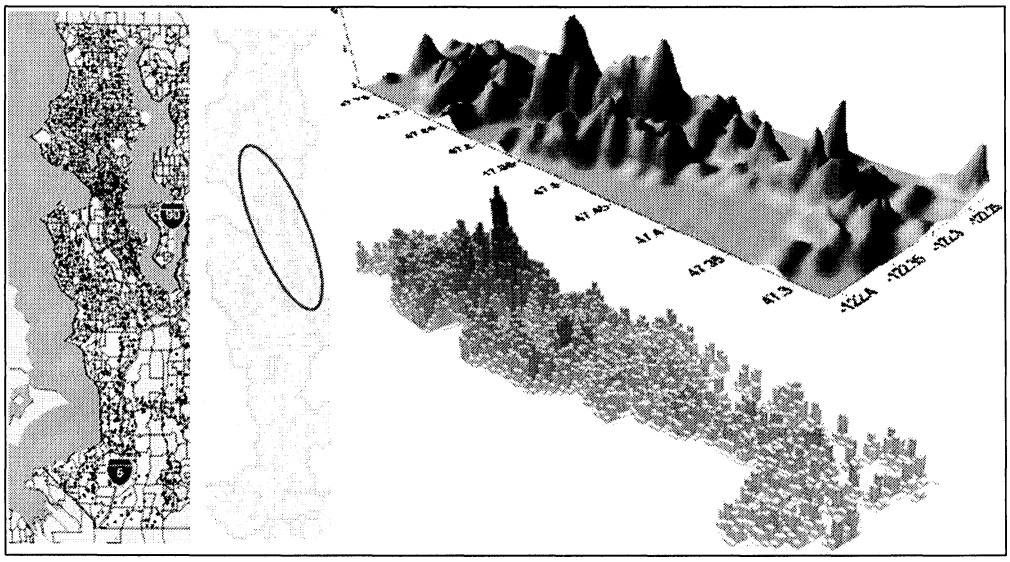


Infant Death Rate and Low Birth Weight in the I-5 Corridor of Seattle and King County

Richard Hoskins
Washington State Department of Health
Olympia, Washington

Although the infant death rate (< 1 year old) has been steadily declining in Washington, the incidence of low birth weight (< 2500 gms) is increasing. This is a significant public health problem, resulting in suffering and high medical cost. If we know where the rates are high at a neighborhood level we can develop more efficient and effective programs. The goal is to determine regions where rates are clustered and to characterize those regions with respect to SES variables from the US Census.

Birth and infant death data were geocoded to the street level. In order to detect clusters of high infant death ***and*** low birth weight, several *CrimeStat* tools were used. We find that using several tools at once helps detect regions where something untoward is going on and also helps develops guesses about where other problems might be expected develop.



I-5 corridor in King County | **Kernel density interpolation** | **Top: 3-D map: empirical Bayes rate Bottom: Prism map: SMR**

The result of a kernel density interpolation using a normal estimator is shown above along with an empirical Bayes rate and standardized mortality ratio (SMR) calculated in SAS and mapped in Maptitude (www.caliper.com). Starting with over 2,500 infant deaths, about 25,000 low weight births (out of over 500,000 live births) occurred in the Seattle I-5 corridor region in King County from 1989-2002. The kernel density method was used to detect high rate regions. A clearly articulated region and ridge appears on the grid of the kernel density map and the 3D and prism maps.

the bandwidth choice. For the three fixed intervals, an interval of 0.5 miles produces a finer mesh interpolation than an interval of 2 miles, which tends to 'oversmooth' the distribution. Perhaps, the intermediate interval of 1 mile gives the best balance between fineness and generality. For the two adaptive intervals, the minimum sample size of 25 gives some very specific peak locations whereas the adaptive interval with a minimum sample size of 100 gives a smoother distribution.

Which of these should be used as the *best* choice would depend on how much confidence the analyst has in the results. A key question is whether the 'peaks' are real or merely byproducts of small sample sizes. The best choice would be to produce an interpolation that fits the experience of the department and officers who travel an area. Again, experimentation and discussions with beat officers will be necessary to establish which bandwidth choice should be used in future interpolations.

Note in all five of the interpolations, there is some bias at the edges with the City of Baltimore (the three-sided area in the central southern part of the map). Since the primary file only included incidents for the County, the interpolation nevertheless has estimated some likelihood at the edges; these are *edge biases* and need to be ignored or removed with an ASCII editor.[3] Further, the wider the interval chosen, the more bias is produced at the edge.

Dual Kernel Estimates

The dual kernel density routine in *CrimeStat* is applied to *two* distributions of point locations. For example, the primary file could be the location of auto thefts while the secondary file could be the centroids of census tracts, with the population of the census tract being an intensity variable. The dual routine must be used with *both* a primary file *and* a secondary file. Also, it is necessary to define a reference file, either an existing file or one generated by *CrimeStat* (see chapter 3). Several parameters need to be defined.

File to be Interpolated

The user must indicate the order of the interpolation. The routine uses the language *first* file and *second* file in making the comparison (e.g., dividing the first file by the second; adding the first file to the second). The user must indicate which is the first file, the primary or the secondary. The default is that the primary file is the first file.

Method of Interpolation

The user must indicate the type of kernel estimator. As with the single kernel density routine, five types of kernel density estimators are used

1. Normal distribution (bell; default)
2. Uniform (flat) distribution
3. Quartic (spherical) distribution

4. Triangular (conical) distribution
5. Negative exponential (peaked) distribution

In our experience, there are advantages to each. The normal distribution produces an estimate over the entire region whereas the other four produce estimates only for the circumscribed bandwidth radius. If the distribution of points is sparse towards the outer parts of the region, then the four circumscribed functions will not produce estimates for those areas, whereas the normal will. Conversely, the normal distribution can cause some edge effects to occur (e.g., spikes at the edge of the reference grid), particularly if there are many points near one of the boundaries of the study area. The four circumscribed functions will produce less of a problem at the edges, although they still can produce some spikes. Within the four circumscribed functions, the uniform and quartic tend to smooth the data more whereas the triangular and negative exponential tend to emphasize 'peaks' and 'valleys'. The differences between these different kernel functions are small, however. The user should probably start with the default normal function and adjust accordingly to how the surface or contour looks.

Choice of Bandwidth

The user must define the bandwidth parameter. There are three types of bandwidths for the single kernel density routine - fixed interval, variable interval, or adaptive interval.

Fixed interval

With a fixed bandwidth, the user must specify the interval to be used and the units of measurement (squared miles, squared nautical miles, squared feet, squared kilometers, or squared meters). Depending on the type of kernel estimate used, this interval has a slightly different meaning. For the normal kernel function, the bandwidth is the standard deviation of the normal distribution. For the uniform, quartic, triangular, or negative exponential kernels, the bandwidth is the radius of the search area to be interpolated. Since there are two files being compared, the fixed interval is applied both to the first file and the second file.

Variable interval

With a variable interval, each file (the first and the second) have different intervals. For both, the units of measurements must be specified (squared miles, squared nautical miles, squared feet, squared kilometers, or squared meters). There is a good reason why a user might want variable intervals. In comparing two kernel estimates, the most common comparison is to divide one by the other. However, if the density estimate for a particular cell in the denominator approaches zero, then the ratio will blow up and become a very large number. Visually, this will be seen as spikes in the distribution, the result, usually, of too few cases. In this case, the user might decide to smooth the denominator more than numerator in order to reduce these spikes. For example, the interval for the first file (the numerator) could be 1 mile whereas the interval for the second file (the denominator) could

be 3 miles. Experimentation will be necessary to see whether this is warranted. But, in our experience, it frequently happens when either there are two few cases or there is an irregular boundary to the region with a number of incidents grouped at one of the edges.

Adaptive interval

An adaptive bandwidth adjusts the bandwidth interval so that a minimum number of points (sample size) is found. This sample size is applied to both the first file and the second file. It has the advantage of providing constant precision of the kernel estimate over the entire region. Thus, in areas that have a high concentration of points, the bandwidth is narrow whereas in areas where the concentration of points is more sparse, the bandwidth will be larger. This is the default bandwidth choice in *CrimeStat* since consistency in statistical precision is important. The degree of precision is generally dependent on the sample size of the bandwidth interval. The default is a minimum of 100 points. The user can make the estimate finer by choosing a smaller number of points (e.g., 25) or smoother by choosing a larger number of points (e.g., 200).

Use kernel bandwidths that produce stable estimates

Note: with a duel kernel calculation, particularly the ratio of one variable to another, be careful about choosing a very small bandwidth. This could have the effect of creating spikes at the edges of the study area or in low population density areas. For example, in low population density areas, there will probably be fewer events than in more built-up area. For the denominator of a ratio estimate, an extremely low value could cause the ratio to be exaggerated (a 'spike') relative to neighboring grid cells. Using a larger bandwidth will produce a more stable average.

Output Units

The user must indicate the measurement units for the density estimate in points per squared miles, squared nautical miles, squared feet, squared kilometers, or squared meters.

Intensity or Weighting Variables

If an intensity or weighting variable is used for either the first file or the second file, these boxes must be checked. Be careful about using both an intensity and a weighting variable to avoid 'double weighting'.

Density Calculations

The user must indicate the type of density output. There are six types of density calculations that can be conducted with the dual kernel density routine. The calculations are applied to each reference cell:

1. There is the *ratio of densities*, that is the first file divided by the second file. This is the default choice. For example, if the first file is the location of auto thefts incidents and the second file is the location of census tract centroids with the population assigned as an intensity variable, then ratio of densities would divide the kernel estimate for auto thefts by the kernel estimate for population and would be an estimate of auto thefts risk.

2. There is also the *log ratio of densities*. This is the natural logarithm of the density ratio, that is

$$\text{Log ratio of densities} = \text{Ln}\left[g(x_j) / g(y_j) \right] \quad (8.10)$$

where $g(x_j)$ is the density estimate for the first file and $g(y_j)$ is the density estimate for the second file. For a variable that has a spatially skewed distribution, such that most reference cells have very low density estimates, but a few have very high density estimates, converting the ratio into a log function will tend to mute the spikes that occur. This measure has been used in studies of risk (Kelsall and Diggle, 1995b).

3. There is the *absolute difference in densities*, that is the first file minus the second file. This can be a useful output for examining differential effects. For example, by using the centroids of census block groups (see example 2 below) with the population of the census block group assigned as an intensity or weighting variable, there is a slight bias produced by the spatial arrangements of the block groups. The U. S. Census Bureau suggests that census units (e.g., census tracts, census block groups) be drawn so that there are approximately equal populations in each unit. Thus, block groups towards the center of the metropolitan area tend to be smaller because there is a higher population density at those locations. Thus, the spatial arrangement of the block groups will tend to produce a kernel estimate which has a higher value towards the center independent of the actual population of the block group; the bias is very small, less than 0.1%, but it does exist. A more precise estimate could be produced by subtracting the kernel estimate for the block group centroids *without* using population as the intensity variable from the kernel estimate for the block group centroids *with* population as the intensity variable. The resulting output could then be read back into *CrimeStat* and used as a more precise measure of population distribution. There are other uses of the difference function, such as subtracting the estimate for the population-at-risk from the incident distribution rather than taking the ratio or by calculating the net change in population between two censuses.

4. There is the *relative difference in densities*. Like the relative density in the single-kernel routine (discussed above), the relative difference in densities first standardizes the densities of each file by dividing by the grid cell area and then subtracts the secondary file relative density from the primary file

relative density. This can be useful in calculating changes between two time periods, for example in calculating a change in relative density between two censuses or a change in the crime density between two time periods.

5. There is the *sum of the densities*, that is, the density estimate for the first file plus the density estimate for the second file. Again, this is applied to each reference cell at a time. A possible use of the sum operation is to combine two different density surfaces, for example the density of robberies plus the density of assaults;

6. Finally, there is the *relative sum of densities* between the primary file and the secondary file. The relative sum of densities first standardizes the densities of each file by dividing by the grid cell area and then subtracts the secondary file relative density from the primary file relative density. This can be useful for identifying the total effects of two distributions. For example, the total impact of robberies and burglaries on an area can be estimated by taking the relative density of robberies and adding it to the relative density of burglaries. The result is the combined relative density of robberies and burglaries per unit area (e.g., robberies and burglaries per square mile).

Output Files

Finally, the user must specify the file formats for the output. The results can be output in three forms. First, the results are displayed in an output table. Second, the results can be output into two raster grid formats for display in a surface mapping program: *Surfer for Windows* format as a '.dat' file (Golden Software, 1994) and *ArcView Spatial Analyst* format as a '.asc' file (ESRI, 1998). Third, the results can be output as polygon grids into *ArcView* '.shp', *MapInfo* '.mif' and *Atlas*GIS* '.bna' format (see footnote 1). All but *Surfer for Windows* require that the reference grid be created by *CrimeStat*.

Example 2: Kernel Density Estimates of Vehicle Thefts Relative to Population

As an example of the use of the dual kernel density routine, the duel routine is applied in both the City of Baltimore and the County of Baltimore to 14,853 motor vehicle theft locations for 1996 relative to the 1990 population of census block groups. Again, a reference grid of 100 columns by 108 rows was generated by *CrimeStat*.

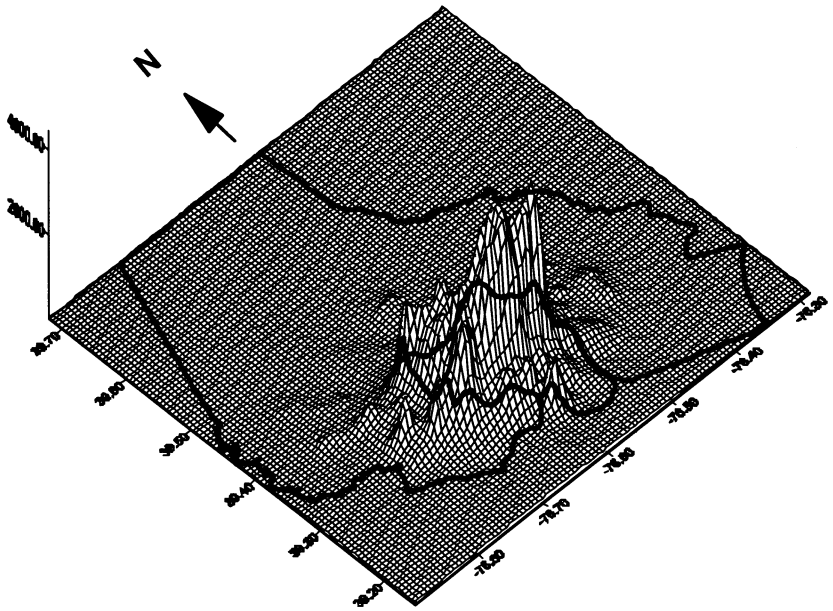
Figure 8.12 shows the resulting density estimate as a *Surfer for Windows* output; again, there is a map view, a surface view, and a contour view. The normal kernel function was used and an adaptive bandwidth of 100 points was selected. As seen, there is a very high concentration of auto theft incidents within the central part of the metropolitan area. The contour view suggest five or six peak areas that are close to each other.

Figure 8.12:

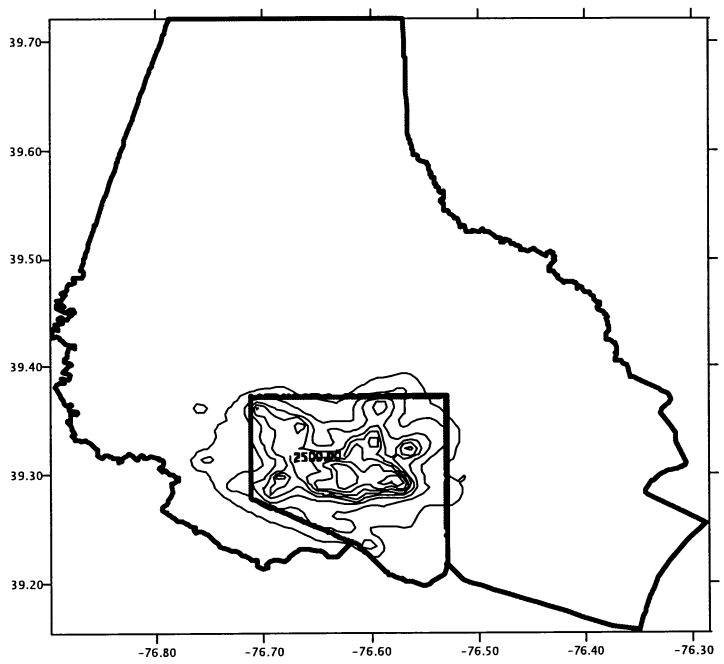
Baltimore County Vehicle Thefts: 1996

Kernel Density Interpolation

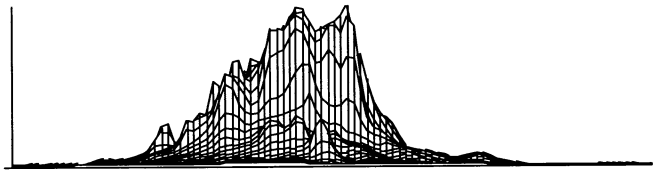
Surface View



Contour View



Ground Level View



Much of this concentration, however, is produced by high population density in the metropolitan center. Figure 8.13, for example, shows the kernel estimate for 1349 census block groups for both the City of Baltimore and the County of Baltimore with the 1990 population assigned as the intensity variable. Again, the normal kernel function was used with an adaptive bandwidth of 100 points being selected. The map shows three views: 1) a surface view; 2) a contour view; and 3) a ground level view looking directly north. The distribution of population is, of course, also highly concentrated in the metropolitan center with two peaks, quite close to each other with several smaller peaks.

When these two kernel estimates are compared using the dual kernel density routine, a more complicated picture emerges (figure 8.14). This routine has conducted three operations: 1) it calculated the distance between each of the 10,800 reference cells and the 14,853 auto theft locations, evaluated the kernel function for each measured distance, and summed the results for each reference cell; 2) it calculated the distance between each of the 10,800 reference cells and the 1349 census block groups with population as an intensity variable, evaluated the kernel function for each intensity-weighted distance, and summed the results for each reference cell; and 3) divided the kernel density estimate for auto thefts by the kernel density estimate for population for each reference cell location.

While the concentration of motor vehicle thefts relative to population ("motor vehicle theft risk") is still high in the metropolitan center, there are bands of high risk that spread outward, particularly along major arterials. There are now many 'hot spot' areas which have a high distribution of motor vehicle thefts relative to the residential population. We could, of course, refine this analysis further by taking, for example, employment as a baseline variable rather than population; employment is a better indicator for the daytime population distribution whereas the residential population is a better indicator for nighttime population distribution (Levine, Kim, and Nitz, 1995a; 1995b).

Example 3: Kernel Density Estimates and Risk-adjusted Clustering of Robberies Relative to Population

The final example shows how the duel kernel interpolation compares with the risk-adjusted nearest neighbor clustering, discussed in chapter 6. Figure 8.15 shows 7 first-order risk-adjusted clusters overlaid on the a duel kernel estimate of 1996 robberies relative to 1990 population.[4] As seen, there is a correspondence between the identified risk-adjusted clusters and the duel kernel interpolation of the ratio of robberies to population. For a broad regional perspective, the interpolation produces an adequate model of where there is a high robbery risk. At the neighborhood level, however, the risk-adjusted clusters are more specific and would be preferable for use by police in identifying high-risk locations.

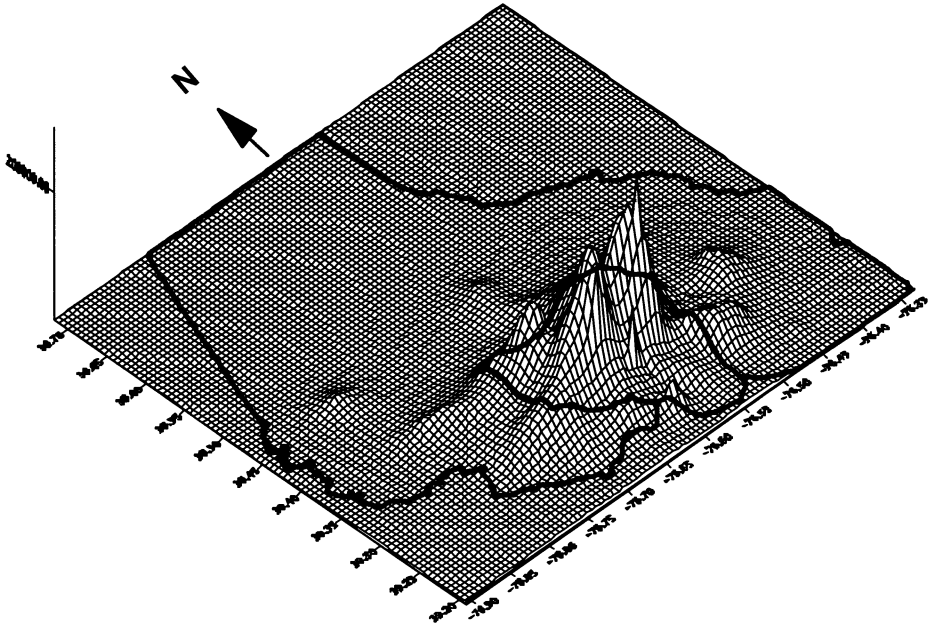
The advantage of a dual kernel density interpolation routine is that two variables can be related together. By interpolating one variable to a reference grid and then interpolating a second variable to the same reference grid, the two variables have been

Figure 8.13:

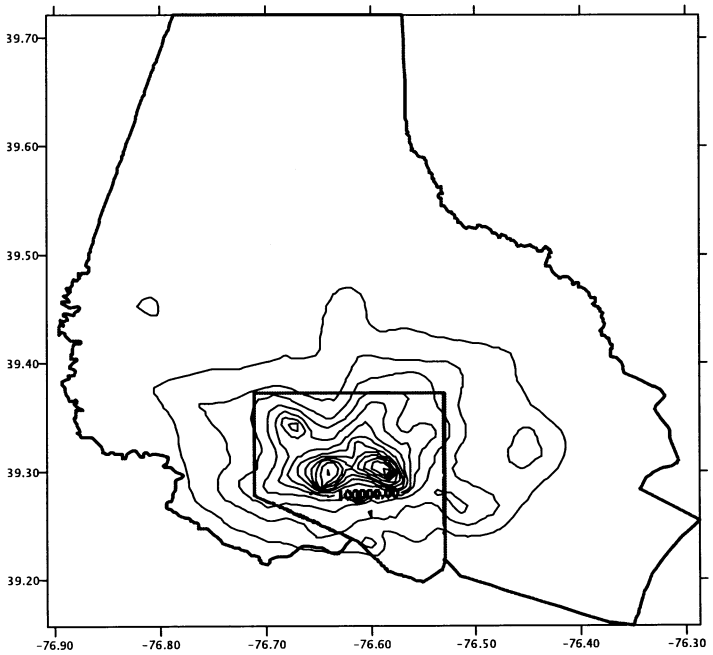
Baltimore Metropolitan Population: 1990

Kernel Density Estimate of Block Group Population

Surface View



Contour View



Ground Level View

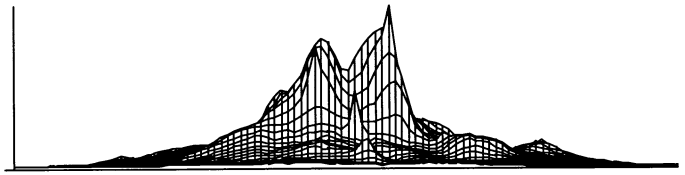
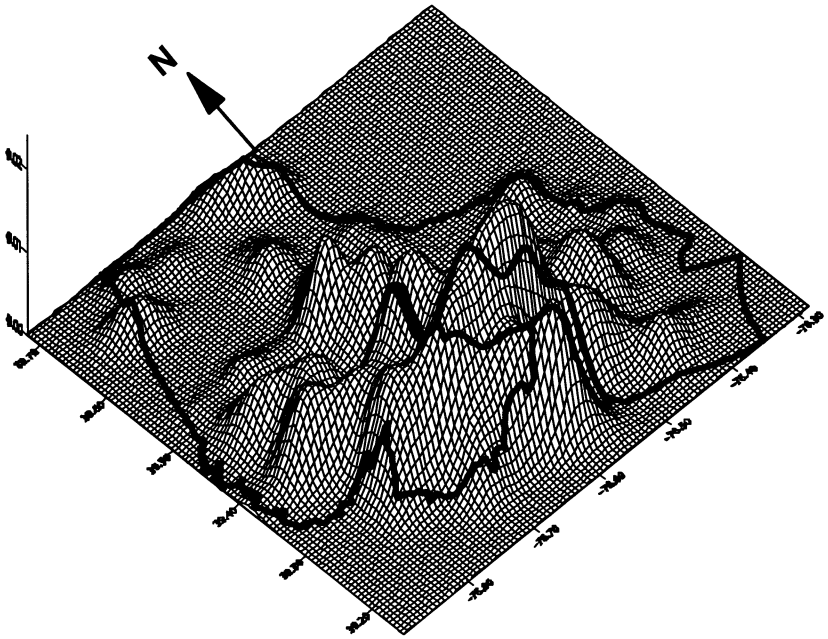


Figure 8.14:

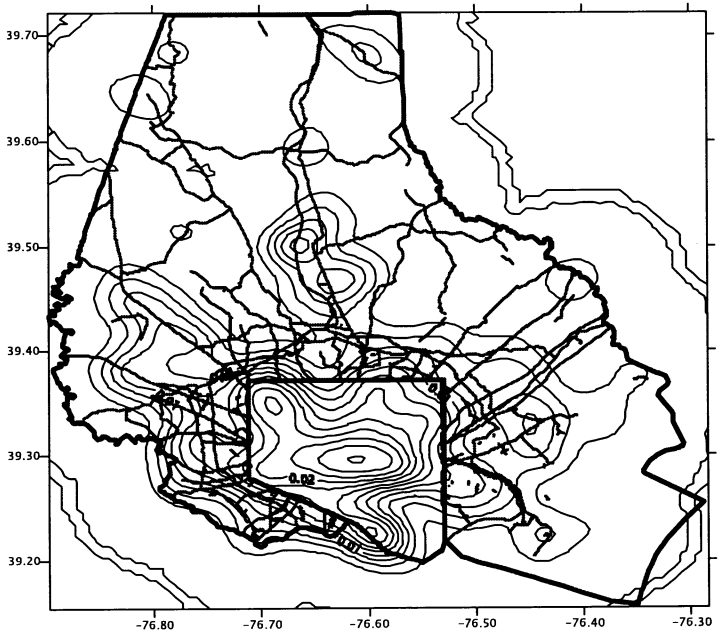
Baltimore County Vehicle Theft Risk

Kernel Density Ratio of 1996 Vehicle Thefts to 1990 Population

Surface View



Contour View



Ground Level View

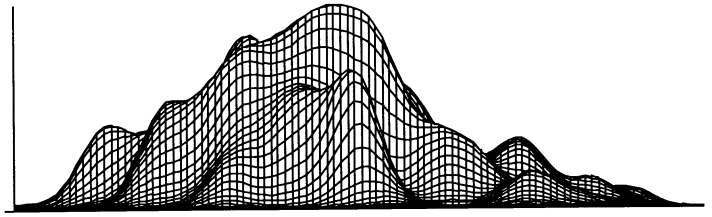
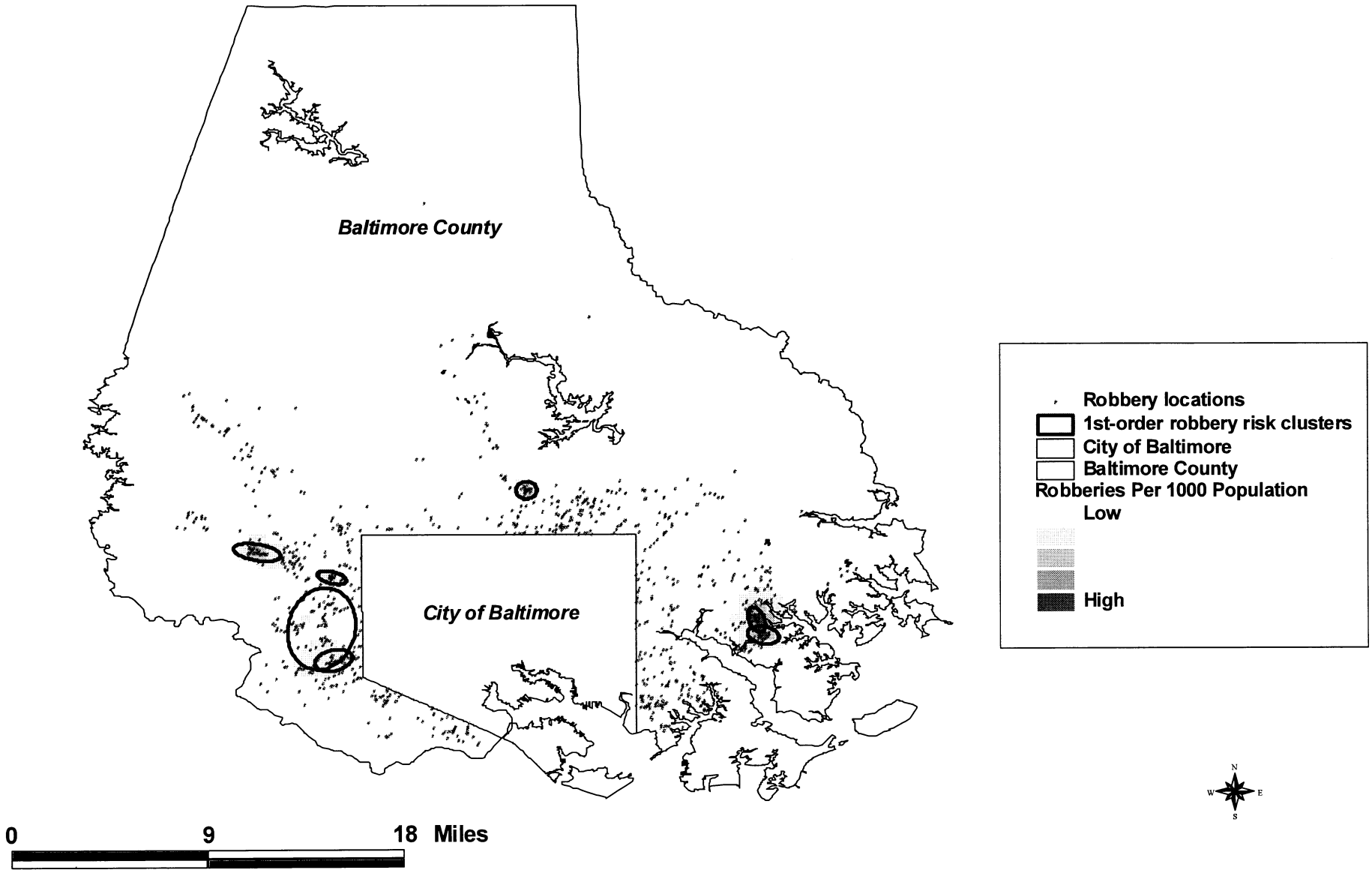


Figure 8.15:
Risk-adjusted Robbery Clusters and Interpolated Robbery Risk
1996 Robberies Relative to 1990 Population



Using Small Area Estimation to Target Health Services

Thomas F. Reynolds, MS
University of Texas-Houston School of Public Health

In Texas, the City of Houston and Harris County organized a Public Health Task force to make recommendations concerning the provision of health services for those without health insurance. Task force members wanted to know approximately how many area citizens did not have health insurance.

Data from the two most recent Current Population Survey Annual Social and Economic Supplements (CPS-ASEC, 2003-04) were used to derive a synthetic estimate using a stratified model. Estimates were calculated at census tract and block group levels. Selected political divisions were clipped from base maps for political officials and legislators.

Percentages are indicative of risk. On the other hand, numbers are essential for targeting physical resources. There is seldom a perfect correspondence between high percentages and large numbers. For example, an area with a concentration of multi-family housing may have a relatively small percentage, but a large number, of uninsured. Percentage maps of the uninsured (figure 1) are generally clustered and informative; however, due to large variations in population numbers at both levels of census geography, maps of the population densities of uninsured proved most valuable to officials (figure 2).

CrimeStat was used to develop the density maps. The single kernel density routine was used to estimate the density of block group values using the centroid to represent the values and the number of uninsured as an intensity value. The Moran Correlogram was used to select the type of kernel for the single-kernel interpolation (a uniform distribution) and an optimal bandwidth.

Fig. 1: Percent Uninsured

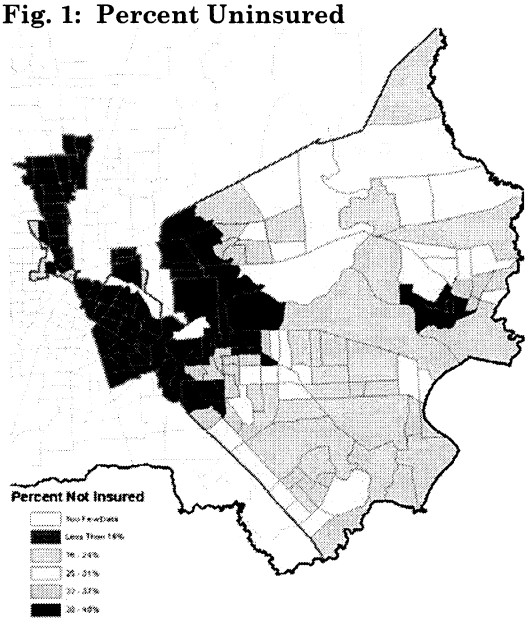
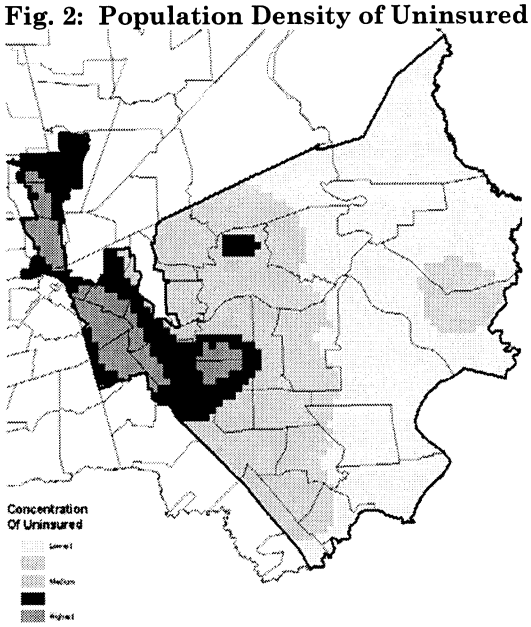


Fig. 2: Population Density of Uninsured



interpolated to the same geographical units. The two interpolations can then be related, by dividing, subtracting, or summing. As has been mentioned throughout this manual, one of the problems with techniques that depend on the concentration of incidents is that they ignore the underlying population-at-risk. With the dual routine, however, we can start to examine the risk and not just the concentration.

Visually Presenting Kernel Estimates

Whether the single- or duel-kernel estimate is used, the result is a grid interpretation of the data. By scaling these values by color in a GIS program, a visualization of the data is obtained. Areas with higher densities can be shown in darker tones and those with lower densities can be shown in lighter tones; some people do the opposite with the high density areas being lighter.

To make the visualization even more realistic, one could use a GIS program to cut out those grid cells that are outside the study area or are on water bodies. Before doing this, however, be sure to re-scale the estimated "Z" values so that they will sum to the total of the original grid. For example, if the original sample size was 1000, then the grid cells will sum to 1000 if the absolute density option is chosen. If, say, 20% of these cells are then removed to improve the visualization, then the grid cell Z values have to be re-scaled so that their sum will continue to be 1000. A simple way to do this is to, first, add up the Z values for the remaining cells and, second, multiply each grid cell Z by the ratio of the original sum to the reduced sum.

The visualization is useful for a broad, regional view. It is not particularly useful for micro analysis. The use of one of the cluster routines discussed in chapters 6 and 7 would be more appropriate for small area analysis.

Conclusion

Kernel density estimation is one of the 'modern' spatial statistical techniques. There is currently research on the use of this technique in both the statistical theory and in developing applications. For crime analysis, the technique represents a powerful way of conducting both 'hot spot' analysis as well as being able to link the 'hot spots' to an underlying population-at-risk. It can be used both for police deployment by targeting areas of high concentration of incidents as well as for prevention by targeting areas with high risk. It can also be used as a research tool for analyzing two or more distributions. More development of this approach can be expected in the next few years.

The Risk of Violent Incidents Relative to Population Density in Cologne Using the Dual Kernel Density Routine

Dietrich Oberwittler and Marc Wiesenhütter
Max Planck Institute for Foreign and International Criminal Law
Freiburg, Germany

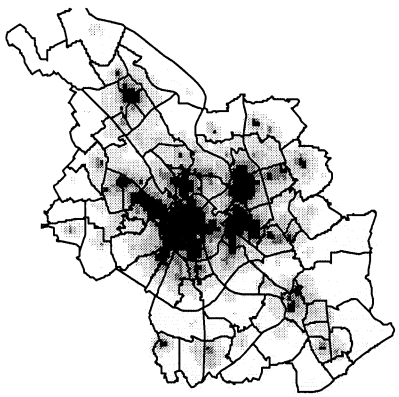
When estimating the density of street crimes within a metropolitan area by interpolating crime incidents, the result is usually a very high concentration in the city center. However, there is also a very high concentration of people either living or pursuing their daily routine activities in these areas. The question emerges how likely is a criminal event when taking into account the number of people spending their time in these areas. The *CrimeStat* duel kernel density routine is able to estimate a ratio density surface of crime relative to the 'population at risk'.

In this example, data on 'calls to the police' for assault and battery from April 1999 to March 2000 (N=6363 calls) and population from Cologne were used. Exact information on the number of people spending their time in the city does not exist. Therefore, 1997 counts of passengers entering and leaving the public transport system at each of 550 stations and bus stops in the city was used as a proxy variable. The number of persons at each station or bus stop was assigned to adjacent census tracts and added to the resident population resulting in a crude measure of the 'population at risk'.

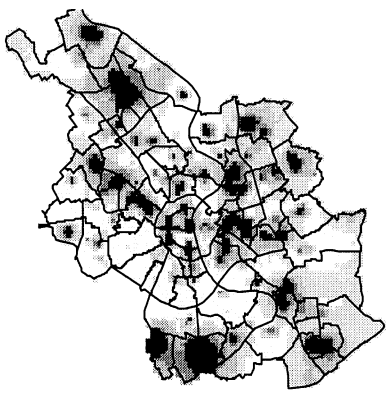
In the dual kernel routine, the density estimate of crime incidents is compared to the density estimate of the population at risk, defined by the centroids of census tracts with the number of persons as an intensity variable. We chose the normal method of interpolation and adaptive intervals with a minimum of five points. The adaptive bandwidth adjusts for the fact that there are fewer incidents and census tracts at the edges of the city, resulting in a relatively smoother density surface for the ratio. The results were output to *ArcView*.

The effect of adjusting the crime distribution for the underlying 'population at risk' becomes quite visible. Whereas the *concentration of crime* is highest in the city center (left map), the *crime risk* (right map) is in fact much higher in several more distant areas that are known for high concentrations of socially disadvantaged persons. Given the imperfect nature of the population data these results should be interpreted as a broad view on the distribution of crime risk that, nevertheless, has important policy implications.

Single kernel density of crime incidences
(assault & battery, Cologne 1999/2000)



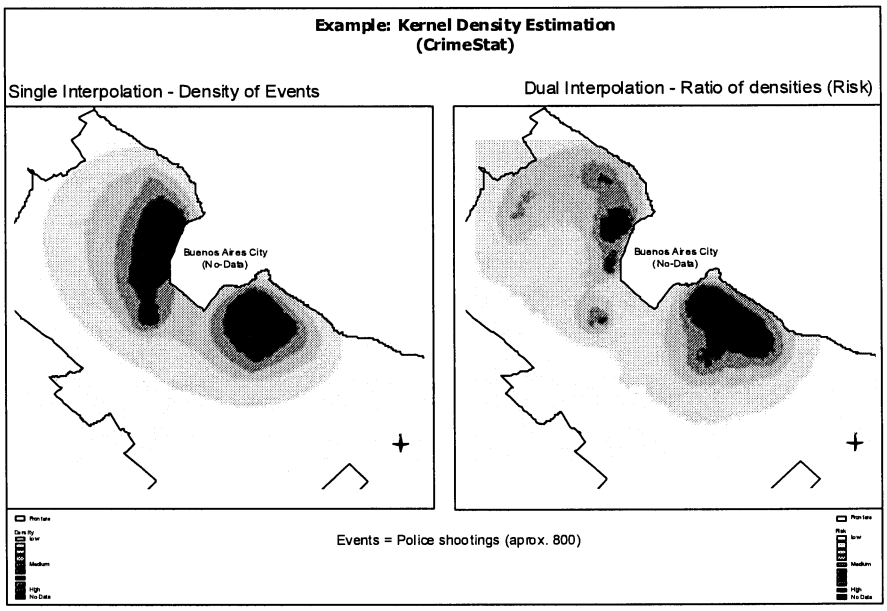
Dual kernel density of crime incidences
relative to population at risk



Kernel Density Interpolation of Police Confrontations in Buenos Aires Province, Argentina: 1999

Gastón Pezzuchi
Crime Analyst
Buenos Aires Province Police Force
Buenos Aires, Argentina

One of our first tryouts with the *CrimeStat* software involved the calculation of both single and dual kernel density interpolations using data on 1999 confrontations with the police within Buenos Aires Province, an area that covers 29 counties around the Federal Capital. The confrontations include mostly gun fights with the police but also other attacks (e.g., knives, rocks, sticks). In the last three years, there has been an increase in confrontations with the police. The single interpolation shows a density surface that gives a good picture of the ongoing level of violence while the duel interpolations shows a risk surface using the personnel deployment data; the latter are confrontations relative to the number of police deployed. Typically, police are allocated to areas according to crime rates.\



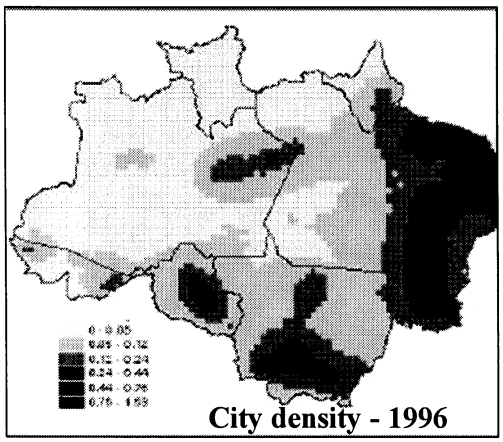
Both images are quite different, suggesting varying policing strategies. For example, though there are two well-defined hot spot areas in the Province (one in the north, the other in the south), the high levels of risk detected in the southern areas came as a complete surprise. The northern area has a higher crime rate than the southern area, hence a high police deployment. However, the level of confrontation are approximately equal between the two areas.

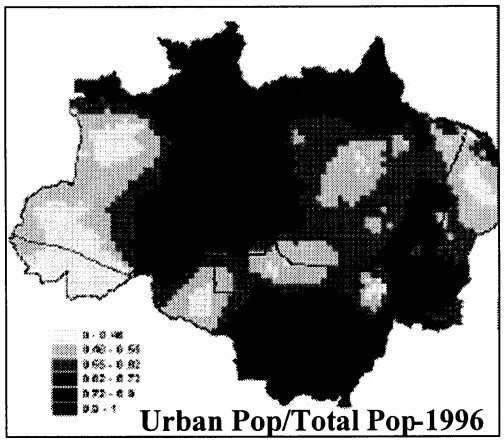
Evolution of the Urbanization Process in the Brazilian Amazonia

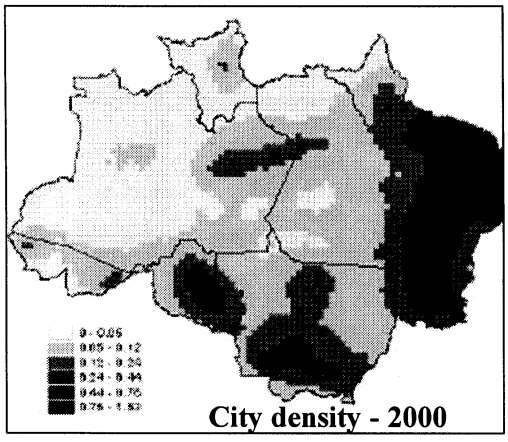
Silvana Amaral, Antônio Miguel V. Monteiro, Gilberto Câmara, José A. Quintanilha
INPE, Instituto Nacional de Pesquisas Espaciais, Brazil

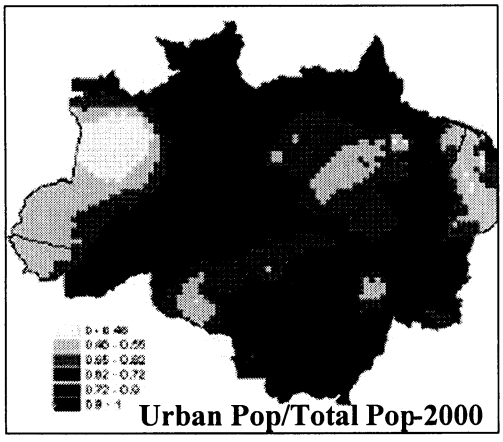
The Brazilian Amazon rain forest is the world's largest contiguous area of tropical rain forest in the world. During the last three decades, the region has experienced the largest urban growth rates in Brazil, a process that has reorganized the network of human settlements in the region. We used the *CrimeStat* single and duel kernel density routines to visualize trends in urbanization from 1996 to 2000 in Amazonia. Two variables were used to measure urbanization: 1) the concentration of urban nuclei (city density); and 2) the ratio of urban to total population.

The concentration of cities was spatially associated withfederal roads in the eastern and southern portions, and along the Amazonas River in the middle of the region. Additionally, the surfaces of urban population show that city density is not always associated with large urban populations. From 1996 to 2000 city density increased in the western Amazonia (Pará state) at a greater rate than the growth of the urban population. In the southeastern part of the region (Rondônia state), there were many urban centers. But the ratio of urban to total population was small, indicating that they are predominately agricultural regions.


City density - 1996


Urban Pop/Total Pop-1996


City density - 2000


Urban Pop/Total Pop-2000

Endnotes to Chapter 8

1. There are differences in opinion about how wide a particular fixed bandwidth should be determined. The smoothing is done for a distribution of values, Z. If there are only unique points (and, hence, there is no Z value at a point), the distances between points can be substituted for Z. Thus, MeanD is the mean distance, sd(D) is the standard deviation of distance, and iqr(D) is the inter-quartile range of distances between points. These would be substituted for MeanZ, sd(Z), and iqr(Z) respectively

   Silverman (1986; 45-47; Härdle, 1991; Farewell, 1999) proposed a bandwidth, h, of:

$$h = 1.06 * \min\left\{ sd(Z), \frac{iqr(Z)}{1.34} \right\} * N^{-1/5}$$

   where *min* is the minimum of the next two terms, *sd(Z)* is the standard deviation of the variable, Z, being interpolated, *iqr(Z)* is the inter-quartile range of Z, and N is the sample size.

   Bowman and Azzalini (1997; 31) defined a slightly different optimal bandwidth for a normal kernel.

$$h = \left\{ \frac{4}{3N} \right\}^{1/5} * sd(Z)$$

   To avoid being influenced by outlier, they suggested using the median absolute deviation estimator for sd(Z)

$$MAD(Z) = \text{median}\left\{ \frac{Z(i) - \text{Median}Z}{0.6745} \right\}$$

   Scott (1992) suggested an upper bound on the normal kernel of

$$h = 1.144 * sd(Z) * N^{-1/5}$$

   Bailey and Gatrell (1995, 85-87) offered a rough choice for the bandwidth of

$$h = 0.68 * N^{-0.2}$$

   but suggested that the user could experiment with different bandwidths to explore the surface.

   On the other hand, the concept of an adaptive bandwidth is based more on sampling

theory (Bailey and Gatrell, 1995). By increasing the bandwidth until a fixed number of points are counted ensures that the level of precision is constant throughout the region. As with all sampling, the standard error of the estimate is a function of the sample size; a larger sample leads to smaller error. In general, if there was independent sampling, the 95% confidence interval of a bandwidth for a normal kernel could be approximated by

$$95\%\ \text{C.I.} = \text{Mean}(Z) \pm 1.96 * \frac{.5}{N(h)^{1/2}} * \text{sd}(Z)$$

where N(h) is the adaptive sample size (the number of points counted within the bandwidth for the adaptive kernel). This assumes that a point has an equal likelihood of falling within the bandwidth of one cell compared to an adjacent cell (i.e., it sits on the boundary of the bandwidth circle). The adaptive bandwidth criteria requires that the bandwidth be increased until it captures the specified number of points. On average, if there are N points in a region of area, A, and if the adaptive sample size is N(p), then the average area required to capture N(p) points is

$$A(p) = \frac{N(p) * A}{N}$$

and the average bandwidth, Mean(h), is

$$\text{Mean}(h) = \text{SQRT}\left[\frac{A(p)}{\pi}\right] = \text{SQRT}\left[\frac{N(p) * A}{N*\pi}\right]$$

Each of these provide different criteria for the bandwidth size with the adaptive being the most conservative. For example, for a standardized distribution with 1000 data points, a standardized mean of Z of 0 and a standardized standard deviation of 1, the Silverman criteria would produce a bandwidth of 0.2663; the Bowman and Azzalini criteria would produce a bandwidth of 0.2661; the Scott criteria would produce a bandwidth of 0.2874 and the Bailey and Gatrell criteria would produce a bandwidth of 0.1708. For the adaptive interval, if the required adaptive sample size is 25, then the average bandwidth would be approximately 0.3162 (this assumes that the area is a circle with a radius of 2 standardized standard deviations).

2. *CrimeStat* will output the geographical boundaries of the reference grid (a polygon grid) and will assign a third-variable (called *Z*) as the density estimate. Of the three polygon grid outputs, *ArcView* '.shp' files can be read directly into the program. For *MapInfo*, on the other hand, the output is in MapInfo Interchange Format (a '.mif' and a '.mid' file); the density estimate (also called *Z*) is assigned to

the '.mid' file. The files must be imported to convert it to a *MapInfo* '.tab' file. For *Atlas*GIS* '.bna' format, however, there are two files that are output - a '.bna' file which includes the boundaries of the polygon grid and a '.dbf' file which includes the grid cell names (called *gridcell*) and the density estimate (also called *Z*). The '.bna' file must be read in first and then the '.dbf' file must be read in and matched to the value of *gridcell*. For all three output formats, the values of Z can be shown as a thematic map but the ranges must be adjusted to illustrate the likely locations for the offender's residence (i.e., the default values in the GIS programs will not display the densities very well). On the other hand, the default interval values for *Surfer for Windows* and *ArcView Spatial Analyst* provide a reasonably good visualization of the densities.

3. All the *CrimeStat* outputs except for *ArcView* 'shp' files are in ASCII. There are usually 'edge effects' and values interpreted outside the actual geographical area. These can be removed with an ASCII editor by substituting '0' for the values at the edges or outside the study region. For 'shp' files, the values at the edges can be edited within the *ArcView* program. Another alternative is to 'cut out' the cells that are beyond the study area. Care must be taken, however, to not edit an output file too much otherwise it will bear little relationship to the calculated kernel estimate.

4. The risk-adjusted hierarchical clustering (Rnnh) method defined the largest search radius but a minimum of 25 points being required to be clustered. The kernel estimate for both the Rnnh and the duel-kernel routines used the normal distribution function with an adaptive bandwidth of 25 points.

Chapter 9
Space-Time Analysis

In this chapter, we discuss three techniques that are used to analyze the relationship between space and time. Up to this point, we have analyzed the distribution of incidents irrespective of the order in which they appeared or in which the time frame in which they appeared. The only temporal analysis that was conducted was in Chapter 4 where several spatial description indices, including the standard deviational ellipse, were compared for different time periods.

As police departments usually know, however, the spatial patterning of incidents doesn't occur uniformly throughout the year, but instead are often clustered together during short time periods. At certain times, a rash of incidents will occur in certain neighborhoods and the police often have to respond quickly to these events. In other words, there is both clustering in time as well clustering in space. This area of research has been developed mostly in the field of epidemiology (Knox, 1963, 1988; Mantel, 1967; Mantel and Bailar, 1970; Besag and Newell, 1991; Kulldorf and Nargawalla, 1995; Bailey and Gattrell, 1995). However, most of these techniques are applicable to crime analysis and criminal justice research as well.

CrimeStat includes four space-time techniques: the Knox index, the Mantel index, the Spatial-temporal moving average, and Correlated Walk Analysis. Figure 9.1 shows the Space-Time Analysis screen.

Measurement of Time in *CrimeStat*

Time can be defined as hours, days, weeks, months, or years. The default is days. However, please note that for any of these techniques, in *CrimeStat*, time must be measured as an *integer* or *real* variable, as mentioned in Chapter 3. Time cannot be defined by a formatted date code (e.g., 11/06/01, July 30, 2002). Each of the three space-time routines expect time to be an integer or real variable (e.g., 1, 2, 34527, 2.8). If given formatted dates, they will calculate an answer, but the result will not be correct.

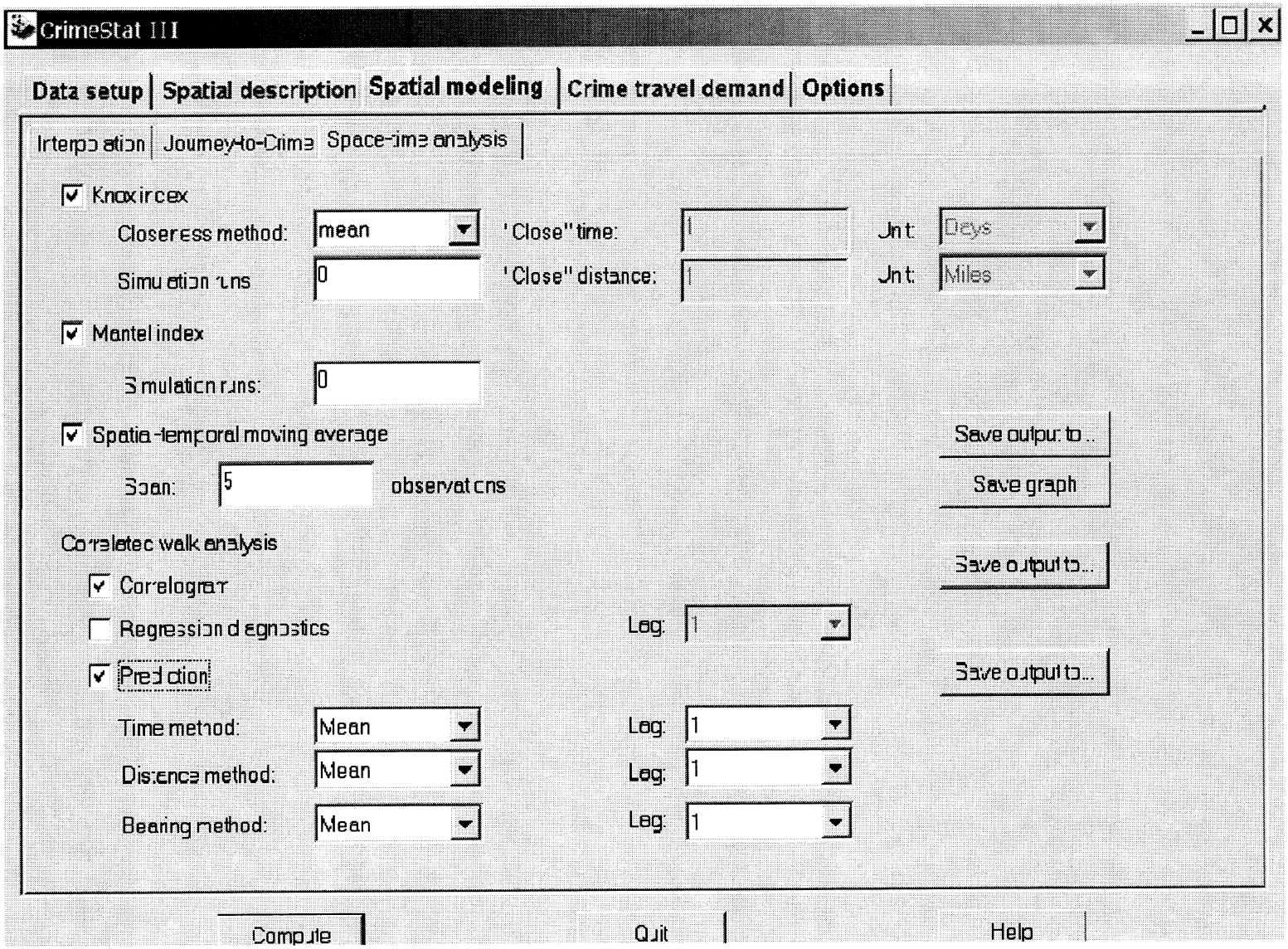
If the time unit is days, a simple transformation is to use the number of days since January 1, 1900. Most spreadsheet and data base programs usually assign an integer number from this reference point. For example, November 12, 2001 has the integer value of 37207 while January 30, 2002 has the integer value of 37286. These are the number of days since January 1, 1900. Any spreadsheet program (e.g., Excel or Lotus 1-2-3) can convert a date format into a real number with the Value function. Also, any arbitrary numbering system will work (e.g., 1, 2, 3).

Space-Time Interaction

There are different types of interaction that could occur between space and time. Four distinctions can be made. First, there could be *spatial clustering* all the time.

Figure 9.1:

Space-Time Analysis Screen



Certain communities are prone to certain events. For example, robberies often are concentrated in particular locations as are vehicle thefts. The hot spot methods that were discussed in chapters 6, 7 and 8 are useful for identifying these concentrations. In this case, there is no space-time interaction since the clustering occurs all the time.

Second, there could be *spatial clustering within a specific time period*. Hot spots could occur during certain time periods. For example, motor vehicle crashes tend to occur with much higher frequencies in the late afternoon and early evening, often as a by-product of congestion on the roads. Crash hot spots will tend to appear at certain times because of the congestion. At most other times, the concentration does not occur because the congestion levels are lower.

Third, there could be *space-time clustering*. A number of events could occur within a short time period within a concentrated area. This type of effect is very common with motor vehicle thefts. A car thief gang may decide to attack a particular neighborhood. After a binge of car thefts, they move on to another neighborhood. In this instance, there are a number of theft incidents that are occurring within a limited period in a limited location. The cluster moves from one location to another. In this case, there is an interaction between space and time in that spatial hot spots appear at particular times, but are temporary. The ability to detect this type of shift is very important to police departments since it affects their ability to respond.

Fourth, there could be *space-time interaction* in which the relationship between space and time in more complex. The interaction could be concentrated, as in the spatial clustering mentioned above, or it could follow a more complex pattern. For example, there could be a diffusion of drug sales from a central location to a more dispersed area. Whereas initially, the drug dealing is concentrated in a few locations, it starts to diffuse to other areas. However, the diffusion may occur at different times of the year (e.g., Christmas and New Years). Alternatively, vehicle thefts may shift towards seaside communities during the summer months when the number of vacationers increases. We saw an example of this in chapter 4 where the ellipse of motor vehicle thefts shifted between June and July to the communities along the Chesapeake River near Baltimore. This type of diffusion is not clustering *per se*, in that it may be spread over a very large coastline. But it is a distinct space-time interaction.

The importance of these distinctions is that many of the space-time tests that exist only measure gross space-time interaction, rather than space-time clustering. For example, the Knox and Mantel tests that follow test for spatial interaction. The interaction could be the result of spatial clustering, but doesn't necessarily have to be. The interaction could occur in a very complex way that would not easily lend itself to more focused intervention by the police. Still, the ability to identify the interaction is an important step in planning an intervention strategy.

Knox Index

The Knox Index is a simple comparison of the relationship between incidents in terms of distance (space) and time (Knox, 1963; 1964). That is, each individual pair is compared in terms of distance and in terms of time interval. Since each pair of points is being compared, there are N*(N-1)/2 pairs. The distance between points is divided into two groups - Close in distance and Not close in distance, and the time interval between points is also divided into two groups - Close in time and Not close in time. The definitions of 'close' and 'Not close' are left to the user.

A simple 2 x 2 table is produced that compares closeness in distance with closeness in time. The number of pairs that fall in each of the four cells are compared (Table 9.1).

Table 9.1

Logical Structure of Knox Index

	Close in time	**Not close in time**	
Close in Distance	O_1	O_2	S_1
Not close in distance	O_3	O_4	S_2
	S_3	S_4	N

where $N = O_1 + O_2 + O_3 + O_4$

$S_1 = O_1 + O_2$

$S_2 = O_3 + O_4$

$S_3 = O_1 + O_3$

$S_4 = O_2 + O_4$

The actual number of pairs that falls into each of the four cells are then compared to the expected number if there was no relationship between closeness in distance and closeness in time. The expected number of pairs in each cell under strict independence between distance and the time interval is obtained by the cross-products of the columns and row totals (table 9.2).

Table 9.2

Expected Frequencies for Knox Index

	Close in time	Not close in time
Close in Distance	E_1	E_2
Not close in distance	E_3	E_4

where $E_1 = S_1 * S_3 / N$
$E_2 = S_1 * S_4 / N$
$E_3 = S_2 * S_3 / N$
$E_4 = S_2 * S_4 / N$

The difference between the actual (observed) number of pairs in each cell and the expected number is measured with a Chi-square statistic (equation 9.1).

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i} \quad \text{with 1 degree of freedom} \qquad (9.1)$$

Monte Carlo Simulation of Critical Chi-square

Unfortunately, the usual probability test associated with the Chi-square statistic cannot be applied since the observations are not independent. Interaction between space and time tend to be compounded when calculating the Chi-square statistic. For example, we've noticed that the Chi-square statistic tends to get larger with increasing sample size, a condition that would normally not be true with the independent observations. To handle the issue of interdependency, there is a Monte Carlo simulation of the chi-square value for the Knox Index under spatial randomness (Dwass, 1957; Barnard, 1963). If the user selects a simulation, the routine randomly selects M pairs of a distance and a time interval where M is the number of pairs in the data set (M = N* [N-1]/2) and calculates the Knox Index and the chi-square test. Each pair of a distance and a time interval are selected from the range between the minimum and maximum values for distance and time interval in the data set using a uniform random generator.

The random simulation is repeated K times, where K is specified by the user and Usually, it is wise to run the simulation 1000 or more times. The output includes:

1. The sample size
2. The number of pairs

3. The calculated chi-square value of the Knox Index from the data
4. The minimum chi-square value of the Knox Index from the simulation
5. The maximum chi-square value of the Knox Index from the simulation
6. Ten percentiles from the simulation:

   a. 0.5%
   b. 1%
   c. 2.5%
   d. 5%
   e. 10%
   f. 90%
   g. 95%
   h. 97.5%
   i. 99%
   j. 99.5%

Methods for Dividing Distance and Time

In the *CrimeStat* implementation of the Knox Index, the user can divide distance and time interval based on the three criteria:

1. The mean (mean distance and mean time interval). This is the default.

2. The median (median distance and median time interval)

3. User defined criteria for distance and time separately.

There are advantage to each of these methods. The mean is the center of the distribution; it denotes a balance point. The median will divide both distance and time interval into approximately equal numbers of pairs. The division is approximate since the data may not easily divide into two equal numbered groups. A user-defined criteria can fit a particular need of an analyst. For example, a police department may only be interested in incidents that occur within two miles of each other within a one week period. Those criteria would be the basis for dividing the sample into 'Close' and 'Not close' distances and time intervals.

Example of the Knox Index

For an example, vehicle thefts in Baltimore County for 1996 were taken. There were 1855 vehicle thefts for which a date was recorded in the data base. The data base was further broken down into twelve separate monthly subsets. Using the median for both distance and time interval, the Knox Index was calculated for the entire set of 1855 incidents. Then, using the median distance for the entire year but a month-specific median time interval, the Knox Index was calculated for each of the twelve months. Table 9.3 presents the Chi-square values and their pseudo-significance levels.

To produce a better test of the significance of the results, 1000 random simulations were calculated for the vehicle theft for the entire year. Table 9.3 below shows the results. Because an extreme value could be obtained by chance with a random distribution, reasonable cut-off points are usually selected from the simulation. In this case, we want a cut-off point that approximates a 5% significance level. Since the Knox Index is a one-tailed test (i.e., only a high chi-square value is indicative of spatial interaction), we adopt an upper threshold of the 95 percentile. In other words, only if the observed chi-square test for the Knox Index is larger than the 95 percentile threshold will the null hypothesis of a random distribution between space and time be rejected.

Table 9.3

Knox Index for Baltimore County Vehicle Thefts
Median Split
N = 1,855 with 1,719,585 comparisons

Month	Actual Chi-square	95 Percentile Simulation Chi-square	Approx. p
January	0.26	6.95	n.s.
February	0.00	6.61	n.s.
March	0.00	6.86	n.s.
April	0.50	6.56	n.s.
May	1.04	7.25	n.s.
June	0.01	6.02	n.s.
July	9.96	9.05	.05
August	5.91	5.55	.05
September	0.27	5.41	n.s.
October	3.33	6.43	n.s.
November	10.79	8.91	.01
December	0.00	6.87	n.s.
All of 1996	**8.69**	**41.89**	**n.s.**

For the entire year, there was not a significant clustering between space and time. Approximately, 26.7% of the incidents were both close in distance (i.e., closer than the median distance between pairs of incidents) and close in time (i.e., closer than the median time interval between pairs of incidents). However, when individual months are examined, only three show significant relationships: July, August, and November. During these months, there is an interaction between space and time. Typically, incidents that cluster together spatially tend also to cluster together temporally. However, it could be the opposite (i.e., events that cluster together temporally tend to be far apart spatially).

The next step would to identify whether there are particular clusters that occur within a short time period. Using one of the 'hot spot' analysis methods discussed in chapters 6 and 7, an analyst could take the events for the three months and try to identify

whether there is spatial clustering during those three months that does not normally occur. We won't do that here, but the point is that the Knox Index is useful to identify *when* there is spatial clustering.

Problems with the Knox Index

The Knox Index is a simple measure of space-time clustering. However, because it is only a 2 x 2 table, different results can be obtained by varying the cut-off points for distance or time. For example, using the mean as the cut-off, the overall Chi-square statistic for all vehicle thefts was 8.67, reasonably close. However, when a cut-off point for distance of 1000 meters and a cut-off point for time of 80 days was used, the Chi-square statistic dropped to 3.16. In other words, the Knox Index will produce different results for different cut-off points.

A second problem has to do with the interpretation. As with any Chi-square test, differences between the observed and expected frequencies could occur in any cell or any combination of cells. Finding a significant relationship does not automatically mean that events that were close in distance were also close in time; it could have been the opposite relationship. However, a simple inspection of the table can indicate whether the relationship is as expected or not. In the above example, all the significant relationships showed a higher proportion of events that were both close in distance and close in time.

Mantel Index

The Mantel Index resolves some of the problems of the Knox Index. Essentially, it is a correlation between distance and time interval for pairs of incidents (Mantel, 1967). More formally, it is a general test for the correlation between two *dissimilarity* matrices that summarizes comparisons between pairs of points (Mantel and Bailar, 1970). It is based on a simple cross-product of two interval variables (e.g., distance and time interval):

$$T = \sum_{i=1}^{N} \sum_{j=1}^{N} (X_{ij} - \text{MeanX})(Y_{ij} - \text{MeanY}) \qquad (9.2)$$

where X_{ij} is an index of similarity between two observations, i and j, for one variable (e.g., distance) while Y_{ij} is an index of similarity between the same two observations, i and j, for another variable (e.g., time interval).

The cross-product is then normalized by dividing each deviation by its standard deviation:

$$r = \frac{1}{(N-1)} \sum_{i=1}^{N} \sum_{j=1}^{N} (X_{ij} - \text{MeanX})/S_x * (Y_{ij} - \text{MeanY})/S_y \qquad (9.3a)$$

$$= \sum_{i=1}^{N} \sum_{j=1}^{N} Z_x * Z_y / (-1) \qquad (9.3b)$$

where X_{ij} and Y_{ij} are the original variables for comparing two observations, i and j, and Z_x and Z_y are the normalized variables.

Example of the Mantel Index

In *CrimeStat*, the Mantel Index routine calculates the correlation between distance and time interval. To illustrate, table 9.4 examines the Mantel correlation for the 1996 vehicle thefts in Baltimore County that was illustrated above. As seen, the correlations are all low. However, as with the Knox Index, July, August and November produce relatively higher correlations. If used as an index, rather than an estimate of variance explained, the Mantel Index can identify time periods when spatial interaction is occurring.

Table 9.4

Mantel Index for Baltimore County Vehicle Thefts
Median Split
N = 1,855 and 1,719,585 Comparisons

Month	r	Simulation 2.5%	Simulation 97.5%	Approx. p-level
January	-.0047	-0.033	0.033	n.s.
February	-.0023	-0.037	0.042	n.s.
March	-.0245	-0.032	0.039	n.s.
April	0.0077	-0.040	0.041	n.s.
May	0.0018	-0.038	0.043	n.s.
June	0.0043	-0.035	0.041	n.s.
July	0.0348	-0.034	0.033	.025
August	0.0544	-0.034	0.035	.01
September	0.0013	-0.044	0.046	n.s.
October	0.0409	-0.037	0.043	n.s.
November	0.0630	-0.042	0.040	.001
December	0.0086	-0.035	0.038	n.s.
All of 1996	0.0015	-0.009	0.010	n.s.

Monte Carlo Simulation of Confidence Intervals

Even though the Mantel Index is a Pearson product-moment correlation between distance and time interval, the measures are not independent and, in fact, are highly interdependent. Consequently, the usual significance test for a correlation coefficient is

not appropriate. Instead, the Mantel routine offers a simulation of the confidence intervals around the index. If the user selects a simulation, the routine randomly selects M pairs of a distance and a time interval where M is the number of pairs in the data set (M = N* [N-1]/2) and calculates the Mantel Index. Each pair of a distance and a time interval are selected from the range between the minimum and maximum values for distance and time interval in the data set using a uniform random generator.

The random simulation is repeated K times, where K is specified by the user. Usually, it is wise to run the simulation 1000 or more times. The output includes:

1. The sample size
2. The number of pairs
3. The calculated Mantel Index from the data
4. The minimum Mantel value from the simulation
5. The maximum Mantel value from the simulation
6. Ten percentiles from the simulation:

    a. 0.5%
    b. 1%
    c. 2.5%
    d. 5%
    e. 10%
    f. 90%
    g. 95%
    h. 97.5%
    i. 99%
    j. 99.5%

To illustrate, 1000 random simulations were calculated for each month using the same sample size as the monthly vehicle theft totals. Table 9.4 above shows the results. Because an extreme value could be obtained by chance with a random distribution, reasonable cut-off points are usually selected from the simulation. In this case, we want cut-off points that approximate a 5% significance level. Since the Mantel Index is a two-tailed test (i.e., one could just as easily get dispersion between space and time as clustering), we adopt a lower threshold of the 2.5 percentile and an upper threshold of 97.5 percentile. Combined, the two cut-off points ensure that approximately 5% of the cases would be either lower than the lower threshold or higher than the upper threshold under random conditions.[1] In other words, only if the observed Mantel Index is smaller than the lower threshold or larger than the upper threshold will the null hypothesis of a random distribution between space and time be rejected.

In Table 9.4, for the entire year, the observed Mantel Index (correlation between space and time) was 0.0015. The 2.5 percentile was -.009 and the 97.5 percentile was 0.01. Since the observed value is between these two cut-off points, we cannot reject the null hypothesis of no relationship between space and time. However, for the individual months, again, July, August and November have correlations above the upper cut-off threshold.

Thus, for those three months *only*, the amount of space-time clustering in the vehicle theft data is most likely greater than what would be expected on the basis of a chance distribution. One would, then, have to explore the data further to find out where those vehicle thefts were occurring, using one the hot spot routines in Chapter 6.

Limitations of the Mantel Index

The Mantel Index is a useful measure of the relationship between space and time. But it does have limitations. First, because it is a Pearson-type correlation coefficient, it is prone to the same types of problems that befall correlations. Extreme values of either space or time could distort the relationship, either positively, if there are one or two observations that are extreme in *both* distance in time interval, or negatively, if there are only one or two observations that are extreme in *either* distance or in time interval.

Second, because the test is a comparison of all pairs of observations, the correlations tend to be small, as noted above. This makes it less intuitive as a measure than a traditional correlation coefficient which varies between -1 and +1 and in which high values are expected. For most analysts, it is not very intuitive to have an index where 0.05 is a high value. This doesn't fault the statistic as much make it a little non-intuitive for users.

Third, as with any correlation coefficient, the sample size needs to be fairly large to produce a stable estimate. In the above, example, one could further break down monthly vehicle thefts by week or, even, day. However, the number of cases will decrease considerably. In the above example, with 1,855 vehicle thefts over a year, the weekly average would be around 36, which is a small sample. Intuitively, a crime analyst wants to know when space-time clustering is occurring and a short time frame is critical for detection; a week would be the largest time interval that would be useful. However, as the sample size gets small, the index becomes unstable. For one thing, the sample size makes the index volatile. While the Monte Carlo simulation will adjust for the sample size, the range of the cut-off thresholds will vary considerably from one week to another with small sample sizes. The analyst will have to run the simulation repeatedly to adjust for the varying sample sizes. For another thing, the shortened time frame allows fewer distinctions in time; if one takes a very narrow time frame (e.g., a day), there can be virtually no time differences observed. One would have to switch to an hourly analysis to produce meaningful differences.

One way to get around this is to have a moving average where the time frame is adjusted to fit a constant number of days (e.g., a 14 day moving average). The advantage is that the sample size tends to remain fairly constant; one could therefore reduce the number of recalculations of the cut-off thresholds since they would not vary much from one day to another. To make this work, however, the data base must be set up to produce the appropriate number of incidents for a moving average analysis.

Nevertheless, the Mantel Index remains a useful tool for analysts. It is still widely used for space-time analysis and it has been generalized to many other types of

dissimilarity analyses than just space and time. If used carefully, the index can be a powerful tool for detection of clusters that are also concentrated in time.

Spatial-Temporal Moving Average

The Spatial-Temporal Moving Average is a simple statistic. It is the moving mean center of M observations where M is a sub-set of the total sample, N. By 'moving', the observations are sequenced in order of occurrence. Hence, there is a time dimension associated with the sequence. The M observations is called the *span* and the default span is 5 observations. The span is centered on each observation so that there are an equal number on both sides. Because there are no data points prior to the first event and after the last event, the first few mean centers will have fewer observations than the rest of the sequence. For example, with a span of 5, the first and last mean centers will have only three observations, the second and next-to-last will have 4 observations, while all others will have 5. In general, it's a good idea to choose an odd number since the middle of the span will be centered on a real observation rather than having to fall between two in the case of an even span.

Though simple, the Spatial-Temporal Moving Average is very useful for detecting changes in behavior by serial offenders. In the next chapter, we will examine journey-to-crime models that attempts to estimate the likely origin location of a serial offender based on the distribution of incidents committed by the offender. However, if the serial offender has either moved residences or else moved the field of operation, then the technique will error because it is assuming a stable field of operations when, in fact, it isn't. The moving average can suggest whether the offender's behavior is stable or not.

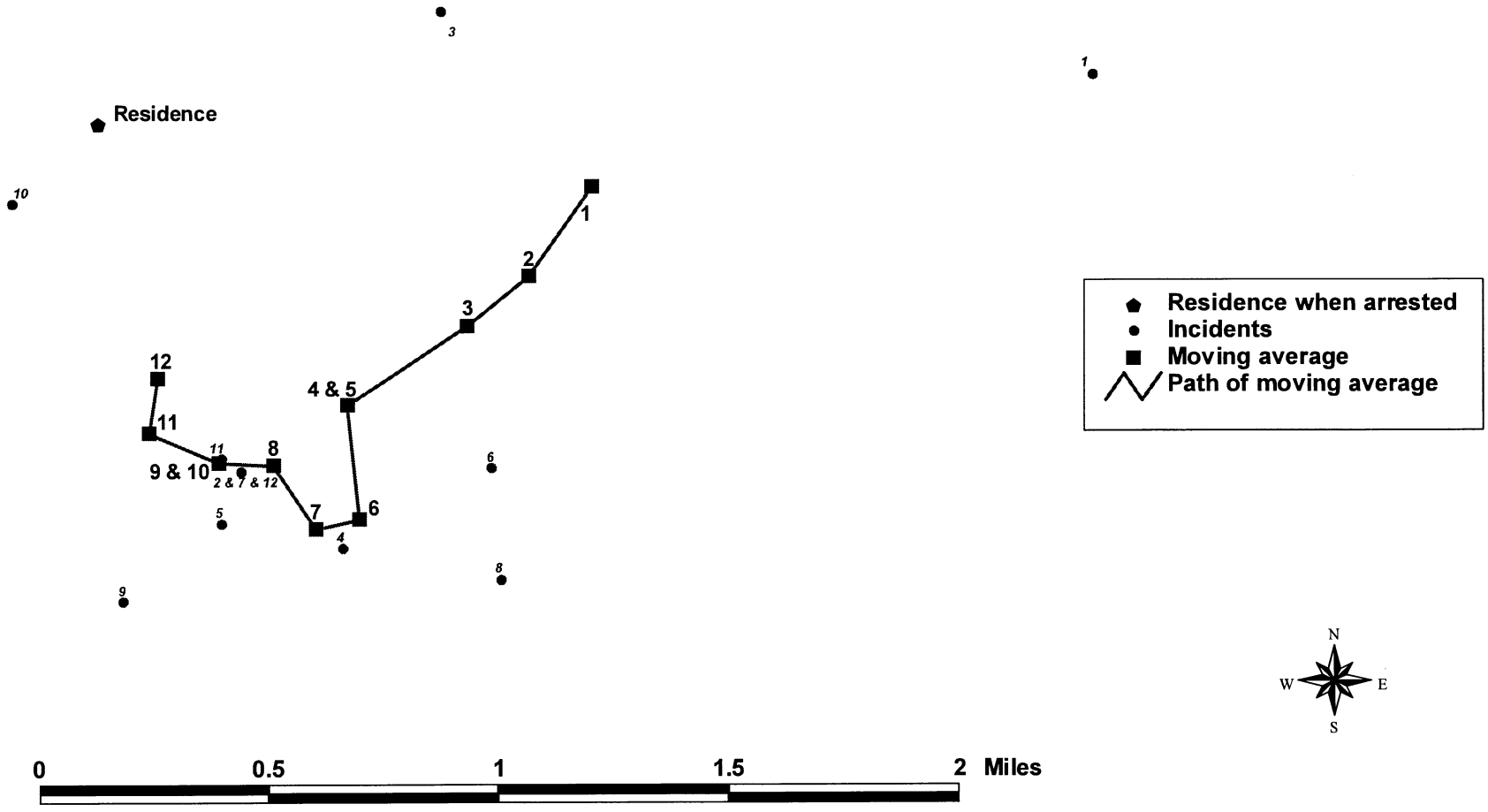
As an example, figure 9.2 below shows the Spatial-Temporal Moving Average of an offender who committed 12 offenses before being arrested. The individual committed eight thefts from vehicles, two thefts from stores, one residential burglary and one highway robbery. The actual incidents are shown in red circles with the sequence number displayed. The moving average is shown in blue square with the sequence number displayed. The path of the moving average is shown as a green line.

As seen, there is a definite shift in the field of operation by this offender. The mean center moves about a mile during this period but the consistency of the trend would suggest that something fundamental changed by the offender, either the person moved residences or the nature of the committed crimes changed. In using the Journey-to-crime tools, an analyst would probably want to focus on the latter events since these are more geographically circumscribed. Notice that the last two moving averages are relatively close to the actual residence location of the offender when arrested (less than three-quarters of a mile away).

In short, the Spatial-Temporal Moving Average simply plots the changes in the mean center of the span and is useful for detecting changes in the behavior pattern of serial offenders.

Figure 9.2:

Moving Path of Serial Offender Sequence of 12 Crimes



Correlated Walk Analysis

Correlated Walk Analysis (CWA) is a tool that is aimed at analyzing the spatial and temporal *sequencing* of incidents committed by a single serial offender. In this sense, it is the 'flip side' of Journey to crime analysis (see chapter 10). Whereas journey to crime analysis makes guesses about the likely origin location for a serial offender, based on the spatial distribution of the incidents committed by the offender, the CWA routine makes guesses about the time and location of a next event, based on both the spatial distribution of the incidents and the temporal sequencing of them. In effect, it is a Spatial-Temporal Moving Average with a prediction of a next event.

The statistical origin of CWA is Random Walk Theory. Random Walk Theory has been developed by physicists to explain the distribution of molecules in a rapidly changing environment (e.g., the movements of a particle in a gas which is diffusing - Brownian movement). Sometimes called a 'drunkard's walk', the theory starts with the premise that movement is random in all directions. From an arbitrary starting point, a particle (or person) moves in any direction in a series of steps. The direction of each step is independent of the previous steps. After each step, a random decision is made and the person moves in a random direction. This process is repeated *ad infinitum* until an arbitrary stopping point is selected (i.e., the observer quits looking). It has been shown mathematically that all one and two dimensional random walks must eventually return to their original starting point (Spitzer, 1963; Henderson, Renshaw, and Ford, 1983).[2] This is called a *recurrent random walk*. On the other hand, independent random walks in more than two dimensions are not necessarily recurrent, a state called *transient random walk*.

Figure 9.3 illustrates a random walk of 2000 steps. For a large number of steps in a two-dimensional walk, the likely distance of a person (or particle) from the starting point is

$$E(d) = d_{rms} * \sqrt{N} \qquad (9.4)$$

where $d_{rms} = \sqrt{(\Sigma d_i^2 / N)}$. The term, d_{rms} is the *root mean square* of distance.

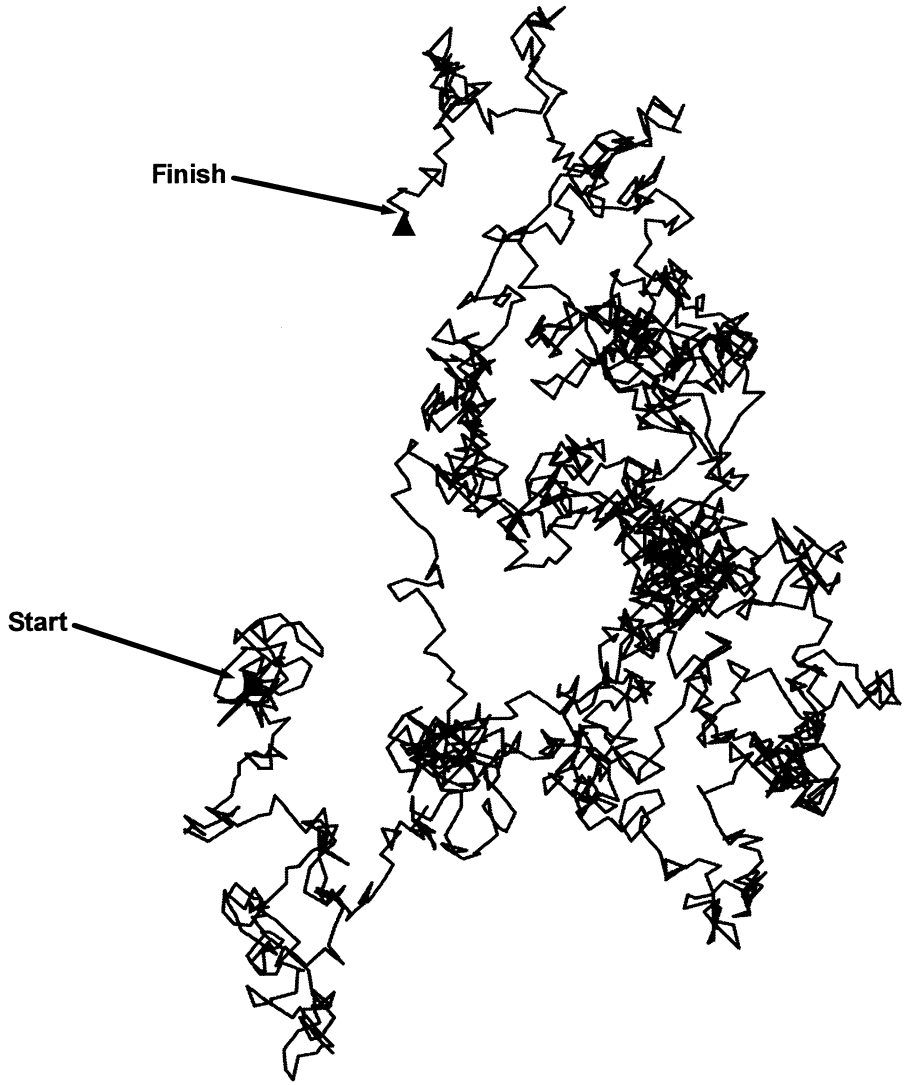
There are a number of different types of random walks. The simplest is a movement of uniform distance only along a grid cell (i.e., a Manhattan geometry). The person can only move North, South, East or West for a unit distance of 1. A more complex random walk allows angular distances and an even more complex random walk allows varying distances (e.g., normally distributed random distances, uniformly random distances). The walk in figure 9.3 was of this latter type. X and Y values were selected randomly from a range of -1 to +1 using a uniform random number generator. For a conceptual understanding of Random Walk Theory, see Chaitin (1990) and, for a mathematical treatment, see Spitzer (1976). Malkiel (1999) applied the concepts of Random Walk Theory to stock price fluctuations in a book that has now become a classic.

Henderson, Renshaw and Ford (1983; 1984) have introduced the concept of a *correlated random walk*. In a correlated random walk, momentum is maintained. If a person is moving in a certain direction, they are more likely to continue in that direction

Figure 9.3:

A Random Walk

2000 Random Steps of -1.0 to +1.0 in X and Y Direction



than to reverse direction or travel orthogonally. In other words, at any one decision point, the probabilities of traveling in any direction are not equal; the same direction has a higher probability than an orthogonal change (i.e., turning 90 degrees) and those, in turn, have a higher probability than completely reversing direction. By implication, the same is true for distance and distance. A longer step than average is likely to be followed by another longer step than average while a shorter step than average is likely to be followed by another short step. Similarly, there is consistency in the time interval between events; a short interval is also likely to be followed by a short interval. In other words, a correlated random walk is a random walk with momentum (Chen and Renshaw, 1992; 1994). These authors have applied the theory to the analysis of the branching of tree roots (Henderson, Ford, Renshaw, and Deans, 1983; Renshaw, 1985).

Correlated Walk Analysis

Correlated Walk Analysis is a set of tools that can help an analyst understand the sequencing of sequential events in terms of time interval, distance and direction. In *CrimeStat*, there are three CWA routines. The first two help the analyst understand whether there are patterns in time, distance or direction while the last routine allows the analyst to make a guess about the next likely event, when it will occur and where it will occur. The three routines are:

1. CWA - Correlogram
2. CWA - Diagnostics
3. CWA - Prediction

CWA - Correlogram

The *Correlogram* routine calculates the correlation in time interval, distance, and bearing (direction) between events. It does this through *lags*. A lag is a separation in the intervals between events. The difference between the first and second event is the first interval. The difference between the second and third events is the second interval. The difference between the third and fourth events is the third interval, and so forth. For each successive interval, there is a time difference; there is a distance and there is a direction. One could extend this to all the intervals, comparing each interval with the next one; that is, we compare the first interval with the second, the second interval with the third, the third interval with the fourth, and so on until the sample is complete. When comparing successive intervals, this is called a *lag of 1*. It is important to keep in mind the distinction between an event (e.g., an incident) and an interval. It takes two events to create an interval. Thus, for a lag of 1, there are M= N-1 intervals where N is the number of events (e.g., for 3 incidents, there are 2 intervals).

A lag of two compares every other event. Thus, the first interval is compared to the third interval; the second interval is compared to the fourth; the third interval is compared to the fifth; and so on until there are no more intervals left in the sample. Again, the comparison is for time difference, distance, and direction separately. We can extend this logic to a lag of 3 (every third event), a lag of 4 (every fourth event), and so forth.

The CWA - Correlogram routine calculates the Pearson Product-Moment correlation coefficient between successive events. For a lag of 1, it compares successive events and correlates the time interval, distance, and bearing separately for these successive events. For a lag of 2, it compares every other event and correlates the time interval, distance, and bearing separately for these successive events. The routine does this until it reaches a maximum of 7 lags (i.e., every seventh event). However, if the sample size is very small, it may not be able to calculate all lags. It will require 12 incidents (events) to calculate all seven lags since it requires at least four observations per lag (i.e., N - L - 4 where N is the number of events and L is the maximum number of lags calculated).

Adjusted Correlogram

The Correlogram calculates the raw correlation between intervals by lag for time, distance, and bearing. One of the problems that may appear, especially with small samples, is for higher-order lags to be very high, either positive or negative. There are probably two reasons for this. For one thing, with each lag, the sample size decreases by one; with a very small sample size, correlations can become very volatile, jumping from positive to negative, and from low to high. Another reason is that periodicity in the data set is compounded with higher-order lags in the form of 'echos'. For example, if a lag of 2 is high, then a lag of 4 will also be somewhat high since there is a compounding of the lag 2 effect. When combined with a small sample size, it is not uncommon to have higher-order lags with very high correlations, sometimes approaching +/- 1.0. The user must be careful in selecting a higher-order lag because there is an apparent effect which may be due to the above reasons, rather than any real predictability. One of the key signs for spurious higher-order effect is a sudden jump in the strength of the correlation from one lag to the next (though sometimes a high higher-order lag can be real; see examples below).

To minimize these effects, the output also includes an adjusted correlogram that adjusts for the loss of degrees of freedom. The formula is:

$$A = \frac{M - L - 1}{M - 1} \tag{9.5}$$

where M is the number of intervals (N-1) and L is the number of lags. For example, for a sample size of 13, there will be 12 intervals (M). For a lag of 1, the adjustment will be

$$A = \frac{12 - 1 - 1}{12 - 1} = \frac{10}{11} = 0.909$$

The effect of the adjustment is to reduce the correlation for higher-order lags. It won't completely eliminate the effect, but it should help minimize spurious effects. As will be shown below, however, sometimes high higher-order lags are real.

CWA - Correlogram Output

The routine outputs 10 parameters:

1. The sample size (number of events);
2. Number of intervals;
3. Information on the units of time, distance, and bearing;
4. Final distance to origin in meters (distance between last and first event);
5. Expected random walk distance from origin (if sequence was strictly random);
6. Drift (the ratio of actual distance from origin to expected random walk distance);
7. Final bearing from origin (direction between last event and first event);
8. Expected random walk bearing. Defined as 0 because there is no expected direction.
9. Correlations by lag for time, distance, and bearing (up to 7 lags); and
10. Adjusted correlations by lag for time, distance, and bearing (up to 7 lags).

The aim of the CWA - Correlogram is to examine repetitive sequences, whether for time interval, distance or direction. It is possible to have separate repetitions for time, distance and direction. For example, an offender may commit crimes every 7 days or so, say, on the weekend. In this case, the individual is repeating himself/herself about once every week. Similarly, an individual may alternate directions, first going East then going West, then going back to the East, and so forth. In other words, what we're asking with the routine is whether there are any repetitions in the sequence of incidents committed by a serial offender. Does he/she repeat the crimes in time? If so, what is the *periodicity* (the repititious sequence? Does he/she repeat the crimes in distance? Is so, what is the periodicity? Finally, does he/she repeat the crimes in direction? If so, what is the periodicity? The CWA-Correlogram, therefore, analyzes the sequence of incidents committed by an individual and does this separately for time interval, distance, and direction.

Offender repetition

Why is this important? Most crime analysis is predicted on the assumption that offenders (people in general) repeat themselves, consciously or unconsciously. That is, individuals have specific behavior patterns that tend to be repeated. If an individual acts in a certain way (e.g., committing a burglary), then, most likely , the person will repeat himself/herself again. There is no guarantee, of course. But, because human beings do not behave spatially or temporally random but tend to operate in somewhat consistent ways, there is a likelihood that the individual will act in a similar manner again.

This assumption is the basis of profiling which aims at understanding the MO of an offender. If offenders were totally random in their behavior, detection and apprehension would be made much more difficult than it already is. So, between the two extremes of a totally random individual (the 'random walk person') and a totally predictable individual

(the 'algorithmic person'), we have the bulk of human behavior, at least in terms of time, distance and direction.

CWA - Diagnostics

The Diagnostics routine is similar to the CWA - Correlogram except that it calculates an Ordinary Least Squares autoregression for a particular lag. That is, it regresses each interval against a previous interval. The user enters the lag number (the default is 1) and the routine produces three regression models for the successive event as the dependent variable against the prior event as the independent variable. There are three equations, for time interval, distance, and bearing separately. The output includes:

1. The sample size (number of events);
2. The number of intervals;
3. Information on the units of time, distance, and bearing;
4. The multiple correlation coefficient;
5. The squared multiple correlation coefficient (i.e., R^2);
6. The overall standard error of estimate;
7. The regression coefficient for the constant and for the prior event;
8. The standard error of the regression coefficients;
9. The t-values for the regression coefficients;
10. The p-value (two-tail) for the regression coefficients;
11. An analysis of variance test for the full model. This includes sum of squares for the regression term and for the residual;
12. The ratio of the regression sum of squares to the residual sum of squares (the F-ratio); and
13. The p-value associated with the F-value.

What the regression diagnostics provides is an indicator of the amount of predictability in the lag. It has the same information as the Correlogram (since the square of the correlation, r^2, is the same as R^2 for a single independent variable regression equation), but it is easier to interpret. Essentially, it is argued below that, unless the R^2 in the regression equation is sufficiently high, that one is better off using the mean or median lag for prediction. Conversely, if the R^2 is very high, then the user should be suspicious about the data.

CWA - Prediction

Finally, after having analyzed the sequential pattern of events, the user can make a prediction about the time and place of the next event. There are three methods for making a prediction, each with a separate lag:

1. Mean difference
2. Median difference
3. Regression equation

The method is applied to the last event in the data set. The *mean difference* applies the mean interval of the data for the specified lag to the last event. For example, for time interval and a lag of 1, the routine calculates the interval between each event and takes the average. It then applies the mean time interval to the last time in the data set as the prediction. The *median difference* applies the median interval of the data for the specified lag to the last event. For example, for bearing and a lag of 1, the routine calculates the direction (bearing) between each event, calculates the median bearing, and applies that median average to the location of the last event in the data set as the predicted value.

The *regression equation* calculates a regression coefficient and constant for the specified lag and uses the data value for the last *interval* as input into the regression equation; the result is the predicted value. For example, for distance and a lag of 1, the routine calculates the regression coefficient and constant for a regression equation in which each event is compared to the previous event. The last distance in the data set (i.e., between the last event and the previous event) is used as an input for the regression equation and the predicted distance is marked off from the coordinates of the last event.

In other words, the routine takes the time and location of the last event and adds a time interval, a direction, and a distance as a predicted next event (next time, next location). The method by which this prediction is made can be the mean interval, the median interval, or the regression equation. If the user species a lag other than 1, that lag is applied to the last event. For example, for time with a mean difference and a lag of 2, the routine calculates the time interval between each event and every other event, calculates the average and applies that average to the last event in the data set.

CWA - Prediction Graphical Output

The CWA - Prediction routine outputs five graphical objects in 'shp', 'mif, or 'bna' formats. The routine adds five prefixes to the file name of the output object:

1. Events - a line indicating the sequence of events. If the user also brings in the points in the data set, it will be possible to number each of these steps;
2. PredDest - the predicted location for the next event;
3. Path - a line from the last location in the data set to the predicted location;
4. POrigL - a point representing the center of minimum distance of the data set. The center of minimum distance is taken as a proxy for the origin location of the offender; and
5. PW - a line from the expected origin to the predicted destination

For example, if the user provides the file name 'NightRobberies' and specifies a 'shp' output, there will be five objects output:

EventsNightRobberies.shp
PathNightRobberies.shp
PWNightRobberies.shp
PredDestNightRobberies.shp
POrigLNightRobberies.shp

Example 1: A Completely Predictable Individual

The simplest way to illustrate the logic of the CWA is to start with a completely predictable individual. This individual commits crimes on a completely systematic basis. Table 9.5 illustrates the behavior of this individual.

Starting at an arbitrary origin with an X coordinate of 1 and a Y coordinate of 1 and on day 1, the individual commits 13 incidents in total. In the table, these are numbered events 1 through 13. Let's start with direction and distance. From the origin, the individual always travels in a Northeast direction of 45 degrees (clockwise from due North - 0 degrees). The individual's second incident is at coordinate X=2, Y=2. Thus, the individual traveled at 45 degrees from the previous incident and for a distance of 1.4142 (the hypotenuse of the right angle created by traveling one unit in the X direction and one unit in the Y direction). For the third incident, the individual commits this at X=4, Y=4. Thus, the direction is also at 45 degrees from the previous location but the distance is now 2.8284 (or the square root of 8 which comes from a step of 2 along the X axis and a step of 2 along the Y axis). For the fourth incident, the individual commits the crime at X=7, Y=7. Again, the direction is 45 degrees, but the distance is 4.2426 (or the square root of 18 which comes from a step of 3 along the X axis and a step of 3 along the Y axis).

Table 9.5

Example of a Predictable Serial Offender: 1

(N = 13 incidents)

Event	X	Y	Distance	Days	Time Interval
1	1	1	-	1	-
2	2	2	1.4142	3	2
3	4	4	2.8284	7	4
4	7	7	4.2426	9	2
5	8	8	1.4142	13	4
6	10	10	2.8284	15	2
7	13	13	4.2426	19	4
8	14	14	1.4142	21	2
9	16	16	2.8284	25	4
10	19	19	4.2426	27	2
11	20	20	1.4142	31	4
12	22	22	2.8284	33	2
13	25	25	4.2426	37	4
Logical prediction for next event	14	26	26	1.4142	39	2

For the fifth incident, again the individual traveled at 45 degrees to the previous incident, but repeated himself/herself with a step of only 1 unit in both the X and Y directions. The individual then continued the sequence, always traveling in a 45 degree orientation to due North. For distance, a step of 1 in both the X and Y directions is followed by a step of 2 in both directions, and is followed by a step of 3 in both directions. In other words, the individual repeats direction every time and repeats distance every third time. There is a periodicity of 1 for direction and 3 for distance.

For time interval, this individual repeats him/herself every other time. The second event occurs 2 days after the first event. The third event occurs 4 days after the second event; the fourth event occurs 2 days after the third event; the fifth events occurs 4 days after the fourth event; and so forth. In other words, for time interval, the individual repeats him/herself every other interval (i.e., the periodicity is 2). Figure 9.4 illustrates the sequence; the number at each event location is the number of the day that the individual committed the offense (starting at an arbitrary day 1).

Since this fictitious individual is completely predictable, we can easily guess when and where the next event will occur (see table 9.5 above). The direction will, of course, be at 45 degrees from the previous location. Looking at the last known event (event 13), the distance traveled was 4.2426. Thus, we predict that the individual will revert to a move of 1 in the X direction and 1 in the Y direction, or coordinates X=26, Y=26. Finally, for time interval, since the last known time interval was 4 days, then this individual will commit the next event 2 days later, or day number 39.

Example 1: Analysis

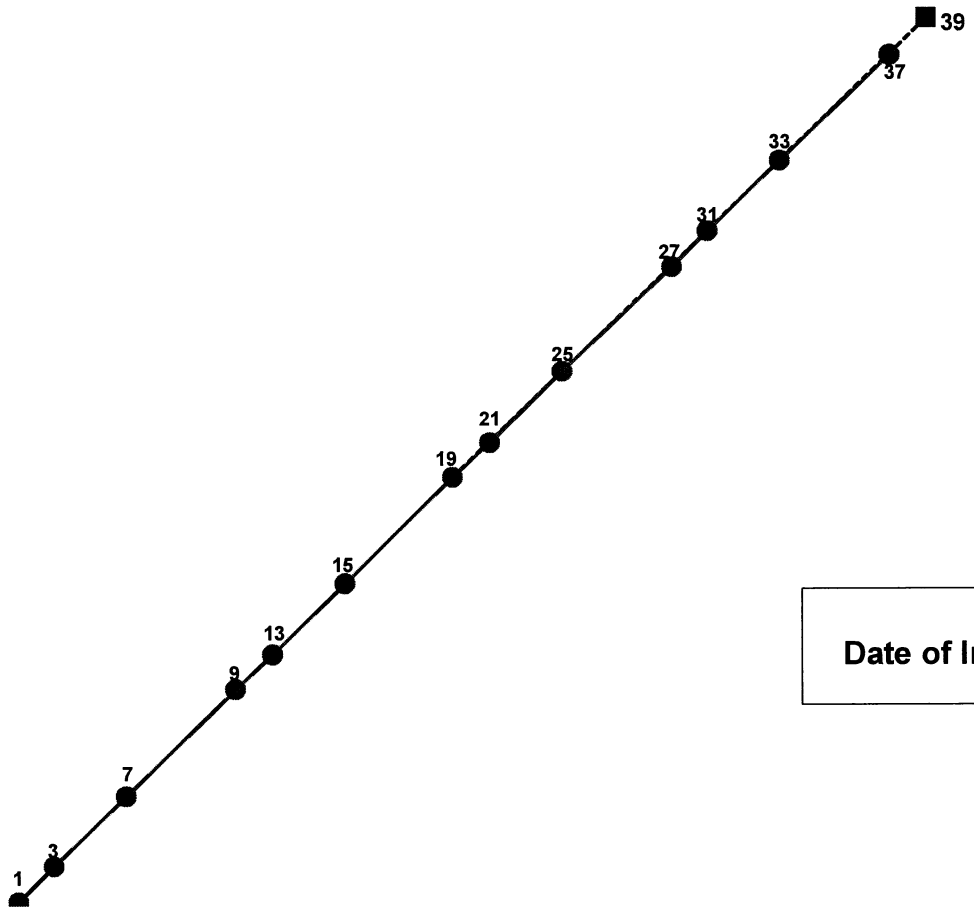
The first step is to analyze the sequencing of the events. There are 13 events and 12 intervals. The correlogram produces the following output (table 9.6).

Looking at the unadjusted correlations, it can be seen that time shows an alternating pattern of perfect correlations. The first repeating positive 1.0 correlation is for lag 2, which is the exact periodicity that was specified in the example. This offender repeats the time sequence every other time. Thus, if the individual alternates between committing offenses 2 and 4 days after the last, then knowing the time interval for the last offense, it can be assumed that the next event will repeat the next-to-the-last time interval.

For distance, the highest correlation is for a lag of 3. This offender repeat himself/herself every third time, which is exactly what was programmed into the example. Thus, knowing the location of the last event, it can be assumed that the individual will choose the same distance for the next interval as three earlier. Finally, all lags show a perfect 1.0 correlation for bearing. The lowest one is taken, which is a lag of 1. That is, this individual repeats the direction every single time (i.e., he/she always travels in the same direction). Thus, in summary, the correlogram shows that the individual repeats the time interval every other time, the distance every third time, and the direction every time.

Figure 9.4:

Example of a Predictable Serial Offender: I (N=13 Incidents)



Date of Incident Shown

Table 9.6
Correlogram of Predictable Serial Offender: 1

Correlated Walk Analysis -- Correlogram:

Sample size .....: 13
Measurement type ...: Direct
Input units .....: Feet
Time units .....: Days
Distance units .....: Feet
Bearing units .....: Degrees

Lag	Correlation		
	Time	Distance	Bearing
0	1.00000	1.00000	1.00000
1	-1.00000	-0.42105	1.00000
2	1.00000	-0.56522	1.00000
3	-1.00000	1.00000	1.00000
4	1.00000	-0.38462	1.00000
5	-1.00000	-0.58824	1.00000
6	1.00000	1.00000	1.00000
7	-1.00000	-0.28571	1.00000

Adjusted:

Lag	Correlation		
	Time	Distance	Bearing
0	1.00000	1.00000	1.00000
1	-0.90909	-0.38278	0.90909
2	0.81818	-0.46245	0.81818
3	-0.72727	0.72727	0.72727
4	0.63636	-0.24476	0.63636
5	-0.54545	-0.32086	0.54545
6	0.45455	0.45455	0.45455
7	-0.36364	-0.10390	0.36364

The CWA - Diagnostics routine merely confirms these correlations. The regression equations yield an R^2 of 1.0 (unadjusted) for each of three variables, for the appropriate lag. For example, table 9.7 below shows the regression results for distance for a lag of 3

Table 9.7

Regression Results for Serial Offender 1: Distance

Variable: distance		Standard error of estimate:		0.00000
Multiple R: 1.00000		Squared multiple R:		1.00000
	Coefficient	Std Error	t	P(2 Tail)
Constant	0.000000	0.00000	0.00000	0.00000
Coefficient	1.000000	0.00000	0.00000	0.00000

Analysis of Variance

Source	Sum-of-Squares	df	Mean-Square	F-ratio	P
Regression	12.00000	1	12.00000	0.00000	0.00000
Residual	0.00000	8	0.00000		
Total	12.00000	9			

The adjusted correlogram show a similar pattern, though the absolute correlations have been reduced. The best decision would still be for a lag of 2 for time, a lag of 3 for distance, and a lag of 1 for bearing. Figure 9.5 shows a graph of the correlogram. *CrimeStat* has a built-in graph function for the correlogram and adjusted correlogram.

Example 1: Prediction

Finally, for prediction, it is apparent that the best method would be to use a regression equation with lags of 2 for time, 3 for distance, and 1 for bearing. Table 9.8 shows the output. As can be seen, the routine predicts exactly the next time and location. The next event for this completely predictable serial offender will be on day 39 at the location with coordinates X=26, Y=26.

Table 9.8
Predicted Results for Serial Offender 1
Regression Equation with
Lags of 2 for Time, 3 for Distance, 1 for Bearing

Variable	Predicted value	From event	Method	Lag
Time interval	2.00000	13	Regression	2
Distance interval	1.41421	13	Regression	3
Bearing interval	44.99997	13	Regression	1
Predicted time .....:	39.00000			
Predicted X coordinate :	26.00000			
Predicted Y coordinate :	26.00000			

The regression equation is the best model in this case. The other methods produce reasonably close approximations, however. Table 9.9 shows the results of using other methods for prediction. As seen, a model where all three components (time, distance, bearing) were lagged by 1 as well as a model where all three components were lagged by 3 also produces the expected correct answer. The mean interval and median interval methods also produce reasonably close, though not exact, answers. In this particular case, the regression method with the best lags produced the optimal solution.

Example 2: Another Completely Predictable Individual

A second example is also a perfectly predictable individual. This time, the directional component changes. The directional trend is northward, but with changes in angle every third event. The time pattern is completely consistent with subsequent events occurring every two days. Table 9.10 presents the pattern and the logical next event while figure 9.6 displays the pattern.

Figure 9.5: **Correlogram of Serial Offender: 1**

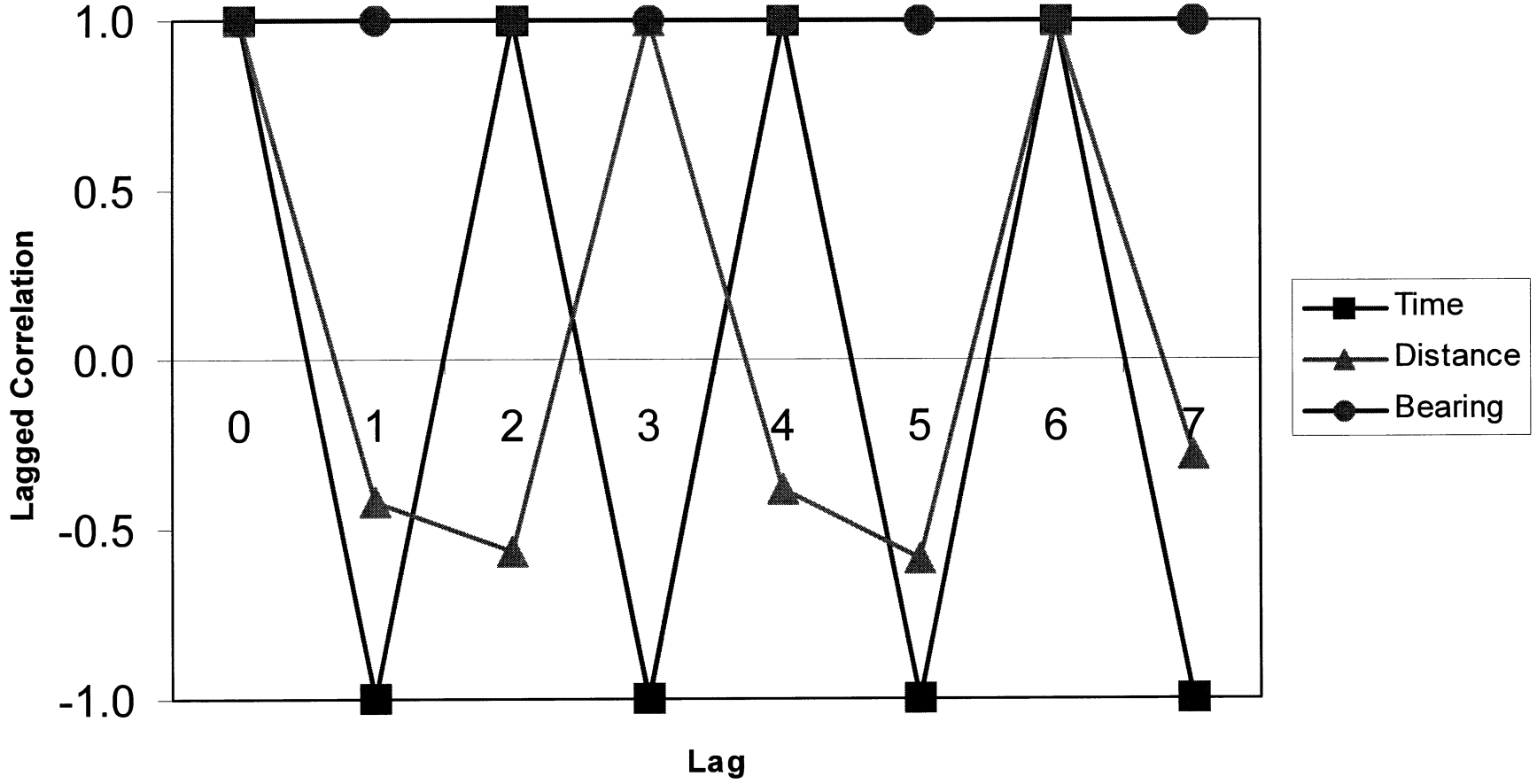


Table 9.9

Comparison of Methods for Predictable Serial Offender 1

	EVENT	X	Y	DISTANCE	DAYS	TIME INTERVAL	DIRECTION
Logical Prediction for next event	**14**	**26**	**26**	**1.4142**	**39**	**2**	**45**
PREDICTION:							
Mean (lag=1)	14	27.0	27.0	2.8	40.0	3.0	45.0
Median (lag=1)	14	27.0	27.0	2.8	41.0	4.0	45.0
Regression:							
Lag=1	14	26.6	26.6	2.3	39.0	2.0	45.0
Lag=2	14	27.0	27.0	2.9	39.0	2.0	45.0
Lag=3	14	26.0	26.0	1.4	39.0	2.0	45.0
Optimal (t=2,d=3,b=1)	14	26.0	26.0	1.4	39.0	2.0	45.0

Table 9.10

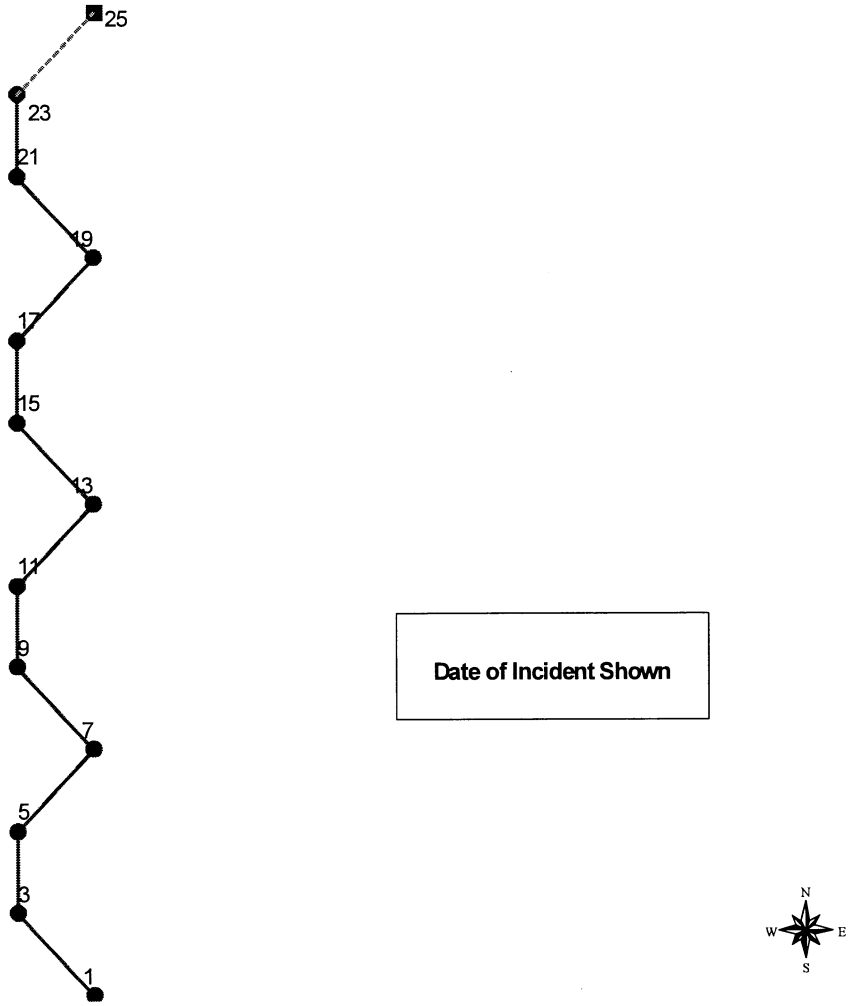
Example of a Predictable Serial Offender: 2

(N = 14 incidents)

Event	X	Y	Distance	Days	Time Interval
1	3	1	-	1	-
2	1	3	2.8284	3	2
3	1	5	2.0000	5	2
4	3	7	2.8284	7	2
5	1	9	2.8284	9	2
6	1	11	2.0000	11	2
7	3	13	2.8284	13	2
8	1	15	2.8284	15	2
9	1	17	2.0000	17	2
10	3	19	2.8284	19	2
11	1	21	2.8284	21	2
12	1	23	2.0000	23	2
Logical prediction for next event	13	3	25	2.8284	25	2

Figure 9.6:

Example of a Predictable Serial Offender: 2 (N=12 Incidents)



The correlogram reveals that both distance and bearing repeat themselves every third event while the time interval is repeated every time. The regression diagnostics show that there is perfect predictability for time and for distance, and high predictability for bearing (not shown). Finally, a regression model is used for prediction with lags of 1 for time, 3 for distance, and 3 for bearing. The model correctly predicts the expected time (days=25) and location (X=3, Y=25). Table 9.11 shows the results.

Methodology for CWA

These two examples illustrate what the CWA routine is doing. There are three steps. First, the sequential pattern is analyzed with the correlogram. This shows which lags have the strongest correlations between lags for time, distance, and bearing separately. Second, the pattern is tested with a regression model. The purpose is to determine how strong a relationship is any particular model. As will be suggested below, if a model is too weak or, conversely, too strong, it most likely will not predict very well. Third, a prediction model is selected. The user can utilize the regression model or use the mean interval or median interval.

Table 9.11
Comparison of Methods for Predictable Serial Offender 2

	EVENT	X	Y	DISTANCE	DAYS	TIME INTERVAL	DIRECTION
Logical Prediction for next event	**13**	**3**	**25**	**2.8284**	**25**	**2**	**45**
PREDICTION:							
Mean (lag=1)	13	2.2	25.2	2.5	25.0	2.0	28.6
Median (lag=1)	13	3.0	25.0	2.8	25.0	2.0	45.0
Regression:							
Lag=1	13	3.0	25.0	2.8	25.0	2.0	45.0
Lag=2	13	1.9	25.2	2.4	25.2	2.0	22.5
Lag=3	13	3.0	25.0	2.8	25.0	2.0	45.0
Optimal (t=1,d=3,b=3)	13	3.0	25.0	2.8	25.0	2.0	45.0

Example 3: A Real Serial Offender

How well does the CWA routine work with real serial offenders? People are not as predictable as these examples; the examples are algorithmic and people don't work like algorithms. But, to the extent to which there is some predictability in human behavior, the CWA routine can be a useful tool for crime analysis, detection, and apprehension.

To illustrate this, a serial offender was identified from a large data set obtained from Baltimore County. The individual committed 16 offenses between 1992 and 1997 when he was eventually apprehended. The profile of crimes committed by this individual were quite diverse. There were 11 larceny incidents (shoplifting and bicycle theft), 1 residential burglary, 1 commercial burglary, 2 assaults, and 1 robbery.

To test the model, the first 15 incidents were used to predict the 16th. This allowed the error between the observed and predicted values for time and location to be used for evaluation. Figure 9.7 shows the sequencing of actions of the first 15 incidents committed by this individual, most of which occurred in the eastern part of Baltimore County.

The correlogram revealed a complicated pattern (figure 9.8). The adjusted matrix was used because of the high correlations at higher-order lags. Nevertheless, the optimal lags appeared to be 1 for time, 3 for distance, and 6 for bearing. A regression model was used to test these parameters. Figure 9.7 also shows the predicted location for the next likely location (the red plus sign) and the location where the individual actually committed the 16th event (green triangle). The error in prediction was good. The distance between the actual and predicted locations was 1.8 miles and the error in predicting the time of the next location was 3.9 days. Overall, the model did quite well for this individual.

Event Sequence as an Analogy to a Correlated Walk

Nevertheless, there are problems in the model for this case. First, this is not a true sequence of actions, but a pseudo-sequence. The individual doesn't go from the first event to the second event to the third event, and so forth. A considerable time may elapse between events. Similarly, distance and direction are conceptual only, not real. For example, in figure 9.7, the individual did not actually travel across the inlets of the Chesapeake Bay as the lines indicate. Distance between the events was actually much greater than estimated by the model and direction was more complex. Nevertheless, to the extent to which an individual makes a spatial decision about where to go, implicitly he or she is making a directional and distance decision. In other words, the decision making process may take into account prior locations. In this case, the CWA routines would be useful.

Example 4: A Second Real Serial Offender

A second real example confirms that the method can produce reasonably close predictions. An offender committed 13 crimes, including three incidents of shoplifting, eight incidents of theft from a vehicle, one residential burglary, and one highway robbery. The correlogram showed that a lag of 1 was strongest for time, distance, and bearing (figure 9.9). The R-squares were moderate (0.45 for time; 0.18 for distance; 0.18 for bearing). Using the regression method with a lag of 1 for each component, the likely location of the next event was predicted (Figure 9.10). The error between the predicted event and the actual event was, again, reasonable with a difference in time of 3.3 days and a difference in distance of 2.4 miles.

Figure 9.7:
Likely Location for Next Crime:
Serial Offender in Baltimore County
N=16 Incidents

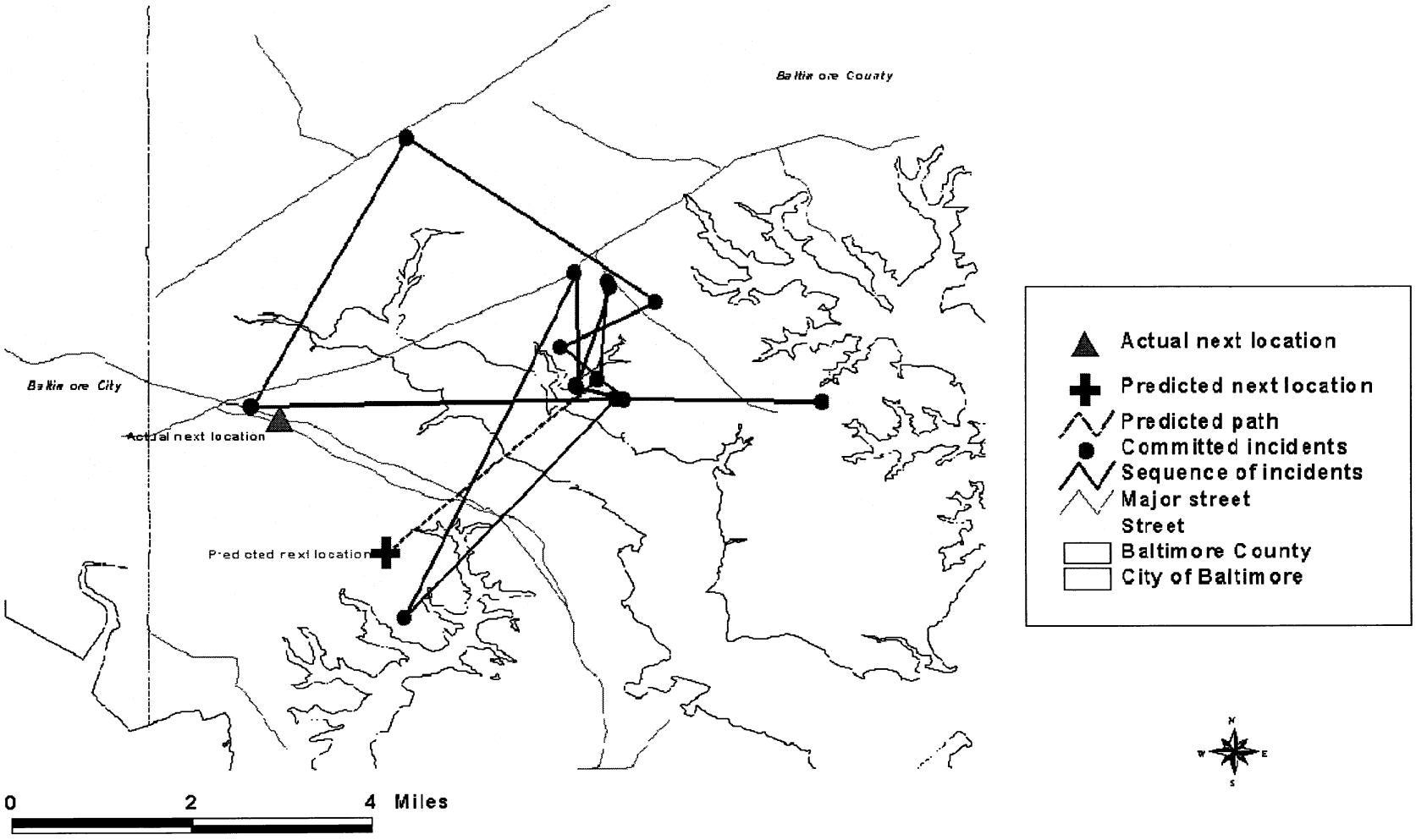


Figure 9.8: Correlogram of Actual Offender

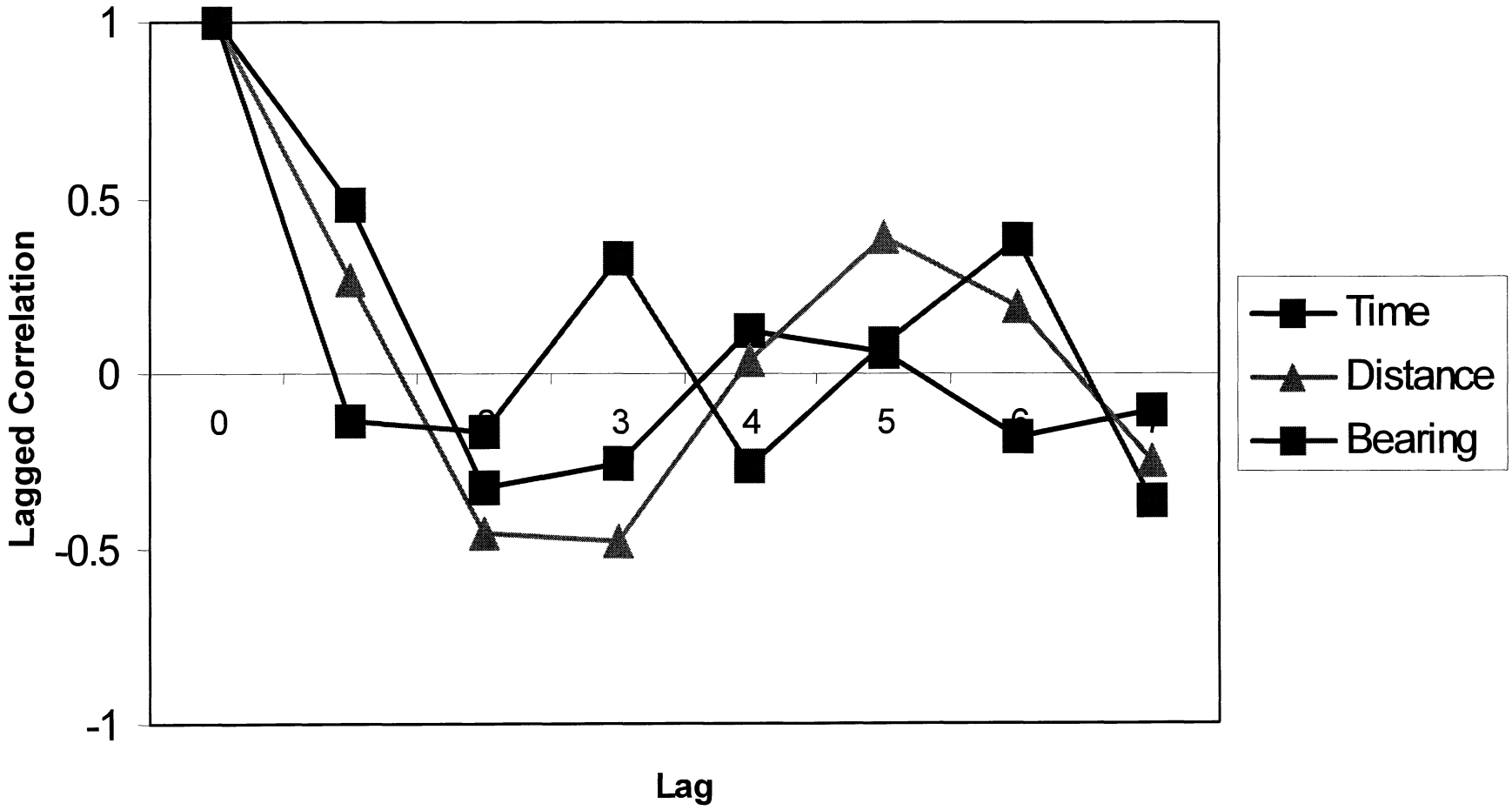


Figure 9.9: Correlogram of Another Offender

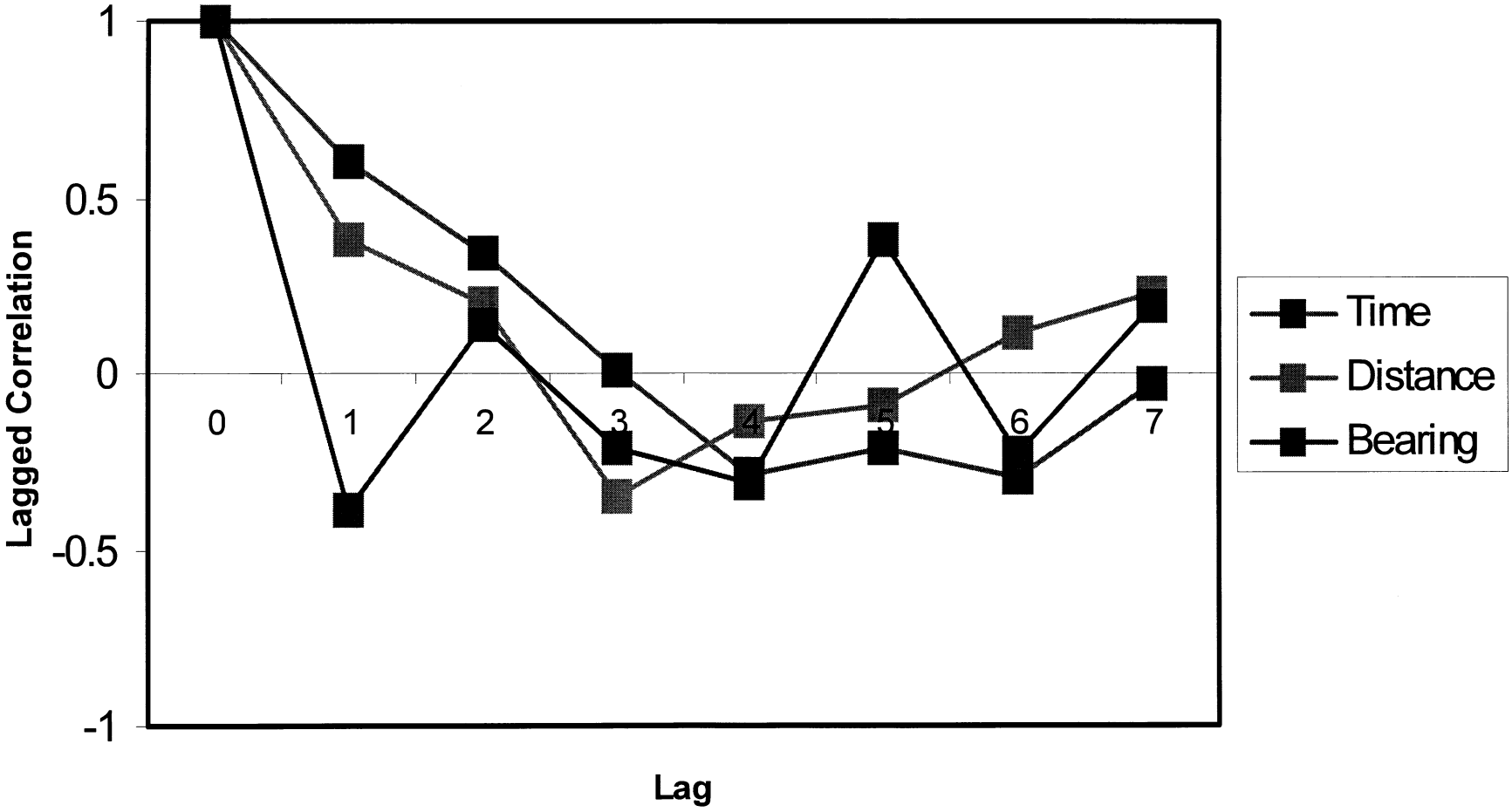
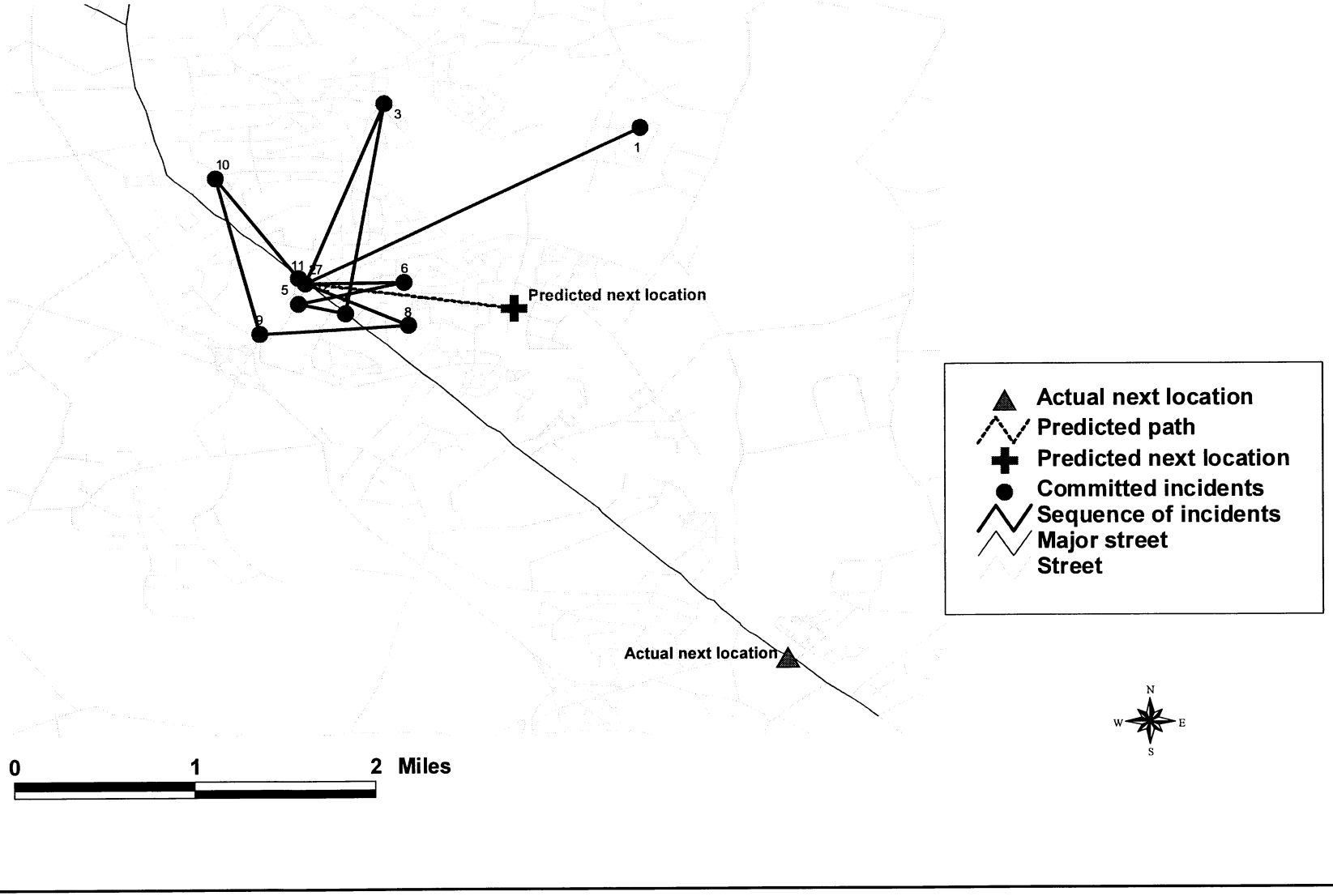


Figure 9.10:

Likely Location for Next Crime: Another Serial Offender in Baltimore County

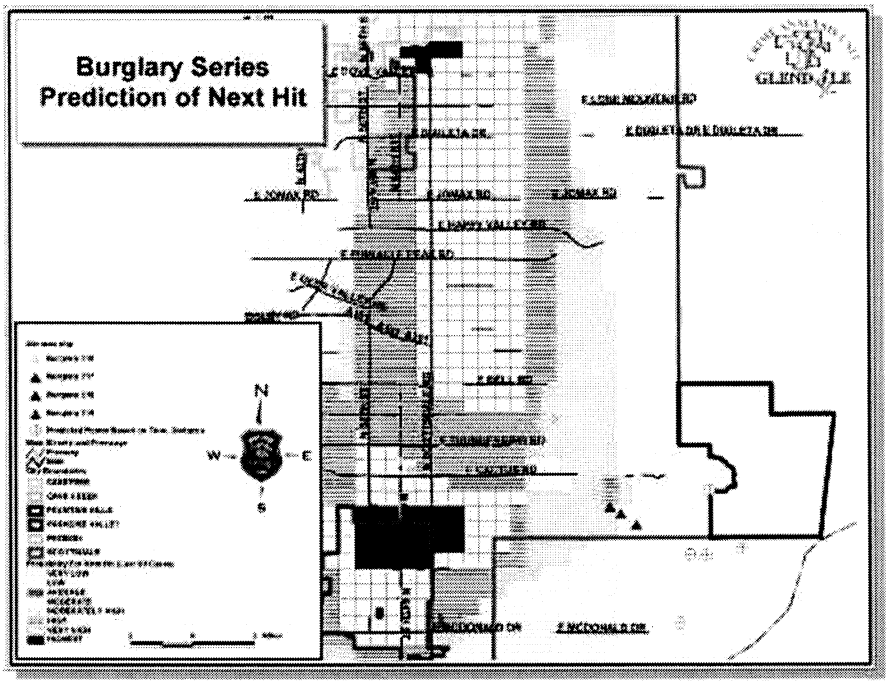


Tracking a Burglary Gang with the Correlated Walk Analysis

Bryan Hill
Glendale Police Department
Glendale, AZ

The space-time analysis tools provided with *CrimeStat II* add an important element to an analyst's review of a tactical prediction effort. Although the method for calculating the Correlated Walk Analysis (CWA) is still more experimental than proven, it allows the analyst to see potential patterns in relation to a suspect's crime travel in terms of time, distance, and direction. In a recent burglary series involving several jurisdictions in our county, the CWA technique was used as part of an aggregate process referred to as the Probability Grid Method. That method combines results from several models to predict the next likely area for a new hit in a crime series. One of the most confusing aspects of these burglaries was the fact that several jurisdictions were involved and the offenders seemed to bounce back and forth from one jurisdiction to the next.

There were also 219 offenses in the series, providing considerable complexity. Because there were so many events, the distances could be anywhere from 0.5 miles to 20 miles, I could never really put my finger on what direction or distance the offender would hit next, but was confident a pattern existed and was likely changing over time. The following map shows the probability grid areas predicted and the CWA points predicted. The triangles shown represent the last four hits. The first hit was near the probability grid prediction in the northern portion of the map; however the subsequent hits were all very close to where the CWA routine predicted they would be. This was also a brand new area for these offenders and was a surprise to the department investigating these incidents. This area was not what was expected based on the SD ellipses and other methods used to predict the next event. The CWA tool requires more testing to determine the accuracy of its predictions, however it may turn out to be a valuable tool in a crime analyst's arsenal.



Accuracy of Predictions

However, it's important not to be overly optimistic about the technique. It is always possible to find cases that fit a method very well. The above mentioned cases appear to do that. Unfortunately, the method is not a magic elixir for predicting serial offenders. Like any method, it has error. It is also a fairly new tool in crime analysis so that we don't have a lot of experience with it. The one example of its use was by Helms (1999), who also is cautious about its utility.

Therefore, at this point, I cannot give conclusive results about whether the method is accurate or not and under what conditions it is best used. It will take some experience to know how effective it is for crime analysis.

To explore the accuracy of the method, 50 serial offenders were identified from a large data base of more than 41,000 incidents in Baltimore County between 1993 and 1997 (see Chapter 10 for details). The 50 offenders were identified based on knowing the dates on which they committed crimes, or at least on which they committed crimes for which they were charged and eventually tried. The number of incidents varied from a low of 7 incidents to a high of 38 incidents. An attempt was made to produce balance in the number of incidents, though the actual distribution of cases did reflect the availability of candidates in the data base. For the fifty individuals, the distribution of incidents was 7 (five individuals), 8 (four individuals), 9 (six individuals), 10 (two individuals), 11 (five individuals), 12 (five individuals), 13 (six individuals), 14 (three individuals), 15 (six individuals), 17 (two individuals), and one individual each for 20, 21, 24, 29 and 38 incidents.

To test the CWA model, the last event committed by these individuals was removed so that N-1 events could be used to predict event N. In this way, it is possible to evaluate the accuracy of the method.

Ten methods were compared:

1. The optimal regression method for time with the lag having the strongest relationship being selected;
2. The optimal regression method for location (distance and bearing) where the with the lags for distance and bearing having the strongest relationship being selected;
3. A regression model for time with a lag of 1;
4. A regression model for location with a lag of 1 (for both distance and bearing);
5. The mean interval for time;
6. The mean interval for location (distance and bearing);
7. The median interval for time;
8. The median interval for location (distance and bearing);
9. The mean center of the incidents (for location only); and
10. The center of minimum distance of the incidents (for location only).

The latter two methods were used for reference. For journey to crime estimation, the center of minimum distance is the best at predicting the origin location of serial offenders (see chapter 10). The reason is because this statistic *minimizes the distance* to all incident locations. The mean center was close behind, though not quite as good. As an estimate, the center of minimum distance is a very good index when there is a single origin that is being predicted. On the other hand, where the purpose is to predict the location of a next event, the center of minimum distance and mean center may be less than useful since they will not generally predict the actual next location. They minimize error, but are rarely accurate. For example, in the above mentioned cases (two theoretical and two real), these statistics did not predict accurately the location of the next event. Instead, they identified a point in the middle of the distribution where the sum of the distances to all incident locations was small.

Error Analysis

Each of the models was compared to the actual time and location of the last, removed incident. For time, the error measure was in days (the absolute difference between the actual day and the predicted day). For location, the error measure was in miles (i.e., absolute distance between the actual and predicted location). The results were mixed. Overall, error was moderate. Table 9.12 summarizes the overall error.

Overall, the center of minimum distance and the mean center do produce, as expected, smaller errors for distance than any of the CWA methods; as noted above, locations in the middle of the distribution of incidents will minimize error, but they won't predict accurately the location of a next event nor indicate in which direction it will occur from the last event. On the other hand, the CWA methods are not particularly accurate, either. They work very well for a completely predictable offender, as was seen in the examples above, but not necessarily for real offenders.

Among the CWA methods, the mean interval, median interval and the lag 1 regression appears to give better results for time than the optimal regression. Overall, the median interval produces the lowest median error, which is about a month and half. In terms of location, the mean interval and median intervals produce slightly better results than the optimal regression, though the lag 1 regression was just as good.

Comparison of CWA Methods

At this point, it is unclear as when it is best to use this technique. Three variables seem to explain part of the error variation. First, a larger sample size leads to better prediction, as would be expected (Table 9.13).

For time, there is definitely an improvement in predictability with larger sample sizes. Among these methods, the mean interval and lag 1 regression show the smallest error for the largest samples (14 cases). For distance, on the other hand, generally, the error increases with increasing sample size. The one exception is for the optimal regression method where medium-sized samples (10-13 cases) produce the lowest error.

Table 9.12

Average and Median Error for CWA Methods
50 Serial Offenders

Method	Average Error	Median Error
Time (days)		
Optimal regression: time	112.2	79.8
Lag 1 regression: time	88.1	70.0
Mean interval: time	89.7	64.9
Median interval: time	91.2	45.5
Distance (miles)		
Optimal regression: location	6.4	5.4
Lag 1 regression: location	5.7	4.2
Mean interval: location	5.8	4.7
Median interval: location	5.3	3.9
Reference Location (miles)		
Mean center	3.3	1.7
Center of minimum distance	3.1	1.2

Variables Affecting Predictability

Long time span

There are a variety of reasons for these strange results, but one reason may be the time span of the events. Some of these offenders committed crimes over a long period, up to five years. Sample size is intrinsically related to the time span (r=0.55). The longer the time span that an offender commits crimes, the more incidents he/she will perpetrate. With increasing time, the individual's behavior patterns may change.

For those offenders with many incidents, a separate analysis was conducted of the events occurring within the last year. Many of these individuals appeared to have moved their base of operation over time, so the isolation of the most recent events was done in order to produce a clearer behavior pattern. The results, while promising, were not dramatic. Accuracy was improved a little compared to using the full sequence, particularly spatial accuracy. However, even with the last few events, these frequently occurred over a long time period (up to two years). Consequently, the idea of isolating a 'clean' set of events did not materialize, at least with these data. On the other hand, with a data set of only recent events, it may be possible to improve predictability.

Table 9.13

Sample Size and Prediction Error
(Average Error)

Time (days)

Sample Size	Optimal Regression	Lag 1 Regression	Mean Interval	Median Interval
6-9	143.4	108.5	116.4	120.8
10-13	108.2	86.8	83.4	79.5
11+	79.8	65.1	65.7	71.2

Distance (miles)

Sample Size	Optimal Regression	Lag 1 Regression	Mean Interval	Median Interval
6-9	7.4	5.2	5.0	4.4
10-13	5.5	6.0	5.7	5.5
11+	6.1	5.9	6.8	6.1

Centrographic: Distance (miles)

Sample Size	Mean Center	Center of Minimum Distance
6-9	2.9	2.4
10-13	2.9	3.1
11+	4.3	4.1

Table 9.14

Regression Diagnostics and Prediction Error
Comparison of CWA Regression Methods

	Time (days)		*Distance (miles)*	
R-Square	**Optimal Regression**	**Lag 1 Regression**	**Optimal Regression**	**Lag 1 Regression**
0-0.29	93.7	90.9	6.7	6.3
0.30-0.59	89.3	33.8	6.0	5.0
0.60+	164.3	122.7	6.3	5.2

Strength of predictability

A second variable that appears to have an effect is the strength of predictability, based on the first N-1 cases. For the diagnostics routine, as the overall R-square for the regression equation increases, the regression equation does better. However, with very high R-square coefficients, the error is worse. Table 9.14 shows the relationship.

The lowest error is obtained with moderate R-square coefficients, for both time and distance. This is why one has to be careful with very high lagged correlations in the correlogram and high R-squares in the diagnostics. Unless one is dealing with a perfectly predictable individual (as the two theoretical examples illustrated), high correlations may be a result of a very small sample size, rather than any inherent predictability.

Limitations of the Technique

In short, users should be careful about using the CWA technique. It can be useful for identifying repeating patterns by an offender, but it won't necessarily predict accurately the offender's next actions. There are a variety of reasons for the lack of predictability. First, there may be intermediate events that are unknown. With each of these offenders in the Baltimore County data base, there is always the possibility that the individuals committed other crimes for which they were not charged. The sequential analysis assumes that all the events are known. But this may not be the case.

A simulation on several cases was conducted by removing events and then re-running the correlogram and prediction models. Removing one event did not appreciably alter the relationship, but removing more than one event did. In other words, if there are unknown events, the true sequential behavior pattern of the offender may not be properly identified. Considering that most offenders commit fewer than 10 incidents before they get caught, the statistical effect of missing information may be critical.

A second reason has been alluded to already. In applying the model to crime events, it is not a true sequential model, but a *pseudo-sequential* model since much time may intervene between events. Distance and direction are conceptual in the sense that the individual doesn't directly orient from one event to the other, but returns to his/her living patterns. Thus, what may appear to be a repeating pattern may not be. Here, the issue of sample size is critical. If there are only a few incidents on which to base an analysis, one could see a pattern which actually doesn't exist. One has to be careful about drawing inferences from very small samples.

A third reason is that people are inherently unpredictable. The two algorithmic examples produced excellent results, but few persons are that systematic about their behavior. Therefore, we must be cautious in expecting too much out of the model.

Conclusion

Nevertheless, the model has utility. First, it can help police identify whether there is a pattern in an offender's behavior. Knowing that there is a pattern can help in planning an arrest strategy. Even if the strategy does not pay off every time, it may improve police effectiveness. In short, the CWA can help a police department analyze the sequential behavior of an offender they are trying to catch. They may be able to anticipate a new event and may be able to warn people who are more likely to be attacked by this individual. If used carefully, the model can be useful for crime analysis and detection.

Second, it can encourage the development of additional predictor tools for individuals. As mentioned above, the center of minimum distance produces a 'best guess' estimate in the sense that it minimizes the distance to the next event. It usually doesn't predict the next event, but it does produce a minimal error. If used in conjunction with the CWA, it may be possible to narrow the search area for the next event.

Third, the CWA model can stimulate research into crime prediction. Police are always trying to predict the next event by an offender and will use multiple techniques and a lot of intuition in trying to 'out-guess' an offender. It is hoped that the CWA model will stimulate more research into predicting the sequence of offender behavior as well into how those sequences aggregate into a large spatial pattern. Most of this text has been devoted to analyzing the spatial patterns of a large number of events. The statistics have, perhaps naively, assumed that each of those events were independent. In reality, they aren't since many crimes are committed by the same individuals. In theory, a distribution of crime incidents could be disaggregated into a distribution of *sequences of events* committed by the same offenders, if we had enough information. Understanding how aggregate distributions is a by-product of the behavior of a limited number of individuals is an important research goal that needs to be addressed.

In the next chapter, we'll look at Journey-to-crime modeling and at the issue of modeling criminal travel behavior.

Endnotes for Chapter 9

1. It would be possible to make a one-tailed test with the simulation. For example, if one is only interested in the degree of clustering, one could adopt the 95 percentile as the threshold. An observed Mantel value that was lower than this threshold would be consistent with the null hypothesis.

2. Henderson, Renshaw and Ford (1981) defined the correlated walk as a two-dimensional walk where the sum of the probabilities in four directions along a lattice are:

$$P = p + q + 2r = 1$$

where P is the total probability (1), p is the probability of continuing in the same direction, q is the probability of moving in an opposite direction, and r is the probability of moving one unit to the right or to the left. The advantage of this formulation is that the probabilities do not have to be equal (i.e., p could exceed q or r). Nevertheless, the individual steps can be considered a special case of a correlated random walk in the plane (Henderson, 1981).

The non-lattice two dimensional case can also be considered a recurrent random walk since a step in any direction (not just along a lattice) can be considered the result of two steps, one in the X direction and one in the Y (or, alternatively, a pairing of all steps in the X direction with all steps in the Y direction). Unfortunately, this logic does not apply to more than two dimensions. Such multi-dimensional walks do not have to return to their origin. However, Spitzer (1963) has shown that an independent walk is recurrent if the second moment around the origin is finite.

Chapter 10
Journey to Crime Estimation

The *Journey to Crime* (Jtc) routine is a distance-based method which makes estimates about the likely residential location of a serial offender. It is an application of *location theory*, a framework for identifying optimal locations from a distribution of markets, supply characteristics, prices, and events. The following discussion gives some background to the technique. Those wishing to skip this part can go to page 10-19 for the specifics of the Jtc routine.

Location Theory

Location theory is concerned with one of the central issues in geography. This theory attempts to find an optimal location for any particular distribution of activities, population, or events over a region (Haggett, Cliff and Frey, 1977; Krueckeberg and Silvers, 1974; Stopher and Meyburg, 1975; Oppenheim, 1980, Ch. 4; Bossard, 1993). In classic location theory, economic resources were allocated in relation to idealized representations (Anselin and Madden, 1990). Thus, von Thünen (1826) analyzed the distribution of agricultural land as a function of the accessibility to a single population center (which would be more expensive towards the center), the value of the product produced (which would vary by crop), and transportation costs (which would be more expensive farther from the center). In order to maximize profit and minimize costs, a distribution of agricultural land uses (or crop areas) emerges flowing out from the population center as a series of concentric rings. Weber (1909) analyzed the distribution of industrial locations as a function of the volume of materials to be shipped, the distance that the goods had to be shipped, and the unit distance cost of shipping; consequently, industries become located in particular concentric zones around a central city. Burgess (1925) analyzed the distribution of urban land uses in Chicago and described concentric zones of both industrial and residential uses. Their theory formed the backdrop for early studies on the ecology of criminal behavior and gangs (Thrasher, 1927; Shaw, 1929).

In more modern use, the location of persons with a certain need or behavior (the 'demand' side) is identified on a spatial plane and places are selected as to maximize value and minimize travel costs. For example, for a consumer faced with two retail shops selling the same product, one being closer but more expensive while the other being farther but less expensive, the consumer has to trade off the value to be gained against the increased travel time required. In designing facilities or places of attraction (the 'supply' side), the distance between each possible facility location and the location of the relevant population is compared to the cost of locating near the facility. For example, given a distribution of consumers and their propensity to spend, such a theory attempts to locate the optimal placement of retail stores, or, given the distribution of patients, the theory attempts to locate the optimal placement of medical facilities.

Predicting Locations from a Distribution

One can also reverse the logic. Given the distribution of demand, the theory could be applied to estimate a central location from which travel distance or time is minimized. One of the earliest uses of this logic was that of John Snow, who was interested in the causes of cholera in the mid-19th century (Cliff and Haggett, 1988). He postulated the theory that water was the major vector transmitting the cholera bacteria. After investigating water sources in the London metropolitan area and concluding that there was a relationship between contaminated water and cholera cases, he was able to confirm his theory by an outbreak of cholera cases in the Soho district. By plotting the distribution of the cases and looking for water sources in the center of the distribution (essentially, the center of minimum distance; see chapter 4), he found a well on Broad Street that was, in fact, contaminated by seepage from nearby sewers. The well was closed and the epidemic in Soho receded. Incidently, in plotting the incidents on a map and looking for the center of the distribution, Snow applied the same logic that had been followed by the London Metropolitan Police Department who had developed the famous 'pin' map in the 1820s.

Theoretically, there is an optimal solution that minimizes the distance between demand and supply (Rushton, 1979). However, computationally, it is an almost impossible task to define, requiring the enumeration of every possible combination. Consequently in practice, approximate, though sub-optimal, solutions are obtained through a variety of methods (Everett, 1974, Ch. 4).

Travel Demand Modeling

A sub-set of location theory models the travel behavior of individuals. It actually is the converse. If location theory attempts to allocate places or sites in relation to both a supply-side and demand-side, travel demand theory attempts to model how individuals travel between places, given a particular constellation of them. One concept that has been frequently used for this purpose is that of the *gravity function*, an application of Newton's fundamental law of attraction (Oppenheim, 1980). In the original Newtonian formulation, the attraction, F, between two bodies of respective masses M_1 and M_2, separated by a distance D, will be equal to

$$F = g \frac{M_1 M_2}{D^2} \qquad (10.1)$$

where g is a constant or scaling factor which ensures that the equation is balanced in terms of the measurement units (Oppenheim, 1980). As we all know, of course, g is the gravitational constant in the Newtonian formulation. The numerator of the function is the *attraction* term (or, alternatively, the attraction of M_2 for M_1) while the denominator of the equation, d^2, indicates that the attraction between the two bodies falls off as a function of their *squared* distance. It is an *impedance* term.

Social Applications of the Gravity Concept

The gravity model has been the basis of many applications to human societies and has been applied to social interactions since the 19th century. Ravenstein (1895) and Andersson (1897) applied the concept to the analysis of migration by arguing that the tendency to migrate between regions is inversely proportional to the squared distance between the regions. Reilly's 'law of retail gravitation' (1929) applied the Newtonian gravity model directly and suggested that retail travel between two centers would be proportional to the product of their populations and inversely proportional to the square of the distance separating them:

$$T_{ij} = \alpha \frac{P_i P_j}{D_{ij}^2} \qquad (10.2)$$

where T_{ij} is the interaction between centers *i* and *j*, P_i and P_j are the respective populations, D_{ij} is the distance between them raised to the second power and α is a balancing constant. In the model, the initial population, P_i, is called a *production* while the second population, P_j, is called an *attraction*.

Stewart (1950) and Zipf (1949) applied the concept to a wide variety of phenomena (migration, freight traffic, exchange of information) using a simplified form of the gravity equation

$$T_{ij} = \alpha \frac{P_i P_j}{D_{ij}} \qquad (10.3)$$

where the terms are as in equation 10.2 but the exponent of distance is only 1. In doing so, they basically linked location theory with travel behavior theory. Given a particular pattern of interaction for any type of goods, service or human activity, an optimal location of facilities should be solvable.

In the Stewart/Zipf framework, the two P's were both population sizes and, therefore, their sums had to be equal. However, in modern use, it's not necessary for the productions and attractions to be identical units (e.g., P_i could be population while P_j could be employment).

The total volume of productions (trips) from a single location, i, is estimated by summing over all destination locations, j:

$$T_i = K \quad P_i \sum_j (P_j/D_{ij}) \qquad (10.4)$$

Over time, the concept has been generalized and applied to many different types of travel behavior. For example, Huff (1963) applied the concept to retail trade between zones in an urban area using the general form of

$$T_{ij} = \alpha \frac{A_j^{\beta}}{D_{ij}^{\lambda}} \tag{10.5}$$

where T_{ij} is the number of purchases in location *j* by residents of location *i*, A_j is the attractiveness of zone j (e.g., square footage of retail space), D_{ij} is the distance between zones *i* and *j*, β is the exponent of S_j, and λ is the exponent of distance, and α is a constant (Bossard, 1993). $D_{ij}^{-\lambda}$ is sometimes called an *inverse distance* function. This is a *single constraint* model in that only the attractiveness of a commercial zone is constrained, that is the sum of all attractions for j must equal the total attraction in the region.

Again, it can be generalized to all zones by, first, estimating the total trips generated from one zone, i, to another zone, j,

$$T_{ij} = \alpha \frac{P_i^{\rho} A_j^{\beta}}{D_{ij}^{\lambda}} \tag{10.6}$$

where T_{ij} is the interaction between two locations (or zones), P_i is productions of trips from location/zone i, A_j is the attractiveness of location/zone j, D_{ij} is the distance between zones i and j, β is the exponent of S_j, ρ is the exponent of H_i, λ is the exponent of distance, and α is a constant.

Second, the total number of trips generated by a location, i, to all destinations is obtained by summing over all destination locations, j:

$$T_i = \alpha P_i^{\rho} \sum_j (A_j^{\beta}/D_{ij}^{\lambda}) \tag{10.7}$$

This differs from the traditional gravity function by allowing the exponents of the production from location i, the attraction from location j, and the distance between zones to vary. Typically, these exponents are calibrated on a known sample before being applied to a forecast sample and the locations are usually measured by zones. Thus, retailers in deciding on the location of a new store can use this type of model to choose a site location to optimize travel behavior of patrons; they will, typically, obtain data on actual shopping trips by customers and then calibrate the model on the data, estimating the exponents of attraction and distance. The model can then be used to predict future shopping trips if a facility is built at a particular location.

This type of function is called a *double constraint* model because the balancing constant, K, has to be constrained by the number of units in both the origin and destination locations; that is, the sum of P_i over all locations must be equal to the total number of productions while the sum of P_2 over all locations must be equal to the total number of attractions. Adjustments are usually required to have the sum of individual productions and attractions equal the totals (usually estimated independently).

The equation can be generalized to other types of trips and different metrics can be substituted for distance, such as travel time, effort, or cost (Isard, 1960). For example, for commuting trips, usually employment is used for attractions, frequently sub-divided into retail and non-retail employment. In addition, for productions, median household income or car ownership percentage is used as an additional production variable. Equation 10.7 can be generalized to include any type of production or attraction variable (10.8 and 10.9):

$$T_{ij} = \alpha_1 P_i^{\rho} \alpha_2 A_j^{\beta}/D_{ij}^{\lambda} \quad (10.8)$$

$$T_i = \alpha_1 P_i^{\rho} \Sigma (\alpha_2 A_j^{\beta}/D_{ij}^{\lambda}) \quad (10.9)$$

where T_{ij} is the number of trips produced by location i that travel to location j, P_i is either a single variable associated with trips produced from a zone or the cross-product of two or more variables associated with trips produced from a zone, A_j is either a single variable associated with trips attracted to a zone or the cross-product of two or more variables associated with trips attracted to a zone, D_{ij} is either the distance between two locations or another variable measuring travel effort (e.g., travel time, travel cost), ρ, β, and λ are exponents of the respective terms, α_1 is a constant associated with the productions to ensure that the sum of trips produced by all zones equals the total number of trips for the region (usually estimated independently), and α_2 is a constant associated with the attractions to ensure that the sum of trips attracted to all zones equals the total number of trips for the region. Without having two constants in the equation, usually conflicting estimates of K will be obtained by balancing the equation against productions or attractions. The summation over all destination locations, j (equation 10.9), produces the total number of trips from zone i.

Intervening Opportunities

Stouffer (1940) modified the simple gravity function by arguing that the attraction between two locations was a function not only of the characteristics of the relative attractions of two locations, but of intervening opportunities between the locations. His hypothesis "..assumes that there is no necessary relationship between mobility and distance... that the number of persons going a given distance is directly proportional to the number of opportunities at that distance and inversely proportional to the number of intervening opportunities"(Stouffer, 1940, p. 846). This model was used in the 1940s to explain interstate and intercounty migration (Bright and Thomas, 1941; Isbell, 1944; Isard, 1979). Using the gravity type formulation, we can write this as:

$$T_{ji} = \alpha \frac{A_j^{\beta}}{\Sigma(A_k^{\xi})\ D_{ij}^{\lambda}} \quad (10.10)$$

where T_{ji} is the attraction of location j by residents of location i, A_j is the attractiveness of zone j, A_k is the attractiveness of all other locations that are *intermediate* in distance between locations i and j, D_{ij} is the distance between zones *i* and *j*, β is the exponent of S_j, ξ is the exponent of S_k, λ is the exponent of distance, and α is a constant. While the

intervening opportunities are implicit in equation 10.5 in the exponents, β and λ, and coefficient, K, equation 10.10 makes the intervening opportunities explicit. The importance of the concept is that the interaction between two locations becomes a complex function of the spatial environment of nearby areas and not just of the two locations.

Urban Transportation Modeling

This type of model is incorporated as a formal step in the urban transportation planning process, implemented by most regional planning organizations in the United States and elsewhere (Stopher and Meyburg, 1975; Krueckeberg and Silvers, 1974; Field and MacGregor, 1987). The step, called *trip distribution*, is linked to a five step model. First, data are obtained on travel behavior for a variety of trip purposes. This is usually done by sampling households and asking each member to keep a travel diary documenting all their trips over a two or three day period. Trips are aggregated by individuals and by households. Frequently, trips by different purposes are separated. Second, the volume of trips produced by and attracted to zones (called traffic analysis zones) is estimated, usually on the basis of the number of households in the zone and some indicator of income or private vehicle ownership. Third, trips produced by each zone are distributed to every other zone usually using a gravity-type function (equation 10.9). That is, the number of trips produced by each origin zone and ending in each destination zone is estimated by a gravity model. The distribution is based on trip productions, trip attractions, and travel 'resistance' (measured by travel distance or travel time). Fourth, zone-to-zone trips are allocated by mode of travel (car, bus, walking, etc); and, fifth, trips are assigned to particular routes by travel mode (i.e., bus trips follow different routes than private vehicle trips). The advantage of this process is that trips are allocated according to origins, destinations, distances (or travel times), modes of travel and routes. Since all zones are modeled simultaneously, all intermediate destinations (i.e., intervening opportunities) are incorporated into the model. Chapters 11-17 present a crime travel demand model.

Alternative Distance Decay Functions

One of the problems with the traditional gravity formulation is in the measurement of travel resistance, either distance or time. For locations separated by sizeable distances in space, the gravity formulation can work properly. However, as the distance between locations decreases, the denominator approaches infinity. Consequently, an alternative expression for the interaction has been proposed which uses the negative exponential function (Hägerstrand, 1957; Wilson, 1970).

$$A_{ji} = S_j^{\beta} e^{(-\alpha D ij)} \qquad (10.11)$$

where A_{ji} is the attraction of location j for residents of location i, S_j is the attractiveness of location j, D_{ij} is the distance between locations *i* and *j*, β is the exponent of S_j, *e* is the base of the natural logarithm (i.e., 2.7183...), and α is an empirically-derived exponent. Sometimes known as *entropy maximization*, the latter part of the equation includes a negative exponential function which has a maximum value of 1 (i.e., $e^{-0} = 1$). This has the

advantage of making the equation more stable for interactions between locations that are close together. For example, Cliff and Haggett (1988) used a negative exponential gravity-type model to describe the diffusion of measles into the United States from Canada and Mexico. It has also been argued that the negative exponential function generally gives a better fit to urban travel patterns, particularly by automobile (Foot, 1981; Bossard, 1993; NCHRP, 1995).

Other functions have also be used to describe the distance decay - negative linear, normal distribution, lognormal distribution, quadratic, Pareto function, square root exponential, and so forth (Haggett and Arnold, 1965; Taylor, 1970; Eldridge and Jones, 1991). Later in the chapter, we will explore several different mathematical formulations for describing the distance decay. One, in fact, does not need to use a mathematical function at all, but could empirically describe the distance decay from a large data set and utilize the described values for predictions. The use of mathematical functions has evolved out of both the Newtonian tradition of gravity as well as various location theories which used the gravity function. A mathematical function makes sense under two conditions: 1) if travel is uniform in all directions; and 2) as an approximation if there is inadequate data from which to calibrate an empirical function. The first assumption is usually wrong since physical geography (i.e., oceans, rivers, mountains) as well as asymmetrical street networks make travel easier in some directions than others. As we shall see below, the distance decay is quite irregular for journey to crime trips and would be better described by an empirical, rather than mathematical function.

In short, there is a long history of research on both the location of places as well as the likelihood of interaction between these places, whether the interaction is freight movement, land prices or individual travel behavior. The gravity model and variations on it have been used to describe the interactions between these locations.

Travel Behavior of Criminals

Journey to Crime Trips

The application of travel behavior theory to crime has a sizeable history as well. The analysis of distance for journey to crime trips was applied in the 1930s by White (1932), who noted that property crime offenders generally traveled farther distances than offenders committing crimes against people, and by Lottier (1938), who analyzed the ratio of chain store burglaries to the number of chain stores by zone in Detroit. Turner (1969) analyzed delinquency behavior by a distance decay travel function showing how more crime trips tend to be close to the offender's home with the frequency dropping off with distance. Phillips (1980) is, apparently, the first to use the term *journey to crime* is describing the travel distances that offenders make though Harries (1980) noted that the average distance traveled has evolved by that time into an analogy with the journey to work statistic.

Rhodes and Conly (1981) expanded on the concept of a *criminal commute* and showed how robbery, burglary and rape patterns in the District of Columbia followed a

distance decay pattern. LeBeau (1987a) analyzed travel distances of rape offenders in San Diego by victim-offender relationships and by method of approach. Boggs (1965) applied the intervening opportunities model in analyzing the distribution of crimes by area in relation to the distribution of offenders. Other empirical descriptions of journey to crime distances and other travel behavior parameters have been studied by Blumin (1973), Curtis (1974), Repetto (1974), Pyle (1974), Capone and Nichols 1975), Rengert (1975), Smith (1976), LeBeau (1987b), and Canter and Larkin (1993). It has generally been accepted that property crime trips are longer than personal crime trips (LeBeau, 1987a), though exceptions have been noted (Turner, 1969). Also, it would be expected that average trip distances will vary by a number of factors: crime type; method of operation; time of day; and, even, the value of the property realized (Capone and Nichols, 1975).

Modeling the Offender Search Area

Conceptual work on the type of model have been made by Brantingham and Brantingham (1981) who analyzed the *geometry of crime* and conceptualized a criminal search area, a geographical area modified by the spatial distribution of potential offenders and potential targets, the awareness spaces of potential offenders, and the exchange of information between potential offenders. In this sense, their formulation is similar to that of Stouffer (1940), who described intervening opportunities, though their's is a behavioral framework. An important concept developed by the Brantingham's is that of decreased criminal activity near to an offender's home base, a sort of a safety area around their near neighborhood. Presumably, offenders, particularly those committing property crimes, go a little way from their home base so as to decrease the likelihood that they will get caught. This was noted by Turner (1969) in his study of delinquency in Philadelphia. Thus, the Brantingham's postulated that there would be a small safety area (or 'buffer' zone) of relatively little offender activity near to the offender's base location; beyond that zone, however, they postulated that the number of crime trips would decrease according to a distance decay model (the exact mathematical form was never specified, however).

Crime trips may not even begin at an offender's residence. Routine activity theory (Cohen and Felson, 1979; 1981) suggests that crime opportunities appear in the activities of everyday life. The routine patterns of work, shopping, and leisure affect the convergence in time and place of would be offenders, suitable targets, and absence of guardians. Many crimes may occur while an offender is traveling from one activity to another. Thus, modeling crime trips as if they are referenced relative to a residence is not necessarily going to lead to better prediction.

The mathematics of journey to crime has been modeled by Rengert (1981) using a modified general opportunities model:

$$P_{ij} = K\, U_i\, V_j\, f(D_{ij}) \qquad (10.12)$$

where P_{ij} is the probability of an offender in location (or zone) i committing an offense at location j, U_i is a measure of the number of crime trips produced at location i (what Rengert called *emissiveness*), V_j is a measure of the number of crime targets (attractiveness) at

location j, and $f(D_{ij})$ is an unspecified function of the cost or effort expended in traveling from location i to location j (distance, time, cost). He did not try to operationalize either the production side or the attraction side. Nevertheless, conceptually, a crime trip would be expected to involve both elements as well as the cost of the trip.

In short, there has been a great deal of research on the travel behavior of criminals in committing acts as well as a number of statistical formulations.

Predicting the Location of Serial Offenders

The journey to crime formulation, as in equation 10.9, has been used to estimate the origin location of a serial offender based on the distribution of crime incidents. The logic is to plot the distribution of the incidents and then use a property of that distribution to estimate a likely origin location for the offender. Inspecting a pattern of crimes for a central location is an intuitive idea that police departments have used for a long time. The distribution of incidents describes an activity area by an offender, who lives somewhere in the center of the distribution. It is a *sample* from the offender's activity space. Using the Brantingham's terminology, there is a search area by an offender within which the crimes are committed; most likely, the offender also lives within the search area.

For example, Canter (1994) shows how the area defined by the distribution of the 'Jack the Ripper' murders in the east end of London in the 1880s included the key suspects in the case (though the case was never solved). Kind (1987) analyzed the incident locations of the 'Yorkshire Ripper' who committed thirteen murders and seven attempted murders in northeast England in the late 1970s and early 1980s. Kind applied two different geographical criteria to estimate the residential location of the offender. First, he estimated the center of minimum distance. Second, on the assumption that the locations of the murders and attempted murders that were committed late at night were closer to the offender's residence, he graphed the time of the offense on the Y axis against the month of the year (taken as a proxy for length of day) on the X axis and plotted a trend line through the data to account for seasonality. Both the center of minimum distance and the murders committed at a later time than the trend line pointed towards the Leeds/Bradford area, very close to where the offender actually lived (in Bradford).

Rossmo Model

Rossmo (1993; 1995) has adapted location theory, particularly travel behavior modeling, to serial offenders. In a series of papers (Rossmo, 1993a; 1993b; 1995; 1997) he outlined a mathematical approach to identifying the home base location of a serial offender, given the distribution of the incidents. The mathematics represent a formulation of the Brantingham and Brantingham (1981) search area model, discussed above in which the search behavior of an offender is seen as following a distance decay function with decreased activity near the offender's home base. He has produced examples showing how the model can be applied to serial offenders (Rossmo, 1993a; 1993b; 1997).

The model has four steps (what he called *criminal geographic targeting*):

1. First, a rectangular study area is defined that extends beyond the area of the incidents committed by the serial offender. The average distance between points is taken in both the Y and X direction. Half the Y inter-point distance is added to the maximum Y value and subtracted from the minimum Y value. Half the X inter-point distance is added to the maximum X value and subtracted from the minimum X value. These are based on projected coordinates; presumably, the directions would have to be adjusted if spherical coordinates were used. The rectangular study defines a grid from which columns and rows can be defined.

2. For each grid cell, the Manhattan distance to each incident location is taken (see chapter 3 for definition).

3. For each Manhattan distance from a grid cell to an incident location, MD_{ij}, one of two functions is evaluated:

   A. If the Manhattan distance, MD_{ij}, is less than a specified buffer zone radius, B, then

$$P_{ij} = \prod_{c=1}^{T} \{k[(1-\phi)(B^{g-f}) / (2B - | x_i - x_c| + | y_i - y_c|)^g] \} \qquad (10.13)$$

   where P_{ij} is the resultant of offender interaction for grid cell, i; c is the incident number, summing to T; $\phi = 0$; k is an empirically determined constant; g is an empirically determined exponent; and f is an empirically determined exponent.

   The Greek letter, Π, is the product sign, indicating that the results for each grid cell-incident distance, MD_{ij}, are *multiplied* together across all incidents, c. This equation reduces to

$$P_{ij} = \prod_{c=1}^{T} \{k(1-0)(B^{g-f}) / (2B - | x_i - x_c| + | y_i - y_c|)^g \} \qquad (10.14)$$

$$P_{ij} = \prod_{c=1}^{T} \frac{KB^{g-f}}{(2B - | x_i - x_c| + | y_i - y_c|)^g} \qquad (10.15)$$

   Within the buffer region, the function is the ratio of a constant, k, times the radius of the buffer, B, raised to another constant (g-f),

divided by the difference between the diameter of the circle (2B) and the Manhattan distance, MD_{ij}, raised to a constant, g. This is a *non-linear* function.

B. If the Manhattan distance, MD_{ij}, is greater than a specified buffer zone radius, B, then

$$P_{ij} = \prod_{c=1}^{T} \{k [\phi / (| x_i - x_c| + | y_i - y_c|)^f \} \quad (10.16)$$

where P_{ij} is the resultant of offender interaction for grid cell, i, and incident location, j; *c* is the incident number, summing to T; $\phi = 1$; *k* is an empirically determined constant (the same as in equation 10.15 above); and *f* is an empirically determined exponent (the same as in equation 10.15 above).

Again, the Greek letter, Π, indicates that the results for each grid cell-incident distance, MD_{ij}, are multiplied together across all incidents, c. This equation reduces to

$$P_{ij} = \prod_{c=1}^{T} \{k [1/(| x_i - x_c| + | y_i - y_c|)^f \} \quad (10.17)$$

$$P_{ij} = \prod_{c=1}^{T} \left\{ \frac{k}{(| x_i - x_c| + | y_i - y_c|)^f} \right\} \quad (10.18)$$

Outside of the buffer region, the function is a constant, k, divided by the distance, MD_{ij}, raised to an exponent, f. It is an inverse distance function and drops off rapidly with distance

4. Finally, for each grid cell, i, the functions evaluated in step 3 above are summed over all incidents.

For both the 'within buffer zone' (near to home base) and 'outside buffer zone' (far from home base) functions, the coefficient, k, and exponents, f and g, are empirically determined. Though he doesn't discuss how these are calculated, they are presumably estimated from a sample of known offender locations where the distance to each incident is known (e.g., arrest records).

The result is a surface model indicating a likelihood of the offender residing at that location. He describes it as a probability surface, but it is actually a *density* surface. Since the probability of interaction between any one grid cell, i, and any one incident, j, cannot be greater than 1, the surface actually indicates the product of individual likelihoods that the

offender uses that location as the home base. To be an actual probability function, it would have to be re-scaled so that the sum of the grid cells was equal to 1.

The second function - 'outside the buffer zone' (equation 10.16) is a classic gravity function, similar to equation 10.5 except there is no attraction definition. It is the distance decay part of the gravity function. The first function, equation 10.13, is an increasing curvilinear function designed to model the area of decreased activity near the offender's home base.

Strengths and weaknesses of the Rossmo model

The Rossmo model has both strengths and weaknesses. First, the model has some theoretical basis utilizing the Brantingham and Brantingham (1981) framework for an offender search area as well as the mathematics of the gravity model and distinguishes two types of travel behavior - near to home and farther from home. Second, the model does represent a systematic approach towards identifying a likely home base location for an offender. By evaluating each grid cell in the study area, an independent estimate of the likelihood is obtained, which can then be integrated into a continuous surface with an interpolation graphics routine.

There are problems with the particular formulation, however. First, the exclusive use of Manhattan distances is questionable. Unless the study area has a street network that follows a uniform grid, measuring distances horizontally and vertically can lead to overestimation of travel distances; further, the more the layout differs from a north-south and east-west orientation, the greater the distortion. Since many urban areas do not have a uniform grid street layout, the method will necessarily lead to overestimation of travel distances in places where there are diagonal or irregular streets.[1]

Second, the use of a product term, Π, complicates the mathematics. That is, the technique evaluates the distance from a particular grid cell, i, to a particular incident location, j. It then *multiplies* this result by all other results. Since the P values are actually densities, which can be greater than 1.0, the process, if strictly applied, would be a compounding of probabilities with overestimation of the likelihood for grid cells close to incident locations and underestimation of the likelihood for grid cells farther away. In the description of the method, however, Rossmo actually mentions summing the terms. Thus, the substitution of a summation sign, Σ, for the product sign would help the mathematics.

A third problem is in the distance decay function (equation 10.16). The use of an inverse distance term has problems as the distance between the grid cell location, i, and the incident location, j, decreases. For some types of crimes, there will be little or no buffer zone around the offender's home base (e.g., rapes by acquaintances). Consequently, the buffer zone radius, B, would approach 0. However, this would cause the model to become unstable since the inverse distance term will approach infinity.

Fourth, the use of a mathematical function to describe the distance decay, while easy to define, probably oversimplifies actual travel behavior. A mathematical function to

describe distance decay is an approximation to actual travel behavior. It assumes that travel is equally likely in each direction, that travel distance is uniformly easy (or difficult) in each direction, and that, similarly, opportunities are uniformly distributed. For most urban areas, these conditions would not be true. Few cities form a perfect grid (Salt Lake City is, of course, an exception), though most cities have sections that are grided. Both physical geography limit travel in certain directions as does the historical street structure, which is often derived from earlier communities. A mathematical function does not consider this structure, but rather assumes that the 'impedance' in all directions is uniform.

This latter criticism, of course, would be true for all mathematical formulations of travel distance. There are corrections that can be made to adjust for this. For example, in the urban travel demand type model, trip distribution between locations is estimated by a gravity model, but then the distributed trips are constrained by, first, the total number of trips in the region (estimated separately), second, by mode of travel (bus v. single driver v. drivers plus passengers v. walk, etc.), and, third, by the route structure upon which the trips are eventually assigned (Krueckeberg and Silvers, 1974; Stopher and Meyburg, 1975; Field and MacGregor, 1987). Calibration at all stages against known data sets ensures that the coefficients and exponents fit 'real world' data as closely as possible. It would take these types of modifications to make the travel distribution type of model postulated by Rossmo and others be more realistic.

Fifth, the model imposes mathematical rigidity on the data. While there are two different functions that could vary from place to place, the particular type of distance decay function might also vary. Specifying a strict form for the two equations limits the flexibility of applying the model to different types of crime or to places where the distance decay does not follow the form specified by Rossmo.

A sixth problem is that opportunities for committing crimes - the attractiveness of locations, are never measured. That is, there is no enumeration of the opportunities that would exist for an offender nor is there an attempt to measure the strength of this attraction. Instead, the search area is inferred strictly from the distribution of incidents. Because the distribution of offender opportunities would be expected to vary from place to place, the model would need to be re-calibrated at each location. In this sense, both the Canter model and my journey to crime model (both described below) also share this weakness. It is understandable in that victim/target opportunities are difficult to define *a priori* since they can be interpreted differently by individuals. Nevertheless, a more complete theory of journey to crime behavior would have to incorporate some measure of opportunities, a point that both Brantingham and Brantingham (1981) and Rengert (1981) have made.

Finally, the 'buffer zone' concept is but one interpretation of the tendency of many crimes not to be committed close to the home location. There are other interpretations that are applicable. For example, the distribution of crime opportunities is often not close to the home location, either. Many crimes occur in commercial areas. In most American cities, residential areas are not located in commercial areas. Thus, there will usually be a

distance between a residential location and a nearby crime opportunity. This does not imply anything about a 'safety zone' for the offender but, instead, may illustrate the distribution of the opportunities. If we could map the travel distance of, say, shopping trips, we would probably find a similar distribution to that seen in most of journey to crime studies (and illustrated below).

The concept of a 'buffer zone' is a hypothesis, not a certainty. The language of it is so appealing that many people believe it to be true. But, to demonstrate the existence of a 'buffer zone' would require interviewing offenders (or offenders who have been arrested) and demonstrating that they did not commit crimes near their residence even though there were opportunities (i.e., they valued safety over opportunity). To my knowledge, there has not been a study that demonstrated this yet. Otherwise, one cannot distinguish between the 'buffer zone' hypothesis and the distribution of available opportunities. They may very well be the same thing.

Canter Model

Canter's group in Liverpool (Canter and Tagg, 1975; Canter and Larkin, 1993; Canter and Snook, 1999; Canter, Coffey and Huntley, 2000) have modified the distance decay function for journey to crime trips by using a negative exponential term, instead of the inverse distance. Their *Dragnet* program uses the negative exponential function

$$Y = \alpha\, e^{(-\beta\, Dij/\, P)} \qquad (10.19)$$

where Y is the likelihood of an offender traveling a certain distance to commit a crime,, D_{ij} is the distance (from a home base location to an incident site), α is an arbitrary constant, β is the coefficient of the distance (and, hence, an exponent of *e*), P is a normalization constant, and *e* is the base of the natural logarithm. The model is similar to equation 10.11 except, like Rossmo, it does not include the attractiveness of the location.

Using the logic that most crimes are committed near the offender's home base, Canter, Coffey and Huntley (2000) use a five step process to estimate a search strategy:

1. The study area is defined by a rectangle that is 20% larger in area than that defined by the minimum and maximum X/Y points. A grid cell structure of 13, 300 cells is imposed over the rectangle. Each grid cell is a reference location, i.

2. A decay coefficient is selected. In equation 10.19, this would be the coefficient, β, for the distance term, D_{ij}, both of which are exponents of e. Unlike Rossmo, Canter uses a series of decay coefficients from 0.1 to 10 to estimate the sensitivity of the model. The equation indicates the likelihood with which any location is likely to be the home base of the offender based on one incident.

3. Because different offenders have different search areas, the measured distances for each cell are divided by a normalization coefficient, P, that adjusts all offenses to a comparable range. Canter uses two different types of normalization function: 1) mean inter-point distance between all offenses (across a group of offenders); and 2) the QRange, which is an index that takes into account asymmetry in the orientation of the incidents.

4. For each reference cell, i, the distance between each grid cell and each incident location is evaluated with the function and the standardized likelihoods are summed to yield an estimate of location potential.

5. A *search cost* index is defined by the proportion of the study area that has to be searched to find the offender. By calibrating the model against known cases, an estimate of search efficiency is obtained.

Additional modifications can be added to the functions to make them more flexible (Canter, Coffey and Huntley, 2000). For example, 'steps' are distances near to home where offenders are not likely to act while 'plateaus' are constant distances near to home where there is the highest likelihood of acting. For example, Canter and Larkin (1993) found an area around serial offenders' homes of about 0.61 mile in radius within which they were less likely to commit crimes.

Canter and Snook (1999) provide estimates of the search cost (or efficiency) associated with various distance coefficients. For example, with the known home base locations of 32 burglars, a β of 1.0 yielded a mean search cost of 18.06%; that is, on average, only 18.06% of the study area had to be searched to find the location of 32 burglars in the calibration sample. Clearly, for some of them, a larger area had to be searched while for others a smaller area; the average was 18.06%. Conversely, the mean search cost index for 24 rapists was 21.10% and for 37 murderers 28.28%. They further explored the marginal increase in locating offenders by increasing the percentage of the study area that had to be searched. They found for their three samples (burglary, rape, homicide) that more than half the offenders could be located within 15% of the area searched.

The Canter model is different from the Rossmo model is that it suggests a search strategy by the police for a serial offender rather than a particular location. The strength of it is to indicate how narrow an area the police should concentrate on in order to optimize finding an offender. Clearly, in most cases, only a small area needs be searched.

Strengths and weaknesses of the Canter model

The model has both strengths and weaknesses. First, the model provides a search strategy for law enforcement. By examining what type of function best fits a certain type of crime, police can target their search efforts more efficiently. The model is relatively easy to implement and is practical. Second, the mathematical formulation is stable. Unlike the inverse distance function in the Rossmo model, equation 10.19 will not have problems associated with distances that are close to 0. Further, the model does provide a search

strategy for identifying an offender. It is a useful tool for law enforcement officers, particularly as they frame a search for a serial offender.

There are also weaknesses to the model. First, it lacks a theoretical basis. Canter's research has provided a great deal in terms of understanding the activity spaces of serial offenders (Canter and Larkin, 1993; Canter and Gregory, 1994; Canter, 1994; Hodge and Canter, 2000). However, the empirical model used is strictly pragmatic. Second, mathematically, it imposes the negative exponential function without considering other distance decay models. In the *Dragnet* program, the decay function is a string of 20 numbers so that, in theory, any function can be explored. However, the default is a negative exponential. The negative exponential has been used in many travel behavior studies (Foot, 1981; Bossard, 1993), but it does not always produce the best fit. Later on, I'll show examples of travel behavior which show a distinctly non-monotonic function, even beyond a home base 'buffer zone'. While the model can be adapted to be more flexible by different exponents and including steps and plateaus, for example, it is still tied to the negative exponential form. Thus, the model might work in some locations, but may fail in others; a user can't easily adjust the model to make it fit new data.

Third, the coefficient of the negative exponential, α, is defined arbitrarily. In the *Dragnet* program, it is usually set as 0.5. While this ensures that the result never exceed 1.0 for any one incident, there is a limit on the location potential summation since the total potential is a function of the number of incidents (i.e., it will be higher for more incidents). Thus, the use of α ends up being arbitrary. It would have been better if the coefficient were calibrated against a known sample.

Fourth, and finally, also similar to the Rossmo model (and to my Jtc model below), criminal opportunities (or attractions) are never measured, but are inferred from the pattern of crime incidents. As a pragmatic tool for informing a police search, one could argue that this is not important. However, in a different location, the distance coefficient is liable to differ as is the search cost index. It would need to be re-calibrated each time.

Nevertheless, the Canter model is a useful tool for police department and can help shape a search strategy. It is different from the other location models in that it is not focused so much on the best prediction for a location of an offender (though the summation discussed above in step 4 can yield that) as it does in defining where the search should be optimized.

Geographic Profiling

Journey to crime estimation should be distinguished from *geographical profiling*. Geographical profiling involves understanding the geographical search pattern of criminals in relation to the spatial distribution of potential offenders and potential targets, the awareness spaces of potential offenders including the labeling of 'good' targets and crime areas, and the interchange of information between potential offenders who may modify their awareness space (Brantingham and Brantingham, 1981). According to Rossmo:

"...Geographic profiling focuses on the probable spatial behaviour of the offender within the context of the locations of, and the spatial relationships between, the various crime sites. A psychological profile provides insights into an offender's likely motivation, behaviour and lifestyle, and is therefore directly connected to his/her spatial activity. Psychological and geographic profiles thus act in tandem to help investigators develop a picture of the person responsible for the crimes in question" (Rossmo, 1997).

In other words, geographic profiling is a framework for understanding how an offender traverses an area in searching for victims or targets; this, of necessity, involves understanding the social environment of an area, the way that the offender understands this environment (the 'cognitive map') as well as the offender's motives.

On the other hand, journey to crime estimation follows a much simpler logic involving the distance dimension of the spatial patterning of a criminal. It is a method aimed at estimating the distance that serial offenders will travel to commit a crime and, by implication, the likely location from which they started their crime 'trip'. In short, it is a strictly statistical approach to estimating the residential whereabouts of an offender compared to understanding the dynamics of serial offenders.

It remains an empirical question whether a conceptual framework, such as geographic profiling, can predict better than a strictly statistical framework. Understanding of a phenomena, such as serial murders, serial rapists, and so forth, is an important research area. We seek more than just statistical prediction in building a knowledge base. However, it doesn't necessarily follow that understanding produces better predictions. In many areas of human activity, strictly statistical models are better in predicting than explanatory models. I will return to this point later in the section.

The *CrimeStat* Journey to Crime Routine

The journey to crime (Jtc) routine is a diagnostic designed to aid police departments in their investigations of serial offenders. The aim is to estimate the likelihood that a serial offender lives at any particular location. Using the location of incidents committed by the serial offender, the program makes statistical guesses at where the offender is liable to live, based on the similarity in travel patterns to a known sample of serial offenders for the same type of crime. The *Jtc* routine builds on the Rossmo (1993a; 1993b; 1995) framework, but extends its modeling capability.

1. A grid is overlaid on top of the study area. This grid can be either imported or can be generated by *CrimeStat* (see chapter 2). The grid represents the entire study area. Unlike Rossmo or Canter and Snook, there is no optimal study area. The technique will model that which is defined. Thus, the user has to select an area intelligently.

2. The routine calculates the distance between each incident location committed by a serial offender (or group of offenders working together) and

each cell, defined by the centroid of the cell. Rossmo (1993a; 1995) used indirect (Manhattan) distances. However, this would be appropriate only when a city falls on a uniform grid. The *Jtc* routine allows both direct and indirect distances. In most cases, direct distances would be the most appropriate choice as a police department would normally locate origin and destination locations rather than particular routes that are taken (see below).

3. A distance decay function is applied to each grid cell-incident pair and sums the values over all incidents. The user has a choice whether to model the travel distance by a mathematical function or an empirically-derived function.

4. The resultant of the distance decay function for each grid cell-incident pair are summed over all incidents to produce a likelihood (or density) estimate for each grid cell.
5. In both cases, the program outputs the two results: 1) the grid cell which has the peak likelihood estimate; and 2) the likelihood estimate for every cell. The latter output can be saved as a *Surfer® for Windows* 'dat', *ArcView Spatial Analyst©* 'asc', ASCII 'grd', *ArcView®* '.shp', *MapInfo®* '.mif', *Atlas*GIS™* '.bna' file or as an Ascii grid 'grd' file which can be read by many GIS packages (e.g., *ARC/INFO®*, *Vertical Mapper©*). These files can also be read by other GIS packages (e.g., *Maptitude*).

Figure 10.1 shows the logic of the routine and figure 10.2 shows the Journey to Crime (Jtc) screen. There are two parts to the routine. First, there is a calibration model which is used in the empirically-derived distance function. Second, there is the Journey to Crime (Jtc) model itself in which the user can select either the already-calibrated distance function or the mathematical function. The empirically-derived function is, by far, the easiest to use and is, consequently, the default choice in *CrimeStat*. The discussion of it is on p. 35. However, the mathematical function can be used if there is inadequate data to construct an empirical distance decay function or if a particular form is desired.

Distance Modeling Using Mathematical Functions

We'll start by illustrating the use of the mathematical functions because this has been the traditional way that distance decay has been examined. The *CrimeStat* Jtc routine allows the user to define distance decay by a mathematical function.

Probability Distance Functions

The user selects one of five probability density distributions to define a likelihood that the offender has traveled a particular distance to commit a crime. The advantage of having five functions, as opposed to only one, is that it provides more flexibility in describing travel behavior. The travel distance distribution followed will vary by crime type, time of day, method of

Figure 10.1: Journey to Crime Interpolation Routine

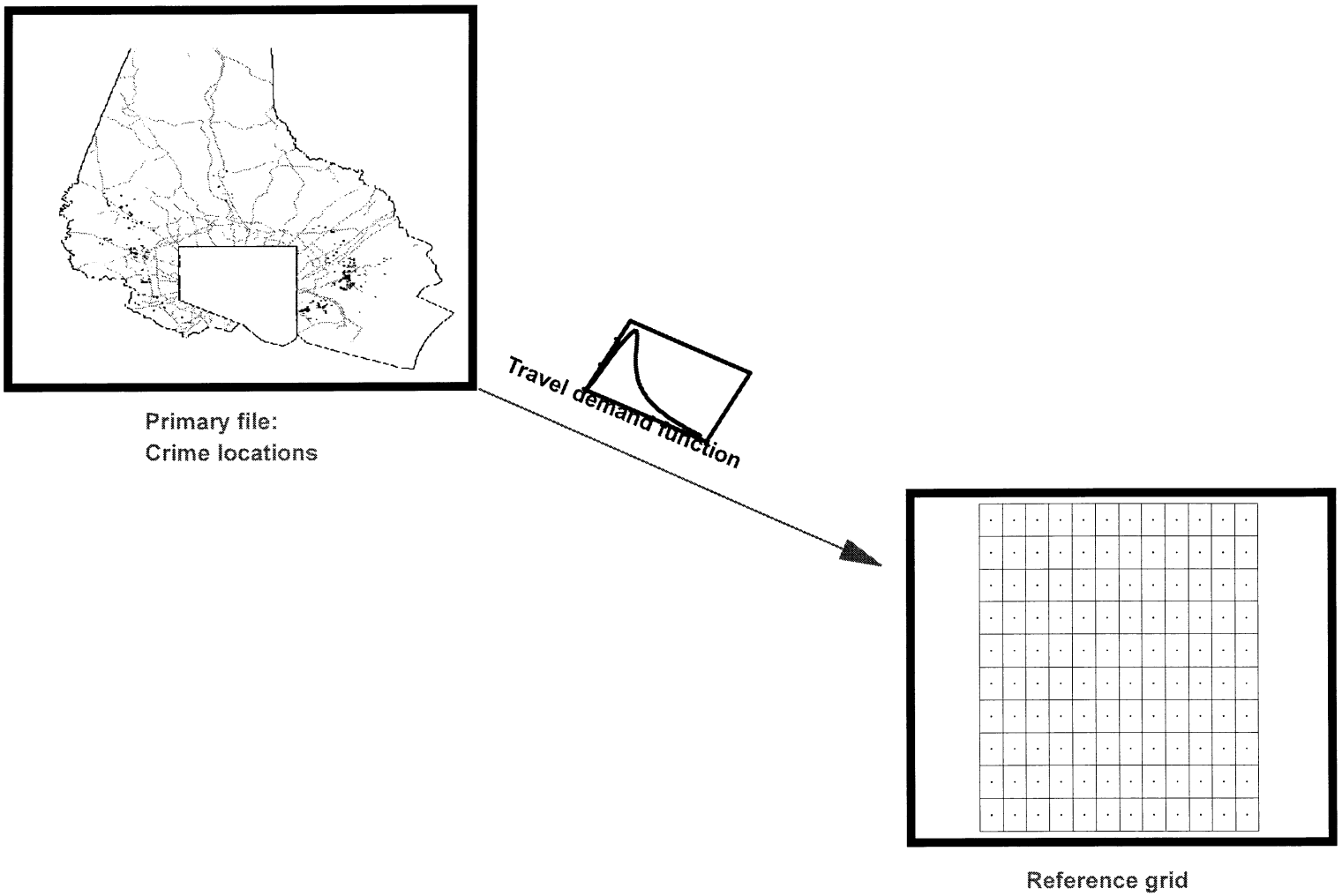
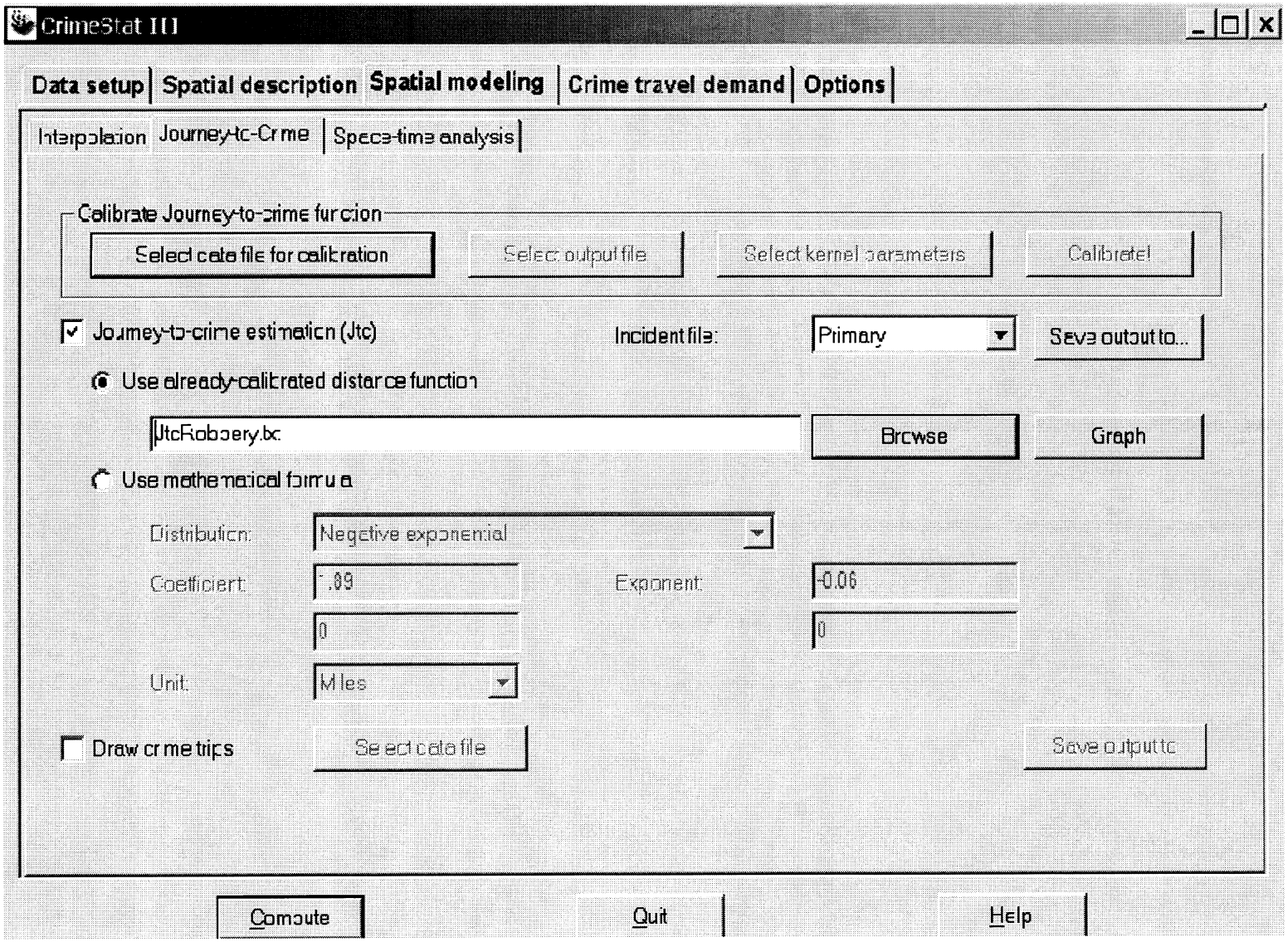


Figure 10.2: Journey to Crime Screen



operation, and numerous other variables. The five functions allow an approach that can simulate more accurately travel behavior under different conditions. Each of these has parameters that can be modified, allowing a very large number of possibilities for describing travel behavior of a criminal.

Figure 10.3 illustrates the five types.[2] Default values based on Baltimore County have been provided for each. The user, however, can change these as needed.

Briefly, the five functions are:

Linear

The simplest type of distance model is a linear function. This model postulates that the likelihood of committing a crime at any particular location declines by a constant amount with distance from the offender's home. It is highest near the offender's home but drops off by a constant amount for each unit of distance until it falls to zero. The form of the linear equation is:

$$f(d_{ij}) = \quad A + B*d_{ij} \tag{10.20}$$

where $f(d_{ij})$ is the likelihood that the offender will commit a crime at a particular location, i, defined here as the center of a grid cell, d_{ij} is the distance between the offender's residence and location i, A is a slope coefficient which defines the fall off in distance, and B is a constant. It would be expected that the coefficient B would have a negative sign since the likelihood should decline with distance. The user must provide values for A and B. The default for A is 1.9 and for B is -0.06. This function assumes no buffer zone around the offender's residence. When the function reaches 0 (the X axis), the routine automatically substitutes a 0 for the function.

Negative Exponential

A slightly more complex function is the negative exponential. In this type of model, the likelihood is also highest near the offenders home and drops off with distance. However, the decline is at a constant *rate* of decline, thus dropping quickly near the offender's home until is approaches zero likelihood. The mathematical form of the negative exponential is

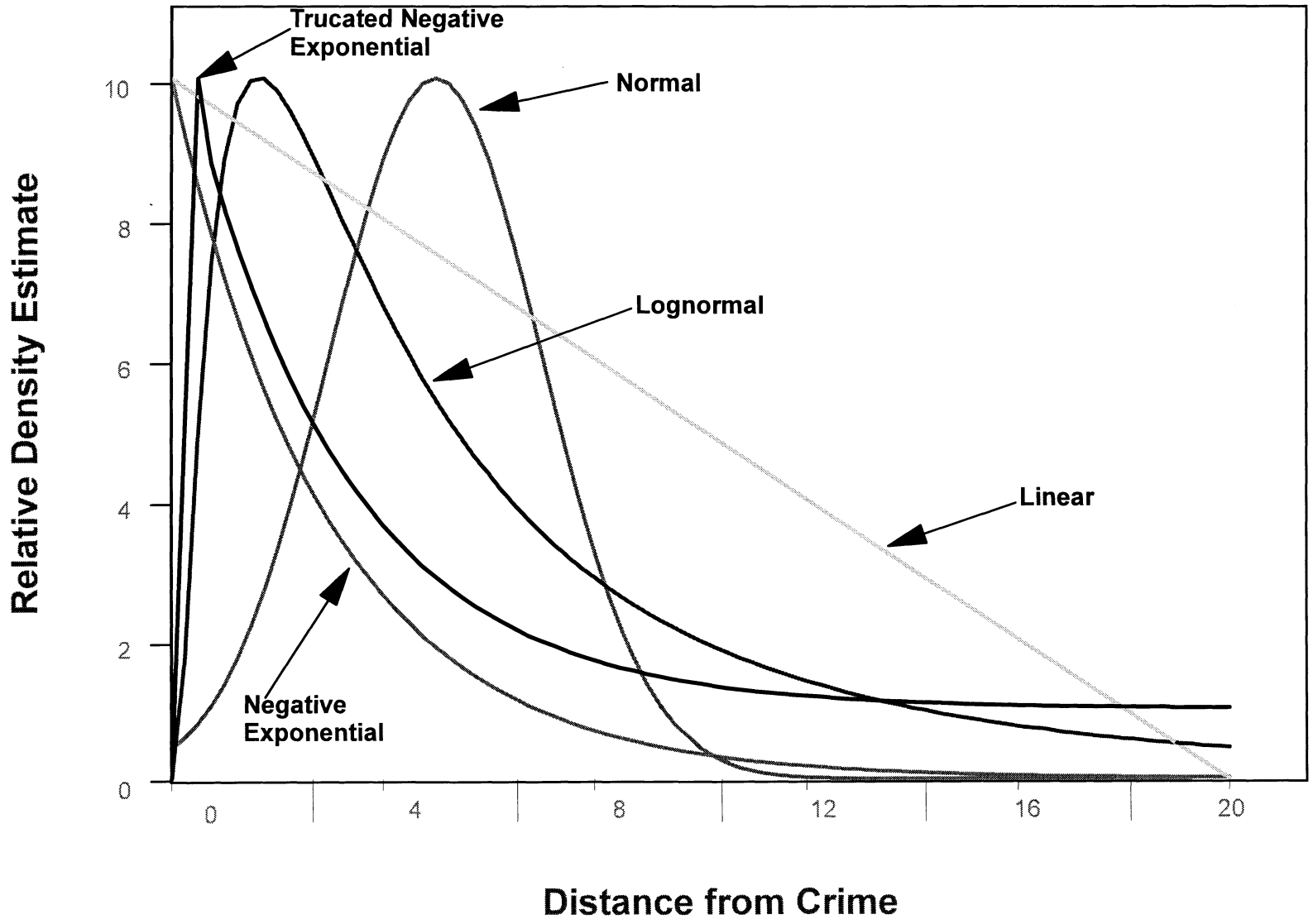
$$f(d_{ij}) = \quad A*e^{-B*d_{ij}} \tag{10.21}$$

where $f(d_{ij})$ is the likelihood that the offender will commit a crime at a particular location, i, defined here as the center of a grid cell, d_{ij} is the distance between each reference location

Figure 10.3:

Journey to Crime Travel Demand Functions

Five Mathematical Functions



and each crime location, e is the base of the natural logarithm, A is the coefficient and B is an exponent of e. The user inputs values for A - the coefficient, and B - the exponent. The default for A is 1.89 and for B is -0.06. This function is similar to the Canter model (equation 10.19) except that the coefficient is calibrated. Also, like the linear function, it assumes no buffer zone around the offender's residence.

Normal

A normal distribution assumes the peak likelihood is at some optimal distance from the offender's home base. Thus, the function rises to that distance and then declines. The rate of increase prior to the optimal distance and the rate of decrease from that distance is symmetrical in both directions. The mathematical form is:

$$Z_{ij} = \frac{(d_{ij} - \text{MeanD})}{S_d} \tag{10.22}$$

$$f(d_{ij}) = A * \frac{1}{S_d * \text{SQRT}(2\pi)} * e^{-0.5 * Z_{ij}^2} \tag{10.23}$$

where $f(d_{ij})$ is the likelihood that the offender will commit a crime at a particular location, i (defined here as the center of a grid cell), d_{ij} is the distance between each reference location and each crime location, MeanD is the mean distance input by the user, S_d is the standard deviation of distances, **e** is the base of the natural logarithm, and A is a coefficient. The user inputs values for MeanD, S_d, and A. The default values are 4.2 for the mean distance, MeanD, 4.6 for the standard deviation, S_d, and 29.5 for the coefficient, A.

By carefully scaling the parameters of the model, the normal distribution can be adapted to a distance decay function with an increasing likelihood for near distances and a decreasing likelihood for far distances. For example, by choosing a standard deviation greater than the mean (e.g., MeanD = 1,S_d = 2), the distribution will be skewed to the left because the left tail of the normal distribution is not evaluated. The function becomes similar to the model postulated by Brantingham and Brantingham (1981) in that it is a single function which describes travel behavior.

Lognormal

The lognormal function is similar to the normal except it is more skewed, either to the left or to the right. It has the potential of showing a very rapid increase near the offender's home base with a more gradual decline from a location of peak likelihood (see Figure 10.3). It is also similar to the Brantingham and Brantingham (1981) model. The mathematical form of the function is:

$$f(d_{ij}) = A * \frac{1}{d^2_{ij} * S_d * \text{SQRT}(2\pi)} * \mathbf{e}^{-[\ln(d^2_{ij}) - \text{MeanD}]^2 / 2 * s_d^2} \tag{10.24}$$

where $f(d_{ij})$ is the likelihood that the offender will commit a crime at a particular location, i, defined here as the center of a grid cell, d_{ij} is the distance between each reference location and each crime location, MeanD is the mean distance input by the user, S_d is the standard deviation of distances, ***e*** is the base of the natural logarithm, and A is a coefficient. The user inputs MeanD, S_d , and A. The default values are 4.2 for the mean distance, MeanD, 4.6 for the standard deviation, S_d, and 8.6 for the coefficient, A. They were calculated from the Baltimore County data (see table 10.3).

Truncated Negative Exponential

The truncated negative exponential is a joined function made up of two distinct mathematical functions - the linear and the negative exponential. For the near distance, a positive linear function is defined, starting at zero likelihood for distance 0 and increasing to d_p, a location of peak likelihood. Thereupon, the function follows a negative exponential, declining quickly with distance. The two mathematical functions making up this spline function are

$$\text{Linear:} \quad f(d_{ij}) = 0 + B*d_{ij} = B*d_{ij} \quad \text{for } d_{ij} \geq 0,\ d_ij \leq d_p \qquad (10.25)$$

$$\text{Negative Exponential:} \quad f(d_{ij}) = A*e^{-C*d_{ij}} \quad \text{for } X_i > d_p \qquad (10.26)$$

where d_{ij} is the distance from the home base, B is the slope of the linear function and for the negative exponential function A is a coefficient and C is an exponent. Since the negative exponential only starts at a particular distance, d_p, A, is assumed to be the intercept *if* the Y-axis were transposed to that distance. Similarly, the slope of the linear function is estimated from the peak distance, d_p, by a peak likelihood function. The default values are 0.4 for the peak distance, d_p, 13.8 for the peak likelihood, and -0.2 for the exponent, C. Again, these were calculated with Baltimore County data (see table 10.3)

This function is the closest approximation to the Rossmo model (equations 10.13 and 10.16). However, it differs in several mathematical properties. First, the 'near home base' function is linear (equation 10.25), rather than a non-linear function (equation 10.13). It assumes a simple increase in travel likelihoods by distance from the home base, up to the edge of the safety zone.[3] Second, the distance decay part of the function (equation 10.26) is a negative exponential, rather than an inverse distance function (equation 10.13); consequently, it is more stable when distances are very close to zero (e.g., for a crime where there is no 'near home base' offset).

Calibrating an Appropriate Probability Distance Function

The mathematics are relatively straightforward. However, how does one know which distance function to use? The answer is to get some data and calibrate it. It is important to obtain data from a sample of known offenders where both their residence at the time they committed crimes as well as the crime locations are known. This is called

the *calibration data set*. Each of the models are then tested against the calibration data set using an approach similar to that explained below. An error analysis is conducted to determine which of the models best fits the data. Finally, the 'best fit' model is used to estimate the likelihood that a particular serial offender lives at any one location. Though the process is tedious, once the parameters are calculated they can be used repeatedly for predictions.

Because every jurisdiction is unique in terms of travel patterns, it is important to calibrate the parameters for the particular jurisdiction. While there may be some similarities between cities (e.g., Eastern "centralized" cities v. Western "automobile" cities), there are always unique travel patterns defined by the population size, historical road pattern, and physical geography. Consequently, it is necessary to calibrate the parameters anew for each new city. Ideally, the sample should be a large enough so that a reliable estimate of the parameters can be obtained. Further, the analyst should check the errors in each of the models to ensure that the best choice is used for the Jtc routine. However, once it has been completed, the parameters can be re-used for many years and only periodically re-checked.

Data Set from Baltimore County

I'll illustrate with data from Baltimore County. The steps in calibrating the *Jtc* parameters were as follows:

1. 49,083 matched arrest and incident records from 1992 through 1997 were obtained in order to provide data on where the offender lived in relation to the crime location for which they were arrested.[4]

2. The data set was checked to ensure that there were X and Y coordinates for both the arrested individual's residence location and the crime incident location for which the individual was being charged. The data were cleaned to eliminate duplicate records or entries for which either the offender's residence or the incident location were missing. The final data set had 41,424 records. There were many multiple records for the same offender since an individual can commit more than one crime. In fact, more than half the records involved individuals who were listed two or more times. The distribution of offenders by the number of offenses for which they were charged is seen in Table 10.1. As would be expected, a small proportion of individuals account for a sizeable proportion of crimes; approximately 30% of the offenders in the database accounted for 56% of the incidents.

3. The data were imported into a spreadsheet, but a database program could equally have been used. For each record, the direct distance between the arrested individual's residence and the crime incident location was calculated. Chapter 2 presented the formulas for calculating direct distances between two locations and are repeated in endnote 5.[5]

Table 10.1

Number of Offenders and Offenses in Baltimore County: 1993-1997
Journey to Crime Database

Number of Offenses	Number of Individuals	Percent of Offenders	Number of Incidents	Percent of Incidents
1	18,174	70.0%	18,174	43.9%
2	4,443	17.1%	8,886	21.5%
3	1,651	6.4%	4,953	12.0%
4	764	2.9%	3,056	7.4%
5	388	1.5%	1,940	4.7%
6-10	482	1.9%	3,383	8.2%
11-15	61	0.2%	757	1.8%
16-20	10	<0.0%	175	0.4%
21-25	3	<0.0%	67	0.2%
26-30	0	<0.0%	0	0.0%
30+	1	<0.0%	33	<0.0%
	25,977		41,424	

4. The records were sorted into sub-groups based on different types of crimes. For the Baltimore County example, eleven categories of crime incident were used. Table 10.2 presents the categories with their respective sample sizes. Of course, other sub-groups could have been identified. Each sub-group was saved as a separate file. The same records can be part of multiple files (e.g., a record could be included in the 'all robberies' file as well as in the 'commercial robberies' file). All records were included in the 'all crimes' file.

5. For each type of crime, the file was grouped into distance intervals of 0.25 miles each. This involved two steps. First, the distance between the offender's residence and the crime location was sorted in ascending order. Second, a frequency distribution was conducted on the distances and grouped into 0.25 mile intervals (often called *bins*). The degree of precision in distance would depend on the size of the data set. For 41,426 records, quarter mile bins were appropriate.

6. For each type of crime, a new file was created which included only the frequency distribution of the distances broken down into quarter mile distance intervals, d_i.

7. In order to compare different types of crimes, which will have different frequency distributions, two new variables were created. First, the frequency in the interval was converted into the percentage of all crimes of in each interval by dividing the frequency by the total number of incidents, N,

and multiplying by 100. Second, the distance interval was adjusted. Since the interval is a range with a starting distance and an ending

Table 10.2

Baltimore County Files Used for Calibration

Crime Type	Sample Size
All crimes	41,426
Homicide	137
Rape	444
Assault	8,045
Robbery (all)	3,787
Commercial robbery	1,193
Bank robbery	176
Burglary	4,694
Motor vehicle theft	2,548
Larceny	19,806
Arson	338

distance but has been identified by spreadsheet program as the beginning distance only, a small fraction, representing the midpoint of the interval, is added to the distance interval. In our case, since each interval is 0.25 miles wide, the adjustment is half of this, 0.125. Each new file, therefore, had four variables: the interval distance, the adjusted interval distance, the frequency of incidents within the interval (the number of cases falling into the interval), and the percentage of all crimes of that type within the interval.

8. Using the regression program in the crime travel demand model (see chapter 12), a series of regression equations was set up to model the frequency (or the percentage) as a function of distance. In this case, I used our routines, but other statistical packages could equally have been used. Again, because comparisons between different types of crimes were of interest, the percentage of crimes (by type) within an interval was used as the dependent variable (and was defined as a percentage, i.e., 11.51% was recorded as 11.51). Five equations testing each of the five models were set up.

Linear

For the linear function, the test was

$$Pct_i = A + Bd_i \qquad (10.27)$$

where Pct_i is the percentage of all crimes of that type falling into interval i, d_i is the distance for interval i, A is the intercept, and B is the slope. A and B are estimated directly from the regression equation.

Negative Exponential

For the negative exponential function, the variables have to be transformed to estimate the parameters. The function is

$$Pct_i = A * e^{-B*d_i} \tag{10.28}$$

A new variable is defined which is the natural logarithm of the percentage of all crimes of that type falling into the interval, $ln(Pct_i)$. This term was then regressed against the distance interval, d_i.

$$\ln(Pct_i) = K - B*d_i \tag{10.29}$$

However, since the original equation has been transformed into a log function, B is the coefficient and A can be calculated directly from

$$\ln(Pct_i) = \ln(A) - B*d_i \tag{10.30}$$

$$A = e^K \tag{10.31}$$

If the percentage in any bin was 0 (i.e., $Pct_i = 0$), then a value of -16 was taken since the natural logarithm of 0 cannot be solved (it approximates -16 as the percentage approaches 0.0000001).

Normal

For the normal function, a more complex transformation must be used. The normal function in the model is

$$Pct_i = A * \frac{1}{S_d * SQRT(2\pi)} * e^{.0.5*Zij^2} \tag{10.32}$$

First, a standardized Z variable for the distance, d_i, is created

$$Z_i = \frac{(d_i - MeanD)}{S_d} \tag{10.33}$$

where MeanD is the mean distance and S_d is the standard deviation of distance. These are calculated from the original data file (*before* creating the file of frequency distributions). Second, a normal transformation of Z is constructed with

$$\text{Normal}(Z_i) = \frac{1}{S_d * \text{SQRT}(2\pi)} * e^{-0.5 * Zij^2} \tag{10.34}$$

Finally, the normalized variable is regressed against the percentage of all crimes of that type falling into the interval, Pct_i with *no* constant

$$\text{Pct}_i = A * \text{Normal}(Z_i) \tag{10.35}$$

A is estimated by the regression coefficient.

Lognormal

For the lognormal function, another complex transformation must be done. The lognormal function for the percentage of all crimes of a type for a particular distance interval is

$$\text{Pct}_i = A * \frac{1}{d^2_{ij} * S_d * \text{SQRT}(2\pi)} * e^{-[\ln(d^2_i) - \text{MeanD}]^2 / 2 * S_d^2} \tag{10.36}$$

The transformation can be created in steps. First, create L

$$L = \ln(d_i^2) \tag{10.37}$$

Second, create M

$$M = (1 - \text{MeanD})^2 \tag{10.38}$$

Third, create O

$$O = \frac{m}{(2 * S_d^2)} \tag{10.39}$$

Fourth, create P by raising **e** to the O^{th} power.

$$P = e^O \tag{10.40}$$

Fifth, create the lognormal conversion, Lnormal

$$\text{Lnormal}(d_i) = A * \frac{1}{d^2_{ij} * S_d * \text{SQRT}(2\pi)} * P \tag{10.41}$$

Finally, the lognormal variable is regressed against the percentage of all crimes of that type falling into the interval, Pct_i with *no* constant

$$Pct_i = A^* \text{ Lnormal}(d_i) \quad (10.42)$$

A is estimated with the regression coefficient.

Truncated Negative Exponential

For the truncated negative exponential function, two models were set up. The first applied to the distance range from 0 to the distance at which the percentage (or frequency) is highest, $Maxd_i$. The second applied to all distances greater than this distance

$$\text{Linear:} \quad Pct_i = A + Bd_i \text{ for } d_{ij} \geq 0,\ d_ij \leq Maxd_{ij} \quad (10.43)$$

$$\text{Negative Exponential:} \quad Pct_i = A^*e^{-C^*di} \text{ for } d_ij > Maxd_{ij} \quad (10.44)$$

To use this function, the user specifies the distance at which the peak likelihood occurs, d_p (the *peak distance*) and the value for that peak likelihood, P (the *peak likelihood*). For the negative exponential function, the user specifies the exponent, C.

In order to splice the two equations together (the spline), the *CrimeStat* truncated negative exponential routine starts the linear equation at the origin and ends it at the highest value. Thus,

$$A = 0 \quad (10.45)$$

$$B = P/d_p \quad (10.46)$$

where P is the peak likelihood and d_p is the peak distance.

The exponent, C, can be estimated by transforming the dependent variable, Pct_i, as in the negative exponential above (equation 10.28) and regressing the natural log of the percentage ($\ln(Pct_i)$ against the distance interval, d_i, *only* for those intervals that are greater than the peak distance. I have found that estimating the transformed equation with a coefficient, A in

$$Pct_i = A * e^{-C^*d_i} \quad (10.47)$$

$$\ln(Pct_i) = Ln(A) - C^*d_i \quad (10.48)$$

gives a better fit to the equation. However, the user need only input the exponent, C, in the Jtc routine as the coefficient, A, of the negative exponential is calculated internally to produce a distance value at which the peak likelihood occurs. The formula is:

$$A = e^{\ln(P) + C*(d_p - d_i)} \quad (10.49)$$

where P is the peak likelihood, d_p is the distance for the peak likelihood, C is an exponent (assumed to be positive) and d_i is the distance interval for the histogram.

9. Once the parameters for the five models have been estimated, they can be compared to see which one is best at predicting the travel behavior for a particular type of crime. It is to be expected that different types of crimes will have different optimal models and that the parameters will also vary.

Examples from Baltimore County

Let's illustrate with the Baltimore County data. Figure 10.4 shows the frequency distribution for all types of crime in Baltimore County. As can be seen, at the nearest distance interval (0 to 0.25 miles with an assigned 'adjusted' midpoint of 0.125 miles), about 6.9% of all crimes occur within a quarter mile of the offender's residence (it can be seen on the Y-axis). However, for the next interval (0.25 to 0.50 miles with an assigned midpoint of 0.375 miles), almost 10% of all crimes occur at that distance (9.8%). In subsequent intervals, however, the percentage decreases, a little less than 6% for 0.50 to 0.75 miles (with the midpoint being 0.625 miles), a little more than 4% for 0.75 to 1 mile (the midpoint is 0.875 miles), and so forth.

The best fitting statistical function was the negative exponential. The particular equation is

$$Pct_i = 5.575 * e^{-0.229*d_i} \quad (10.50)$$

This is shown with the solid line. As can be seen, the fit is good for most of the distances, though it underestimates at close to zero distance and overestimates from about a half mile to about four miles. There is only slight evidence of decreased activity near to the location of the offender.

However, the distribution varies by type of crime. With the Baltimore County data, property crimes, in general, occur farther away than personal crimes. The truncated negative exponential generally fit property crimes better, lending support for the Brantingham and Brantingham (1981) framework for these types. For example, larceny offenders have a definite safety zone around their residence (figure 10.5). Fewer than 2% of larceny thefts occur within a quarter mile of the offender's residence. However, the percentage jumps to about 4.5% from a quarter mile to a half. The truncated negative exponential function fits the data reasonably well though it overestimates from about 1 to 3 miles and underestimates from about 4 to12 miles.

Figure 10.4:

Journey to Crime Distances: All Crimes
Negative Exponential Distribution

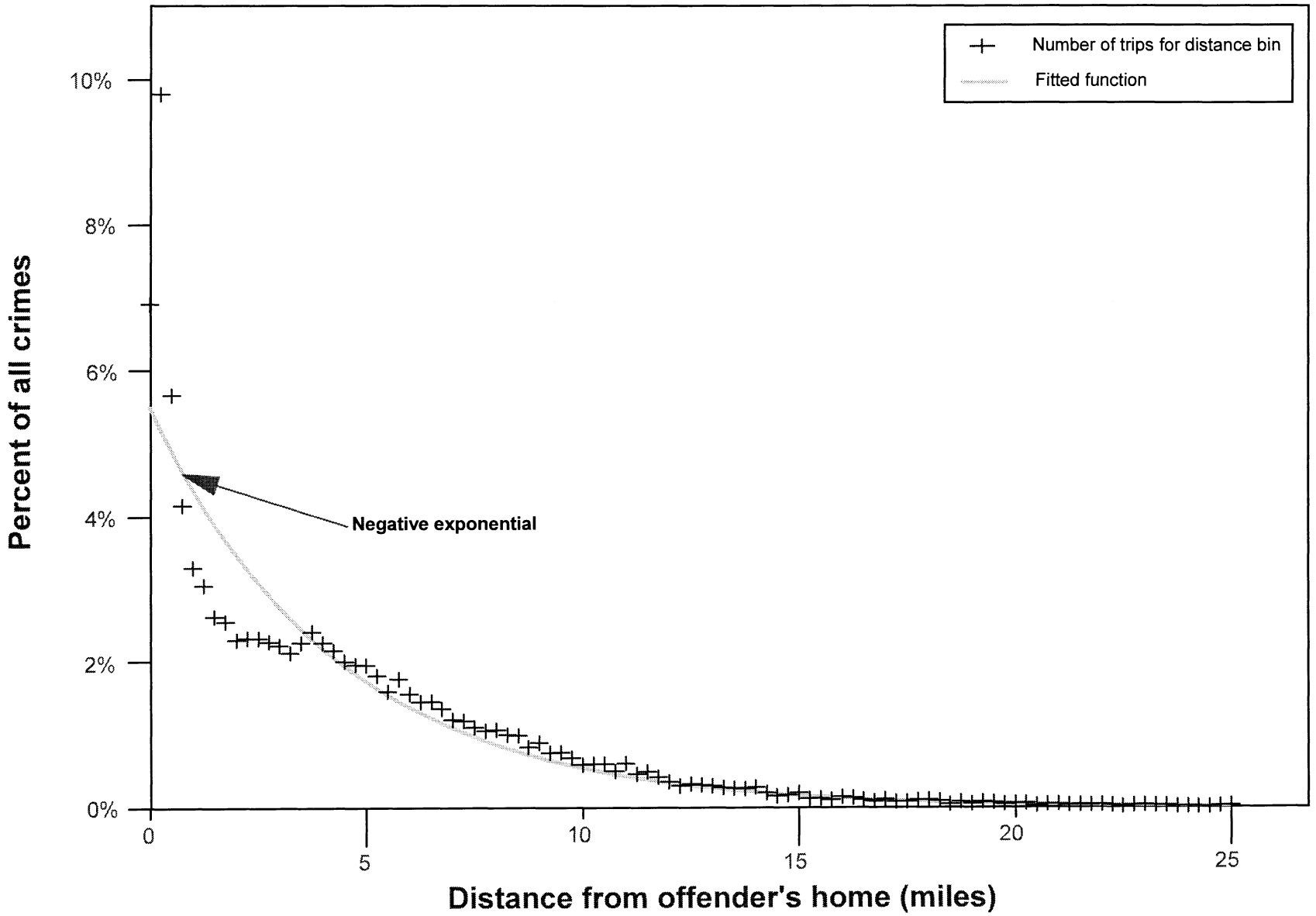
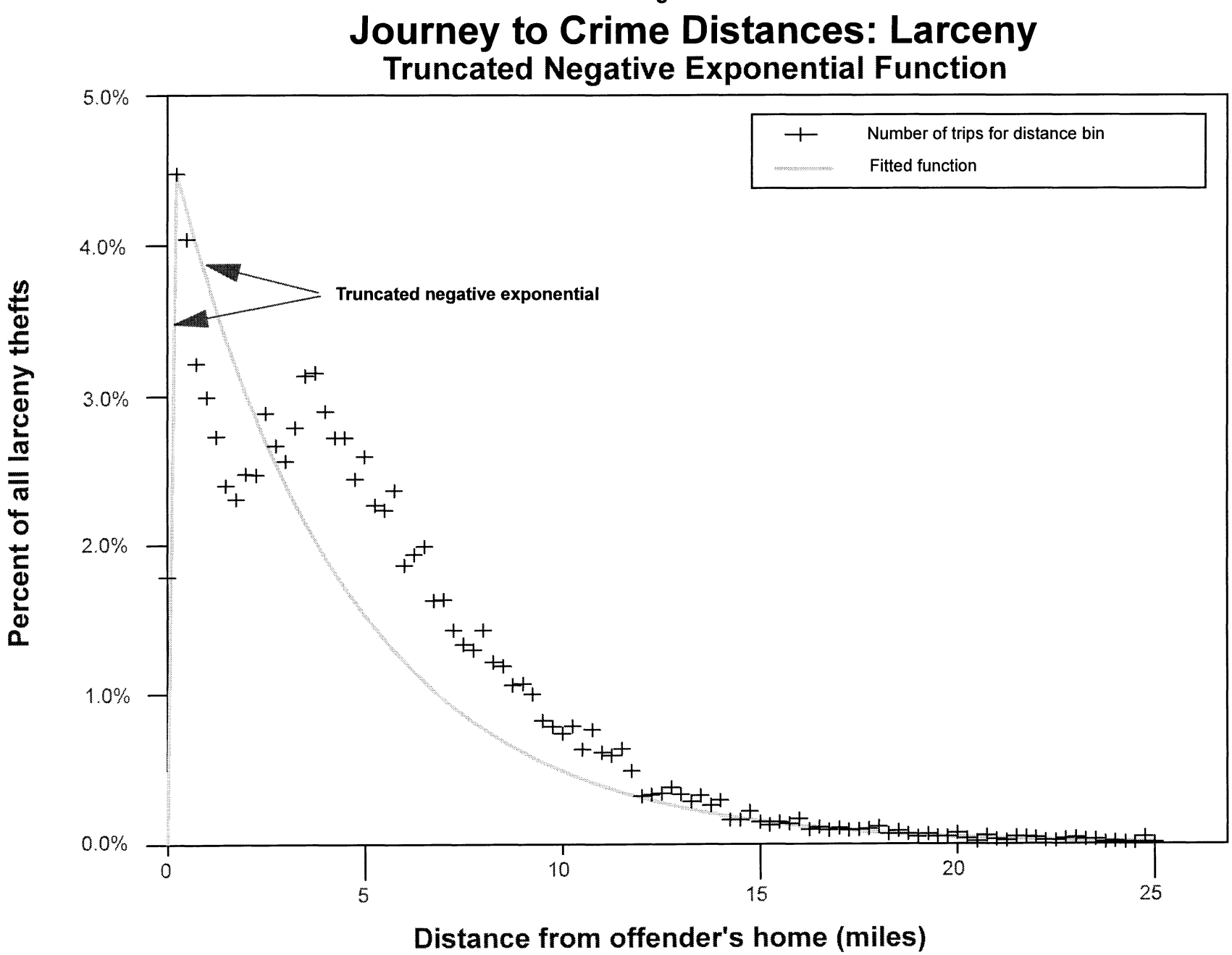


Figure 10.5:



Similarly, motor vehicle thefts show decreased activity near the offender's resident, though it is less pronounced than larceny theft. Figure 10.6 shows the distribution of motor vehicle thefts and the truncated negative exponential function which was fit to the data. As can be seen, the fit is reasonably good though it tends to underestimate middle range distances (approximately 3-12 miles).

Some types of crime, on the other hand, are very difficult to fit. Figure 10.7 shows the distribution of bank robberies. Partly because there were a limited number of cases (N=176) and partly because it's a complex pattern, the truncated negative exponential gave the best fit, but not a particularly good one. As can be seen, the linear ('near home') function underestimates some of the near distance likelihoods while the negative exponential drops off too quickly; in fact, to make this function even plausible, the regression was run only up to 21 miles (otherwise, it underestimated even more).

For some crimes, it was very difficult to fit any single function. Figure 10.8 shows the frequency distribution of 137 homicides with three functions being fitted to the data - the truncated negative exponential, the lognormal, and the normal. As can be seen each function fits only some of the data, but not all of it.

Testing for Residual Errors in the Model

In short, the five mathematical functions allow a user to fit a variety of distance decay distributions. Each of the models will predict some parts of the distribution better than others. Consequently, it is important to conduct an error analysis to determine which model is 'best'. In an error analysis, the residual error is defined as

$$\text{Residual error} = Y_i - E(Y_i) \qquad (10.51)$$

where Y_i is the observed (actual) likelihood for distance i and $E(Y_i)$ is the likelihood predicted by the model. If raw numbers of incidents are used, then the likelihoods are the number of incidents for a particular distance. If the number of incidents are converted into proportions (i.e., probabilities), then the likelihoods are the proportions of incidents for a particular distance.

The choice of 'best model' will depend on what part of the distribution is considered most important. Figure 10.9, for example, shows the residual errors on vehicle theft for the five fitted models. That is, each of the five models was fit to the proportion of vehicle thefts by distance intervals (as explained above). For each distance, the discrepancy between the actual percentage of vehicle thefts in that interval and the predicted percentage was calculated. If there was a perfect fit, then the discrepancy (or residual) was 0%. If the actual percentage was greater than the predicted (i.e., the model underestimated), then the residual was positive; if the actual was smaller than the predicted (i.e., the model overestimated), then the residual was negative.

Figure 10.6:

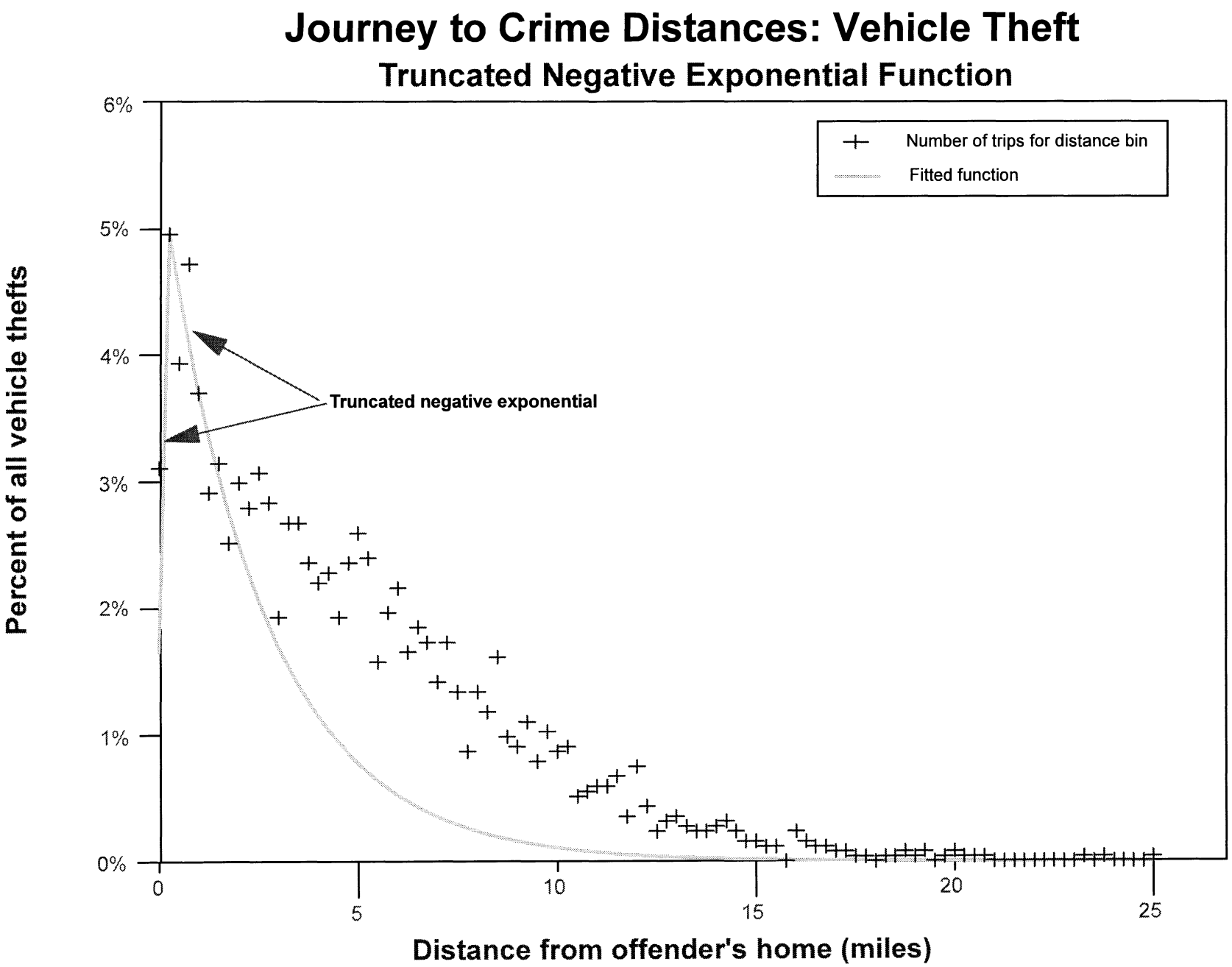


Figure 10.7:

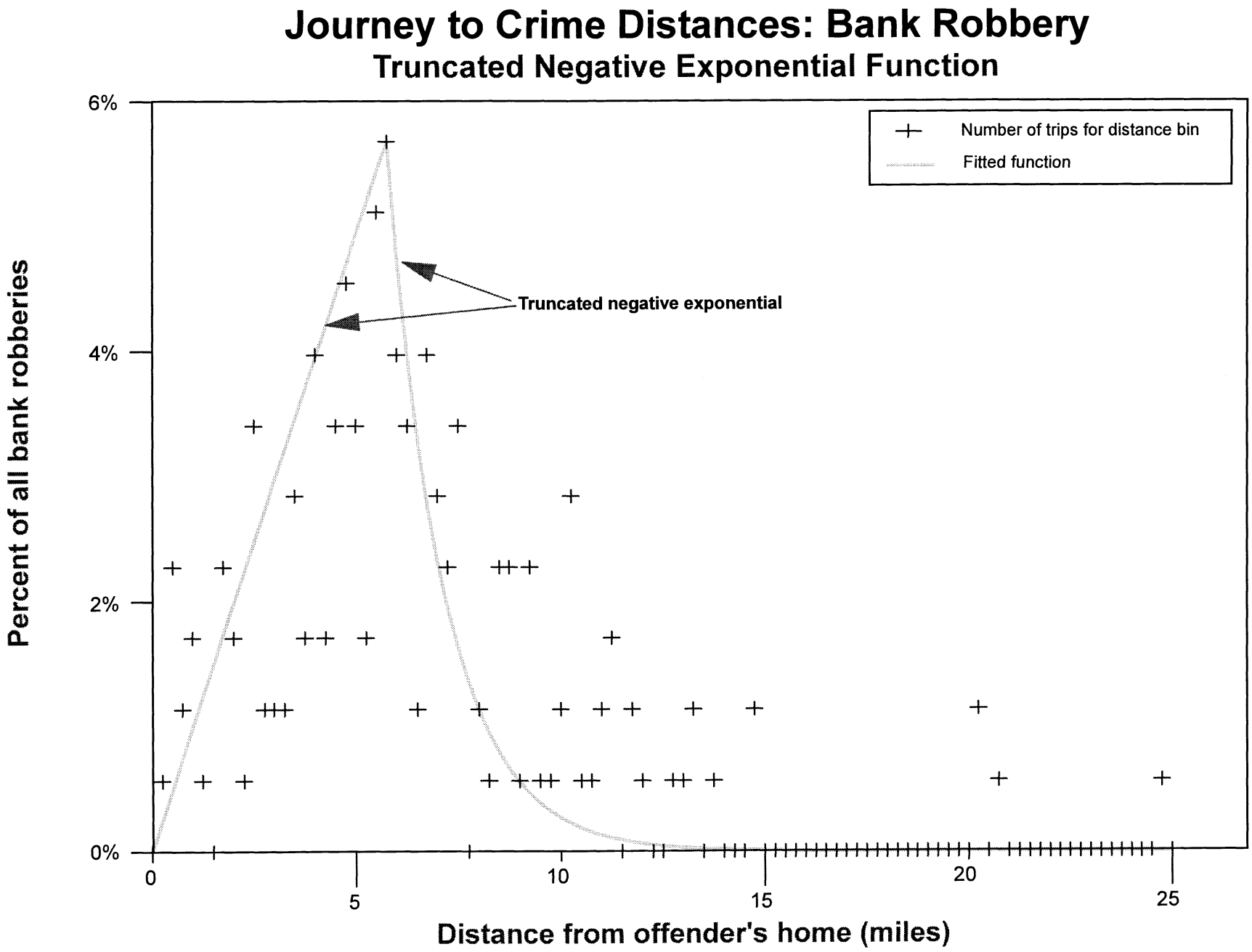


Figure 10.8:

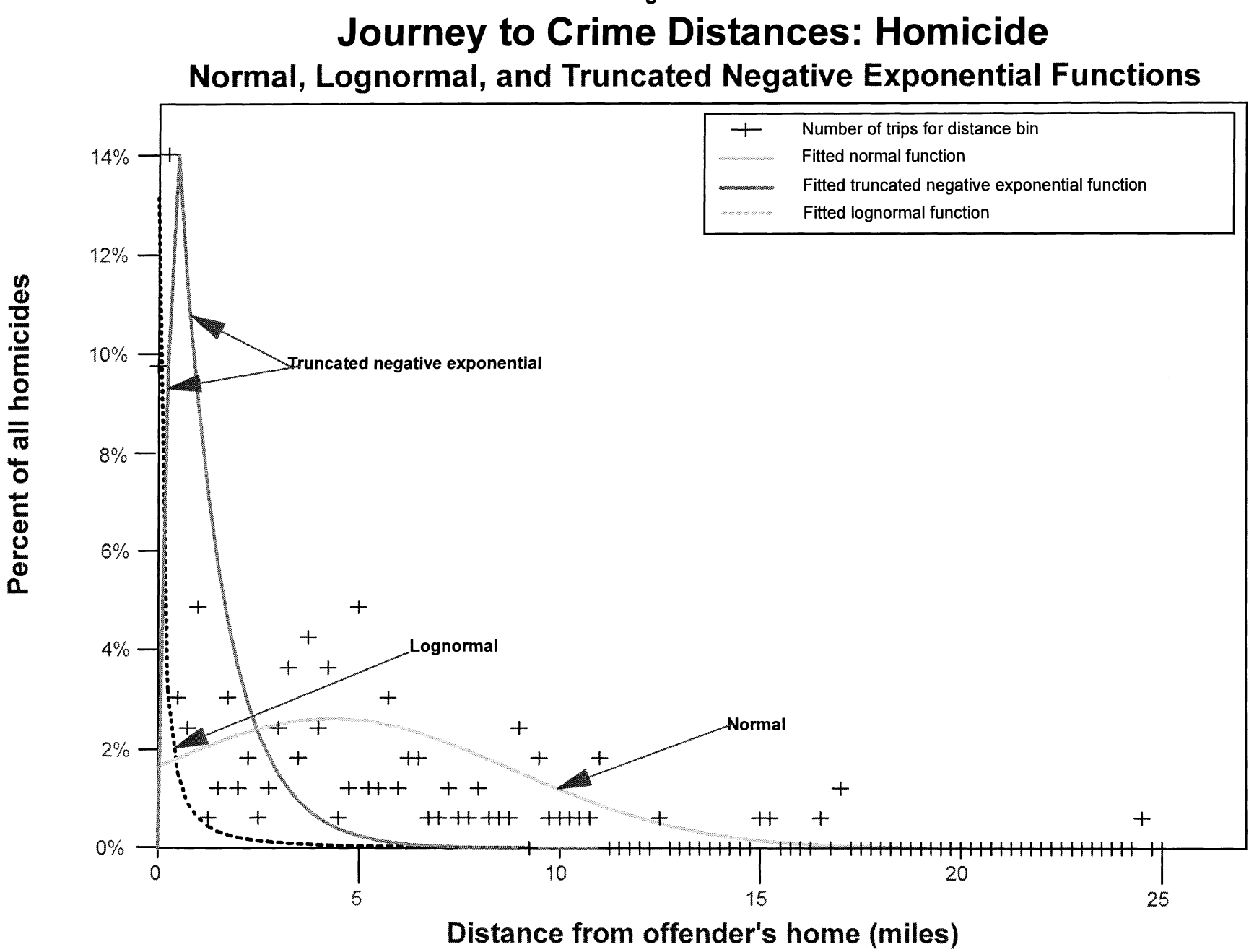
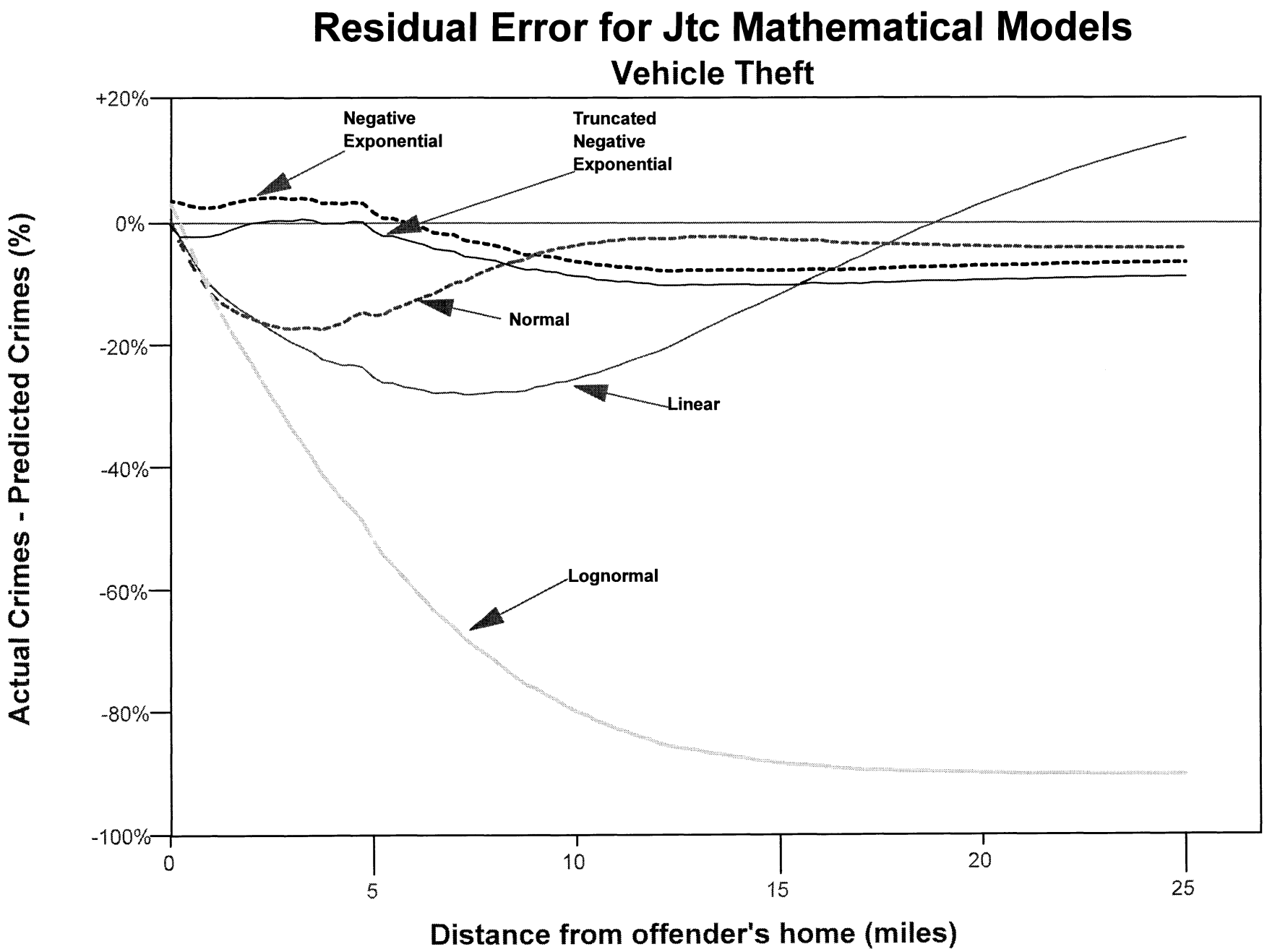


Figure 10.9:

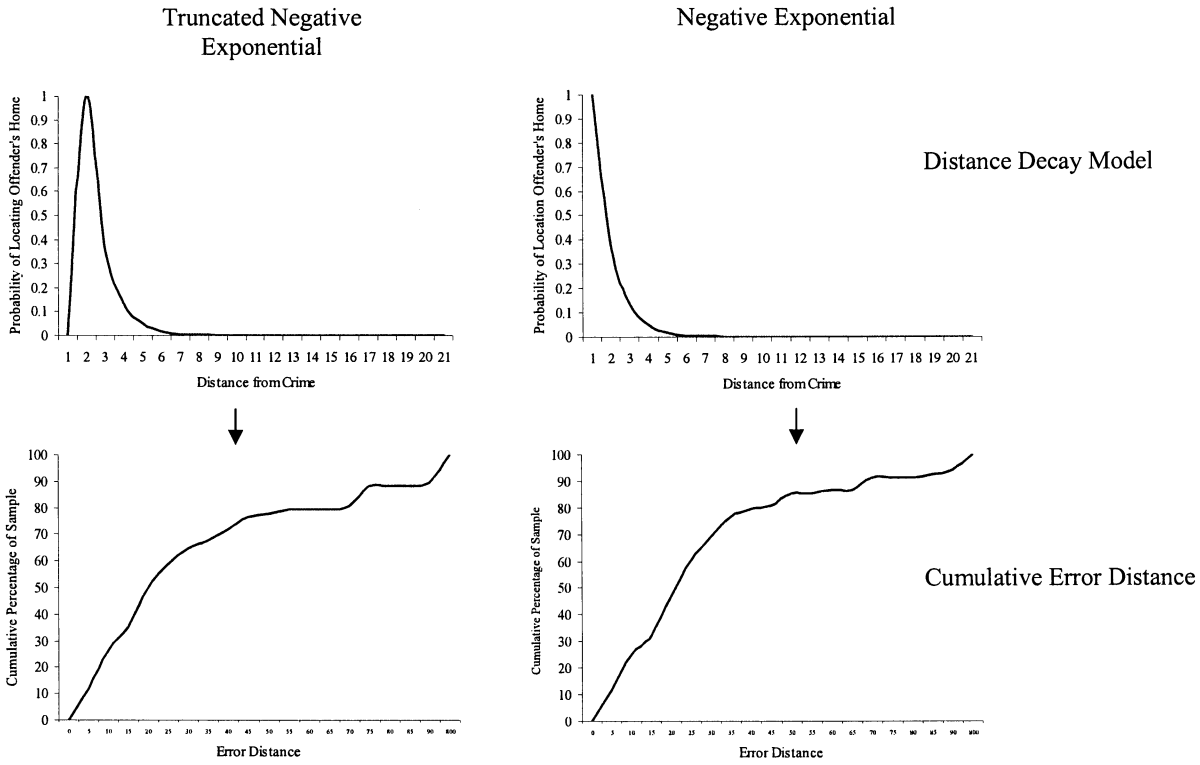


Using *CrimeStat* for Geographic Profiling

Brent Snook, Memorial University of Newfoundland,
Paul J. Taylor, University of Liverpool, Liverpool
Craig Bennell, Carleton University, Ottawa

A challenge for researchers providing investigative support is to use information about crime locations to prioritize geographic areas according to how likely they are to contain the offender's residence. One prescient solution to this problem uses *probability distance functions* to assign a likelihood value to the activity space around each crime location. A research goal is to identify the function that assigns the highest likelihood to the offender's actual residence, since this should prove more efficient in future investigations.

CrimeStat was used to test of the effectiveness of two functions for a sample of 68 German serial murder cases, using a measure known as *error distance*. The top figures below illustrate the two functions used and the bottom figures portray the corresponding effectiveness of the functions by plotting the percentage of the sample 'located' by error distance. A steeper effectiveness curve indicates that home locations were closer to the point of highest probability and that, consequently, the probability distance function was more efficient. In this particular test, no difference was found between the two functions in their ability to classify geographic areas.



Original Article: Taylor, P.J., Bennell, C., & Snook B. (2002) *Problems of Classification in Investigative Psychology.* Proceedings of the 8th Conference of the International Federation of Classification Societies, Krakow, Poland

As can be seen in figure 10.9, the truncated negative exponential fit the data well from 0 to about 5 miles, but then became poorer than other models for longer distances. The negative exponential model was not as good as the truncated for distances up to about 5 miles, but was better for distances beyond that point. The normal distribution was good for distances from about 10 miles and farther. The lognormal was not particularly good for any distances other than at 0 miles, nor was the linear.

The degree of predictability varied by type of crime. For some types, particularly property crimes, the fit was reasonably good. I obtained R^2 in the order of 0.86 to 0.96 for burglary, robbery, assault, larceny, and auto theft. For other types of crime, particularly violent crimes, the fit was not very good with R^2 values in the order of 0.53 (rape), 0.41 (arson) and 0.30 (homicide). These R^2 values were for the entire distance range; for any particular distance, however, the predictability varied from very high to very low.

In modeling distance decay with a mathematical function, a user has to decide which part of the distribution is the most important as no simple mathematical function will normally fit all the data (even approximately). In these cases, I assumed that the near distances were more important (up to, say, 5 miles) and, therefore, selected the model which 'best' fit those distances (see table 10.2). However, it was not always clear which model was best, even with that limited criteria.

Problems with Mathematical Distance Decay Functions

There are several reasons that mathematical models of distance decay distributions, such as illustrated in the Jtc routine, do not fit data very well. First, as mentioned earlier, few cities have a completely symmetrical grid structure or even one that is approximately grid-like (there are exceptions, of course). Limitations of physical topography (mountains, oceans, rivers, lakes) as well as different historical development patterns makes travel asymmetrical around most locations.

Second, there is population density. Since most metropolitan areas have much higher intensity of land use in the center (i.e., more activities and facilities), travel tends to be directed towards higher land use intensity than away from them. For origin locations that are not directly in the center, travel is more likely to go towards the center than away from it.

This would be true of an offender as well. If the person were looking for either persons or property as 'targets', then the offender would be more likely to travel towards the metropolitan center than away from it. Since most metropolitan centers have street networks that were laid out much earlier, the street network tends to be irregular. Consequently, trips will vary by location within a metropolitan area. One would expect shorter trips by an offender living close to the metropolitan center than one living farther away; shorter trips for offenders living in more built-up areas than in lower density areas; shorter trips for offenders in mixed use neighborhoods than in strictly residential neighborhoods; and so forth. Thus, the distribution of trips of any sort (in our case, crime trips from a residential location to a crime location), will tend to follow an irregular,

distance decay type of distribution. Simple mathematical models will not fit the data very well and will make many errors.

Third, the selection of a best mathematical function is partly dependent on the interval size used for the bins. In the above examples, an interval size of 0.25 miles was used to calculate the frequency distribution. With a different interval size (e.g., 0.5 miles), however, a slightly different distribution is obtained. This effects the mathematical function that is selected as well as the parameters that are estimated. For example, the issue of whether there is a safety zone near the offender's residence from which there is decreased activity or not is partly dependent on the interval size. With a small interval, the zone may be detected whereas with a slightly larger interval the subtle distinction in measured distances may be lost. On the other hand, having a smaller interval may lead to unreliable estimates since there may be few cases in the interval. Having a technique depend on the interval size makes it vulnerable to mis-specification.

Uses of Mathematical Distance Decay Functions

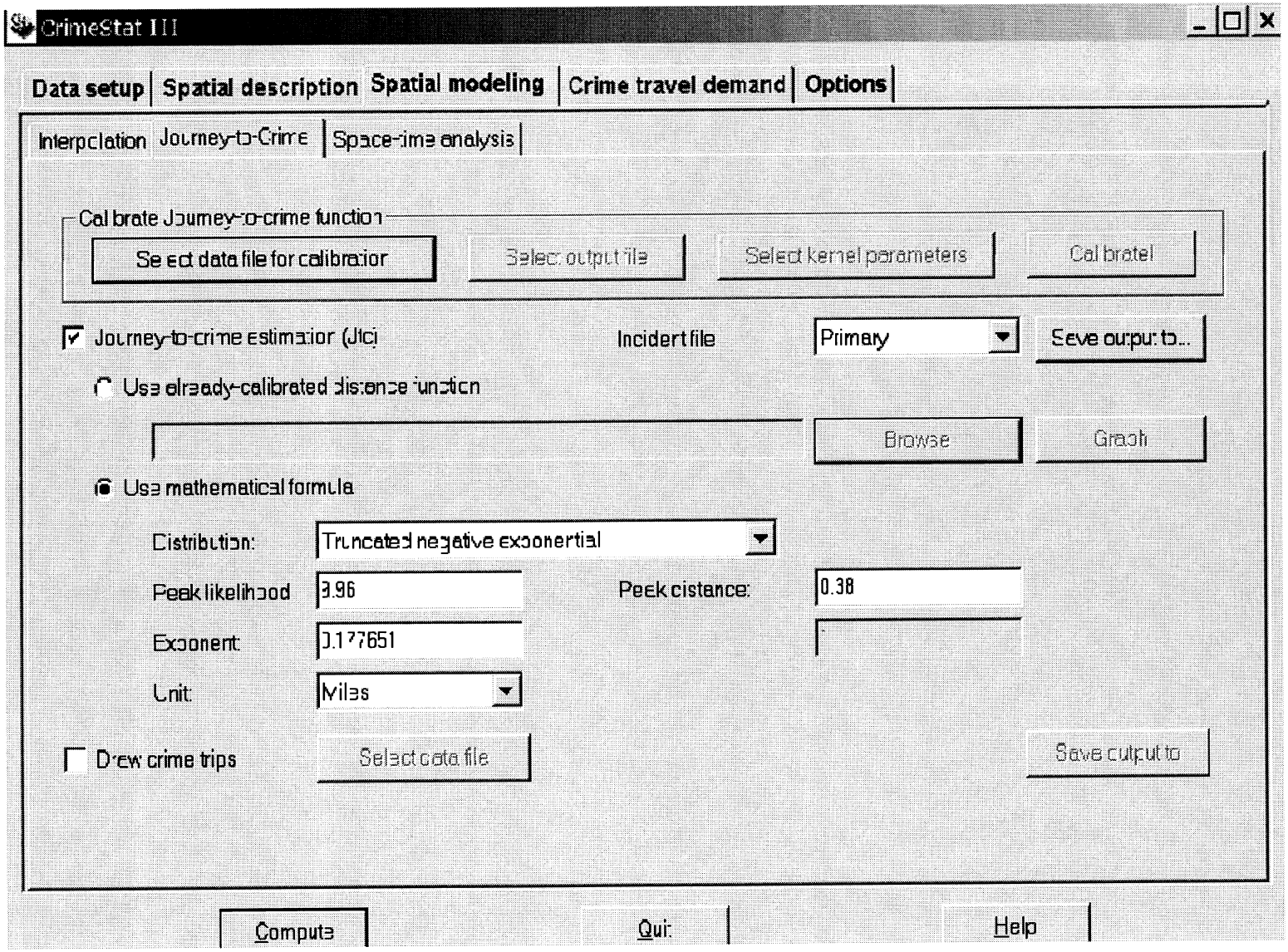
Does this mean that one should not use mathematical distance functions? I would argue that under most circumstances, a mathematical function will give less precision than an empirically-derived one (see below). However, there are two cases when a mathematical model would be appropriate. First, if there is either no data or insufficient data to model the empirical travel distribution, the use of a mathematical model can serve as an approximation. If the user has a good sense of what the distribution looks like, then a mathematical model may be used to approximate the distribution. However, if a poorly defined function is selected, then the selected function may produce many errors.

A second case when mathematical models of distance decay would be appropriate is in theory development or application. Many models of travel behavior, for example, assume a simple distance decay type of function in order simplify the allocation of trips over a region. This is a common procedure in travel demand modeling where trips from each of many zones are assigned to every other zone using a gravity type of function (Stopher and Meyburg, 1975; Field and MacGregor, 1987). Even though the model produces errors because it assumes uniform travel behavior in all directions, the errors are corrected later in the modeling process by adjusting the coefficients for allocating trips to particular roads (traffic assignment). The model provides a simple device and the errors are corrected down the line. Still, I would argue that an empirically-derived distribution will produce fewer errors in allocation and, thus, require less adjustment later on. Errors can never help a model and its better to get it more correct initially to have to adjust it later on; the adjustment may be inadequate. Nevertheless, this is common practice in transportation planning.

The Journey to Crime Routine Using a Mathematical Formula

The Jtc routine which allows mathematical modeling is simple to use. Figure 10.10 illustrates how the user specifies a mathematical function. The routine requires the use of a grid which is defined on the reference file tab of the program (see chapter 3). Then, the

Figure 10.10: Jtc Mathematical Distance Decay Function



user must specify the mathematical function and the parameters. In the figure, the truncated negative exponential is being defined. The user must input values for the peak likelihood, the peak distance, and the exponent (see equations 10.43 and 10.44 above). In the figure, since the serial offenses were a series of 18 robberies, the parameters for robbery have been entered into the program screen. The peak likelihood was 9.96% (entered as a whole number - i.e., 9.96); the distance at which this peak likelihood occurred was the second distance interval 0.25-0.50 miles (with a mid-point of 0.38 miles); and the estimated exponent was 0.177651. As mentioned above, the coefficient for the negative exponential part of the equation is estimated internally.

Table 10.3 gives the parameters for the 'best' models which fit the data for the 11 types of crime in Baltimore County. For several of these (e.g., bank robberies), two or more functions gave approximately equally good fits. Note that these parameters were estimated with the Baltimore County data. They will not fit any other jurisdiction. If a user wishes to apply this logic, then the parameters should be estimated anew from existing data. Nevertheless, once they have been calibrated, they can be used for predictions.

The routine can be output to *ArcView*, *MapInfo*, *Atlas*GIS*, *Surfer for Windows*, *Spatial Analyst*, and as an Ascii grid file which can be read by many other GIS packages. All but *Surfer for Windows* require that the reference grid be created by *CrimeStat*.

Distance Modeling Using an Empirically Determined Function

An alternative to mathematical modeling of distance decay is to empirically describe the journey to crime distribution and then use this empirical function to estimate the residence location. *CrimeStat* has a two-dimensional kernel density routine that can calibrate the distance function if provided data on trip origins and destinations. The logic of kernel density estimation was described in chapter 8, and won't be repeated here. Essentially, a symmetrical function (the 'kernel') is placed over each point in a distribution. The distribution is then referenced relative to a scale (an equally-spaced line for two-dimensional kernels and a grid for three-dimensional kernels) and the values for each kernel are summed at each reference location. See chapter 8 for details.

Calibrate Kernel Density Function

The *CrimeStat* calibration routine allows a user to describe the distance distribution for a sample of journey to crime trips. The requirements are that:

1. The data set must have the coordinates of *both* an origin location and a destination location; and

2. The records of all origin and destination locations have been populated with legitimate coordinate values (i.e., no unmatched records are allowed).

Table 10.3

Journey to Crime Mathematical Models for Baltimore County
Parameter Estimates for Percentage Distribution
(Sample Sizes in Parentheses)

ALL CRIMES

Negative Exponential:	Coefficient:	5.575107
	Exponent:	0.229466

HOMICIDE

Truncated		
Negative Exponential:	Peak likelihood	14.02%
	Peak distance	0.38 miles
	Exponent	0.064481

RAPE

Lognormal:	Mean	3.144959
	Standard Deviation	4.546872
	Coefficient	0.062791

ASSAULT

Truncated		
Negative Exponential:	Peak likelihood	27.40%
	Peak distance	0.38 miles
	Exponent	0.181738

ROBBERY

Truncated		
Negative Exponential:	Peak likelihood	9.96%
	Peak distance	0.38 miles
	Exponent	0.177651

COMMERCIAL ROBBERY

Truncated		
Negative Exponential:	Peak likelihood	4.9455%
	Peak distance	0.625 miles
	Exponent	0.151319

Table 10.3 (continued)

BANK ROBBERY

Truncated Negative Exponential:	Peak likelihood	9.96%
	Peak distance	5.75 miles
	Exponent	0.139536

BURGLARY

Truncated Negative Exponential:	Peak likelihood	20.55%
	Peak distance	0.38 miles
	Exponent	0.162907

AUTO THEFT

Truncated Negative Exponential:	Peak likelihood	4.81%
	Peak distance	0.63 miles
	Exponent	0.212508

LARCENY

Truncated Negative Exponential:	Peak likelihood	4.76%
	Peak distance	0.38 miles
	Exponent	0.193015

ARSON

Truncated Negative Exponential:	Peak likelihood	38.99%
	Peak distance	0.38 miles
	Exponent	0.093469

Data Set Definition

The steps are relatively easy. First, the user defines a calibration data set with both origin and destination locations. Figure 10.11 illustrates this process. As with the primary and secondary files, the routine reads *ArcView* 'shp', *dBase* 'dbf', Ascii 'txt', and *MapInfo* 'dat' files. For both the origin location (e.g., the home residence of the offender) and the destination location (i.e., the crime location), the names of the variables for the X and Y coordinates must be identified as well as the type of coordinate system and data unit (see chapter 3). In the example, the origin locations has variable names of HomeX and HomeY and the destination locations has variable names of IncidentX and IncidentY for the X and Y coordinates of the two locations respectively. However, any name is acceptable as long as the two locations are distinguished.

The user should specify whether there are any missing values for these four fields (X and Y coordinates for both origin and destination locations). By default, *CrimeStat* will ignore records with blank values in any of the eligible fields or records with non-numeric values (e.g.,alphanumeric characters, #, *). Blanks will always be excluded unless the user selects <none>. There are 8 possible options:

1. <blank> fields are automatically excluded. This is the default
2. <none> indicates that no records will be excluded. If there is a blank field, *CrimeStat* will treat it as a 0
3. 0 is excluded
4. –1 is excluded
5. 0 and –1 indicates that both 0 and -1 will be excluded
6. 0, -1 and 9999 indicates that all three values (0, -1, 9999) will be excluded

Any other numerical value can be treated as a missing value by typing it (e.g., 99)Multiple numerical values can be treated as missing values by typing them, separating each by commas (e.g., 0, -1, 99, 9999, -99).

The program will calculate the distance between the origin location and the destination location for each record. If the units are spherical (i.e., lat/lon), then the calculations use spherical geometry; if the units are projected (either meters or feet), then the calculations are Euclidean (see chapter 3 for details).

Kernel Parameters

Next, the user must define the kernel parameters for calibration. There are five choices that have to be made (Figure 10.12):

1. The method of interpolation. As with the two-dimensional kernel technique described in chapter 8, there are five possible kernel functions:

Figure 10.11: Jtc Calibration Data Input

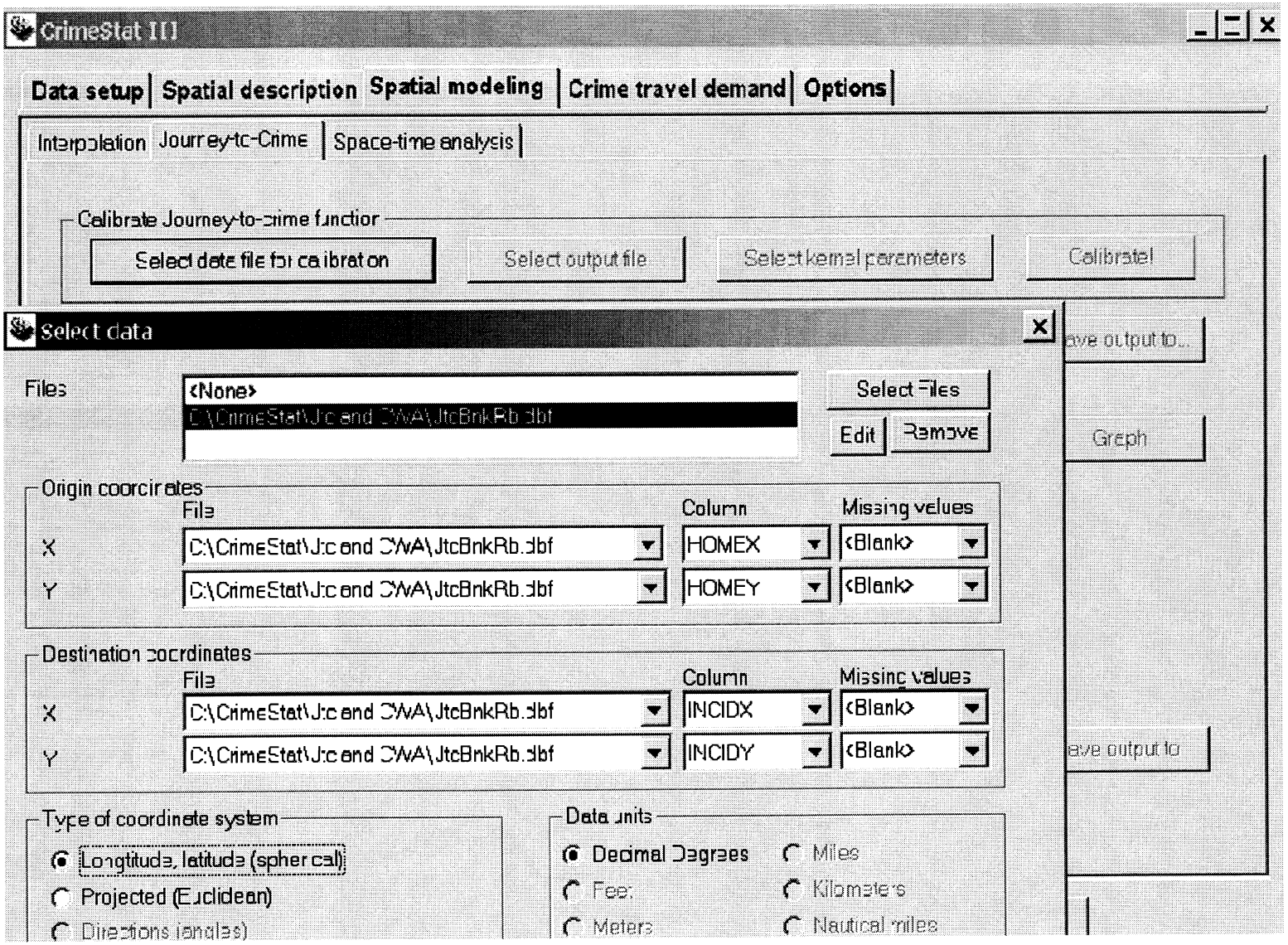
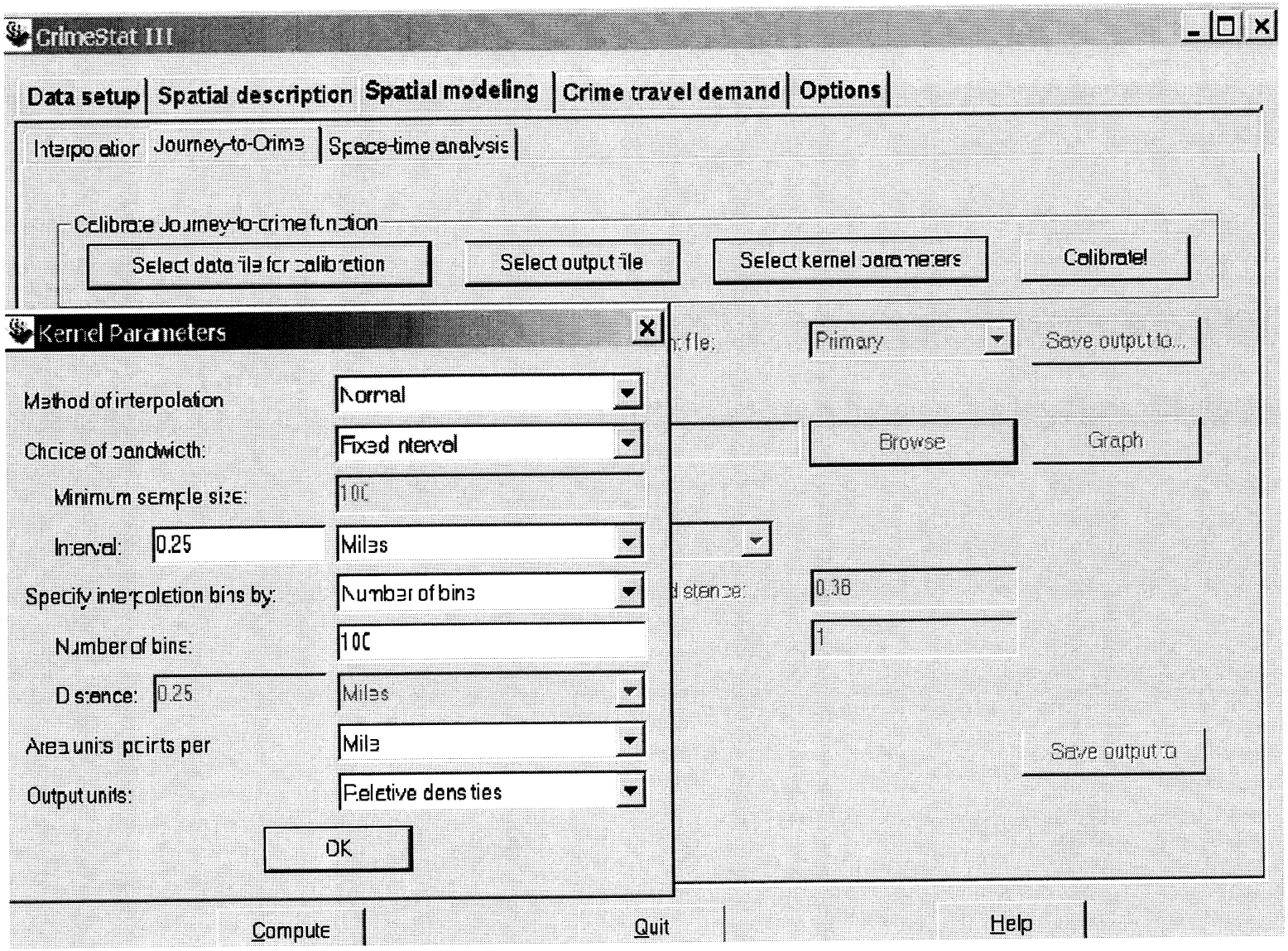


Figure 10.12: Jtc Calibration Kernel Parameters



A. Normal (the default);
B. Quartic;
C. Triangular (conical);
D. A negative exponential (peaked); and
E. A uniform (flat) distribution.

2. Choice of bandwidth. The bandwidth is the width of the kernel function. For a normal kernel, it is the standard deviation of the normal distribution whereas for the other four kernels (quartic, triangular, negative exponential, and uniform), it is the radius of the circle defined by the kernel. As with the two-dimension kernel technique, the bandwidth can be fixed in length or adaptive (variable in length). However, for the one-dimensional kernel, the fixed bandwidth is the default since an even estimate over an equal number of intervals (bins) is desirable. If the fixed bandwidth is selected, the interval size must be specified and the units (in miles, kilometers, feet, meters, and nautical miles). The default is 0.25 mile intervals. If the adaptive bandwidth is selected, the user must identify the minimum sample size that the bandwidth should incorporate; in this case, the bandwidth is widened until the specified sample size is counted.

3. The number of interpolation bins. The bins are the intervals along the distance scale (from 0 up to the maximum distance for a journey to crime trip) and are used to estimate the density function. There are two choices. First, the user can specify the number of intervals (the default choice with 100 intervals). In this case, the routine calculates the maximum distance (or longest trip) between the origin location and the destination location and divides it by the specified number of intervals (e.g., 100 equal-sized intervals). The interval size is dependent on the longest trip distance measured. Second, the user can specify the distance between bins (or the interval size). The default choice is 0.25 miles, but another value can be entered. In this case, the routine counts out intervals of the specified size until it reaches the maximum trip distance.

4. The output units. The user specifies the units for the density estimate (in units per mile, kilometer, feet, meters, and nautical miles).

5. The output calculations. The user specifies whether the output results are in probabilities (the default) or in densities. For probabilities, the sum of all kernel estimates will equal 1.0. For densities, the sum of all kernel estimates will equal the sample size.

Saved Calibration File

Third, the user must define an output file to save the empirically determined function. The function is then used in estimating the likely home residence of a particular

function. The choices are to save the file as a 'dbf' or Ascii text file. The saved file then can be used in the Jtc routine. Figure 10.13 illustrates the output file format.

Calibrate

Fourth, the calibrate button runs the routine. A calibration window appears and indicates the progress of the calculations. When it is finished, the user can view a graph illustrating the estimated distance decay function (Figure 10.14). The purpose is to provide quick diagnostics to the user on the function and selection of the kernel parameters. While the graph can be printed, it is not a high quality print. If a high quality graph is needed, the output calibration file should be imported into a graphics program.

Examples from Baltimore County

Let's illustrate this method by showing the results for the same data sets that were calculated above in the mathematical section (figures 10.4-10.8). In all cases, the normal kernel function was used. The bandwidth was 0.25 miles except for the bank robbery data set, which had only 176 cases, and the homicide data set, which only had 137 cases; because of the small sample sizes, a bandwidth of 0.50 miles was used for these two data sets. The interval width selected was a distance of 0.25 miles between bins (0.5 miles for bank robberies and homicides) and probabilities were output.

Figure 10.15 shows the kernel estimate for all crimes (41,426 trips). A frequency distribution was calculated for the same number of intervals and is overlaid on the graph. It was selected to be comparable to the mathematical function (see figure 10.4). Note how closely the kernel estimate fits the data compared to the negative exponential mathematical function. The fit is good for every value but the peak value; that is because the kernel averages several intervals together to produce an estimate.

Figure 10.16 shows the kernel estimate for larceny thefts. Again, the kernel method produces a much closer fit as a comparison with figure 10.5 will show. Figure 10.17 shows the kernel estimate for vehicle thefts. Figure 10.18 shows the kernel estimate for bank robberies and figure 10.19 shows the kernel estimate for homicides. An inspection of these graphs shows how well the kernel function fits the data, compared to the mathematical function, even when the data are irregularly spaced (in vehicle thefts, bank robberies, and homicides). Figure 10.20 compares the distance decay functions for homicides committed against strangers compared to homicides committed against known victims.

In short, the Jtc calibration routine allows a much closer fit to the data than any of the simpler mathematical functions. While it's possible to produce a complex mathematical function that will fit the data more closely (e.g., higher order polynomials), the kernel method is much simpler to use and gives a good approximation to the data.

Figure 10.3:

Journey to Crime Travel Demand Functions

Five Mathematical Functions

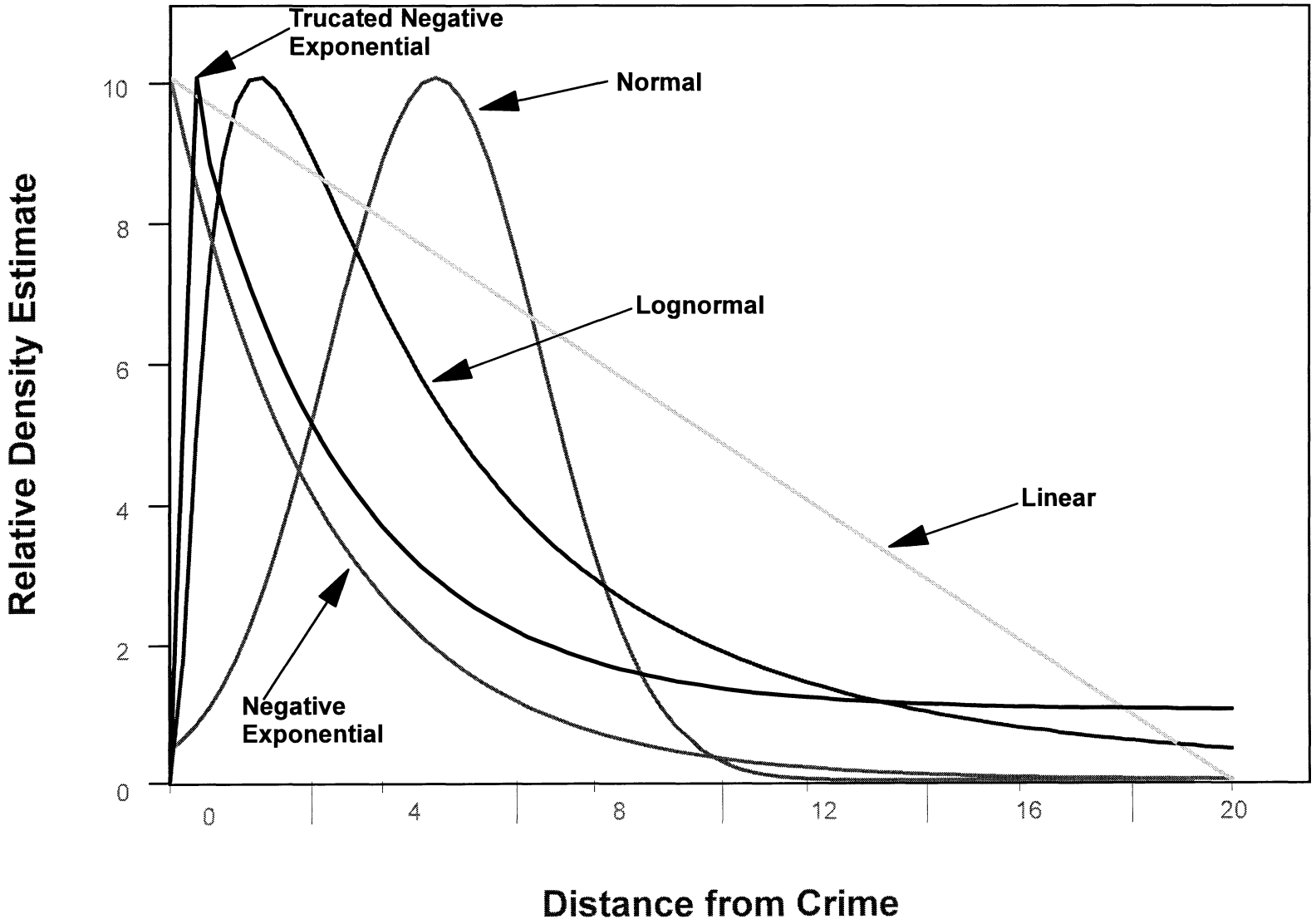


Figure 10.14: Jtc Calibration Graphic Output

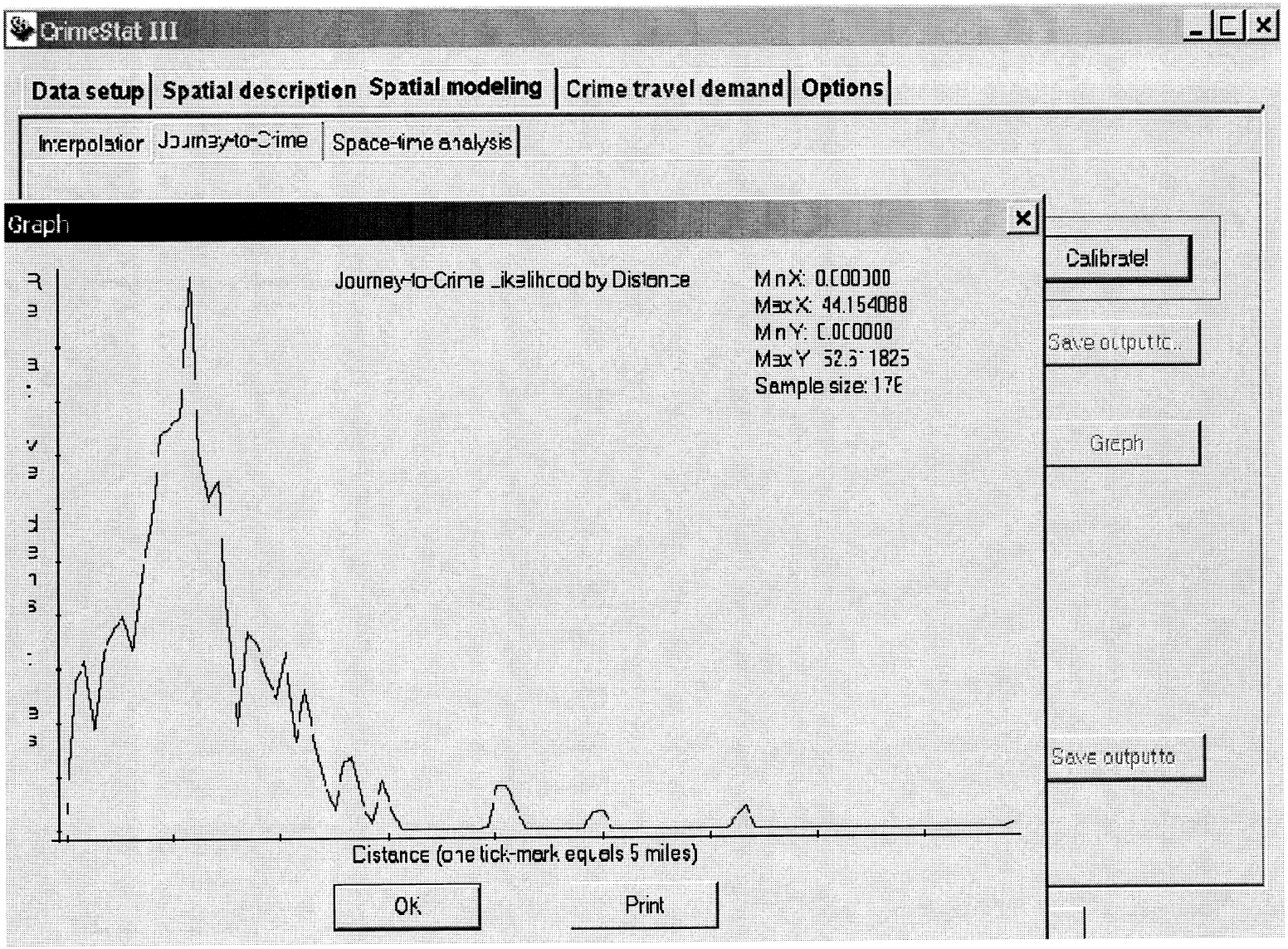


Figure 10.15:

Journey to Crime Distances: All Crimes

Frequencies and Kernel Density Estimate

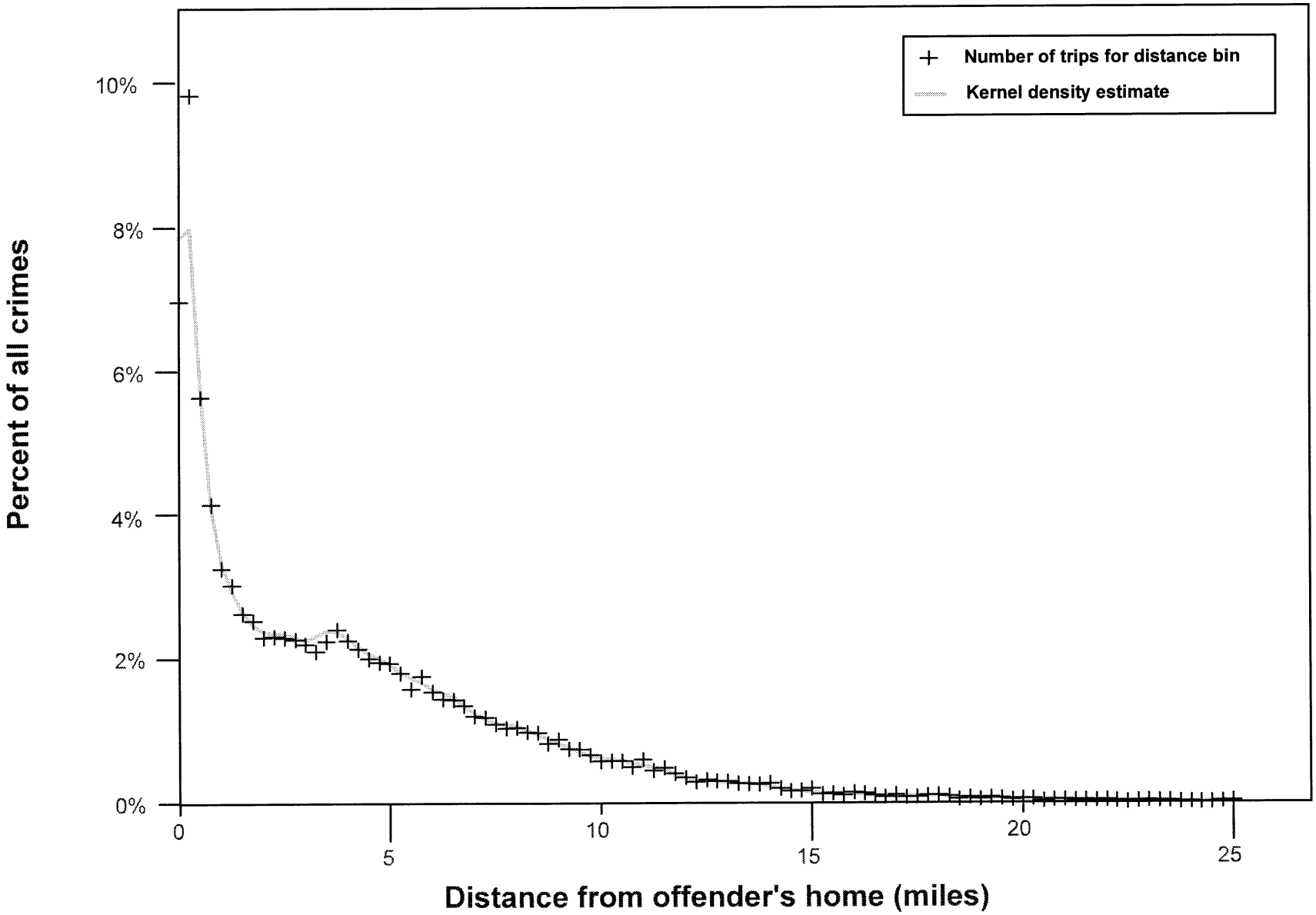


Figure 10.16:

Journey to Crime Distances: Larceny

Frequencies and Kernel Density Estimate

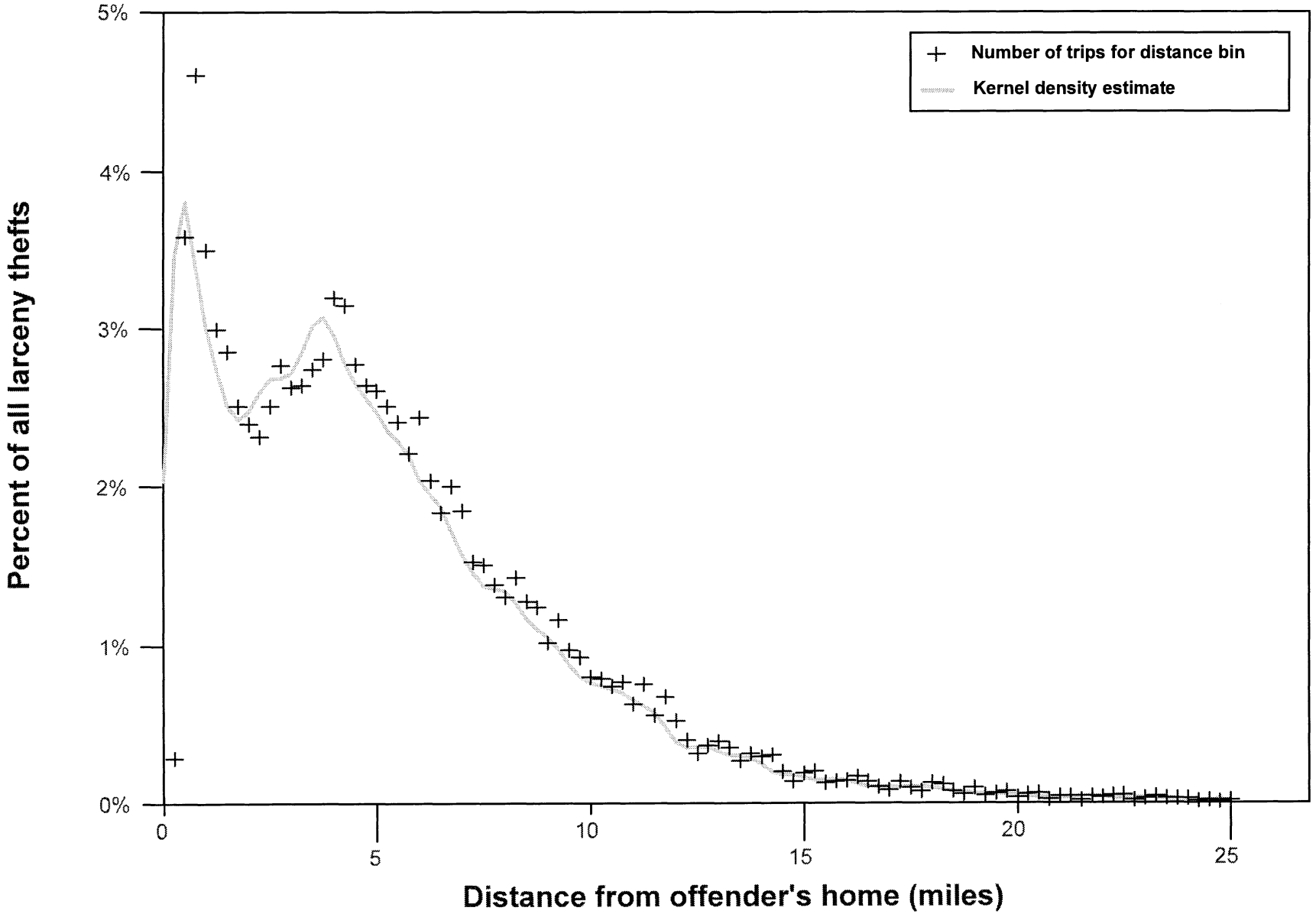


Figure 10.17:

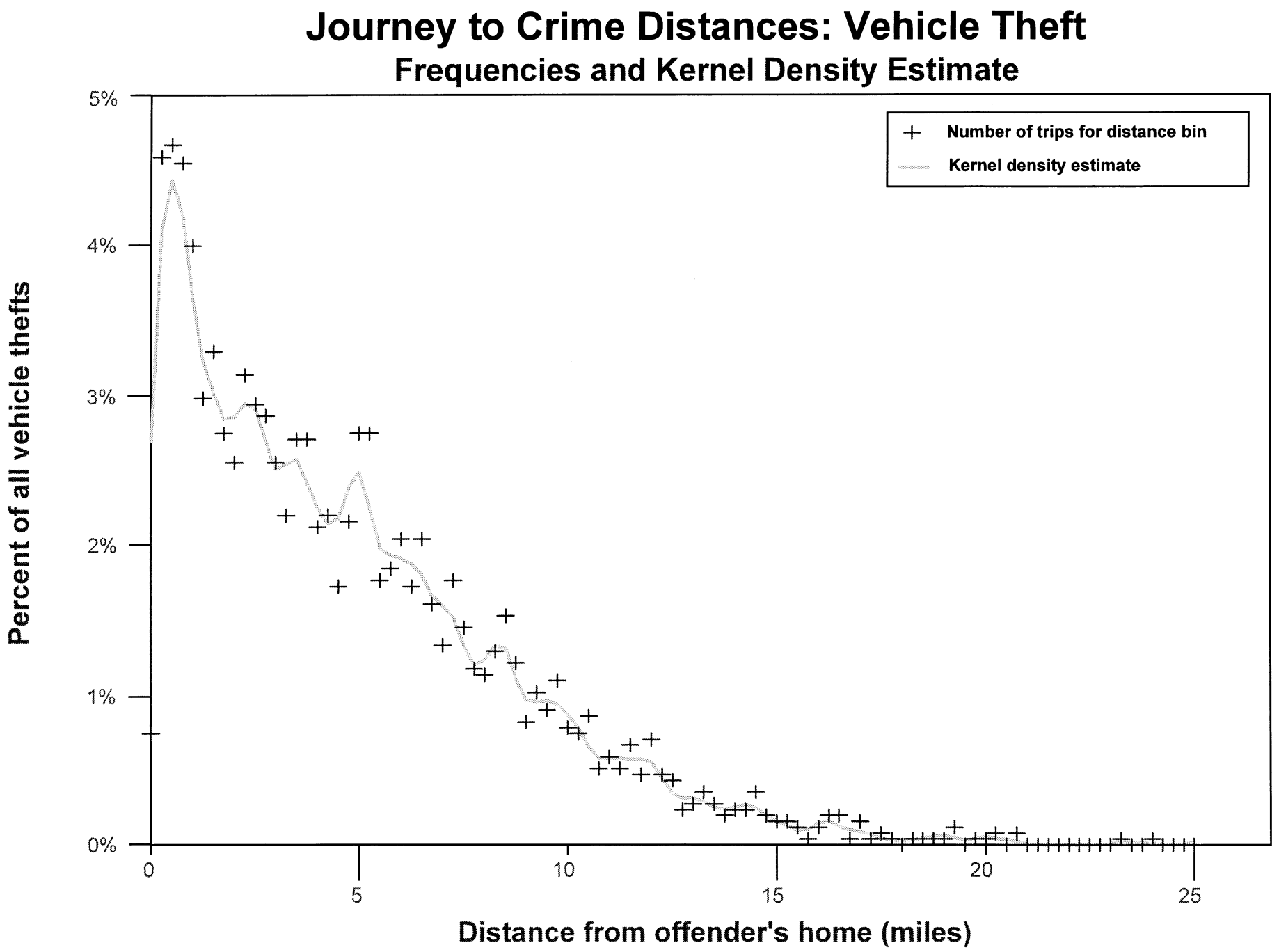


Figure 10.18:

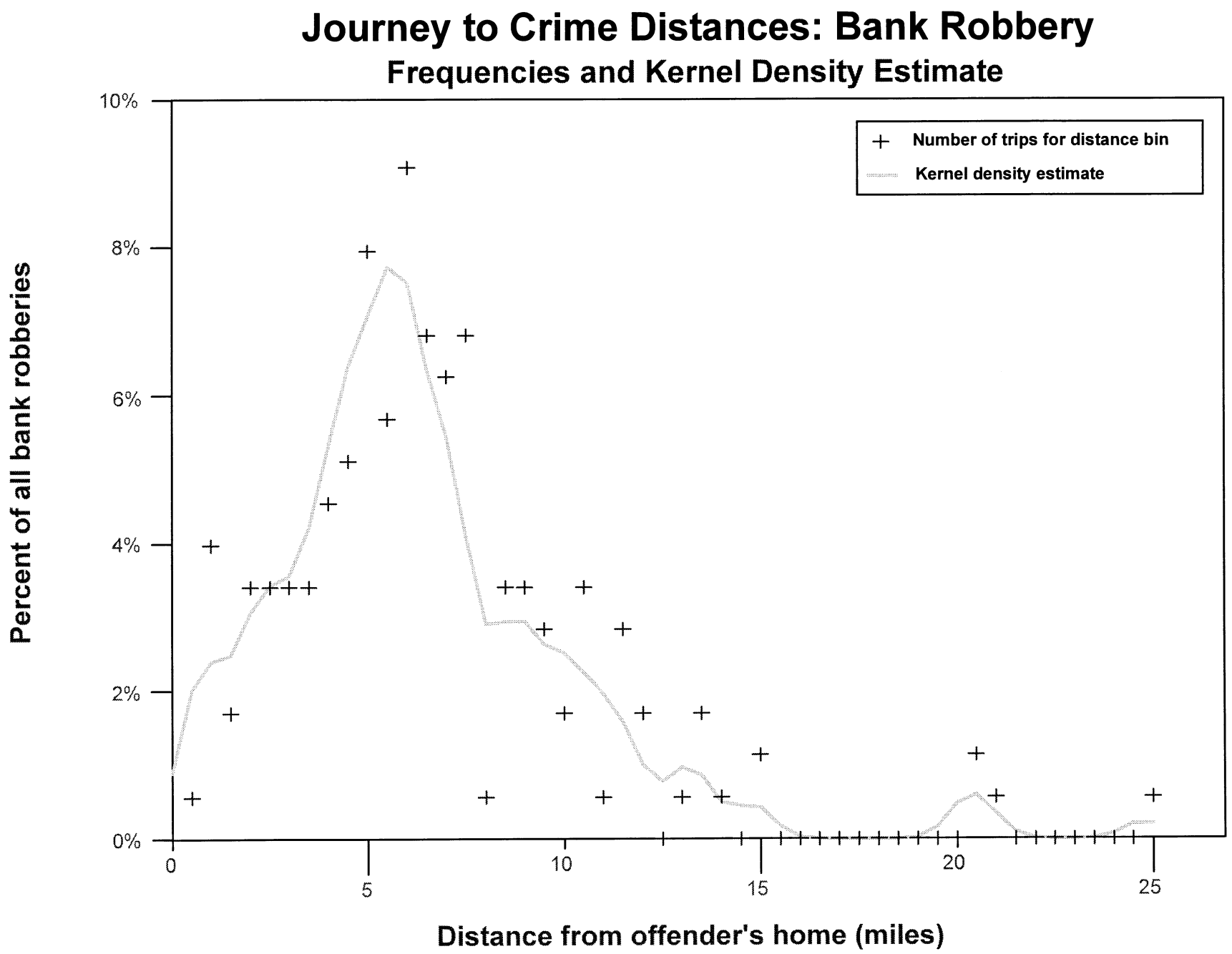


Figure 10.19:

Journey to Crime Distances: Homicide

Frequencies and Kernel Density Estimate

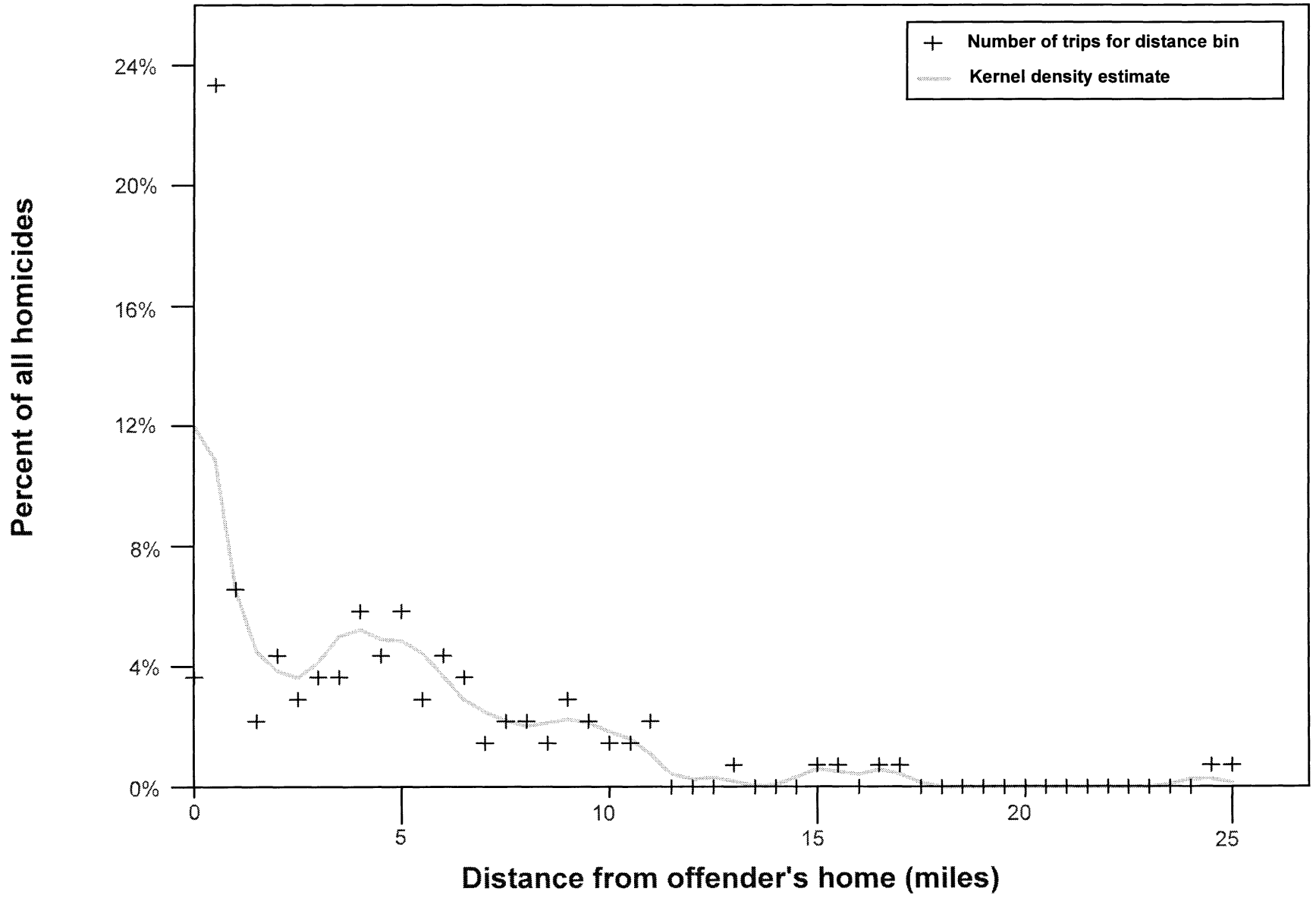
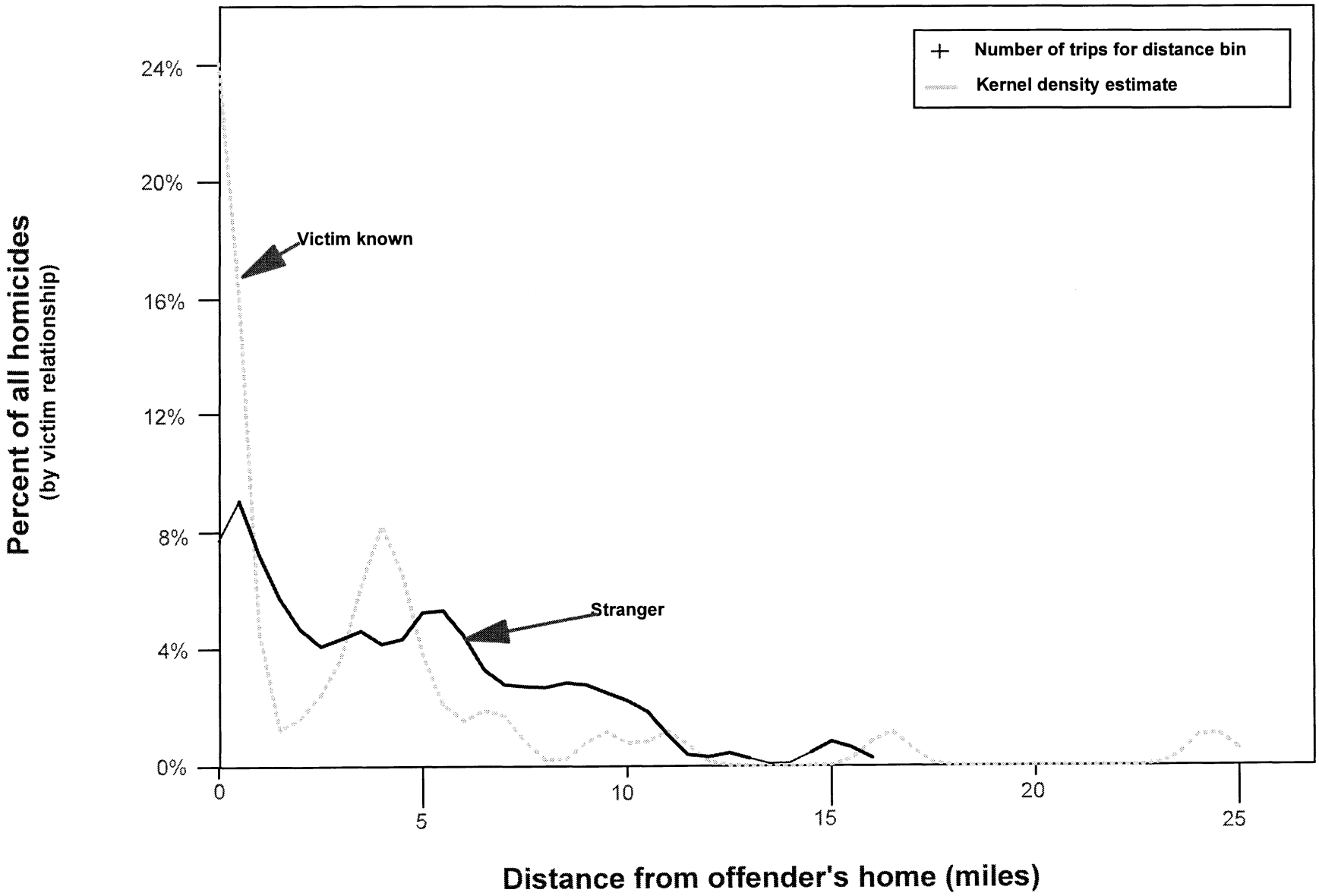


Figure 10.20:

Journey to Crime Distances: Homicide by Victim Relationship

Frequencies and Kernel Density Estimate

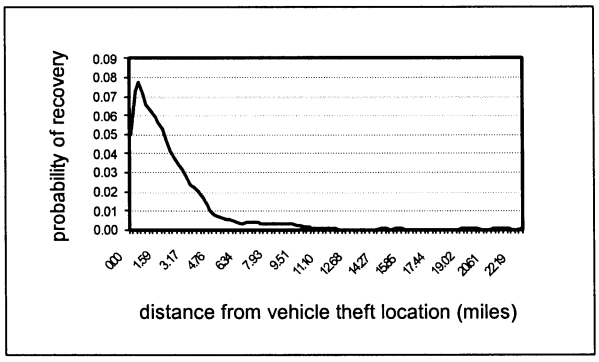


Using Journey-To-Crime Routine for Journey-After-Crime Analysis

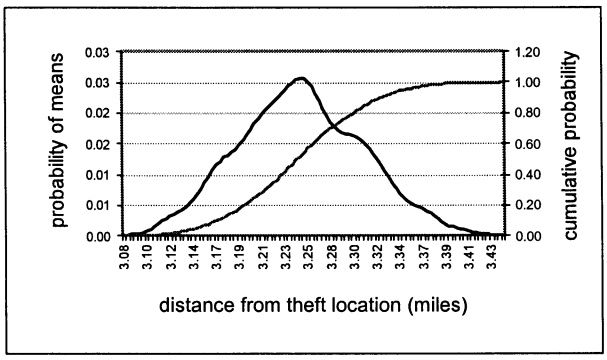
Yongmei Lu
Department of Geography
Southwest Texas State University
San Marcos, TX

The study of vehicle theft recovery locations can fill a gap in the knowledge about criminal travel patterns. Although the journey-to-crime routine of *CrimeStat* was designed to analyze the distance between offense location and offender's residential location, it can be used to describe the distance between vehicle theft location and the corresponding recovery location.

There were more than 3000 vehicle thefts in the City of Buffalo in 1998. Matching the offenses with vehicle recoveries in the same year, 1600 location pairs were identified for a journey-after-vehicle-theft analysis. To evaluate the randomness of the distances, 1000 groups of simulations were conducted. Every group contains 1600 simulated trips of journey-after-vehicle-theft. The results indicate that 1) short distances dominate journey-after-vehicle-theft, and 2) the observed trips are significantly shorter than the random trips given the distribution of possible vehicle theft and recovery locations.



Probability of recovering a stolen vehicle by distance from vehicle theft location



Distribution of mean distances of simulated vehicle theft-recovery location pairs.

Using Journey to Crime for Different Age Groups of Offenders

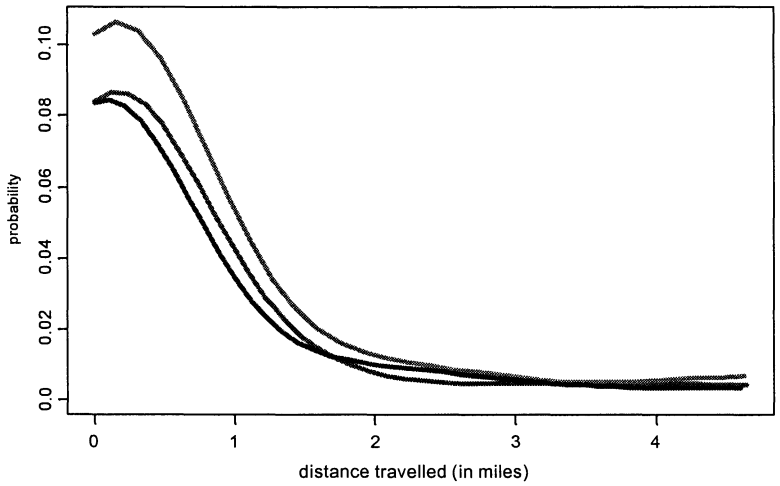
Renato Assunção, Cláudio Beato, Bráulio Silva
CRISP, Universidade Federal de Minas Gerais , Brazil

CrimeStat offers a method for analysing the distance between the crime scene and the residence of the offender using the journey to crime routine within the spatial modeling module. We analysed homicide incidents in Belo Horizonte, a Brazilian city of 2 million inhabitants, for the period January 1996 – December 2000. We used 496 homicide cases for which the police identified an offender who was living in Belo Horizonte, and for which both the crime location and offender residence could be identified. The cases were divided into three groups according to the offender's age: 1) 14 to 24 (N=201); 2) 25 to 34 (N=176); and 3) 35 or older (N=119). The journey to crime calibration routine was used to produce a probability curve *P(d)* that gives the approximate chance of an offender travelling approximately distance *d* to commit the crime.

We used the normal kernel, a fixed bandwidth of 1000 meters, 100 output bins, and the probability (or proportion of all points) option, rather than densities. This is to allow comparisons between the three age groups since they have different number of homicides. We tested for each age group separately and directed the output to a text file to analyse the three groups simultaneously.

The green, blue, and purple curves are associated with the 14-24, 25-34, 35+ year olds respectively. There are more similarities than differences between the groups. Most homicides are committed near to the residence of the offenders with between 60% t o 70% closer than one mile from their home. However, the curve does not vanish totally even for large distances because there are around 15% of offenders, of any age group, travelling longer than 3 miles to commit the crime. The oldest offenders travel longer distances, on average, followed by the youngest group, with the 25-34 year olds travelling the shortest distances.

Journey to homicide probabilities in Belo Horizonte, Brazil



The Journey to Crime Routine Using the Calibrated File

After the distance decay function has been calibrated and saved as a file, the file can be used to calculate the likelihood surface for a serial offender. The user specifies the name of the already-calibrated distance function (as a 'dbf' or an Ascii text file) and the output format. As with the mathematical routine, the output can be to *ArcView*, *MapInfo*, *Atlas*GIS*, *Surfer for Windows*, *Spatial Analyst*, and as an Ascii grid file which can be read by many other GIS packages. All but *Surfer for Windows* require that the reference grid be created by *CrimeStat*.

The result is produced in three steps:

1. The routine calculates the distance between each reference cell of the grid and each incident location;

2. For each distance measured, the routine looks up the calculated value from the saved calibration file; and

3. For each reference grid cell, it sums the values of all the incidents to produce a single likelihood estimate.

Application of the Routine

To illustrate the techniques, the results of the two methods on a single case are compared. The case has been selected because the routines accurately estimate the offender's residence. This was done to demonstrate how the techniques work. In the next section, I'll ask the question about how accurate these methods are in general.

The case involved a man who had committed 24 offenses. These included 13 thefts, 5 burglaries, 5 assaults, and one rape. The spatial distribution was varied; many of the offenses were clustered but some were scattered. Since there were multiple types of crimes committed by this individual, a decision had to be made over which model to use to estimate the individual's residence. In this case, the theft (larceny) model was selected since that was the dominant type of crime for this individual.

For the mathematical function, the truncated negative exponential was chosen from table 10.3 with the parameters being:

Peak likelihood	4.76%
Peak distance	0.38 miles
Exponent	0.193015

For the kernel density model, the calibrated function for larceny was selected (figure 10.16).

Figure 10.21 shows the results of the estimation for the two methods. The output is from *Surfer for Windows* (Golden Software, 1994). The left pane shows the results of the mathematical function while the right pane shows the results for the kernel density function. The incident locations are shown as circles while the actual residence location of the offender is shown as a square. Since this is a surface model, the highest location has the highest predicted likelihood.

In both cases, the models predicted quite accurately. The discrepancy (error) between the predicted peak location and the actual residence location was 0.66 miles for the mathematical function and 0.36 miles for the kernel density function. For the mathematical model, the actual residence location (square) is seen as slightly off from the peak of the surface whereas for the kernel density model the discrepancy from the peak cannot be seen.

Nevertheless, the differences in the two surfaces show distinctions. The mathematical model has a smooth decline from the peak likelihood location, almost like a cone. The kernel density model, on the other hand, shows a more irregular distribution with a peak location followed by a surrounding 'trough' followed a peak 'rim'. This is due to the irregular distance decay function calibrated for larceny (see figure 10.16). But, in both cases, they more or less identify the actual residence location of the offender.

Choice of Calibration Sample

The calibration sample is critical for either method. Each method assumes that the distribution of the serial offender will be similar to a sample of 'like' offenders. Obviously, distinctions can be made to make the calibration sample more or less similar to the particular case. For example, if a distance decay function of all crimes is selected, then a model (of either the mathematical or kernel density form) will have less differentiation than for a distance decay function from a specific type of crime. Similarly, breaking down the type of crime by, say, mode of operation or time of day will produce better differentiation than by grouping all offenders of the same type together. This process can be taken on indefinitely until there is too little data to make a reliable estimate. An analyst should try to find as close a calibration sample to the actual as is possible, given the limitations of the data.

For example, in our calibration data set, there were 4,694 burglary incidents where both the offender's home residence and the incident location were known. The approximate time of the offense for 2,620 of the burglaries was known and, of these, 1,531 occurred at night between 6 pm and 6 am. Thus, if a particular serial burglar for whom the police are interested in catching tends to commit most of his burglaries at night, then choosing a calibration sample of nighttime burglars will generally produce a better estimate than by grouping all burglars together. Similarly, of the 1,531 nighttime burglaries, 409 were committed by individuals who had a prior relationship with the victim. Again, if the analysts suspect that the burglar is robbing homes of people he knows or is acquainted with, then selecting the subset of nighttime burglaries committed against a known victim

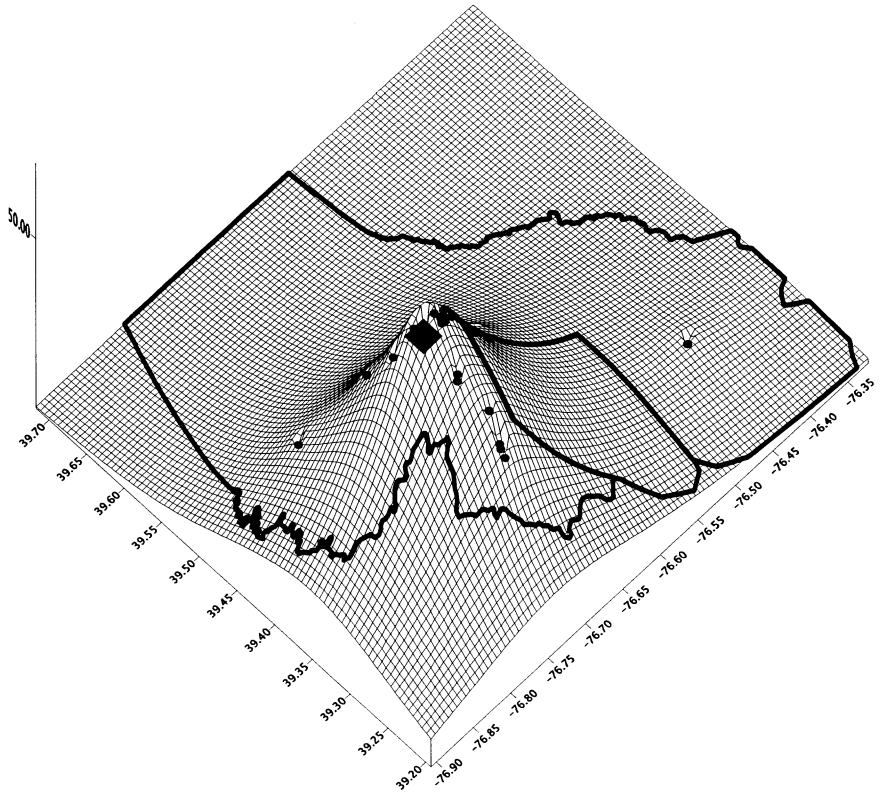
Figure 10.21:

Predicted and Actual Location of Serial Thief

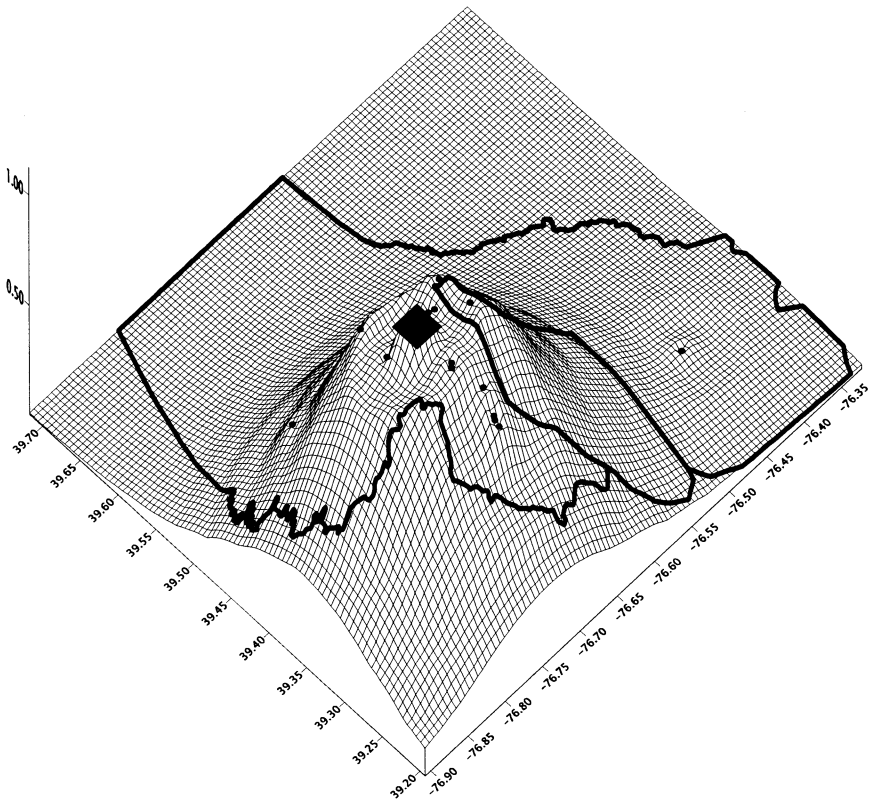
Man Charged with 24 Offenses in Baltimore County

Predicted with Mathematical and Kernel Density Models for Larceny

Residence Location = Square
Crime Locations = Circles



Mathematical Model:
Truncated Negative Exponential



Kernel Density Model

would produce even better differentiation in the model than taking all nighttime burglars. However, eventually, with further sub-groupings there will be insufficient data.

This point has been raised in a recent debate. Van Koppen and De Keijser (1997) argued that a distance decay function that combined multiple incidents committed by the same individuals could distort the estimated relationship compared to selecting incidents committed by different individuals.[6] Rengert, Piquero and Jones (1999) argued that such a distribution is nevertheless meaningful. In our language, these are two different sub-groups - persons committing multiple offenses compared to persons committing only one offense. Combining these two sub-groups into a single calibration data set will only mean that the result will have less differentiation in prediction than if the sub-groups were separated out.

Actually, there is not much difference, at least in Baltimore County. From the 41,426 cases, 18,174 were committed by persons who were only listed once in the database while 23,251 offenses were committed by persons who were listed two or more times (7,802 individuals). Categorizing the 18,174 crimes as committed by 'single incident offenders and the 23,251 crimes as committed by 'multiple incident offenders', the density distance decays functions were calculated using the kernel density method (Figure 10.22).

The distributions are remarkably similar. There are some subtle differences. The average journey to crime trip distance made by a single incident offender is longer than for multiple incident offenders (4.6 miles compared to 4.0 miles, on average); the difference is highly significant ($p \leq .0001$), partly because of the very large sample sizes. However, a visual inspection of the distance decay functions shows they are similar. The single incident offenders tend to have slightly more trips near their home, slightly fewer for distances between about a mile up to three miles, and slightly more longer trips. But, the differences are not very large.

There are several reasons for the similarity. First, some of the 'single incident offenders' are actually multiple incident offenders who have not been charged with other incidents. Second, some of the single incident offenders are in the process of becoming multiple incident offenders so their behavior is probably similar. Third, there may not be a major difference in travel patterns by the number of offenses an individual commits, certainly compared to the major differences by type of crime (see graphs above). In other words, the distinction between a single offender crime trip and a multiple offender crime trip is just another sub-group comparison and, apparently, not that important. Nevertheless, it is important to choose an appropriate sample from which to estimate a likely home base location for a serial offender. The method depends on a similar sample of offenders for comparison.

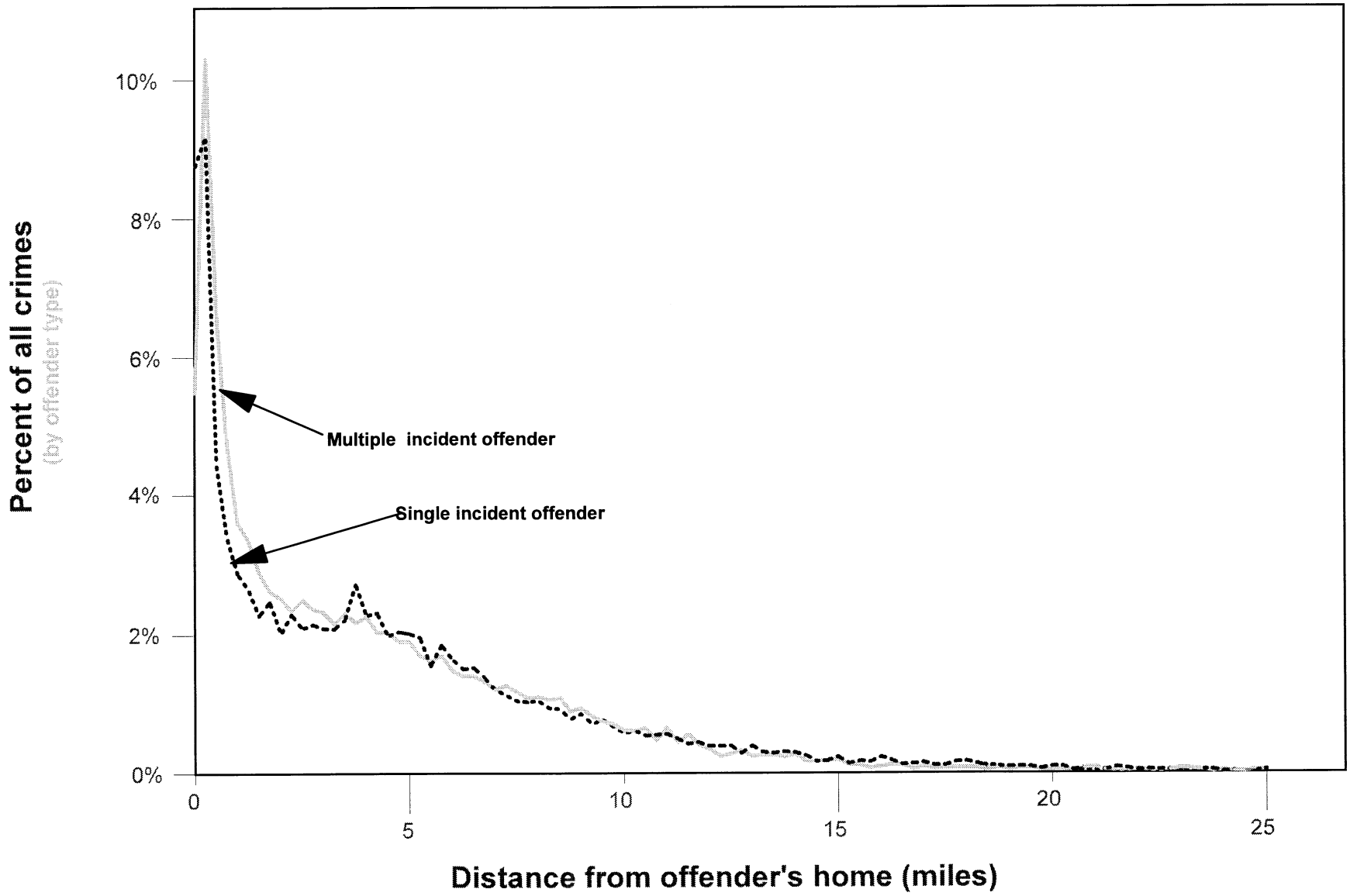
Sample Data Sets for Journey to Crime Routines

Three sample data sets from Baltimore County have been provided for the journey to crime routine. The data sets are simulated and do not represent real data. The first file - JtcTest1.dbf, are 2000 simulated robberies while the second file - JtcTest2.dbf, are 2500

Figure 10.22:

Journey to Crime Distances

Kernel Density Estimate of Single and Multiple Incident Offenders



Hot Spot Verification in Auto Theft Recoveries

Bryan Hill
Glendale Police Department
Glendale, AZ

We use *CrimeStat* as a verification tool to help isolate clusters of activity when one application or method does not appear to completely identify a problem. The following example utilizes several *CrimeStat* statistical functions to verify a recovery pattern for auto thefts in the City of Glendale (AZ). The recovery data included recovery locations for the past 6 months in the City of Glendale which were geocoded with a county-wide street centerline file using *ArcView*.

First, a spatial density "grid" was created using *Spatial Analyst* with a grid cell size of 300 feet and a search radius of 0.75 miles for the 307 recovery locations. We then created a graduated color legend, using standard deviation as the classification type and the value for the legend being the *CrimeStat* "Z" field that is calculated.

In the map, the K-means (red ellipses), Nnh (green ellipses) and *Spatial Analyst* grid (red-yellow grid cells) all showed that the area was a high density or clustering of stolen vehicle recoveries. Although this information was not new, it did help verify our conclusion and aided in organizing a response

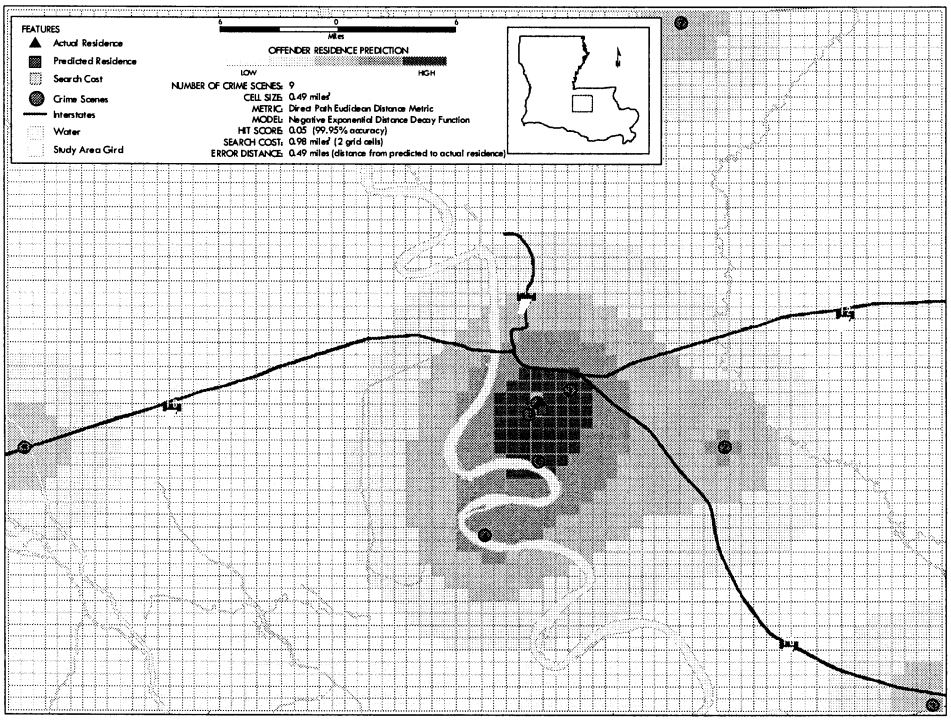
Constructing Geographic Profiles Using the *CrimeStat* Journey-To-Crime Routine

Josh Kent,
Michael Leitner,
Louisiana State University
Baton Rouge, LA

The map below shows a geographic profile constructed from nine crime sites associated with a Baton Rouge serial killer, Sean Vincent Gillis, who was apprehended on April 29, 2004 at his residence in Baton Rouge. Eight of the nine are body dump sites and the ninth is a point of fatal encounter. All crime sites were located in the City of Baton Rouge and surrounding parishes. Gillis's hunting style can best be described as that of a typical 'localized marauder'.

The Journey-to-crime routine, implemented in *CrimeStat*, was applied to simulate the travel characteristics of Gillis to and from the known crime sites. Gillis's travel behavior was calibrated with different mathematical functions that were derived from the known travel patterns of 301 homicide cases in Baton Rouge.

The profile was estimated using Euclidean distance and the negative exponential distance decay function. It predicts the actual residence of Gillis extremely accurately. The straight-line error distance between the predicted and the actual residence is only 0.49 miles. The proportion of the entire study area that must be searched in order to successfully identify the serial offender's residence is 0.05% (approximately 0.98 square miles out of a 2094.75 square miles study area).



simulated burglaries. Both files have coordinates for an origin location (HomeX, HomeY) and a destination location (IncidentX, IncidentY). Users can use the calibration routine to calculate the travel distances between the origins and the destinations. A third data set - Serial1.dbf, are simulated incident locations for a serial offender. Users can use the Jtc estimation routine to identify the likely residence location for this individual. In running this routine, a reference grid needs to be overlaid (see chapter 3). For Baltimore County, appropriate coordinates for the lower-left corner are -76.91^0 longitude and 39.19^0 latitude and for the upper-right corner are -76.32^0 longitude and 39.72^0 latitude.

Draw Crime Trips

The Journey to Crime module includes one utility that can help visualize the pattern before selecting a particular estimation model. This is a Draw Crime Trips routine that simply draws lines between the origin and destination of individual crime trips. The X and Y coordinates of an origin and destination location are input and the routine draws a line in *ArcView* 'shp', *MapInfo* 'mif', *Atlas*GIS* 'bna' or Ascii format.

Figure 10.23 illustrates the drawing of the known travel distances for 444 rape cases for which the residence location of the rapist was known. Of the 444 cases, 113 (or 25.5%) occurred in the residence of the rapist. However, for the remaining 331 cases, the rape location was not the residence location. As seen, many of the trips are of quite long distances. This would suggest the use of an journey to crime function that has many trips at zero distance but with a more gradual decay function.

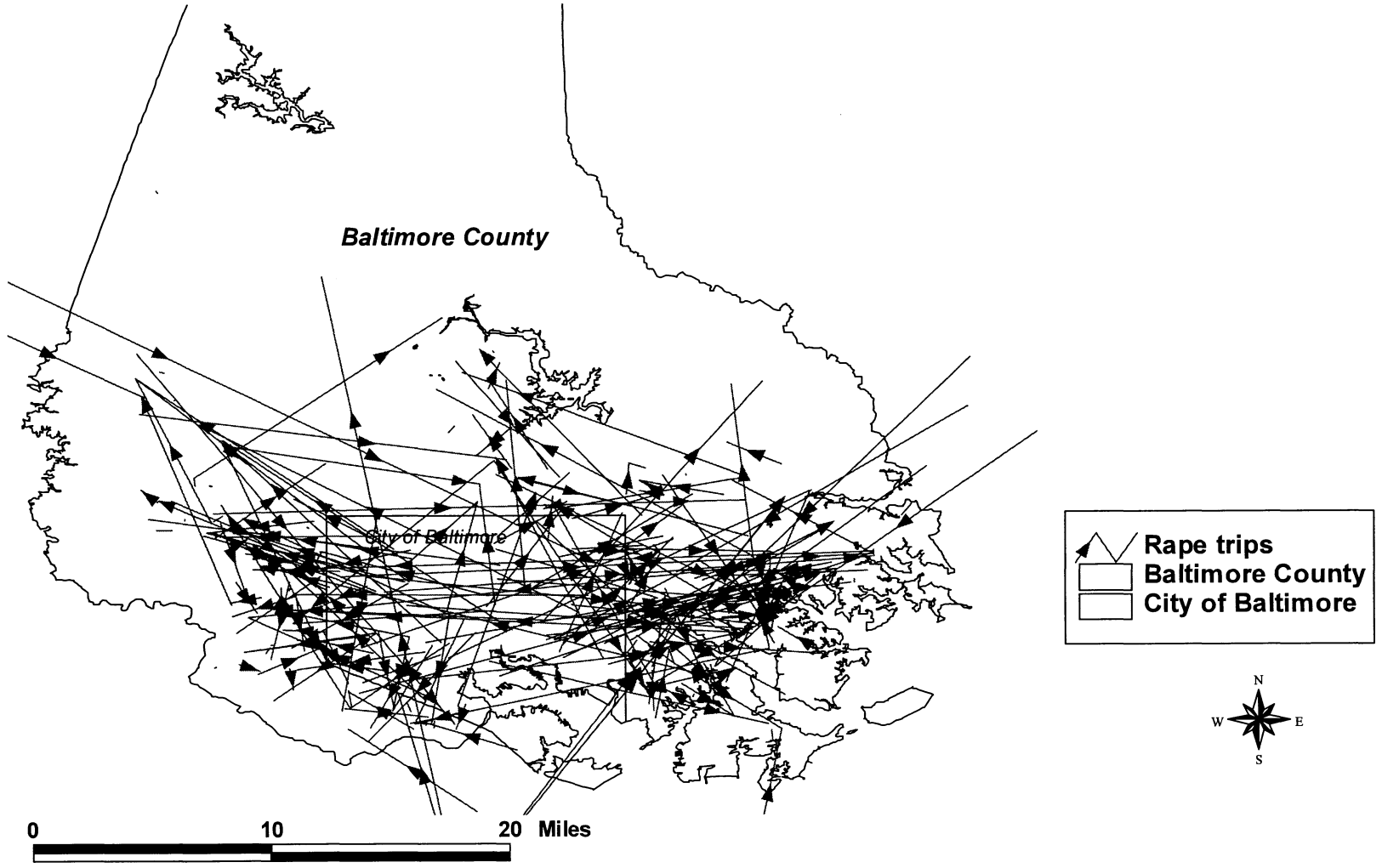
How Accurate are the Methods?

A critical question is how accurate are these methods? The journey to crime model is just that, a model. Whether it involves using a mathematical function or an empirically-derived one, the assumption in the Jtc routine is that the distribution of incidents will provide information about the home base location of the offender. In this sense, it's not unlike the way most crime analysts will work when they are trying to find a serial offender. A typical approach will be to plot the distribution of incidents and routinely search a geographic area in and around a serial crime pattern, noting offenders who have an arrest history matching case attributes (MO, type weapon, suspect description, etc.). Because a high proportion of offenses are committed within a short distance of offender residence's, the method can frequently lead to their apprehension. But, in doing this method, the analysts are not using a sophisticated statistical model.

Test Sample of Serial Offenders

To explore the accuracy of the approach, a small sample of 50 serial offenders was isolated from the database and used as a target sample to test the accuracy of the methods. The 50 offenders accounted for 520 individual crime incidents in the database. To test the Jtc method systematically, the following distribution was selected (table 10.4). The sample was not random, but was selected to produce a balance in the number of incidents

Figure 10.23:
Journey to Rape Trips
Distance from Residence to Rape Location for 444 Known Offenders



committed by each individual and to, roughly, approximate the distribution of incidents by serial offenders. Each of the 50 offenders was isolated as a separate file so that each could be analyzed in *CrimeStat*.

Identifying the Crime Type

Each of the 50 offenders was categorized by a crime type. Only two of the offenders committed the same crime for all their offenses; most committed two or more different types of crimes. Arbitrarily, each offender was typed according to the crime type that he/she most frequently committed; in the two cases where there was a tie between two crime types, the most severe was selected (i.e., personal crime over property crime). While I recognize that there is arbitrariness in the approach, it seemed a practical solution. Any error in categorizing an offender would be applicable to all the methods. The crime types for the 50 offenders approximately mirrored the distribution of incidents: larceny (29); vehicle theft (7); burglary (5); robbery (5); assault (2); bank robbery (1); and arson (1).

Identifying the Home Base and Incident Locations

In the database, each of the offenders was listed as having a residence location. For the analysis, this was taken as the *origin* location of the journey to crime trip. Similarly, the incident location was taken as the *destination* for the trip. Operationally, the crime trip is taken as the distance from the origin location to the destination location. However, it is very possible that some crime trips actually started from other locations. Further, many of these individuals have moved their residences over time; we only have the last known residence in the database. Unfortunately, there was no other information in the digital database to allow more accurate identification of the home location. In other words, there may be, and probably are, numerous errors in the estimation of the journey to crime trip. However, these errors would be similar across all methods and should not affect their relative accuracy.

Evaluated Methods

Ten methods were compared in estimating the likely residence location of the offenders. Four of the methods used the Jtc routines and six were simple spatial distribution methods (table 10.5).

The mean center and center of minimum distance are discussed in chapter 4. The center of minimum distance, in particular, is more or less the geographic center of distribution in that it ignores the values of particular locations; thus, locations that are far away from the cluster (extreme values)

Table 10.4

Serial Offenders Used in Accuracy Evaluation

Number of Offenders	Number of Crimes Committed by Each Person
4	3
4	4
4	5
4	6
4	7
4	8
3	9
3	10
3	11
2	12
2	13
2	14
2	15
1	16
1	17
1	18
1	19
1	20
1	21
1	22
1	24
1	33
50	520

have no effect on the result. The directional mean and triangulated mean is part of the directional mean routine, discussed in chapter 4 and in the update release notes; the routine has now been modified so that it can be used with ordinary X/Y coordinates. The geometric and harmonic means are discussed in the update release notes; they are both means which discount extreme values.

Table 10.5

Comparison Methods for Estimating the Home Base of a Serial Offender

Journey to Crime Methods

Mathematical model for all crimes

Mathematical model for specific crime type

Kernel density model for all crimes

Kernel density model for specific crime type

Spatial Distribution Methods

Mean center

Center of minimum distance

Directional mean (weighted) calculated with 'lower left corner' as origin

Triangulated mean

Geometric mean

Harmonic mean

The Test

Each of these ten methods were run against each of the files created for the serial offenders. For the six 'means' (mean center, geometric mean, harmonic mean, directional mean, triangulated mean, center of minimum distance), the mean was itself the best guess for the likely residence location of the offender. For the four journey to crime functions, the grid cell with the highest likelihood estimate was the best guess for the likely residence location of the offender.

Measurement of Error

For each of the 50 offenders, error was defined as the distance in miles between the 'best guess' and the actual location. For each offender, the distance between the estimated home base (the 'best guess') and the actual residence location was calculated using direct distances. Table 10.6 presents the results. The data show the error by method for each of

Table 10.6

Accuracy of Methods for Estimating Serial Offender Residences

(N= 50 Serial Offenders)

Dataset	Number of Crimes	Primary Crime Type	Mean Center Error (miles)	Center of Minimum Distance Error (miles)	Triangulated Mean Error (miles)	Geometric Mean Error (miles)	Harmonic Mean Error (miles)	Jtc Kernel: All Crimes Error (miles)	Jtc Kernel: Crime Type Error (miles)	Jtc Math: All Crimes Error (miles)	Jtc Math: Crime Type Error (miles)	Average Error	All Methods Minimum Error	Maximum Error
3A	3	Larceny	**31.5991**	32.4477	32.4109	31.5995	31.6000	32.7824	32.7880	32.7824	32.7880	**32.3109**	**31.5991**	**32.7880**
3B	3	Larceny	13.2303	12.1683	24.1531	13.2311	13.2319	**10.7526**	14.4929	**10.7526**	11.2501	**13.6959**	**10.7526**	**24.1531**
3C	3	Bank robbery	2.8348	0.9137	2.7767	2.8335	2.8322	**0.6775**	5.8416	**0.6775**	6.0946	**2.8313**	**0.6775**	**6.0946**
3D	3	Burglary	2.9733	3.2603	6.1013	2.9728	**2.9724**	4.6038	3.3883	3.3882	3.7931	**3.7170**	**2.9724**	**6.1013**
4A	4	Vehicle theft	4.2436	4.2670	**3.8217**	4.2436	4.2436	4.2527	4.2364	4.2527	4.2590	**4.2022**	**3.8217**	**4.2670**
4B	4	Larceny	1.9618	0.3100	2.0563	1.9621	1.9623	0.3125	**0.2018**	0.3125	0.2784	**1.0397**	**0.2018**	**2.0563**
4C	4	Larceny	4.4733	4.4733	4.6789	4.4733	4.4733	4.9681	4.3563	**4.2637**	4.3563	**4.5018**	**4.2637**	**4.9681**
4D	4	Assault	0.2925	0.1905	**0.0466**	0.2925	0.2926	0.0703	0.0703	0.0703	0.4560	**0.1979**	**0.0466**	**0.4560**
5A	5	Larceny	17.3308	16.6459	17.8985	17.3292	17.3276	**15.9738**	17.8655	15.9739	16.4526	**16.9775**	**15.9738**	**17.8985**
5B	5	Larceny	1.3609	0.2481	1.7733	1.3586	1.3564	**0.2068**	0.6974	0.5140	0.6974	**0.9126**	**0.2068**	**1.7733**
5C	5	Larceny	2.2458	2.6832	16.4518	2.2450	2.2442	2.7886	2.4205	2.7886	3.0922	**4.1067**	**2.2442**	**16.4518**
5D	5	Larceny	0.9169	0.2250	0.2371	0.9171	0.9174	**0.1577**	0.4267	**0.1577**	0.4267	**0.4869**	**0.1577**	**0.9174**
6A	6	Larceny	5.1837	5.2081	7.9621	5.1837	5.1837	5.1271	**4.8554**	4.9393	5.2256	**5.4298**	**4.8554**	**7.9621**
6B	6	Vehicle theft	1.3720	1.1869	**0.9625**	1.3710	1.3700	3.1126	2.3800	1.3566	2.0831	**1.6883**	**0.9625**	**3.1126**
6C	6	Larceny	1.3199	0.3157	1.7928	1.3192	1.3184	**0.2580**	0.5272	**0.2580**	0.5272	**0.8485**	**0.2580**	**1.7928**
6D	6	Larceny	3.2458	2.3324	6.5209	3.2431	3.2405	**1.2506**	2.6253	1.9718	1.9718	**2.9336**	**1.2506**	**6.5209**
7A	7	Larceny	3.9023	3.4185	**2.3176**	3.9022	3.9021	2.7419	3.0532	3.1364	3.0532	**3.2697**	**2.3176**	**3.9023**
7B	7	Larceny	12.4100	9.2973	14.8293	12.4107	12.4115	**8.5357**	8.6148	**8.5357**	8.8275	**10.6525**	**8.5357**	**14.8293**
7C	7	Burglary	5.0501	7.1477	10.8567	5.0481	**5.0460**	7.9975	7.9975	7.9975	7.6274	**7.1965**	**5.0460**	**10.8567**
7D	7	Larceny	2.2686	0.7733	75.7424	2.2684	2.2682	**0.0892**	0.7191	**0.0892**	0.7191	**9.4375**	**0.0892**	**75.7424**
8A	8	Larceny	6.0298	**6.0165**	6.2653	6.0264	6.0229	8.4210	6.2962	6.2022	6.1166	**6.3774**	**6.0165**	**8.4210**
8B	8	Larceny	**1.0041**	1.1437	2.1776	1.0042	1.0042	1.7475	1.3510	1.5298	1.3510	**1.3681**	**1.0041**	**2.1776**
8C	8	Larceny	1.3059	1.6944	1.3684	1.3043	1.3027	2.1513	**1.2020**	2.1513	1.8707	**1.5946**	**1.2020**	**2.1513**
8D	8	Vehicle theft	3.5794	2.3780	5.5915	3.5809	3.5825	**0.5900**	1.3340	1.9133	1.3340	**2.6537**	**0.5900**	**5.5915**
9A	9	Robbery	5.2527	5.7156	**4.8574**	5.2529	5.2532	7.8257	7.1961	6.2520	5.9265	**5.9480**	**4.8574**	**7.8257**
9B	9	Larceny	8.1923	10.6555	**6.9916**	8.1886	8.1850	12.4578	10.3957	12.4578	12.0514	**9.9529**	**6.9916**	**12.4578**
9C	9	Robbery	3.7778	3.8454	11.0042	3.7758	**3.7738**	4.9015	5.1862	4.6206	4.3445	**5.0255**	**3.7738**	**11.0042**
10A	10	Larceny	0.9358	0.5159	1.1003	0.9355	0.9353	**0.0606**	0.3720	0.2601	0.7172	**0.6481**	**0.0606**	**1.1003**
10B	10	Larceny	2.8581	3.4940	14.2219	2.8536	**2.8491**	6.4051	6.5709	10.3095	6.4758	**6.2264**	**2.8491**	**14.2219**
10C	10	Larceny	0.8052	**0.7251**	5.5938	0.8050	0.8049	0.9059	0.8404	0.9060	1.2786	**1.4072**	**0.7251**	**5.5938**
11A	11	Vehicle theft	**2.9127**	3.2715	3.1192	2.9130	2.9134	3.6936	3.4335	3.4282	3.2087	**3.2104**	**2.9127**	**3.6936**
11B	11	Robbery	0.3250	0.3250	0.2513	0.3250	0.3250	0.4235	**0.2263**	0.4235	0.7011	**0.3695**	**0.2263**	**0.7011**
11C	11	Vehicle theft	1.2689	1.7157	1.4750	1.2709	1.2729	2.8945	**0.6984**	2.8945	2.2049	**1.7440**	**0.6984**	**2.8945**
12A	12	Larceny	3.3881	4.2334	10.9241	3.3867	3.3852	6.4050	**3.2639**	5.5843	5.2132	**5.0871**	**3.2639**	**10.9241**
12B	12	Larceny	0.5562	0.5361	2.8003	0.5562	0.5562	0.7897	0.6709	0.7897	0.9631	**0.9132**	**0.5361**	**2.8003**
13A	13	Larceny	6.3282	7.2857	**6.0244**	6.3248	6.3213	7.6438	7.4607	7.6438	7.9915	**7.0027**	**6.0244**	**7.9915**
13B	13	Assault	**1.4943**	**1.4943**	1.5279	1.4944	1.4944	1.6501	1.5954	1.6501	2.0824	**1.6092**	**1.4943**	**2.0824**
14A	14	Larceny	1.9363	0.8706	1.4498	1.9365	1.9368	0.3434	0.6058	**0.2596**	0.7631	**1.1224**	**0.2596**	**1.9368**
14B	14	Arson	0.6898	0.3727	0.8086	0.6899	0.6900	**0.3359**	**0.3359**	**0.3359**	0.6213	**0.5422**	**0.3359**	**0.8086**
15A	15	Vehicle theft	0.7282	0.7189	**0.3362**	0.7277	0.7271	0.8155	0.4855	0.8155	1.5128	**0.7630**	**0.3362**	**1.5128**
15B	15	Robbery	0.4914	0.4914	0.8254	0.4914	0.4914	0.6468	0.5693	0.6468	0.6546	**0.5898**	**0.4914**	**0.8254**
16A	16	Vehicle theft	2.1107	2.0995	8.2311	2.1107	2.1107	**1.5957**	1.6404	2.5911	2.4033	**2.7659**	**1.5957**	**8.2311**
17A	17	Burglary	1.6484	0.3093	1.0227	1.6461	1.6438	**0.2879**	**0.2879**	**0.2879**	0.5268	**0.8512**	**0.2879**	**1.6484**
18A	18	Larceny	0.6308	0.4196	1.0876	0.6329	0.6349	**0.2132**	0.3383	**0.2132**	0.6985	**0.5410**	**0.2132**	**1.0876**
19A	19	Larceny	**8.6462**	9.4195	8.6772	8.6486	8.6511	10.2869	9.2708	9.7022	9.5548	**9.2064**	**8.6462**	**10.2869**
20A	20	Burglary	6.3520	5.7969	28.3094	6.3486	6.3452	**0.5934**	0.8673	**0.5934**	0.7945	**6.2223**	**0.5934**	**28.3094**
21A	21	Burglary	1.2396	0.8861	1.2776	1.2393	1.2390	0.5243	0.5243	1.0253	**0.4965**	**0.9391**	**0.4965**	**1.2776**
22A	22	Larceny	3.6828	2.6232	**2.0949**	3.6803	3.6777	2.4937	2.8944	2.4937	2.8944	**2.9484**	**2.0949**	**3.6828**
24A	24	Larceny	1.7959	0.5892	2.3033	1.7975	1.7991	**0.2658**	0.3574	0.4222	0.6587	**1.1099**	**0.2658**	**2.3033**
33A	33	Robbery	3.9901	5.0481	7.2505	3.9940	3.9979	7.9485	7.6939	8.1907	7.9439	**6.2286**	**3.9901**	**8.1907**
		Median Error =	**2.5517**	**2.2159**	**3.4704**	**2.5509**	**2.5502**	**1.9494**	**2.0102**	**2.0615**	**2.1440**			
		Mean Error =	**4.0434**	**3.8441**	**7.6472**	**4.0429**	**4.0424**	**4.0395**	**4.0305**	**4.0163**	**4.1467**			
		SD Error =	**5.2166**	**5.3845**	**12.0642**	**5.2166**	**5.2166**	**5.5678**	**5.6237**	**5.5398**	**5.4177**			

the 50 offenders. The three right columns show the average error of all methods and the minimum error and maximum errors obtained by a method. The method with the minimum error is boldfaced; for some cases, two or three methods are tied for the minimum. The bottom three rows show the median error, the average error and the standard deviation of the errors for each method across all 50 offenders.

There are a number of conclusions from the results. First, the degree of precision for any of these methods varies considerably. The precision of the estimates vary from a low of 0.0466 miles (about 246 feet) to a high of 75.7 miles. The overall precision of the methods is not very high and is highly variable. There are a number of possible reasons for this, some of which have been discussed above. Each of the methods produces a single parameter from what is, essentially, a probability distribution whereas the distribution of many of these incidents are widely dispersed. Few of the offenders had such a concentrated pattern that only a single location was possible. Since these are probability distributions, not everyone follows the 'central tendency'. Also, some of these offenders may have moved during the period indicated by the incidents, thereby shifting the spatial pattern of incidents and making it difficult to identify the last residence.

A second conclusion is that, for any one offender, the methods produce similar results. For many of the offenders the difference between the best estimate (the minimum error) and the worst estimate (the maximum error) is not great. Thus, the simple methods are generally as good (or bad) as the more sophisticated methods.

Third, across all methods, the center of minimum distance had the lowest average error. Thus, the approximate geographic center of the distribution produced as good an estimate as the more sophisticated methods. However, it wasn't particularly close (3.8441 miles, on average). The worst method was the triangulated mean; it had an average error of 7.6472 miles. The triangulated mean is produced by vector geometry and will not necessarily capture the center of the distribution. Other than this, there were not great differences. This reinforces the point above that the methods are all, more or less, describing the central tendency of the distribution. For offenders that don't live in the center of their distribution, the error of a method will necessary be high.

Looking at each of the 50 offenders, the methods vary in their efficacy. For example, the Jtc kernel function for all crimes was the best or tied for best for 17 of the offenders, but was also the worst or tied for worst for 9. Similarly, the Jtc kernel function for the specific crimes was best or tied for best for 8 of the offenders, but worse for 4. Even the most consistent was best for 4 offenders, but also worst for one. On the other hand, the triangulated mean, which had the worst overall error, produced the best estimate for 9 of the individuals while it produced the worst estimate for 25 of the individuals. Thus, the triangulated mean tends to be very accurate or very inaccurate; it had the highest variance, by far.

Fourth, the median error is smaller than the average error. That is, the median is the point at which 50% of the cases had a smaller error and 50% had a larger error. Overall, most of the cases were found within a shorter distance than the average would

indicate. This indicates that several cases had very large errors whereas most had smaller errors; that is, they were *outliers*. Over all methods, the Jtc kernel approach for all crimes had the lowest median error (1.95 miles). In fact, all four Jtc methods had smaller median errors than the simple centrographic methods. In other words, they are more accurate than the centrographic methods most of the time. The problem in applying this logic in practice, however, is that one would not know if the case being studied is typical of most cases (in which case, the error would be relatively small) or whether it was an outlier. In other words, the median would define a search area that captured about 50% of the cases, but would be very wrong in the other 50%. If we could somehow develop a method for identifying when a case is 'typical' and when it isn't, increased accuracy will emerge from the Jtc methods. But, until then, the simple center of minimum distance will be the most accurate method.

Fifth, the amount of error varies by the number of incidents. Table 10.7 below shows the average error for each method as a function of three size classes: 1-5 incidents; 6-9 incidents; and 10 or more incidents. As can be seen, for each of the ten methods, the error decreases with increasing number of incidents. In this sense, the measured error is responsive to the sample size from which it is based. It is, perhaps, not surprising that with only a handful of incidents no method can be very precise.

Sixth, the relative accuracy of each of these methods varies by sample size. The method or methods with the minimum error are boldfaced. For a limited number of incidents (1-5), the Jtc mathematical function for all crimes (i.e., the negative exponential with the parameters from table 10.5) produced the estimate with the least error, followed by the Jtc kernel function for all crimes; the was the third best. The differences in error between these were not very great. For the middle category (6-9 incidents), the center of minimum distance produced the least error followed by the Jtc mathematical function for the specific crime type. For those offenders who had committed ten or more crimes, the Jtc kernel function for the specific crime type produced the best estimate, followed by the center of minimum distance. The two mathematical functions produced the least accuracy for this sub-group, though again the differences in error are not very big (2.2 miles for the best compared to 2.7 miles for the worst). In other words, only with a sizeable number of incidents does the Jtc kernel density approach for specific crimes produce a good estimate. It is better than the other approaches, but only slightly better than the simple measure of the center of minimum distance.

Search Area

A number of researchers have been interested in the concept of a search area for the police (Rossmo, 2000; Canter, 2003). The concept is that the journey to crime method can define a small search area within which there is a higher probability of finding the offender. The average or median error discussed above can be used to define such a search area if treated as a radius of a circle. While intuitive, I'm not sure whether this represent a meaningful statistic. For example, taking the average error of the center of minimum distance (3.84 miles) would produce a search area of 46.4 square miles, not exactly

Table 10.7

Method Estimation Error and Sample Size: Average Error of Method by Number of Incidents (miles)

Number of Incidents	Mean Center	Center of Minimum Distance	Triangulated Mean	Geometric Mean	Harmonic Mean	Jtc Kernel: All	Jtc Kernel: Crime types	Jtc Math: All	Jtc Math: Crime types	All Methods Average Error	Minimum Error
3-5	6.9553	6.4861	9.3672	6.9160	6.9545	6.4622	7.2321	**6.3278**	6.9954	**7.0774**	**6.3278**
6-9	4.2596	**4.0753**	10.6160	4.3331	4.2576	4.4805	4.2489	4.2274	4.2020	**4.9667**	**4.0753**
10+	2.3832	2.3149	4.8136	2.4575	2.3827	2.4880	**2.2176**	2.6725	2.6243	**2.7060**	**2.2176**

a small area in which to find a serial offender. Even if we take the median error of 1.94 miles from the Jtc kernel approach for all crimes (1.94 miles) will still produce a search area of 11.9 square miles, and it would be correct only half the time

In other words, these methods are still very imprecise. Further, the error is liable to increase over time, rather than decrease. With about 50% of the U.S. population living in suburbia (Demographia, 1998) and with 90% of American households owning at least one motor vehicle (U.S. Census Bureau, 2000), the average distances traveled by offenders has probably been increasing over time since most types of trips have also shown increases in travel over time. This means that unless police can find a way to narrow down the search area considerably, the methods don't really help beyond what police intuitively do anyway, namely look near the distribution of the incidents committed by serial offenders.

Cautionary Notes

Of course, this is a limited test. It was a small sample (only 50 cases) from a single jurisdiction (Baltimore County). The sample wasn't even randomly selected, but chosen to examine the accuracy by a range of sample sizes. Thus, the conclusions are only tentative and must be seen as hypotheses for further work. Clearly, more research is needed.

Nevertheless, there are certain cautions that must be considered in using either of these journey to crime methods (the mathematical or the empirical). First, a simple technique, such as the center of minimum distance, may be as good as a more sophisticated technique. It doesn't always follow that a sophisticated method will produce any more accuracy than a simple one. For the time being, I would advise crime analysts who are trying to detect a pattern in the distribution of the incidents of a serial offender to do exactly what they have been doing, basically looking at the data and making a subjective guess about where the offender may be residing. The kernel density Jtc routine needs an adequate amount of information (i.e, at least 10 incidents) to produce somewhat precise estimates. These techniques should be seen for now as research tools rather than as diagnostics for identifying the whereabouts of an offender. They are just too imprecise and unreliable to depend on, at least until more definitive results are obtained.

Second, there are other limitations to the technique. The model must be calibrated for each individual jurisdiction. Further, it must be periodically re-calibrated to account for changes in crime patterns. For example, in using the mathematical model, one cannot take the parameters estimated for Baltimore County (Table 10.3) and apply them to another city or if using the kernel density method take the results found at one time period and assume that they will remain indefinitely. The model is a probability model, not a guarantee of certainty. It provides guesses based on the similarity to other offenders of the same type of crime. In this sense, a particular serial offender may not be typical and the model could actually orient police wrongly if the offender is different from the calibration sample. It will take insight by the investigating officers to know whether the pattern is typical or not.

Third, as a theoretical model, the journey to crime approach is quite simple. It is based on a distribution of incidents and an assumed travel distance decay function. From the perspective of modeling the travel behavior of offenders, it is limited. As mentioned above, the method does not utilize information on the distribution of target opportunities nor does it utilize information on the travel mode and route that an offender takes. It is purely a statistical model. The research area of geographic profiling attempts to go beyond statistical description and understand the cognitive maps that offenders use as well as how these interact with their motives. This is good and should clearly guide future research. But it has to be understood that the theory of offender travel behavior is not very well developed, certainly compared to other types of travel behavior. Further, some types of crime trips may not even start from an offender's residence, but may be referenced from another location, such as vehicle thefts occurring near disposal locations. Routine activity theory would suggest multiple origins for crimes (Cohen and Felson, 1979).

The existing models of travel demand used by transportation planners (which have themselves been criticized for being too simple) measure a variety of factors that have only been marginally included in the crime travel literature - the availability of opportunities, the concentration of offender types in certain areas, the mode of travel (i.e., auto, bus, walk), the specific routes that are taken, the interaction between travel time and travel route, and other factors. It will be important to incorporate these elements into the understanding of journey to crime trips to build a much more comprehensive theory of how offenders operate. Travel behavior is very complicated and we need more than a statistical distance model to adequately understand it. The next seven chapters look at an application of travel demand theory to crime travel.

Also, it's not clear whether knowing an offender's 'cognitive map' will help in prediction. There have been no evaluations that have compared a strictly statistical approach with an approach that utilizes information about the offender as he or she understands the environment. It cannot be assumed that integrating information about the perception of the environment will aid prediction. In most travel demand forecasts that transportation engineers and planners make, cognitive information about the environment is not utilized except in the definition of trip purpose (i.e., what the purpose of the trip was). The models use the actual trips by origin and destination as the basis for formulating predictions, not the understanding of the trip by the individual. Understanding is important from the viewpoint of developing theory or for ways to communicate with people. But, it is not necessarily useful for prediction. In short, understanding and prediction are not the same thing.

On the other hand, the journey to crime routine, particularly the kernel density approach, can be useful for police departments *if* used carefully. If there are sufficient cases to build an estimate (i.e., 10 or more incidents), it can provide additional information to officers investigating a serial offender by reducing the number of possible suspects that might be linked to a series of crimes. It can also provide some direction in orienting the deployment of officers and detectives investigating what appear to be serial offenses. It provides guesses about where the offender might be living, but based on similarities with previous offenders for the same type of crime. It's not going to give an exact estimate of

where an offender is living, but will provide some insights into which areas the individual might be located. The Jtc model should be seen as a supplement to other techniques, not a complete solution. Like all the statistical tools in *CrimeStat*, it must be used carefully and intelligently. The philosophy of crime analysis must always be to use a technique with thought and with a systematic procedure.

Endnotes for Chapter 10

1. It should also be pointed out that the use of direct distances will underestimate travel distances particularly if the street network follows a grid.

2. There are, of course, many other types of mathematical functions that can be used to describe a declining likelihood with distance. In fact, there are an infinite number of such functions. However, the five types of functions presented here are commonly used. We avoided the inverse distance function because of its potential to distort the likelihood relationship.

$$f(d) = \frac{1}{d_{ij}^{\,k}}$$

where *k* is a power (e.g., 1, 2, 2.5). For large distances, this function can be a useful approximation of the lessening travel interaction with distance. However, for short distances, it doesn't work. As the distance between the reference cell location and an incident location becomes very small, approaching zero, then the likelihood estimate becomes very large, approaching infinity. In fact, for $d_{ij} = 0$, the function is unsolvable. Since many distances between reference cells and incidents will be zero or close to zero, the function becomes unusable.

3. It is actually the inverse of the inverse distance function. If a distance decay function drops off proportional to the inverse of the distance,

$$Y_{ij} = A^* \, 1/d_{ij}$$

where Y_{ij} is the travel likelihood, A is coefficient, and d_{ij} is the distance from the home base, then the opposite - a distance increase is just the inverse of this function

$$Z_{ij} = \frac{1}{A^* \, 1/d_{ij}} = \frac{d_{ij}}{A} = B^* \, d_{ij}$$

4. There are several sources of error associated with the data set. First, these records were arrest records prior to a trial. Undoubtedly, some of the individuals were incorrectly arrested. Second, there are multiple offenses. In fact, more than half the records were for individuals who were listed two or more times in the database. The travel pattern of repeat offenders may be slightly different than for apparent first-time offenders (see figure 10.19). Third, many of these individuals have lived in multiple locations. Considering that many are young and that most are socially not well adjusted, it would be expected that these individuals would have multiple homes. Thus, the distribution of incidents could reflect multiple home bases, rather than one. Unfortunately, the data we have only gives a single residential location, the place at which they were living when arrested.

5. If the coordinate system is projected with the distance units in feet, meters or miles, then the distance between two points is the hypotenuse of a right triangle using Euclidean geometry.

$$d_{AB} = \sqrt{(X_A - X_B)^2 + (Y_A - Y_B)^2} \quad \text{repeat} \tag{3.1}$$

where each location is defined by an X and Y coordinate in feet, meters, or miles.

If the coordinate system is spherical with units in latitudes and longitudes, then the distance between two points is the Great Circle distance. All latitudes and longitudes are converted into radians using

$$\text{Radians } (\phi) = \frac{2\pi\,\phi}{360} \quad \text{repeat} \tag{3.2}$$

$$\text{Radians } (\lambda) = \frac{2\pi\,\lambda}{360} \quad \text{repeat} \tag{3.3}$$

Then, the distance between the two points is determined from

$$d_{AB} = 2^* \text{Arcsin}\ \{\text{Sin}^2[(\phi_B - \phi_A)/2] + \text{Cos}\ \phi_A{}^*\text{Cos}\phi_B{}^*\text{Sin}^2[(\lambda_B - \lambda_A)/2]^{1/2}\} \quad \text{repeat} \tag{3.4}$$

with all angles being defined in radians (Snyder, 1987, p. 30, 5-3a).

6. They also argued that the combination of incidents - which they called 'aggregation', would distort the relationship between distance and incidence likelihood because of the ecological fallacy. To my mind, they are incorrect on this point. Data on a distribution of incidents by distance traveled is an individual characteristic and is not 'ecological' in any way. An ecological inference occurs when data are aggregated with a *grouping* variable (e.g., state, county, city, census tract; see Langbein and Lichtman, 1978). A frequency distribution of individual crime trip distances is an individual probability distribution, similar, for example, to a distribution of individuals by height, weight, income or any other characteristic. Of course, there are sub-sets of the data that have been aggregated (similar to heights of men v. heights of women, for example). Clearly, identifying sub-groups can make better distinctions in a distribution. But, it is still an individual probability distribution. This doesn't produce bias in estimating a parameter, only variability. For example if a particular distance decay function implies that 70% of the offenders live within, say, 5 miles of their committed incidents, then 30% don't live within 5 miles. In other words, because the data are individual level, then a distance decay function, whether estimated by a mathematical or a kernel density model, is an individual probability model (i.e., an attempt to describe the underlying distribution of individual travel distances for journey to crime trips).

CrimeStat III

Part IV: Crime Travel Demand Modeling

Chapter 11
Overview of
Crime Travel Demand Modeling

The next seven chapters present a module on crime travel demand modeling. Crime travel demand modeling is a framework for examining crime trips over an entire metropolitan area. In this chapter, an overview is presented. In the next five chapters, each of the separate components of crime travel demand modeling are presented. Finally, in chapter 17, Richard Block and Dan Helms present case studies of the method applied to Chicago and Las Vegas crime data.

Much of the theoretical background was discussed in chapter 10 (journey to crime). Readers would be advised to review that material before proceeding with the crime travel demand model.

Travel Demand Forecasting

Crime travel demand modeling is an application of travel demand forecasting (or travel demand modeling). It is used by transportation planners for examining travel patterns over an entire metropolitan area and for forecasting future trends. It is a model of transportation patterns in a metropolitan area and is used for both forecasting and the analysis of the likely effects of building new roadways or installing new transit facilities. In the United States, it is required by Federal law to be used in every metropolitan area greater than 50,000 population as a basis for making decisions on highway and transit expenditures (USDOT, 2003: 23CFR450). It is also used for transportation planning in the metropolitan areas of many other countries of the world (Field and MacGregor, 1987).

The aim is to model travel over an urban area as a means for coordinating the approximately $36 billion dollars in transportation highway funds and $8.6 billion in transit funds that are spent *every* year in the U.S. (Tea3.org, 2004). Rather than waste funds by building new roadways and transit facilities that will be little used, it is a lot more effective to first model the likely benefit of a new facility as a basis for making a decision to build it. In essence, Congress requires a transportation model be developed for every metropolitan area as part of an evaluation of the benefits to be obtained from particular transportation investments.

The framework has emerged slowly since the 1950s and is now starting its "third generation". For the "first generation" - what is used by most Metropolitan Planning Organizations (MPO) today, modeling is conducted entirely at a zonal level. The "second generation" involves modeling individual level choices in travel mode taken within a zonal framework (Horowitz, Koppelman, and Lerman, 1986; McFadden, 2002), while a "third generation" involves modeling individual-level trips in a framework known as "activity-based" modeling (Goulias, 1996; Miller, 1996; Pas, 1996; FHWA, 2001a). In *CrimeStat III*, we implement a modified "first generation" model, primarily due to the lack of data on

individual-level crime trips. In later versions, we may add individual-level choice components.

Need for More Complex Travel Model of Crime

Crime travel demand modeling is an application of travel demand theory targeted specifically to crime analysis. There are many reasons why such an approach is appropriate. First, current models of criminal travel behavior are too simple with respect to travel. As chapter 10 discussed, journey to crime models assume that many offenders commit crimes in their neighborhoods. While this assumption is frequently empirically found, it is not a realistic model of modern day crime travel. Prior to World War II, Americans tended to live and shop almost exclusively in their residential community. Many people would grow up and live in a single community for most of their lives. Since World War II, however, American society has become very mobile. People move frequently, not just within metropolitan areas, but between metropolitan area. For example, between March 1999 and March 2000, 43.4 million Americans moved (Schachter, 2001): over half were within the same county and 20 percent were between different counties in the same state, but 19 percent were moves to a different state.

Second, the almost universal use of personal automobiles has increased daily mobility. For example, in the 2000 census, 90% of households owned at least one motor vehicle. For certain metropolitan areas, particularly in the west and in the south, motor vehicle ownership was greater than 92% (U.S. Census, 2002). Further, per capita vehicle travel has consistently increased over time. Since at least 1960, and probably before, the growth in vehicle miles traveled (VMT) has increased at a much faster rate than population, a trend that does not seem to be abating (FHWA, 1996; Patterson, 1998; FHWA, 2001b; BTS, 2003). Essentially, automobile use has become almost ubiquitous. There is no reason to think that offenders would not be affected by these trends. Since there is no data available that could test whether offenders are less likely to own an automobile than non-offenders, it has to be assumed that more and more offenders have access to an automobile for the use of committing a crime. Clearly, the existence of an automobile makes crime travel much more fluid and difficult to model. While offenders will probably commit crimes in locales for which they are familiar, there is no reason to think that those locales will necessarily be the communities in which they live.

Third, the widespread availability of motor vehicles has allowed major shifts in intra-urban travel patterns. In the last census (2000), approximately half the U.S population lived in areas that would normally be called 'suburbs', even though the U.S. Census Bureau does not use this nomenclature (non-central city, metropolitan population; U.S. Census Bureau, 2000; Demographia, 1998). Within metropolitan areas, approximately two-thirds of the population lives in suburban areas. Much of the community-oriented crime patterns that were described by the so-called "Chicago School of Criminology" in the 1920s and 1930s are no longer true (Burgess, 1925; Thrasher, 1927). Crimes have increased substantially in the suburbs of many metropolitan areas and the differences between the central city and suburbs is decreasing (Demographia, 1999);

Demographia, for example, estimates that 1999 crimes rates in the suburbs were about half those in the central cities.

Figure 11.1 below shows a sample of 200 crime trips in Baltimore County that occurred between 1993 and 1997. As seen, there is a complex pattern. Some of the trips are short; for some, the origin and destination are the same location. But, for other trips, the travel distances are substantial. In other words, there is a complex pattern of crime trips in Baltimore County which is not easily modeled by a simple distance decay-type function.

Fourth, an empirical examination of travel patterns shows considerable temporal variation. There are hourly variations, daily variations, and seasonal variation in crimes. Some of this can be understood as reflecting existing travel patterns in congested metropolitan areas. For example, in Baltimore County, crime travel distances were generally shorter during the peak afternoon "rush hours" (4-7 PM) than at other times. Such a pattern suggests an adaptation to traffic by offenders, a not unreasonable assumption given the difficulties of traversing a metropolitan area during peak travel times.

Fifth, crime travel behavior represents a complex pattern in itself. Especially for personal crimes, there is an interaction in the travel patterns of offenders and victims that is very difficult to even describe, least of all model. Many crimes are committed by multiple offenders and the existence of intermediate locations (e.g., 'fences' for the distribution of stolen goods, auto theft drop locations) makes crime travel even more of a complex pattern to be understood.

In short, American society has become a very mobile society, leading to larger travel distances, more frequent trips, and more complex trips. Again, offenders are going to be affected by these trends. Because of this, there is a need to understand crime patterns in terms of the complexity of travel rather than continue to rely on overly simple models of travel 'distance decay'.

Crime Travel Demand Framework

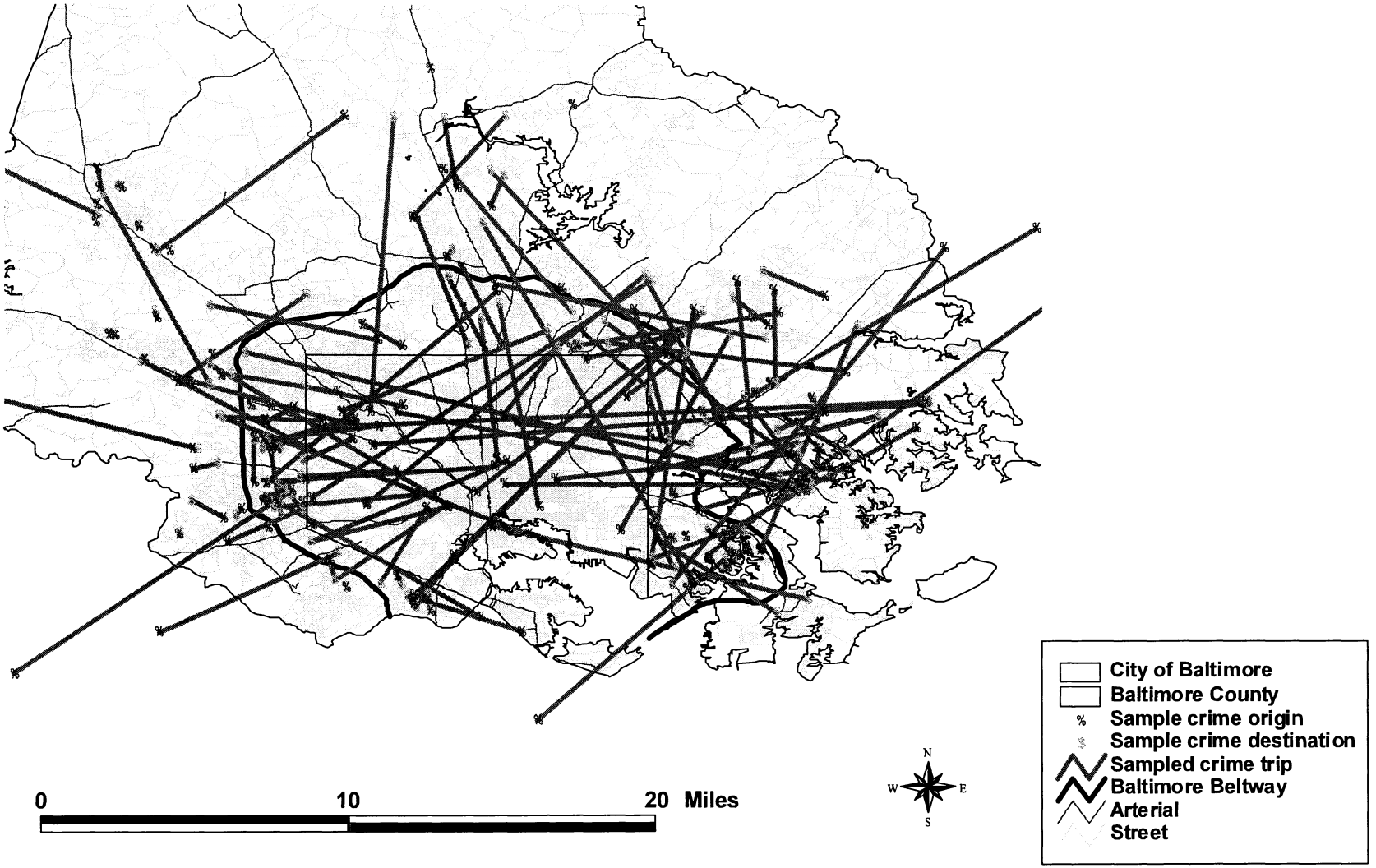
Crime travel demand theory is a framework for understanding this complexity. There are two phases:

1. An inventory (or data gathering) phase; and
2. A modeling phase.

The data gathering involves putting together the necessary data to estimate the model. This involves selecting an appropriate zone system (since the model is estimated at the zonal level), obtaining data on crime 'trips' and allocating it to the zones, obtaining zonal variables that will predict trips (both on the production side and on the attraction side), creating possible policy or policing interventions, and obtaining one or more modeling networks.[1]

Figure 11.1:

Baltimore County Crime Trips: 1993-1997
Origins and Destinations
Sample of 200 Crime Trips



The modeling phase involves four distinct modeling steps (or stages) that represent a logical 'causative' pattern:

1. **Trip generation** - separate models are produced of crime trip productions (i.e., the number of crime trips that originate in each zone) and crime trip attractions (i.e., the number of crime trips that occur in each zone). The model may include policy or intervention variables as predictors as well as socio-economic variables. One of the major uses of the model is to explore how different interventions might alter the number of trips taken.

2. **Trip distribution** - a model that predicts the number of crime trips that will begin in every production zone and will end in every attraction zone.

3. **Mode split** - a model that predicts, for each production-attraction zone pair, which travel modes will be taken (e.g., walking, bicycle, driving, bus).

4. **Network assignment** - a model that predicts, for each production-attraction zone pair by travel model, which route is liable to be taken.

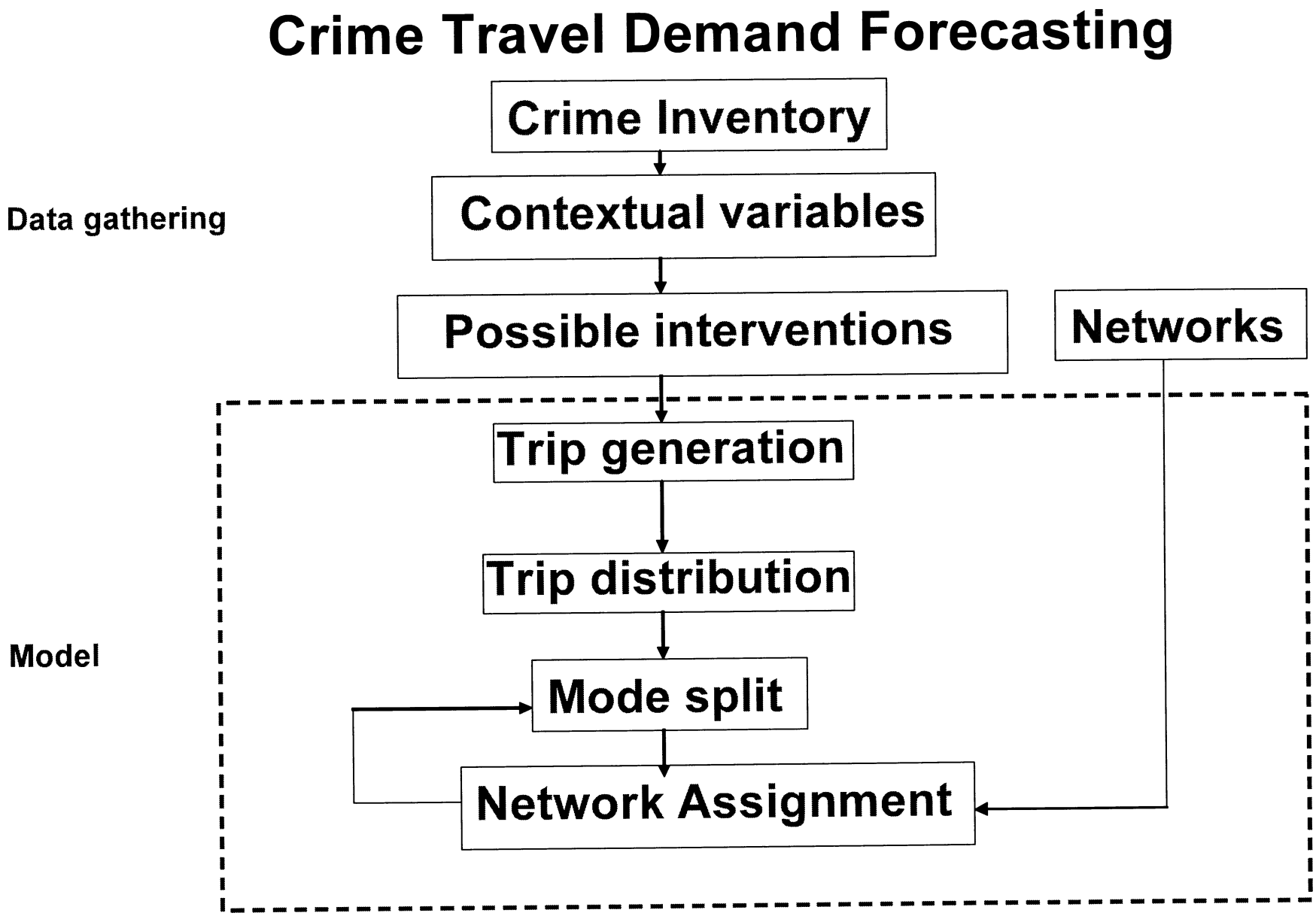
The modeling is typically sequential following these steps. The output from each stage is then used as an input for the subsequent stage. Figure 11.2 below shows the sequence.

One can think of the model as a *plausible* behavioral representation. First, someone decides to make a trip (e.g., an offender decides to commit a robbery to get some money to purchase drugs. That would be the first stage. Second, that individual decides where to go to commit the robbery. This is the second stage. Third, the individual decides how to travel to that location (walk, drive, or take the bus). This is the third stage. Finally, the individual chooses a route; in the case of walking, biking, or driving, that is a deliberate choice whereas in the case of transit trips, it is dependent on the actual bus or rail network. This is the fourth stage.

However, the analogy to behavioral decisions quickly breaks down as alternative behavioral sequences can be generated (e.g., the offender first makes a trip and then decides to commit a crime; the offender first decides to commit a crime and chooses a destination, but then commits a crime at an intermediate location in the trip). As a behavioral model, this type of framework is actually not very accurate for predicting individual behavior as a number of studies have suggested (Domencich and McFadden, 1975; Ortuzar and Willumsen, 2001).

Consequently, it's important to understand this framework as a *zonal* model, rather than a behavioral explanation. The data are aggregated at the zonal level and the model is applicable to that level. The model is good at predicting total trips in a metropolitan area and for predicting the major trip links, and should be used only at that level.

Figure 11.2:



Note in figure 11.2 that there is feedback from the network assignment stage to the mode split stage. This is a function of transit use since the choice of travel mode is dependent on the availability of an appropriate network (e.g., one cannot have train trips if there are no trains nearby).[2]

Also, crime travel demand modeling is a framework, rather than a specific theory. There is more than one way to implement the framework. In transportation modeling, there are many variations of the model and each transportation planning organization implements it in slightly different ways. For this reason, it is best thought of as a framework.

In this version of *CrimeStat*, we implement one particular version of the framework. It is a framework that is consistent and appears to produce reasonable predictions of crime travel behavior. But, clearly, it is not the only way that this could have been implemented.

The "second-" and "third-generation" travel demand models represent alternative ways of modeling travel in a metropolitan area. In the following chapters, these alternatives will be mentioned when appropriate. Nevertheless, the type of framework implemented in this version should be seen as a first step in developing a more realistic model of crime travel behavior.

Crime Travel Definitions

Let's start with two definitions.

Crime Trip

In the *CrimeStat* implementation, a **crime trip** is a round-trip journey from an offender's residence that includes a committed crime at a specified location. From a modeling perspective, the offender's residence will be considered the **origin** of the trip and the crime location will be considered the **destination**. Note that there may be intermediate trips between the origin and the destination, as figure 11.3 illustrates. But, it is assumed that at some point, the offender will return home to the initial origin. Defining a crime trip in this way avoids the issue of what was the actual origin of the trip. As mentioned in chapter 10, routine activity theory suggests that many crime trips occur while the offender is en route from some other location as part of their ordinary activity. The possibilities can become quite complex (e.g., an offender stays overnight at some other location than his/her residence and commits a crime as a part of that stay rather than while en route to home).

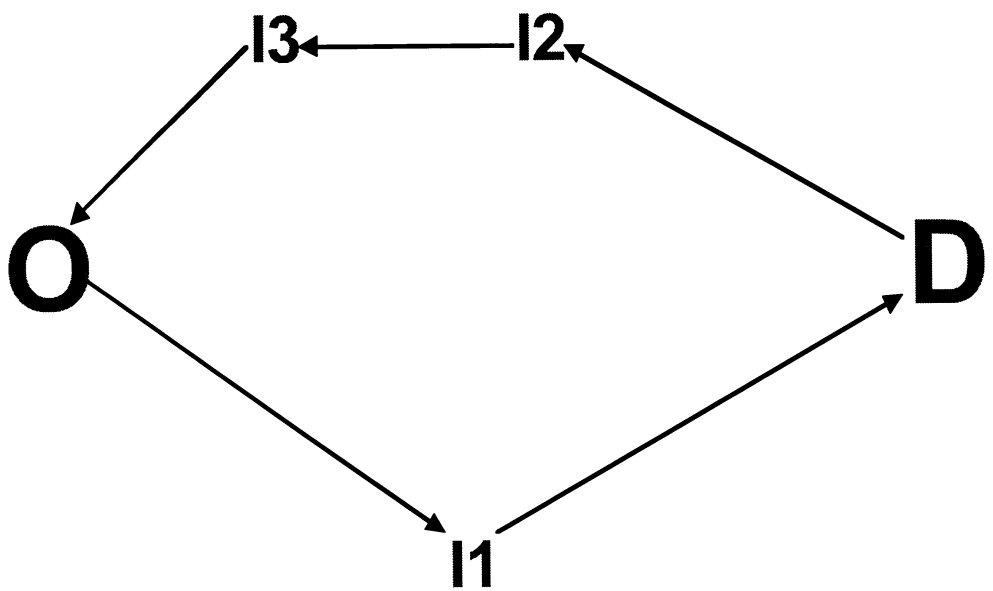
Nevertheless, by referencing all trips with respect to the offender residence, a consistent set of estimates can be obtained. This definition is required by the limitations of crime data whereby intermediate locations are usually not known. It is a hypothetical question whether modeling origins from offender residences will produce better estimates than modeling origins from other locations. But until some alternative data is produced, it

Figure 11.3:

There is an *origin* (residence)

There is a *destination* (crime location)

There may be *intermediate* links



is a speculative question.[3] Consequently, for this analysis, the offender residence is considered the origin of the crime.

In the usual travel demand forecasting framework, transportation modelers usually distinguish **productions** and **attractions** from origins and destinations. The reason is that origins are asymmetrical in time. For example, for a home-to-work (commuting) trip, the origin location in the morning is the residence while the destination is the work location. On the return trip, however, the origins and destinations are reversed (i.e., the work location is the origin while the home location is the destination). The models are referenced in the same way that is done here, namely from the residence locations, and the trips are assumed to be reciprocal. Thus, the production end of a trip is always the residence location and the attraction end of a trip is always the work location. The round-trip journey can be broken into different time sequences (e.g., morning home-to-work trips; afternoon work-to-home trips), but the production and attraction ends are always the same.

In crime travel demand modeling, there is not usually data on intermediate trips. Consequently, some of the finer analysis cannot be done. Therefore, we adopt a similar logic, but with a slightly different terminology. As with the usual travel demand modeling, the production end is *always* the home location and the attraction end is *always* the crime location. However, we use origin and destination interchangeably with production and attraction since we cannot document the return part of a crime trip.

Crime Travel Demand

Crime travel demand is the number of offenders per unit time that are expected to travel on a given segment of the transportation system under a set of socioeconomic, land-use, and environmental conditions. That is, the final model output is an estimate of the number of trips (or offenders) that travel on any given segment of the transportation system at a given time under a given set of conditions. Let's explain this in steps.

Number of trips = number of offenders

First, as mentioned above, the model is estimated sequentially. In the first stage, trip generation, there is a prediction of the number of crime trips that originate from each origin zone and the number of crime trips that occur (end) in each destination zone. In this case, a crime **trip** is equated with an offender because of the nature of arrest records from which these estimates come. With most arrest records, there is a single record for each crime that an individual commits. Thus, the origin is the residence location of the offender while the destination is the crime location. If the individual committed more than one crime, there will be an separate record for each crime (or, at least, those that are known). If two individuals commit a single crime and both are arrested, then there will be two records in the data base. In other words, the nature of the data equates a crime trip with a single offender. Thus, the total number of crime trips estimated (whether from the production or attraction end) is equivalent to the number of offenders.

Aggregate volume/count model

Second, by 'a set of socioeconomic, land-use, and environmental conditions' is meant correlates of crime trips. At the aggregate level of a zone, predictors of crime trips (whether productions or attractions) are correlates of those trips. Since the number of trips are being predicted, the model estimates **volumes** (or counts), not rates.[4] That is, the number of crime trips originating in a zone or ending in a zone is a count of events. Aggregate counts, in turn, tend to be related to other aggregates, particularly population. Thus, in developing a predictive model, population is almost always one of the dominant variables. Sometimes it can be a sub-set of population, such as number of households, number of vehicles, or number of males aged 16-25. But, since the number of incidents is usually a function of the size of the population, one usually finds that variable in the equation. Hence, the model is an **ecological** one, not a behavioral one (as mentioned above). There is a sizeable body of literature that shows difficulties in inferring individual characteristics from ecological models.[5] It is important to keep this distinction in mind and not make inferences about individuals.

In addition to population, variables that predict crime trips are also ecological variables - employment, retail space, number of bars, number of pawn shops, existence of a freeway, number of arterial lane miles, and so forth.

O-D zone pairs

In the second stage, trip distribution, a model is estimated of the number of crime trips that occur from any particular origin zone to any particular destination zone. Since the input to the second stage is the number of predicted crimes originating in each origin zone and the number of predicted crimes ending in each destination zone, the second stage estimates how many trips will be distributed from each origin zone to each destination zone. The result is an estimate of crime trips between zone pairs (an origin zone and a destination zone). There are different names that are used for this combination - zone-to-zone trips; zone pairs; zone-to-zone links, O-D links (for origin-destination links), O-D pairs, but in all cases it refers to the number of trips that start in any one origin zone that go to any one destination zone.

Travel modes

In the third stage, mode split, the number of trips by any O-D combination are then split into different travel modes - walking, biking, driving, bus (if available) or train (if available). In the usual travel demand modeling done by transportation modelers, some of these modes are broken down very finely (e.g., drive alone trips, car pooling trips, park-and-ride trips). There is no logical reason why mode split can't be defined in multiple ways. For our purposes, simple transportation choices are probably adequate because of a lack of data that would allow finer distinctions to be made.

Estimating travel routes by mode

Finally, in the fourth stage, the number of trips from any O-D pair by separate travel model are assigned to a route on the transportation network. Thus, if the trip is by walking, biking, or driving, the model may predict a different route than if the trip is by transit since a transit system is limited to particular bus or rail routes. Hence, the final stage is an estimate of the total number of crimes that occur on any segment of a transportation network by separate travel mode.

The *CrimeStat* Crime Travel Demand Module

The *CrimeStat* crime travel demand module follows this logic fairly closely, but adapts it to the nature of crime data. Figure 11.4 below shows a screen image of the module. There are five main sections (tabs). Four of them correspond to the four stages. Each of the four sections has several routines associated with them. These will be explained in the subsequent chapters.

In addition, there is a "File worksheet" section. This allows the user to save the file names in order to keep track of them. The module is very complicated and there are a lot of files used - 38 of them, many used multiple times. In addition, there are a variety of parameters that are used for the different files. The result is a complicated model whereby not only is the model tested sequentially, but there are multiple options available for each stage. The subsequent chapters, the file worksheet tab, and the online help menu will try to make the routines easy to understand. But, the user has to realize that it will take time to gather the data and to construct the model.

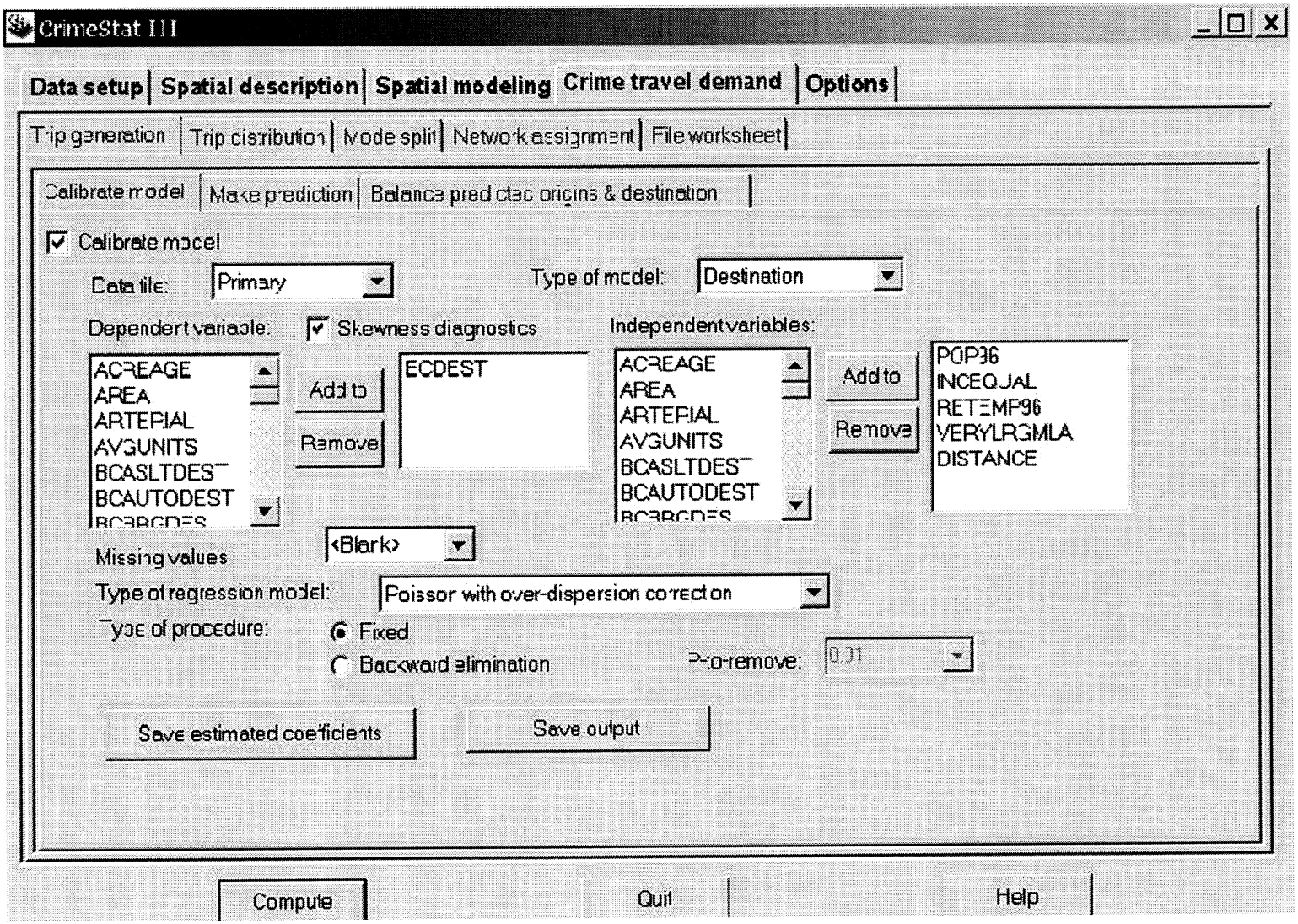
Crime Travel Demand v. Journey to Crime

A distinction should be made between crime travel demand and journey to crime. Crime travel demand modeling is not journey to crime modeling. Journey to crime modeling (and its use in geographical profiling) is a much simpler system. Research on journey to crime has been conducted since the 1930s (see chapter 10). For the most part, journey to crime modeling is a descriptive framework. Estimates are made of the distance that offenders travel during particular crime trips. A distance decay-type function is estimated from these trips and comparisons are made between different types of crime or the same type of crime for different time periods.

There is very little in the way of theory for this type of model. Crime trips are a function of distance plus some other characteristics, such as the crime type or whether there is or is not a 'buffer zone' around the offender's residence (see chapter 10). Most of the journey to crime studies have compared different types of crime by distance traveled, whether measured as average distance or by type of function as was used in chapter 10. Almost exclusively, the key variable is travel distance. There are very few studies that have looked at travel time (see Kent, Leitner and Curtis, 2004).

Figure 11.4:

Crime Travel Demand Module



In other words, journey to crime modeling is a single-stage model, essentially a description, with the primary variable being distance. It is also 'non-adjustable' in the sense that the conditions can't be varied since there is no model that predicts distance other than crime type (or buffer zone, which we didn't find evidence for; see chapter 10). There is very little in the way of predictions that the model can make other than to estimate the likely origin location of an offender (for events committed by a single offender).

Crime travel demand modeling, on the other hand, is a predictive framework. Crime trips are a function of productions, attractions, and impedance. Productions are a function of some socio-economic and policy variables. Attractions are a function of some other socio-economic and policy variables. Impedance is a function of cost and availability variables. Each of these components is predicted by different variables. Hence, the model can be adjusted (e.g., by adding or subtracting a policy intervention variable). One of its benefits is the ability to adjust conditions. For example, if it can be shown that the amount of policing in a zone impacts the number of crimes that either originate or end in that zone, then a subsequent run can 're-assign' police personnel to impact crimes in other zones.

The model is multi-stage since it is estimated sequentially and, therefore, can be used for prediction. Thus, once the model is estimated on one data set, it can be used on another set. Thus, it represents a calibration against a known data set. For example, one could calibrate the model on one year's worth of data and then use the estimated coefficients and parameters on a second year's worth of data. This, in fact, is how it is used in transportation modeling. The model is calibrated on a current year and then applied to a future year to make a forecast of future travel demand.

In short, crime travel demand is a theory of travel behavior whereas journey to crime modeling is but a simple description. In many ways, crime travel demand modeling is a 'quantum' leap in complexity and analysis, requiring gathering a lot more data and calibrating many individual steps. Nevertheless, that complexity allows a far greater use of the model than the traditional journey to crime.

Models v. Description

A key distinction in the crime travel demand framework is that of an empirical description versus a model. The framework can be applied both as an empirical description and as a model, assuming that data can be obtained. An empirical description describes the data that have been collected. For example, for trip generation, it is a count of the number of crimes that originate in each zone and the number of crimes that occur in each zone. For trip distribution, it is the actual number of trips that go from each origin zone to each destination zone. For mode split, assuming that data could be obtained on travel mode, it is a count of the number of trips for each origin-destination pair that are taken by each travel mode. Finally, for network assignment, again assuming that data could be obtained on the actual routes taken, it is a documentation of the actual routes that are taken and a count of the number of trips/offenders using each segment of the transportation network.

In other words, an empirical description is a count of the number of offenders, whether by origin location, destination location, O-D pair, travel mode, or route.

A model, on the other hand, is a simplified set of relationships that approximate the most important features of the actual count. The model is not reality, but is a rough approximation to it. Because it's rough, a model inevitably makes errors. Consequently, there always will be a difference between a model and the actual events to which the model approximates.

The two differ on other dimensions as well. A model has only a few variables whereas the actual events have many (perhaps hundreds). A model has a simplified set of relationships among the variables whereas the actual events have very complex relationships among the variable, often too complex to describe properly. By simplifying the relationships, a model produces, what Herbert Simon used to called, an *analogy* to the actual situation, whereas the actual events are literal (Newell, Shaw and Simon, 1957).

The *CrimeStat* crime travel demand routines can be applied both to empirical data as well as modeled relationships. In fact, two of the routines are directly concerned with the differences between the model and the actual data. Both sets of endeavors have value in their own right, but these differ. An empirical description is most relevant to the present. For a police department trying to mitigate crime and catch offenders, the empirical description is probably of more use than an abstract model. As will be seen, the empirical description of crime trips will always be more complex than the estimated model. If the only purpose is to describe the actual patterns that are occurring, then a model is not needed.

On the other hand, a model has definite advantages that a description does not. First, it can be used for predictions. If a model is calibrated against a known data set, that model can then be applied to a new data set. For example, one could create an estimate of crime trip productions based on existing socioeconomic and land use data. Then, one could apply that model to a forecast data set of future socioeconomic and land use conditions. The result is a prediction of future crime levels. Of course, since the model was never completely correct in the first place, it will inevitably make errors.[6] Further, since there is no guarantee that past relationships will necessarily hold in the future, there is no certainty about whether the most important part of the predicted relationships will actually hold. Nevertheless, there has been enough success in demographic, economic, and transportation modeling that new fields of forecasting have emerged as legitimate research activities.

A second advantage of a model is that it can be manipulated. Variables can be modified to explore their effect. Distributions can be re-arranged to, again, understand their effect. For example, if relationships can be established between the number of crimes produced or attracted to zones, on the one hand, and the number of police personnel in a zone or to the existence of a large shopping mall, or to the existence of a drug treatment center, on the other hand, then scenarios could be run that explored the different arrangements. These "What if?" types of scenarios can be very useful. For example, if a

relationship exists between shopping malls and crime trips, what is liable to happen when a new shopping mall is built? One could take the model, add the new shopping mall (or the retail employment or acreage associated with the mall) and run the model to make a prediction about its likely impact. Or to take another example, if it can be shown that there is a negative relationship between the number of beat police officers and the number of crimes originating in zones, then it would be possible to evaluate the likely consequences of re-arranging police personnel across different beats.

In short, a model is a very powerful tool for evaluating policy or intervention type strategies. Rather than speculate or gather evidence from other metropolitan areas (which is valuable, of course), a model can be used to simulate the likely consequences of an action on crime levels. In transportation planning, the travel demand model is used all the time to evaluate the likely consequences of implementing particular projects. This doesn't mean that it is the only factor considered in making a decision or even the most important factor; clearly, politics, financing, and community support are also major components of any decision. Nevertheless, the travel model is a very important input into any decisions about future investments.

Uses of a Crime Travel Demand Model

Table 11.1 illustrates some possible uses of the crime travel demand model, assuming that data could be obtained.

Table 11.1:
Possible Uses of Crime Travel Demand Model

	Trip Generation	Trip Distribution	Mode Split	Network Assignment
Description	Identify correlates of crimes	Identify crime trip links	Identify crime travel models	Identify routes taken by offenders
Calibration	Estimate coefficients of predictor variables for crime origins & destinations	Estimate origin-to-destination coefficients for crime trips	Estimate formula for travel modes used by offenders	Estimate model for routes taken by offenders
Prediction	Predict future crime levels	Predict future crime trip links	Predict future crime travel modes	Predict future routes used by offenders

The model could be used for description, calibration, or prediction. In description, the emphasis is on describing the travel behavior of offenders. For trip generation, it involves identifying the correlates of crimes, both by origin zone and by destination zonel For trip distribution, it involves describing the actual crime trips taken between specific origin zones and specific destination zones. For mode split, it involves identifying the different modes that offenders are using, describing the proportion of each mode that are used, as well as describing the modes used for particular origin-destination links. Finally, network assignment involves describing the actual routes taken by offenders. In other words, the emphasis on description is identifying the specifics used in crime trips.

On the other hand, calibration involves selecting variables that can approximate the description and estimating coefficients for their use. The emphasis is on finding a limited number of general variables and coefficients that can produce a reasonable approximation to the actual travel behavior. Thus, in trip generation, the aim is to find a few variables that can predict reasonably accurately the number of crimes by origin zone and destination zone. In trip distribution, the aim is to estimate coefficients that can approximately reasonably accurately the trips that are taken from particular origin zones to particular destination zones. In mode split, the aim is to develop coefficients that can approximate the travel modes used while in network assignment, the aim is to find an algorithm that approximates the actual travel routes used by offenders. The result of a calibration is a model that can be generalized whereas a description cannot be generalized.

Finally, in prediction, the calibration models are applied to other data, either forecast values of future levels of the predictive variables or data from other jurisdictions to see the similarities or differences. The existence of a model (ideally calibrated against a real data set) allows the forecast to be made whereas a description cannot be forecast.

Research Uses of a Crime Travel Demand Model

For research, a crime travel demand model has many different uses, only some of which we explore in the next five chapters. First, it organizes crime travel information in a systematic manner. The model is logical and proceeds in a systematic way. As opposed to the journey to crime-type model, which is just a description, the crime travel demand model systematically steps through the four modeling stages in an understandable way. It is a very good way to organize information on crime travel, though, clearly, it's not the only way.

Second, compared to the journey to crime literature, it is a more realistic model of offender travel. For one thing, it incorporates information about origin locations. This helps answer the question of why certain areas produce more crimes than others (remember, it's not a behavioral explanation, but an ecological model). For another thing, it incorporates information about destination locations and helps answer the question of why certain areas attract more crimes than others. For a third thing, it models travel choice in a more complex manner. Instead of assuming that all offenders will travel to a crime in exactly the same way (e.g., by walking), the model allows the separation of different travel modes. For journey to crime models, distance is the only impedance

variable, whereas for crime travel demand modeling, travel time and travel cost are often better predictors of travel behavior, especially in relationship to an available network. In short, it is a much more complex, yet realistic, representation of crime travel behavior.

Third, it is a dynamic analysis of travel behavior. Crime trips are seen as a product of neighborhood production factors, attractions, and travel costs (impedance). And since these change by various hours of the day, so too does the travel pattern change. The ability to model travel at different times of the day is one of the strengths of the travel demand type of framework.

Fourth, and finally, a crime travel demand model can allow comparisons between different types of crimes in the productions, the attractions, and the costs. So, too, can journey to crime models be used to compare different type of crime. But those comparisons are uni-dimensional, essentially comparing different distance decay functions. The crime travel demand model can explain the 'distance decay' function and hence allow a more structural interpretation than was previously possible. For example, in comparing data sets from Baltimore, Chicago, and Las Vegas, Richard Block, Dan Helms and myself are finding that there may be very little difference in the cost function used for different types of crime trips, but that differences in these trips are more a function of the distribution of opportunities (attractions). To link this up to the early theme of this chapter, American society has become so mobile and the automobile so ubiquitous that distances are not as much a barrier to offenders as they used to be. In other words, the distribution of opportunities appears to be the more dominant factor predicting types of crimes than the limitations of neighborhoods and small communities. If this turns out to be true, then we're in for a major shift in the type of crimes that our society will experience over the next few decades. Mobility may replace neighborhood as a determining factor in crime behavior. In other words, the local 'community-based' offender is 'morphing' into a metropolitan-wide and, perhaps, regional offender, a not very desirable prospect.

Crime travel demand modeling allows for a more complex, more interventionist and, perhaps, deeper understanding of crime travel than previous types of model, particularly the journey to crime and serial walk type of model (see chapter 10).

Utility for Policing and Law Enforcement

For police department and other law enforcement agencies, crime travel demand modeling has some advantages as well. First, it can be used to model different policing strategies, as suggested above. For example, it could be used to evaluate the likely effect of shifting patrol deployment. The "What if?" nature of crime travel demand modeling makes it useful to explore alternative arrangements before they are actually implemented.

Second, it could be used for forecasting. As mentioned, if a model has been calibrated on one set of data, then it could be applied to another set to predict, for example, the distribution of crimes five or ten years later. Typically, police departments have not done forecasting, but they are often expected to be able to anticipate changes. This type of model can be useful for that purpose since Councils of Governments (COG) and

Metropolitan Planning Organizations (MPO) systematically make forecasts of future population and employment levels.

Third, it can be used for modeling interventions. Aside from modeling different policing strategies, a range of land use and communities changes could be explored. For example, what would be the effect of introducing more drug treatment centers or more 'weed and seed' adolescent facilities? The logic is similar to forecasting. A model is calibrated against one data set. But, in addition to socioeconomic and land use variables, variables on facilities are added to the equation as predictors. If it can be shown that they have any effect (which we hope they do), then these can be used as variables in a modeling scenario.

Fourth, these types of models can be used for anticipating changes in the community. Again, this is similar to the forecasting purpose mentioned above. But, it's slightly different in that it anticipates structural changes. An example was given of anticipating changes from new shopping malls. In Baltimore County, for example, shopping malls were shown to be the strongest attractors of crime trips. In that context, what would happen if a new mall was built? This type of model can be used to model this scenario. Conversely, a lack of employment opportunities appears to be correlated with crime productions, at least in Baltimore County. What would happen to crime if local employment was increased in certain zones? Again, this type of model is useful for exploring that type of question.

Again, going back to an earlier point, there is, of course, a difference between a model and reality (an actual situation). Reality is complex; models are not, or are a lot simpler. Still, models as analogies can provide insight into mechanisms and allow police, law enforcement, and the policy community as a whole to try to simulate changes without having to commit to expensive, and perhaps disastrous, changes with little information. In other words, modeling in general, and crime travel demand modeling in particular, is a tool that may have wide utility for the law enforcement community.

References on Travel Demand Modeling

In this final section, some sources on travel demand forecasting are listed. There are a large number of sources, though there are few actual textbooks. A very good textbook on the subject is by Ortuzar and Willumsen (2001), while an older, out of print book is by Stopher and Meyburg (1975). There are several major handbooks on the topic (Hensher and Button, 2003; ITE, 2003). Some good chapters on the subject are found in Beimborn (1995), Field and MacGregor (1987, ch. 6) and by Engelen (1986, ch 17). Discussions of "second" generation models can be found in Domencich and McFadden (1975) and Ben-Akiva and Lerman (1985).

However, probably the best source for articles on the subject are found on the Federal Highway Administration web site (http://www.fhwa.dot.gov). Among the articles/presentations that can be found on that site are introductions by Beylyon and Culp (2001) and by Culp (2002). Of particular interest on the FHWA site are discussion of

bicycle and pedestrian travel modeling (Turner, Shunk, and Hottenstein, 1998), which may be relevant for crime analysis, and "third" generation models (RDC, Inc., 1995; Pas, 1996).

Older sources, which are still good are by Oppenheim (1975, ch. 4) and Krueckeberg and Silvers (1974, ch. 10), aside from the Stopher and Meyburg text mentioned above.

Endnotes for Chapter 11

1. In the usual travel demand modeling framework, data gathering is called a *land use inventory* and involves estimating population and employment by different land uses, particularly retail trade and several other types of industry.

2. In classic travel demand modeling, there are several feedback loops. One is from the network to the mode choice, as in the crime travel demand version. A second is from the network to both mode choice and trip distribution stage. If a particular route becomes very congested (having a traffic volume-to-capacity ratio greater than 1.0), it's been noted alternative destinations become more attractive. For example, people will often travel farther and more out of the way to avoid congested corridors. In short, there are a variety of feedbacks from later stages to earlier stages, and the model is quite flexible in being able to accommodate the different sequences.

3. *If* it were possible to obtain data on intermediate locations during crime trips by offenders, then it would be possible to test whether modeling the origin with respect to these intermediate locations produces more stable and clearer predictions than with respect to the residences of the offenders. But, until that data is obtained, the question is speculative.

4. Some agencies have actually used it to predict rates. Since a rate is an event relative to a baseline, population is factored into the dependent variable. It is possible to apply the model as a rate, though the user needs to ensure that all the predictor variables are also rates.

5. The question of whether an ecological inference is valid or not has been studied extensively. Sometimes it holds and sometime it doesn't. An ecological inference occurs when data are aggregated with a *grouping* variable (e.g., state, county, city, census tract; see Langbein and Lichtman, 1978). The relationship is often called an *ecological fallacy*, but that is an oversimplification. Typically, if the between-group variance (i.e., differences) is greater than those within groups, then the ecological relationship will be a lot stronger than at an individual level. Conversely, if the within-group variance is greater than the between-group variance, a relationship that holds at the individual level will not be seen at the aggregate level. There are other ecological characteristics that account for typically higher R^2 at the aggregate level - spatial autocorrelation, skewness in the dependent variable, and hetereoscedasticity (unequal estimation errors around a statistical estimate).

6. Simon and Newell described two kinds of errors: 1) errors of commission (Type I errors); and 2) errors of omission (Type II errors). The first kind represent relationships and predictions that don't exist (to use our terminology) while the second kind represent the failure to detect relationships that do exist. Any model will have both sets of errors. The point to keep in mind is whether a model captures the most important relationships and doesn't make too many Type I errors. Newell, Shaw, and Simon, 1957.

Chapter 12
Data Preparation for Crime Travel Demand Modeling

In this chapter, the data requirements for the crime travel demand model are discussed. At the minimum, there are four types of data that are needed for the crime travel demand module:

1. A zonal system;
2. Matched crime data listing both crime location and likely origin location. This can be, further, broken down by crime types, time of day, day of week, and other sub-sets of the total number of crimes;
3. Socioeconomic and land use data for the zones which are used as predictor variables; and
4. Network data on the road system and the transit system.

In addition, there can be supplementary data that help expand the predictive models. These include:

5. Policy-related data (e.g., strategic or planned interventions)
6. Crime data on the actual distribution of crimes by zones, which is used to correct the implied distribution from 2 above.

The following is a discussion of each of these requirements.

Choice of a Zonal System

The crime travel demand model is a zonal model. That is, it analyzes crime trips by zones. For all four stages, the estimates are for zones (not for individuals). Thus, at the trip generation stage, there are two zonal models - one predicting the number of crimes originating in each origin zone and one predicting the number of crime ending in each destination zone. At the trip distribution stage, there is a prediction of the number of crimes which originate in each origin zone that end up in each destination zone (the implicit number of *trips*). At the mode split stage, the trips for each origin-destination zone pair are, further, sub-divided into different travel modes. Finally, each origin-destination zone pair by travel model is assigned a route. But, at all stages, the estimates are for zones.

Typical Zone Systems

This makes the choice of a zonal system very critical. In practice, three types of zone system have been used:

1. Census geography
2. Traffic analysis zones

3. Grid cells

Census geography follows the geography used by the U.S. Census Bureau (in the United States) or by other national census agencies. Traffic analysis zones are used by most transportation planning agencies for modeling transportation in a metropolitan area. They are typically super-sets of census geography (e.g., two census tracts combined). Finally, grid cells are uniform zones imposed on a metropolitan area. While they have desirable statistical properties, they are rarely used in practice.

Problems with Large Zones

In deciding on a choice of a zonal system, there are several important issues that must be balanced. The first problem one faces is that of zone size. Large zones can distort relationships. It can be shown that the size of a zone has an impact on the statistical relationships between the predictor variables and the dependent variables, which are the number of crime trips by either origin or destination zone. Typically, the larger the zone size, the stronger the relationship. The reason for this effect is complex and has to do with a number of factors, for example minimizing within-zone differences in travel behavior and, therefore, maximizing the between-zone variance relative to the within-zone variance (Langbein and Lichtman, 1978) or aggregating spatial autocorrelation to minimize adjacency effects (Anselin, 1992). But, the effect is well known. The cost of having this stronger statistical relationship is to produce a less precise estimate for the region since within-zone differences are minimized.

One can think of this in terms of an arbitrary point within a zone (e.g., the centroid of the zone though it doesn't have to be the centroid). All the data in the zone are assigned to that point. Thus, the total number of crimes originating within the zone or ending within the zone are assigned to a single point. This means that whether a crime occurred at the edge of the zone or directly in the middle, it is assigned geographically to a single point. Similarly, any of the predictive socioeconomic or land use variables are also assigned to that point (e.g., median household income). Hence, any differences within the zone are eliminated as all events and households are assumed to 'live' at that point. If there are two adjacent zones, for example, that differ in income levels, most likely there is a gradient of income from one to the other; however, putting the measurement of income at a single point in each zone exacerbates the differences between the zones, while ignoring the similarities (e.g., at the edges of the zones where the population on both sides are liable to be more similar). It should be clear that the larger the zone size, the greater the exaggeration between the zones. In other words, larger zones exacerbate differences between zones while minimizing similarities. The result is an oversimplification of the distribution of characteristics of those neighborhoods.

In addition, larger zones have too many trips that both originate and end in the same zone (intra-zonal or 'local' trips). Clearly, the larger the average size of a zone, the more likely that a trip will be entirely within the zone. Thus, there is a strong relationship between average zone size and the number of intra-zonal trips. This will be less useful since it minimizes the complexity of travel. The extreme would be to divide a metropolitan

area into only a few zones (e.g., 4 or 5). The result would detect large scale travel patterns, but would lead to a majority of trips occurring within each zone. One would not be able to say very much about crime trip other than a few general patterns (e.g., crime trips from the central city to the suburbs).

On the other hand, if the zones are too small, there is a danger that there would be more cells in the trip distribution stage (see chapter 14) than there are actual events. The result would be inadequate degrees of freedom in a model and unreliable coefficients. A zone model has to balance the need for increased precision with the ability to produce stable estimates.

Problems in Obtaining Data for Small Zones

In theory, the ideal zone size would be small, say on the order of a block or two. This would allow precision in estimates and the ability to examine the complexity of travel in a metropolitan area. The reason that this is not done very often, however, is the lack of data at the block or block group level. While crime data can be allocated to blocks or block groups, it is often difficult to obtain socioeconomic data at that level. In the United States, for example, while the U.S. Census Bureau will release data down to the block level, confidentiality requirements require that no data be able to identify individuals. Hence, there is very limited data at the block level, typically gender and race distribution. Block group data, on the other hand, is often easily available, including critical income factors.

The biggest problem with a block group zonal system is in obtaining employment data. The U.S. Census Bureau does not normally collect employment data (and won't release if they did) while the Bureau of Labor Statistics, which does collect employment information, won't release it at such a small geography. Thus, obtaining these data depends on local organizations, such as a Council of Government (COG) or a Metropolitan Planning Organization (MPO). Till now, these data have not typically been released at small geographies such as block groups, but, instead, at a larger geographical unit called a *traffic analysis zone* (TAZ). However, because of the widespread use of GIS and the increasing incorporation of high resolution aerial photography into GIS-based land information systems, this situation is changing. For example, at the Houston-Galveston Area Council, the MPO for the greater Houston area, employment estimates are made for as small a geography as a 1000 foot by 1000 foot grid cell, essentially a couple of city blocks. Thus, it is starting to become possible to obtain employment data at very small geographical levels. In the next decade, more and more data will be available for small geographical units and the size limitation mentioned above will slowly disappear.

There is a converse problem with size, however, that also occurs. If the zones are too small (e.g., if data could be obtained at a block face level), there will be too many cells with no crime events. The smaller the geography, the more likely that there will be no events. For example, to illustrate the crime travel demand model, I've used data from Baltimore County. The crime data were 41,974 incidents that occurred between 1993 and 1997 for which both a crime location and a crime origin were known. To model these incidents, traffic analysis zones (TAZ) were used. For Baltimore County, there were 325

destination TAZ's while for both Baltimore County and Baltimore City, there were 532 origin TAZ's. Let's take the origin TAZ's. With 41,974 incidents, the average number per TAZ was 78.9. However, in practice, 27 zones had no crimes originate from them (or approximately 5%). If a smaller geography was used (e.g., block groups), the number of zones with no crime originating in them would increase substantially, as would the percentage. At some point, if the geography becomes very small, a high proportion of the zones will have no crimes originating from them. This makes modeling very difficult as the average number of events will tend towards zero. While there are techniques for modeling a skewed distribution (which will be discussed in Chapter 13), the more skewed the distribution, the less accurate typically is the estimate. Extremely skewed distributions are more problematic for modeling than mildly skewed distributions as the variance terms become very complex to estimate.

Still, on average, a small zone system is preferable to a large one. There is so little data for very small geographies that the problem of zones being too small is an unlikely one, at least for the foreseeable future. Where possible, users should try to obtain data at the smallest geographical level for which data can be obtained.

Problems with Irregular Size and Shape

Another problem facing the choice of a zonal system is the irregular sizes and shapes of most zonal data. For example, the U.S. Census Bureau uses a unit called the *census tract* for the collection of census information. The census tract is supposed to be an area of approximately equal population (though it's never entirely equal). They generally are wholly within jurisdictions (though there are exceptions) and they are made up of blocks and block groups (collections of blocks), but in turn are aggregated upward to form enumeration areas within each jurisdiction. This logic makes sense in terms of the mission of the U.S. Census Bureau, which is take the census; the geography respects political jurisdictions (counties and cities), but is fine enough to help manage the data that is collected during the decennial census.

But, from a modeling viewpoint, this geography has problems. First, the size of census tracts typically increases from the central city outward to the far suburban edges of a metropolitan area. Because the logic of the census tract is to approximate an area of equal population, by necessity the tract area will increase with the lower densities in most suburban communities. Thus, any data assigned to a tract (or to a block or block group within a tract) will be less precise in the suburbs than in the central city. In a travel demand model, one can end up with absurdities whereby trips appear to originate at locations where there are no people simply because the centroid of the zone falls at a location where there are no households (e.g., in a reservoir). The uneven size of zones usually means that a travel model will be more precise in the center of a metropolitan area than in a suburb.

Second, because census tracts are often defined with respect to principal arterial roads (which form their edge), they often will have irregular shapes. This could add a potential source of error in that all events and household characteristics within a boundary

are assigned to a single point in the zone. On the other hand, if the zones have been selected to represent a neighborhood which is relatively uniform, such irregularity may not be a problem.. Nevertheless, if two zones have very different shapes (e.g., one is square while the other is pointed), allocation error (and, hence, modeling error) is liable to be greater in the one that is more irregular, all other things being equal, than in the one that is square.

Again, ideally, a zone system should be a grid whereby each zone is a square of equal size; shape and area effects are constant for all zones. While geographers recognize the value of a grid cell for zonal allocation, in practice, it is rarely used. Among the transportation planning agencies in the country, very few use a grid system. Of the ones with which we are familiar, only the Chicago Area Tranportation Survey (CATS) uses a grid system.[1] In chapter 17, Richard Block discusses applying the crime travel demand model to Chicago.

Therefore, to sum up, in practice, one has to balance four different criteria in selecting a zone system for a crime travel demand model:

1. Zone size (generally, smaller is better within limits)
2. Consistency of zone size (less variability is better)
3. Distortion due to shape (more regular is better)
4. Availability of data

Unfortunately, it is the fourth criterion - the availability of data, that is usually the determining factor in the choice of a modeling zonal system. Hopefully, this will change in the future as more data at the smaller geographical level become available.

Trips from Outside the Study Area

One other problem confronts the choice of a zone system. Irrespective of which zone system is used (census geography, TAZ, grid cells), a decision has to be made about the extent of the area to be used in modeling. The choice of destination zones is made by the availability of crime data. Typically, data are collected by police departments for their jurisdiction. Unless data sets from several adjacent jurisdictions can be obtained and combined, the analyst typically will be restricted to modeling the jurisdiction for which the crime data has been collected. We'll call this the *Modeled Jurisdiction*.

Modeling the origin zones is a decision about which zones contribute to the crimes occurring in the modeled jurisdiction. That is, some of the origins of the crime trips occurring within the modeled jurisdictions may come from outside that jurisdiction. For example, in the case of Baltimore County, approximately 42% of the crimes occurring within that jurisdiction had origins outside that jurisdiction. In such a case, it's very important to include zones beyond the modeled jurisdiction in the crime origin model. That is, to use Baltimore County as an example, if the predictive model for crime origins only included the 325 TAZ's within that jurisdiction, the model would not adequately assess the factors predicting crime origins.

Thus, it's important to widen the scope of the study area to include other areas outside the modeled jurisdiction that contribute to crimes occurring within that jurisdiction. In the case of Baltimore County, 38% of the crimes occurring within that jurisdiction originated from the City of Baltimore.

But where does one draw the line? Eventually, because of limitations due to data or due to the need to restrict the analysis, a boundary has to be drawn around the study region. Some crimes will inevitably occur from outside that line. These are called *External Trips* and refer to the trips that originate from outside the study area. While there is no 'hard and fast' principle, generally transportation planners recommend that the study area include at least 95% of the trips that end in the modeled jurisdiction (Ortuzar and Willumsen, 2001). With such coverage, the 5% (or less) that are external trips will have little effect on the model parameters, and the amount of bias will be small (but will always exist unless 100% of the trips can be measured).

We'll come back to this point in the next chapter. But, the critical point is that the zone system must incorporate a sizeable area in which the vast majority of the crimes originate from within, at least 95%. Going back to the Baltimore County example, adding in the City of Baltimore increased the percentage of trips originating within the study area from 58% (for just Baltimore County) to 96%, an acceptable level to 'draw a boundary' around the study region.

Small Area Limitations

A travel demand model is aimed at modeling travel patterns in a metropolitan-wide area. The model is particularly good at estimating travel for the region as a whole and for large sub-areas of the region. The model is not particularly good at estimating travel within small geographical areas. The problem of intra-zonal trips - trips in which the origin and the destination are within the same zone, represents trips for which the model cannot describe the travel pattern. These are trips that the model detects are within a small area, but cannot estimate where these occur. Similarly, trips between adjacent zones are often imprecise in a travel demand model; the model can indicate the level of short trips, but the level of precision is low.

In other words, the crime travel demand model is good at capturing major travel patterns over a large area and not very good at localized travel. There are other modeling tools for small area travel analysis that provide much more detail about the neighborhoods and road system in which this travel occurs, such as microsimulation software of travel behavior in a neighborhood.

Therefore, in order to apply a travel demand model to crime analysis, it is important to model a substantial part of a metropolitan area. The model will not be very good if a small city or area within a metropolitan area is chosen. In these chapters, crime travel in Baltimore County is used as an example case in order to illustrate the different components of the model. Baltimore County is a large jurisdiction covering approximately 640 square miles; it represents a sizeable part of the Baltimore metropolitan area.

Combining the origin zones of Baltimore City with those of Baltimore County provides a very large proportion of the metropolitan area. In other words, Baltimore County is large enough to model the crime destinations while the origin zones represent much of the metropolitan area.

On the other hand, if we attempted to apply the model to a small part of the region, for example the town of Towson, the model would be less precise and accurate since that town represents a very small proportion of the overall region. In short, a crime travel demand model is useful for modeling either an entire metropolitan region or a sizeable part of a metropolitan region, but should not be considered for a small geographical area. It is a regional travel model, not a local model.

Calculation Limits for the Number of Zones

A final consideration has to do with the number of zones that can be modeled with the *CrimeStat* crime travel demand model. Because of the available RAM in a computer and the limits of the Windows 32 bit operating system, the routine can only handle a certain number of zones. If M is the number of origin zones and N is the number of destination zones, then a trip distribution matrix, which is subsequently used in the mode split and network assignment stages, involves N*M cells. Each field in a cell requires 8 bytes of RAM and there are seven fields output. Thus, a trip distribution output file requires approximately M*N*8*7 bytes of RAM.

To use an example, if the user has 1 Gb of RAM available, then approximately 19,173,877 grid cells could be handled (or a square matrix of 4,378 x 4,378). However, Windows requires some overhead as does *CrimeStat*. Thus, the actual number of grid cells that could be processed will be a little less.

One could, of course, add more RAM. In this case, the file size of the trip distribution matrix could be increased. However, there are limits to this. First, the calculations will slow down, at a rate that is exponential to the file size. At some point, the calculations would take so long as to be impractical. Second, the Windows operating system, which processes information in 32 bit chunks, has a 4 Gb limit. Thus, the maximum file size would be a square matrix of about 8,757 x 8,757.

This means that, even if data at the block level could be obtained, the actual number of zones that could be processed might make this an unrealistic zone model. For example, in Chicago there are 21,068 blocks. Using these blocks as a zone model in the crime travel demand would be impossible since the matrix routines could not handle such a large matrix, even assuming that it's desirable to do so. Therefore, any zonal model that is selected must be compatible with the calculation limits of the available RAM and the Windows operating system. In the case of Chicago, using block groups was an acceptable choice since there are only 2400 of those.

Obtaining Crime Data

There are four types of data that need to be obtained.

Crime Data by Origins and Destinations

First, there is crime data. But, in order to estimate a crime trip, it is essential that these data have information on both crime origins as well as crime destinations. The most likely source of these data will be arrest records whereby both the crime location and the charged offender's residence are given. Only the police are liable to have these data. Thus, it will be necessary to obtain cooperation from the local police department for access to arrest records.

In the data, the residence location is taken as the origin while the crime location is taken as the destination of the trip. As mentioned in chapter 11, the "true" origin of the crime may not be known. First, the offender may not even have been living at the same residence as when arrested. Many offenders are highly transitory persons and a residence at the time of the arrest may not be the actual one from which the crime occurred. Second, the offender may not have traveled directly from home to the crime location, but may have committed the crime as part of his/her daily activities (intermediate trips). However, without any alternative data on the actual origins, there is little that can be done except assume that the residence when arrested is the origin. As long as this definition is kept, a consistent estimate can be obtained.[2] In effect, one is asking the question, "What is the likelihood that an offender who lives in zone i will commit a crime in zone j at some point during a day?". It really doesn't matter whether the offender traveled from the home location to the crime location as opposed to going to the crime location from an intermediate location. The model is simply constructed with respect to residence location.

The data has to be organized so that the X and Y coordinates of both the residence location (the origin) and the crime location (the destination) are given. Figure 12.1 illustrates a typical data set. It will be necessary to geocode both locations in order to establish a 'crime trip', an assumed trip from a particular origin location to a particular destination location.

Figure 12.2 shows the location of 41,974 crimes committed in Baltimore County between 1993 and 1997 while figure 12.3 shows the assumed origin location of the offenders who committed these 41,974 crimes. As seen, the origins are all over the region, but most (96%) are in either Baltimore County or Baltimore City. In other words, a 'crime trip' links the origin location of each crime with the actual destination where it occurred. If arrows were to be drawn from the origin to the destination, the entire map would be swamped with a series of lines.

Choosing a Zonal Model

The zone used for Baltimore County were traffic analysis zones (TAZ). The reason for selecting these was the availability of both population and employment data. The

Figure 12.1:

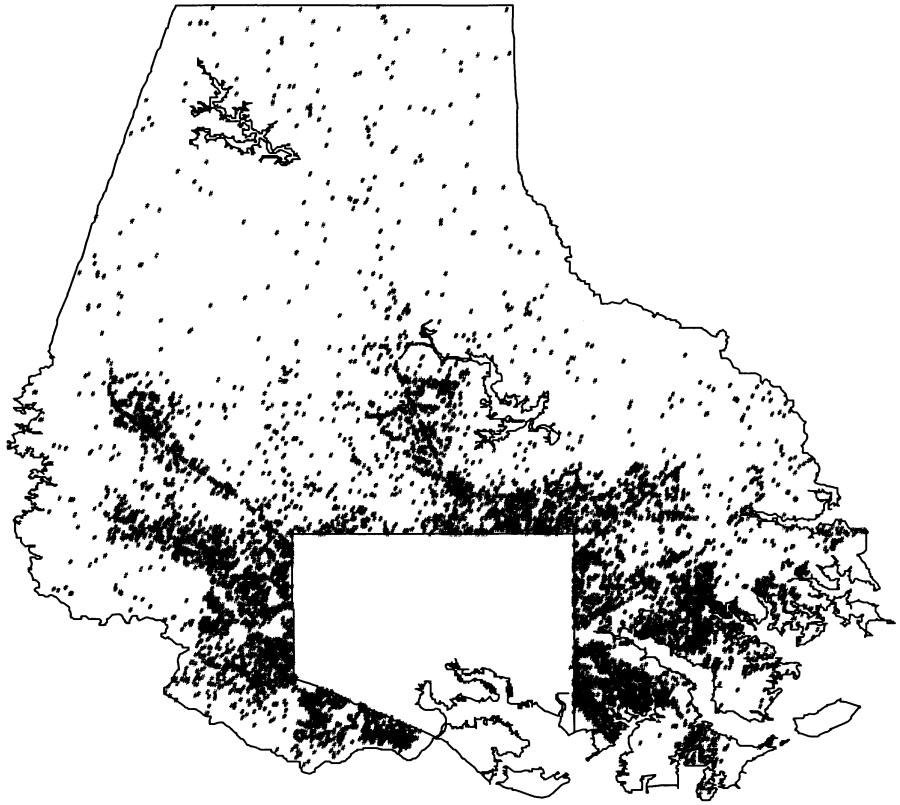
Crime Data Requirements

Minimum data requires origin and destination location

UCR	DATE	INCIDX	INCIDY	HOMEX	HOMEY
430	1/5/97	-76.8131	39.3822	-76.8131	39.3822
440	5/17/95	-76.4490	39.3355	-76.4489	39.3355
210		-76.4068	39.3388	-76.5281	39.3085
210		-76.4142	39.2801	-76.4142	39.2801
430		-76.5527	39.3908	-76.4410	39.3080
440		-76.7581	39.3131	-76.7709	39.3105
440	3/29/94	-76.5095	39.2735	-76.5095	39.2735
440	1/22/96	-76.7344	39.3212	-76.6899	39.3364
690	7/13/93	-76.4525	39.3012	-76.6050	39.3020
690	10/8/94	-76.5278	39.2584	-76.5051	39.3970
690	8/10/97	-76.7384	39.3275	-76.7384	39.3275
690	3/10/96	-76.7325	39.3018	-76.7325	39.3018

Figure 12.2:

Baltimore County Crime Locations: 1993-1997
Location of Crimes Committed by Offenders (N=41,974)



0 20 Miles

N
W E
S

Crime locations (destinations)
City of Baltimore
Baltimore County

Figure 12.3:

Baltimore County Offender Residences: 1993-1997
Location of Baltimore County Offenders When Arrested (N=41,974)

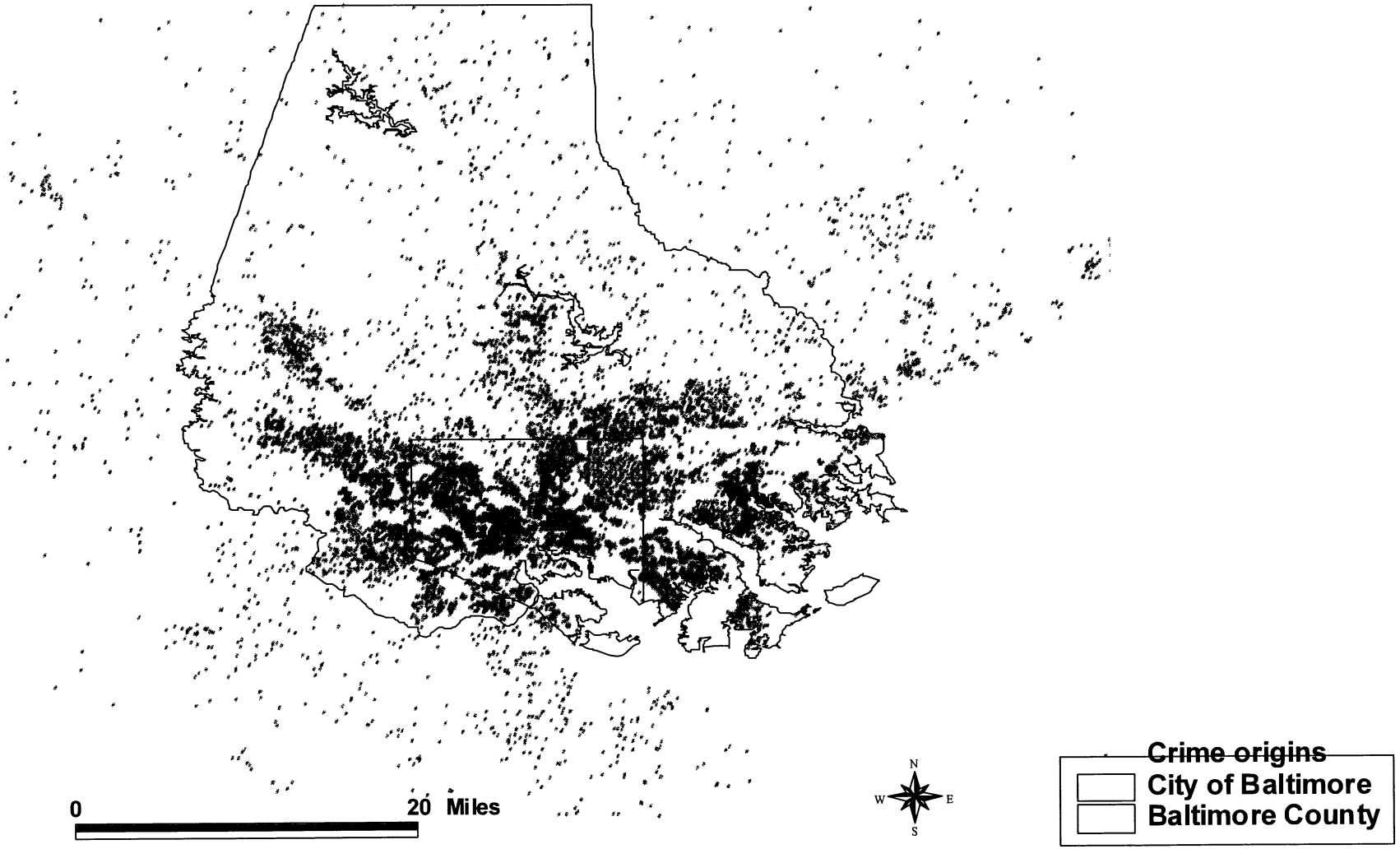
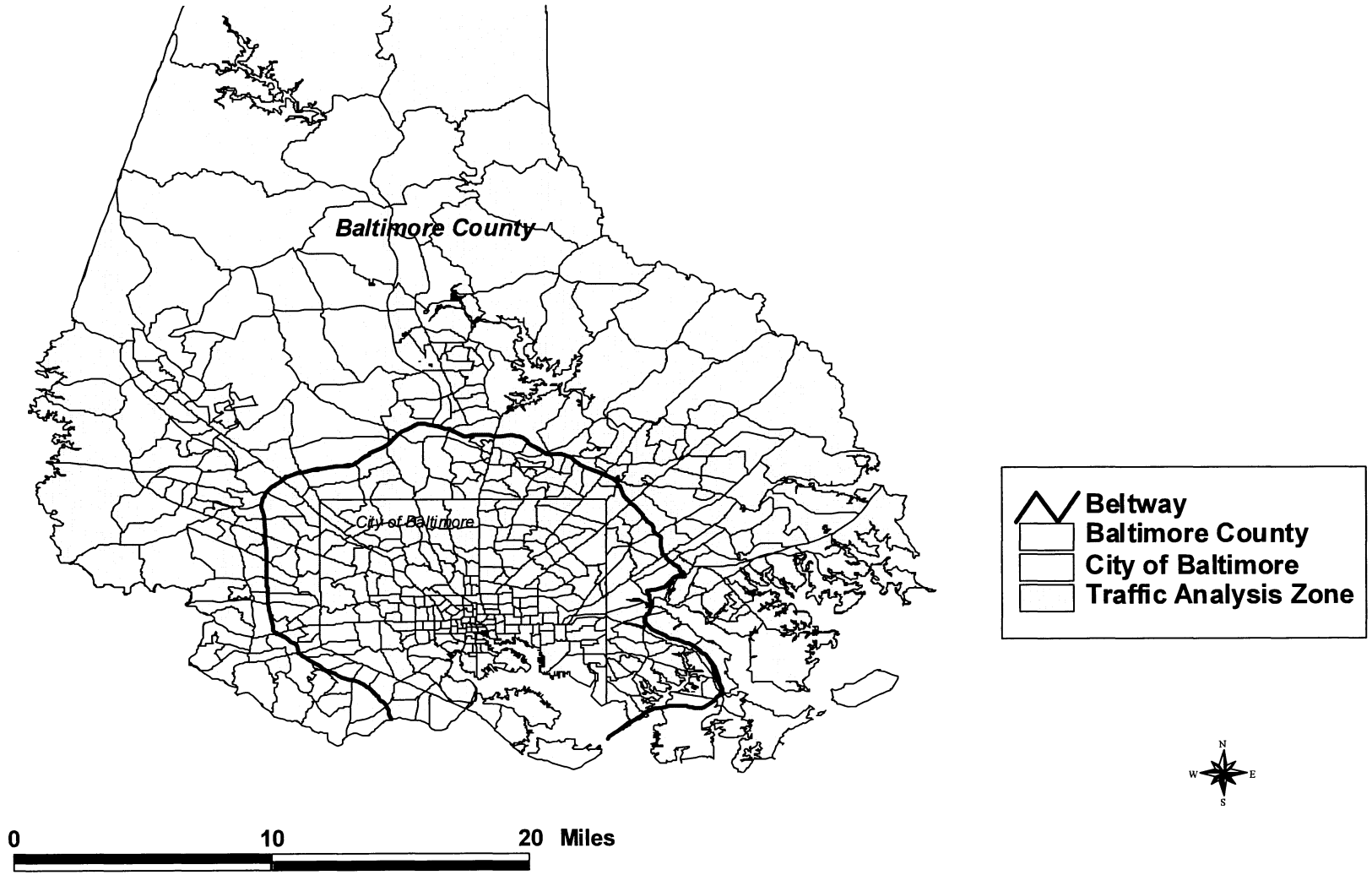


Figure 12.4:

Metropolitan Baltimore Traffic Analysis Zones: 1998



Baltimore Metropolitan Council is the Council of Governments and the Metropolitan Planning Organization for the greater Baltimore region. They use TAZ's for their transportation model. Since data were available by the TAZ's, it seemed like a plausible decision. But, as mentioned above, there are advantages and disadvantages to this decision. Approximately, 20% of all crime trips occur within the same zone (intra-zonal trips). Such a high proportion makes the overall model estimates prone to some error. Figure 12.4 shows the TAZ's for both Baltimore City and Baltimore County.

Note that there is a difference between the zones used for the origins and the zones used for the destinations. In the case of Baltimore County, there are 325 TAZ's that cover the County. However, as mentioned above, since many of the crimes occurring in Baltimore County originate in the City of Baltimore, the origin zones include those of the City as well as the County. Thus, there are 532 origin zones.

Assigning Crime Events to Zones

The next step involves assigning the crime origins and the crime destinations separately to the zonal model. That is, each crime event is assigned to zones twice, once for the origins and once for the destinations. Since an arrest record is an implicit crime trip, the residence location is assigned to a zone and the destination location is assigned to a zone. Then, the number of crimes originating in each zone are summed over all records to produce a distribution of crimes by origin zone. Similarly, the number of crimes ending in each zone are summed over all records to produce a distribution of crimes by destination zone. The result is two distributions of crimes by zone, one for origins and one for destinations.

How does one assign crime events to a zone? There are two general ways to do this:

1. Nearest zone centroid - events are assigned to the zone centroid that is closest.

2. Point-in-polygon - events are assigned to the polygon within which it falls.

With the nearest zone centroid method, an incident is assigned to a zone to which it is closest whereas with the point-in-polygon method, an incident is assigned to a zone in which it falls within the boundary of that zone. Most GIS packages have a point-in-polygon routine and can implement that method.

In *CrimeStat*, on the Distance Analysis I page, there is an Assign Primary Points to Secondary Point routine that will make this assignment based on either method (see chapter 5). In both cases, the incident file must be the primary file and the zonal file must be the secondary file. In the nearest zone centroid method, the routine will assign each event to the centroid to which it is nearest. It will then sum the number of incidents assigned by zone and will add this as a new field to the secondary file (called *Freq*). In the point-in-polygon method, the user must also provide the boundary file for the zones as an *ArcView* shape file. The routine will read the boundary file and will determine in which

polygon an incident falls, and will then assign the incident to that zone. As with the nearest zone centroid method, it will then sum the number of incidents assigned by zone and will add this as a new field (*Freq*) to the secondary file. Chapter 5 presents details of these two routines, and isn't repeated here.

There are advantages and disadvantages to each method. The nearest zone centroid has attributes that are probably closest to the location where the incident occurs. This is important in relating socioeconomic and land use characteristics to the events during the trip generation stage (see chapter 13). Typically, social characteristics change gradually over an urban landscape so that an incident is probably closer to its nearest zone centroid than to any other zone centroid. In the case of the point-in-polygon method, incidents are not necessarily assigned to the nearest centroid since zonal polygons are frequently irregular in shape. Thus, to represent the underlying characteristics of the location in which the incident occurs by a point-in-polygon may end up assigning an incident to a zone that is quite different from where it should be located.

On the other hand, the main advantage of a point-in-polygon assignment is if the zone has a meaning in terms of containment or membership. For example, if a police reporting district (which could be a sub-set of a larger police precinct) is used as the zonal model, assigning incidents to the reporting district within which they fall will ensure that the incidents are assigned to the correct police precinct.

In other words, if it's important that events be assigned to the area to which they belong, then the point-in-polygon method is usually the best. On the other hand, if it is important that the incidents be assigned to the zone to which they are most similar, then the nearest centroid method is usually the best.

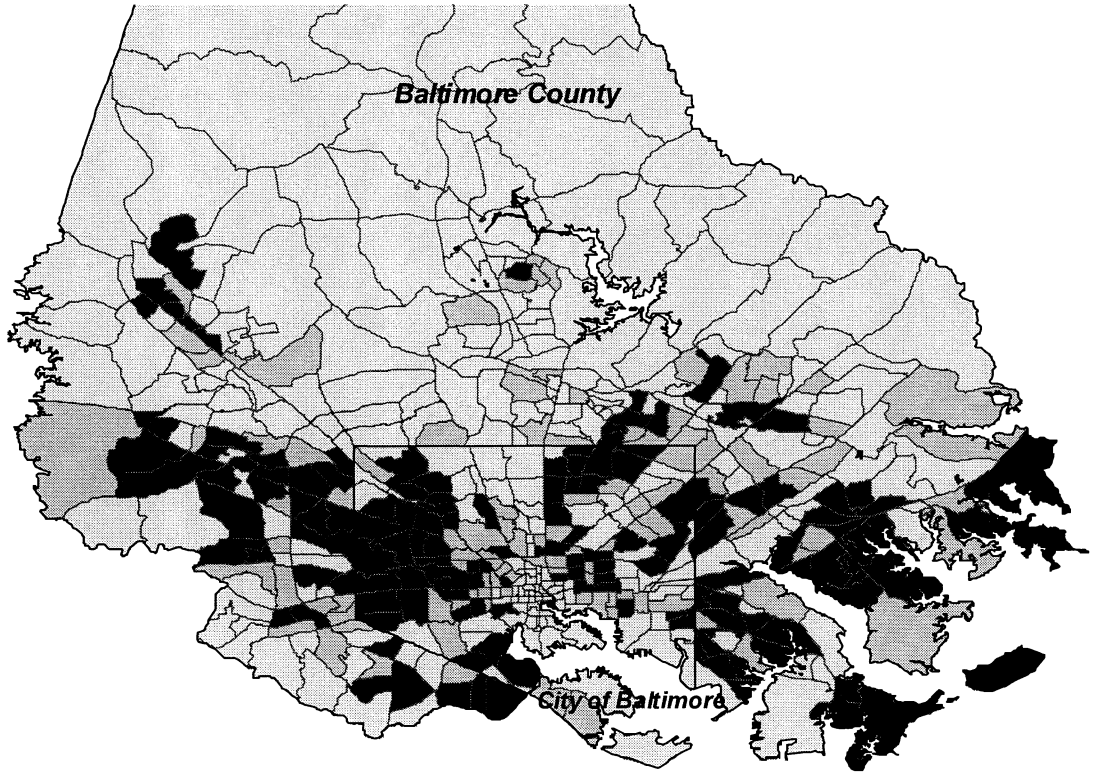
Figure 12.5 shows the number of crimes by origin zone while figure 12.6 shows the number of crimes by destination zone. In both cases, events were assigned by the nearest zone centroid method.

Adjusting Crime Events Estimated from Arrest Records for Accuracy

There is another subtlety that affects the assignment to a zone. The method that has been described assigns records in which there is both an origin and a destination location, such as an arrest record. The reason for doing this is that there is an implied trip between the origin and the destination, as was discussed above and in chapter 11. However, there may be a difference between the distribution of crimes by destination from the arrest records and the actual distribution of crimes from all incidents. The reason is that arrest records represent only a sub-set of all the crime records and, often, a small sub-set. If there are any spatial differences in the arrest likelihood across a metropolitan area, it is possible that some areas will have a higher proportion of offenders being arrested than other areas. The result would be a discrepancy between the distribution of crimes by arrested individuals and the actual distribution of crimes. In other words, the distribution of crimes as identified by the arrest records could be a biased estimate of the actual distribution of crimes. The result could be that the origins of those offenders who were

Figure 12.5:

Crimes Origins by TAZ
Number of Crimes Originating in TAZ
Baltimore County: 1993-1997



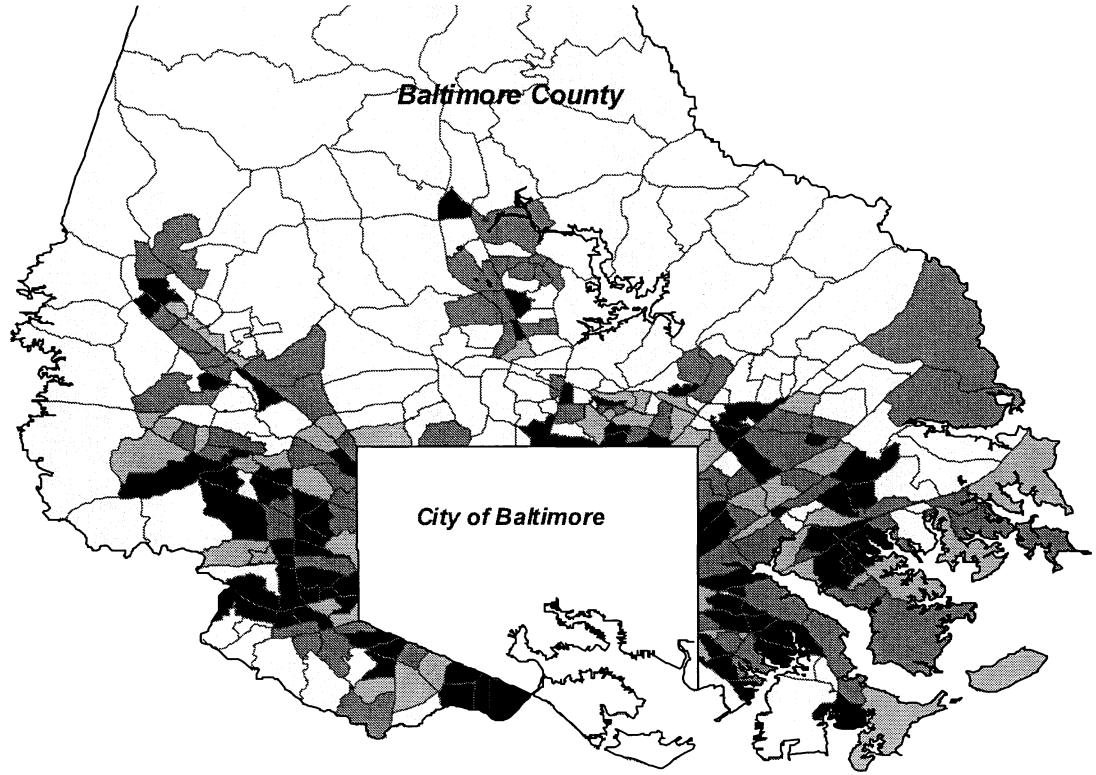
0 10 20 Miles

N
W E
S

City of Baltimore
Baltimore County
Number of crimes occurring in TAZ
Less than 50
50 - 99
100 - 149
150 - 199
200 or more

Figure 12.6:

Crimes Destinations by TAZ
Number of Crimes Occurring in TAZ
Baltimore County: 1993-1997



0 10 20 Miles

N
W E
S

City of Baltimore
Baltimore County
Number of crimes occurring in TAZ
Less than 50
50 - 99
100-149
150 - 199
200 or more

caught will be exaggerated relative to the origins of those offenders who were not caught, and the entire model could end up being biased.[3]

If there is a sizeable discrepancy between the distribution of crimes from the arrest records and the actual distribution of crimes, it is important to correct this. In the Assign Primary Points to Secondary Points routine on the Distance Analysis II page, it is possible to weight the assignment by another variable. This variable can reside on either the secondary (zone) file or on another file. A typical correction weight variable would be a proportion that adjusts the empirical distribution of crime destinations by the true distribution. Thus, a weight greater than 1.0 would increase the proportion whereas a weight smaller than 1.0 would decrease the proportion. A weight of 1.0 would maintain the same proportion.

In order to do this, however, one has to convert the number of crime destinations into proportions. Let's take an example. Suppose the empirical and true distribution of crime destinations was as follows (table 12.1):

Table 12.1

Proportional Weighting Empirical Assignment of Crime Destinations

Zone	Empirical Distribution	True Distribution	Proportional Weight
101	.04	.05	1.25
102	.03	.025	0.83
103	.015	.015	1.00
etc.			

In the example, the actual (true) distribution of crimes for zone 101 is greater than what was measured in the incident-to-zone assignment by a factor of 1.25 to 1 (i.e., .05/.04). Thus, the weight assigned to zone 101 is 1.25. In zone 102, on the other hand, the actual distribution of crime destinations was smaller than what was estimated from the incident-to-zone assignment by a factor of 0.83. Thus, the weight assigned to zone 102 is 0.83. Finally, the proportion of crimes in the empirical and actual distributions for zone 103 is exactly the same. Thus, the weight assigned to zone 103 is 1.00.

The weight variable will be typically a column in the secondary file that corrects the empirical distribution. Naturally, the first time this is done, an analyst would probably not know the empirical distribution. Thus, it will be necessary to repeat the incident-to-zone assignment, the first time in order to count the empirical distribution while the second time to weight that count by the correction factor (which will have been added as a variable to the secondary - zonal, file). See chapter 5 for a more complete discussion of weighting a primary points (incidents) to secondary points (zones) assignment.

Note, the adjustment of the empirical count (assignment) is done usually for the destination variable, not the origin variable. In the case of crime events, police will know the destination of the crime a lot more accurately than they will the origin since there is a crime record on file for the incident. Hence, any discrepancy between the empirical distribution of crimes and the actual distribution will only be known for crime locations (destinations). Therefore, in correcting the empirical distribution, we are assuming that we are also correcting the true distribution of origins, too. It should be obvious, though, that we really don't know. Unless one can obtain a "true" distribution of crime origins and, thereby, correct the origin distribution as well as the destination distribution, one has to assume that the adjustment in the destinations will also correct the distribution of the origins during the balancing stage (see chapter 13 on trip generation).

Obtaining Crime Data by Sub-Types

Till now, the discussion has focused on the total number of crimes that occur within a zone. Clearly, it is possible (and preferable) to break this down into distinct sub-groups. Thus, a separate distribution for robberies, burglaries, vehicle thefts, homicides, and other crime types can be complied. In each case, the separate distribution is being assembled in order to produce distinct models of crime travel by that type. The journey to crime literature has long illustrated the differences in travel distance by crime type, and it would be expected that there are substantial differences in travel patterns as well. Most crime analysts and researchers will want to break down crimes into these distinct categories. Similarly, an analysis by time of day or day of week also would require breaking down crimes by these different temporal categories. In general, an analysis of all crimes is not very meaningful for most police departments. Instead, the focus has to be on crime types and, perhaps, times of day with other sub-sets also being important (e.g., method of operation, use of weapons).

The method used to assign these individual crimes to zones would be, however, exactly the same as for the total number of crimes that was illustrated above. As with the total number of crimes, there would be differential weighting of zones in order to correct any bias in the distribution of crimes calculated from the arrest records compared to the actual distribution of incidents as identified by total crime reports.

Adequate Sample Size

A problem with this approach arises, however. By breaking down crimes into distinct sub-groups (by crime type, time of day, day of week, method of operation, etc), smaller samples are produced. As the sample size decreases, the likelihood of modeling error increases. If the sample is too small, then any of the zonal estimates that are produced in the trip generation stage will be subject to considerable sampling error. Similarly, in subsequent stages (trip distribution and mode split), these small samples are further broken down into cells with very small samples, with most having zero incidents. In other words, sampling error becomes a problem if the total number of crimes is broken down into very small sub-sets, and the model becomes unreliable.

How would one know whether a model is unreliable or not? Probably the simplest way is to repeat the model on two different years worth of data. That is, the analyst constructs the travel demand model on one year's worth of data and then repeats it on another year. If the variables selected during the trip generation stage are the same and if their coefficients are approximately equal, then the model would appear to be reasonably stable. On the other hand, if there are substantial differences in the selected variables and in their coefficients, most likely the data set was too small for the construction of a stable model. One could do formal tests on differences between the coefficients to see whether they are similar or different. But, a general review of the coefficients should indicate whether there is stability or variability. It's not a 'hard and fast' rule since any differences could be due to real changes in the environment creating crime. But, unless there is some obvious explanation for the differences, most likely they indicate that a model is too unreliable to be used from one year to the next (i.e., the sample size is probably too small).

Thus, there is a balance that has to be maintained between having a large enough sample to produce reasonably reliable trip generation and trip distribution coefficients, and breaking down the data into more meaningful categories for analysts and researchers. In general, I believe it is a good idea to model all crimes first before modeling specific sub-types. The reason is to establish baseline characteristics - variables and coefficients. It will become easier to understand how different crime sub-types vary once the overall distribution is known.

Developing a Predictive Model

The above discussion dealt with summarizing crime incidents by zones, both the location where the crimes occurred (the destinations) as well as the locations where the offender was living (the assumed origins). In order to develop a predictive model of crime origins and destinations, it is also necessary to put together a data set of predictive variables. Typically, these will be socioeconomic and land use variables, though other types of variables can be included as well.

Obtaining Socioeconomic Data

Population

The most common type of predictive data will be socioeconomic variables. Among these are population, employment, income levels, poverty data, and household characteristics. At the minimum, population will be an important variable. As mentioned in chapter 11, the crime travel demand model is an aggregate (volume) model. That is, it counts the total number of crime trips (by origin and by destination). Since, the number of trips is generally a function of the total number of persons in a zone, all other factors being equal, population inevitably will enter as either the most important or among the most important variables, as both an origin and a destination variable.

Population could be measured by sub-sets (or proxy) variables, too. For example, the number of households, the number of teenagers, and the number of married couples are also

sub-sets of the total population; the correlation among these variables is usually very high. Which variable is chosen will depend on what type of crime is being predicted. For the total number of crimes, probably the total population should be used because it is a larger and more stable estimate of the total "at risk" population. For specific crimes, however, it may be desirable to choose a sub-set of population. For example, for car thefts, the distribution of males, ages 16-30, might be a more intuitive baseline variable since those age groups contribute disproportionately to vehicle thefts (as they do to most crime types). The disadvantage in using this variable may be the smaller sample sizes that are obtained for some zones. A good way to test this is to model it twice, once with total population and once with the sub-set variable. If the overall predictability of the model is about the same (or, better, if the sub-set variable predicts better than the total population), then the use of the sub-set population will be preferable to the total population. On the other hand, if there isn't much difference, stick with total population as it is a larger, and more stable, variable.

Employment

A second variable that usually comes up is total employment. This is particularly valuable as a predictor of crime destinations since many crimes are attracted to employment areas (e.g., robberies, burglaries, vehicle thefts). Usually a distinction is made between *retail* and *non-retail* employment, though other distinctions can also be made (e.g., office employment, government employment, military employment). The reason is that retail employment is usually found in commercial areas (e.g., shopping malls, strip malls, retail centers). In the case of Baltimore County, for example, retail employment is the strongest predictor of crime destinations.

As an origin variable, too, employment could be important. In the three models that were compared for this version of *CrimeStat* (Baltimore County, Chicago, Las Vegas), employment was seen as a predictor variable for crime origins in several cases, too, usually as a negative predictor (i.e., less employment is associated with more crime). The reason may be less clear, but may have to do with the lack of opportunities in certain districts and neighborhoods.

Income levels

Another obvious variable is income measured in some way. The relationship between crime and low income has long been noted. There are several possible income-type variables that could be used in a model. The most obvious is the total income level of a zone. The U. S. Census Bureau has a total income variable that is part of their SF 3A release. This measures the total of all household incomes in the census. While this variable captures the total available income in the zone, it is not a very intuitive measure. Consequently, other measures are usually used, such as income per capita or median household income. Median household income is usually a more typical measure since the average income per person can be affected by extreme values.

An important issue about income levels, no matter how measured, is that they inflate over time. That is, since income reflects monetary value at any one point, it does not

have a fixed reference point. What this could mean in a model is that, over time, income levels will increase (in absolute terms) due simply to inflation. A model that established, for example, a negative relationship between income and crime (i.e., the higher the income of the zone, the less crime) for one year would end up predicting lower crime levels for another year simply due to inflation.

It's important to standardize income in order to prevent the impact of inflation affecting the model. There are two ways that this is usually done. First, one can standardize income by subtracting the mean and dividing by the standard deviation. That is,

$$Z_i = \frac{Income_i - MeanIncome}{SdIncome} \quad (12.1)$$

where $Income_i$ is the income of each zone, i, MeanIncome is the mean income of all zones, and SdIncome is the standard deviation of all zones. This is a classic standardized measure.

A second way to standardize income is to define *relative* income. That is, the income level of each zone is compared to the income level of the zone with the highest income. That is,

$$I_i = \frac{Income_{highest} - Income_i}{Income_{highest}} \times 100 \quad (12.2)$$

where $Income_{highest}$ is the income level of the zone with the highest income. This index measures the income of a zone relative to the income of the highest income zone. The closer the income level of the zone is to the highest income zone, the smaller the index. Thus, this is an *income inequality index*, similar to the Gini index though more simply calculated. The zone with the highest income will have a value of 0 whereas the zone with the lowest income will have a positive value roughly reflecting the relative differences in income levels between the lowest and the highest.

Each of these measures will prevent a shift in the predicted values due to inflation, though they each measure slightly different attributes; the first measures just absolute income levels (standardized) while the second measures the degree of inequality.

Another type of income variable is the number of persons living under poverty. Again, the relationship between poverty and crime has long been noted (Bursik and Grasmick, 1993).. Thus, a variable that measures poverty directly could add sensitivity to a model that simple income might not detect. The issue of measuring poverty, however, is a complex one. Different government agencies use different measures. For a discussion, see Citro and Michael (1995).

In general, typically median household income and the number of persons (or households) living under the poverty line do correlate quite well. Therefore, it's unlikely that both variables would be significant in a regression equation without, essentially, measuring the same thing. The same is true for education and income, which tend to correlate quite highly. Again, both variables in a regression equation would, essentially, be measuring the same thing. Thus, in a regression model, it's important to select only the strongest and most stable income variable in order to avoid duplicate measures. We'll return to this point in the next chapter.

Other socioeconomic variables

Other socioeconomic variables might be useful in a predictive model. Among these are race or ethnicity, vehicle ownership, number of single parent households, number of unemployed workers, number of persons living in large rental buildings, and others. Again, these variables might produce greater differentiation in a model. But, at the same time, they tend to overlap with income variables and may be measuring the same thing.

Obtaining Land Use Data

Aside from socioeconomic variables, there are land use variables that could be important in predicting both crime origins and destinations. Among these are parks, bars, pawn shops, cheque cashing businesses, the location of shopping malls, retail space, stadiums, train stations, intra-urban metro stations, bus stations, parking lots, hospitals, and adjacency to major freeways or arterial roads. There are a wide variety of land use variables that appear to be important in attracting crime as well as in providing an environment that may encourage people to commit crimes. A thorough elaboration of potential land use variables would help to identify particular attributes associates with crime and, thereby, increase the predictive ability of a model.

There are two ways to document these land uses. One is as a simple categorical ('dummy') variable whereby the field is given a '1' if that land use exists in the zone and a '0' otherwise (e.g., there is a park in the zone; a freeway runs through the zone; there is a stadium in the zone). The second is a count of the level of that land use variable (e.g., the number of bars; retail square footage; park acreage; number of parking stalls in a parking lot). The second variable is, clearly, more precise than the first, but is much harder to document. The availability of data will be a constraining factor in building up a set of land use variables that might predict crime origins or destinations.

Still, before an extensive data inventory is initiated, some cautionary words are in order. In the three studies illustrated in this version of *CrimeStat*, however, few land use variables survived once population, employment and income levels were included. The reason is that many land use variables correlate with these basic variables (e.g., the amount of retail space correlates with retail employment; bars correlate with low income). Thus, in spite of intuitively being related, it was found that most of the land use variables did not improve the models beyond the basic variables.

Special generators

There are exceptions, however. Particularly, there are *special generators* that attract crimes out of proportion to the amount of employment at those locations. Among these are stadiums, major train stations, airports, and large parks. Because these are major regional facilities and, in the case of stadiums and parks, used unevenly from day-to-day, they may attract more crimes that would be expected on the basis of the level of employment at those locations. Traditional travel demand models have incorporated these as special variables because they can account for variability that is not general throughout the study area. More on this in the next chapter.

Spatial Location Variables

Centrality

In addition to socioeconomic and land use variables, spatial location variables *might* be relevant. There are two types of spatial location variables that might be relevant. The first is the *centrality* of the metropolitan area. In most American cities, the central downtown area has a uniqueness that is greater than that which is explained by any one variable. For example, not only is there a large amount of employment in most Central Business Districts (CBD), but there are amenities that are associated with a central location. Usually, there is a greater concentration of restaurants and stores in CBD's and other employment centers. Entertainment often is more concentrated in the CBD; this is not true in many large metropolitan areas (e.g., Los Angeles), but it is true in enough of them to make the CBD an entertainment center as well as an employment center. Similarly, transit lines tend to concentrate in the CBD.

In other words, the CBD is a unique place that affects crime trips. Some CBD's have a large number of crime incidents whereas other don't. Nevertheless, measuring it in a predictive equation *might* increase the predictability of a production or attraction model. A simple variable is the distance from some point within the CBD, for example distance from the City Hall. Zones that are close are liable to have a greater number of crime productions and crime attractions, especially, than zones farther away. This type of spatial effect is very similar to the *first-order* effect described in chapter 5.

Local Spatial Autocorrelation

Another type of spatial effect is a localized similarity between adjacent zones. In other words, there frequently is a spatial autocorrelation in crime productions or attractions between adjacent zones. It is the *second-order* spatial effects described in chapter 5. Zones that have a lot of crimes occurring within them are frequently located next to zones that also have a lot of crimes occurring, and the converse. Spatial regression models (e.g., spatial lag model, geographically-weighted regression) explicitly incorporate this type of spatial effect as a predictor variable.

In this version of *CrimeStat*, there is not a spatial regression model. Therefore, there is not a simple way to handle local spatial effects in the current version, at least in the trip generation stage. The second stage of the model - trip distribution, incorporates an explicit spatial component by weighting distance in estimating the interaction between zones. Thus, any spatial error produced during the trip generation stage is frequently compensated for during the trip distribution stage.

But, if the user wants to incorporate local spatial autocorrelation explicitly in the trip generation stage, then the use of a Local Moran (see chapter 7) or a simple adjacency measure (e.g., '1' if the average of adjacent zones is greater than the mean for all zones and '0' if it is not) may be sufficient in account for the localized spatial autocorrelation.

There are advantages and disadvantages to including first- or second-order spatial effects in a travel model. Since the trip distribution stage has an explicit spatial interaction term, any errors from the first stage (trip generation) are usually accounted for during the second stage. Thus, there is very little advantage to be gained from including a second-order (spatial autocorrelation) variable. However, including a first-order variable can improve predictability of the trip generation model. But, in postulating a simple relationship (e.g., distance from the City Hall), one is assuming that the relationship will hold over time. Over a short period of time, it probably will. But for a longer forecast (e.g., 20 years), it may not. Nevertheless, it represents a possible predictor of crimes.

Defining Policy or Intervention Variables

Aside from socioeconomic and land use variables, a model might include some policy or intervention variables. One of the best uses of a travel demand model is to model the likely effect of a change in one of the predictive variables. A simple one would be the likely effect of building a new facility, for example a shopping mall. In the estimation stage, if the analyst can show that shopping malls are associated with higher (or lower) crimes occurring (i.e., destinations), then a theoretical mall could be placed in a zone and the model run with that as a new input for the zone (with every other variable being the same for all zones). Since the travel demand model is sequential, the impact of new crime trips being attracted to the zone can be followed through the different stages of the model.

There may be other policy or intervention *experiments* that can be conducted with a crime travel demand model. In each case, it is necessary to include the variable in the estimation model to establish a coefficient for it. Then, in the simulated experiment, the variable is re-arranged or allocated differentially and the model is recalculated. Again, the result can be used to estimate what the likely effects of the intervention could be on crime travel patterns.

Among the possible policy or interventions are the construction of a particular type of facility (as mentioned above with a new shopping mall), changing the level of policing in a zone, the creation of a drug treatment center, the establishment of a job retraining center, or the reduction in the number of adult book shops. There are a large number of possible interventions that might affect the level of crime - either produced (origins) or attracted

(destinations). Further, not all of the interventions might reduce crime levels, but some could even increase it (e.g., add new shopping malls). Nevertheless, the ability to add interventions in the model make it a useful device to estimate the likely effects on crime levels without having to actually implement the changes.

In the three studies presented in this version of *CrimeStat*, there were no interventions that were estimated. The model is still very new in its applicability to crime analysis and there hasn't been time to create such a scenario. Still, this type of experiment or 'variable' is an important one and which could make the crime travel demand model a very powerful analysis tool.

Where to Obtain these Data?

Many of these data are easily founds, while others are more difficult. A lot of socioeconomic data is available in the decennial census and distributed by the U.S. Census Bureau. Data on population, households, and income levels can be obtained from the Census Bureau for geographies as small as blocks or block groups. One of the deficiencies of the census data, however, is the lack of information on employment.

An alternative is to obtain data from a Council of Governments or Metropolitan Planning Organization. A Council of Governments (COG) is a regional association of cities and counties that is involved in planning; sometimes it is called an Association of Governments. Virtually every metropolitan area in the United States has a COG that can be a source of information on both population, employment, and, occasionally, land use. Many COG's have a forecasting group that estimates both population and employment, sometimes for very small geographical units. The Houston-Galveston Area Council, for example, has an extensive database of all firms with 10 or more employees and updates this continually utilizing business permits, purchased lists from other organizations, and aerial photographs for identifying new commercial developments. They produce estimates of employment for small grid cells that are approximately 1000 feet on a side; however, these data are released only at the Traffic Analysis Zone (TAZ) level. For more information, see the National Association of Regional Councils (http://www.narc.org).

A Metropolitan Planning Organization (MPO) is a regional transportation planning agency. In many metropolitan areas (e.g., Los Angeles, Houston, Washington, DC), the MPO is part of the COG while in other metropolitan areas (e.g., San Francisco, Chicago), it is not. They will obtain both population and employment data for the TAZ's as part of their travel modeling functions. For more information, see the Association of Metropolitan Planning Organizations (http://www.ampo.org). In short, it is generally possible to obtain data on population and employment from either COGs or MPOs.

Land use data is more difficult to obtain. Simple information can often be obtained from Yellow Pages or online business directories, for example the location of bars and nightclubs. More detailed data may have to be obtained from particular cities and counties. Generally, larger cities have a planning department or a public works department who maintains some land use data. The quality of this information will vary, however, and may

not be consistent across jurisdictions. In a large metropolitan area, it may be possible to obtain regional land use information from the COG, the MPO, a regional utility company, a database of business permits, the tax assessor's office, or even the Army Corps of Engineers.

The point that has to be realized is that a lot of effort is needed to put together a data base for modeling crime travel. Once developed, however, it can be used repeatedly as predictors for different types of crime and can be updated more easily. Like a GIS system, there is a substantial amount of effort 'up front' in order to build a model. But, once collected, the information can be very useful for a multitude of purposes.

Creating an Integrated Data Set

The information that has been collected - both data on crime origins and destinations as well as socioeconomic, land use and policy interventions, needs to be integrated into a single zonal model. That is, the data need to be allocated to zones, both origin zones and destination zones. The result will be ***two*** different data sets, one for crime origins and one for crime destinations. The origin data set will cover the origin zones while the destination data set will cover the destination zones. The same predictor variables, however, can be in both data sets as these variables could predict either origins or destinations, or both.

Allocating data to zones

There are two steps in assembling the data into two data sets. First, the data have to be allocated to the zonal system used. In some cases, these data may be easily available (e.g., obtaining population and employment data by TAZ's when the TAZ is the zonal unit used). In other cases, it may be necessary to allocate the data from one geographical zonal unit to another (e.g., from census block groups to TAZ's). GIS is a very powerful tool for allocating data from one "layer" to another. However, it has to be realized that errors will result from an allocation. For example, breaking up a larger zone into small sub-zones (e.g., breaking up a large census tract into four small grid cells) will lead to some error in the allocation. The GIS splitting routines usually assume that the data are split proportionately between the four 'pieces'. Thus, if employment from a census tract is allocated to two grid cells, one assumes that the workers are uniformly distributed within the census tract and the two grid cells will each capture a share equal to their area relative to the larger tract. This may or may not be true. Where it's not true, adjustments need to be made to ensure that zones represent relatively uniform populations.

The point is, there is error in allocating data from one type of unit to another, and the analyst has to be aware of these potential sources. It is generally better to obtain data at the smallest possible geographical unit in order to minimize the splitting problem described above. Aggregation usually causes less error than splitting. On the other hand, as mentioned at the beginning of this chapter, the larger the zonal unit that is used, the greater the likelihood that there will be within-zone (intra-zonal) trips.

Combining data into origin and destination data sets

The second step is the combining of all the data into two separate data sets, one for origins and one for destinations. All the data that are used for the origin model should be together while all the data that are used in the destination model should be together. Many variables will be in both data sets (e.g., population, employment, income) whereas some variables only make sense as an origin or a destination variable (e.g., residential areas as an origin variable for bank robberies; a rail station as a destination variable for larceny or robbery). Since the origin zones will usually be more numerous than the destination zones (because they include the destinations and those from surrounding jurisdictions), the data have to be consistent across all zones that are used.

For use in the *CrimeStat* crime travel demand module, these data sets should be in one of the acceptable formats (dbf, dat, or ODBC-compliant). I have found that building the data first in a spreadsheet (e.g., Excel or Lotus 1-2-3) is easier to do because variables can be more easily added. Once constructed, the spreadsheet is converted into a dbf file for use by *CrimeStat*.

Obtaining Network Data

The final type of data that needs to be obtained is a network. This is important for the third and fourth stages in the crime travel demand model - mode split and network assignment. In the mode split routine, trips from each origin zone to each destination zone are divided into different travel modes. For driving travel modes, travel has to go along a road network. For walking or biking, there may be additional segments that are not in the road network (e.g., bike paths, short cuts for pedestrians); these can usually be added to the road network to make a more realistic representation. However, for transit modes, the trips have to go along a transit route. In the network assignment routine, all zone-to-zone trips by each travel mode are assigned to particular routes. For this, a network is needed, one for each mode.

In both these cases, travel occurs along a network. That is, the distance (or travel time or travel cost) from one location to another is calculated using the network, rather than as direct or indirect distance. A network is a collection of segments that are interconnected. Travel can only occur on the segments. Each segment has two or more nodes and one or more connecting lines. Travel is from segment to segment. Hence, the *end nodes* have a special status as the connectors which allow travel from one segment to another.

In chapter 16, a more extensive discussion of the shortest cost/path algorithm used for network travel is explained. But, essentially, a 'trip' goes from the origin location to the closest location on the network. It then proceeds along the network, taking the shortest path, until it reaches a node closest to the destination. It then travels from that node to the final destination. Thus, the *representation* of the network is very critical. It has to be accurate and reasonably comprehensive.

There are three types of basic networks that need to be considered:

1. Road network (with additional walking or biking segments)
2. Bus network
3. Train network (if appropriate).

In addition, there can be specialized bicycle networks that are distinct from the road network. However, most transportation agencies model bike trips using the road network. Let's discuss each of these.

Road network

In a GIS system, there are typically two types of road networks that are used:

1. A single-directional (or linear) network
2. A bi-directional network.

Single-directional road network

In a single-directional network, travel can occur in both directions along a segment. A typical example is the TIGER system created by the U.S. Census Bureau (2004a). In this system, each segment typically represent the travel along a road from one intersection to another (i.e., a block in length), though there are exceptions. Travel can occur in both directions in the network unless there are special codes added to indicate a one-way street. The TIGER system, in particular, has a number of attributes associated with it - sides (left side, right side), address ranges (on both sides), census and political designators (again, by sides), and other attributes. This type of network is very common in GIS systems and is widely used in police departments. Because of the address ranges and because it is easily available from the U.S. Census Bureau or companies who improve the TIGER system, this type of network forms the basis of most geo-coding systems.

There are problems with a single-directional network, however. Among these are the inability to distinguish direction or a one-way street. From a network modeling perspective, travel can occur in either direction. It is possible to put a field in the data base that identifies whether the street is one-way or not, and to indicate the direction of travel. But, this has to be added by the user since the TIGER system does not specify that information.

A second problem is the lack of information about travel time or cost on the network. The only metric in the TIGER system are address ranges and, implicitly, distance. However, since travel varies substantially by type of road (larger functional classes have higher speeds) and by time of day due to differing levels of congestion, such a system lacks very important information for modeling travel. The TIGER (or similar) system does have functional class codes that distinguish different levels of road capacity (e.g., Interstate highways, state highways, principal arterial roads, collector roads, etc). It is possible to assign arbitrary average speeds to each of these classes (e.g., 45 miles per hour to an interstate highway; 30 miles per hour to a principal arterial; 20 miles an hour to a collector

road; and so forth). By doing so, a reasonable approximation to actual travel can be obtained. However, there is still not a sensitivity to travel time by time of day. For example, in an urban area, travel at the peak afternoon 'rush hour' (e.g., 3:30 PM - 7 PM) will be, on average, a lot slower than at off-peak hours.

This brings up a third problem, namely that there is no interaction between the direction of travel and the travel time. On most principal arterial roads, travel is unequal in speed at any one time. For example, in many metropolitan areas, travel towards the downtown area is much slower in the morning than in the opposite direction, whereas the reverse is true in the afternoon. A single-directional network cannot distinguish this and the analysts have to add multiple fields to the attribute file in order to make these distinctions (e.g., PM peak from node A to node B direction; PM peak from node B to node A direction; etc).

A fourth problem may or may not exist with a single-directional network. These networks were designed to allow the U.S. Census Bureau to carry out the decennial census. Thus, a lot of attention has given to accuracy of streets and address ranges. Much less attention has been paid to the connectivity of the streets. A lot of the digitizing that goes into the network has been done by local governments, and the quality of this digitizing varies considerably. Some jurisdictions have very precise networks that are updated frequently while other jurisdictions have poorly defined networks that are often out of date. Drivers may know that they can travel from point A to point B via road C, but the network may not have been sufficiently updated to allow that trip to occur in a representation. In some cases, gaps between segments have been noted; the gaps may be very small, but they would prevent a model from 'traveling' from one segment to the next.

Bi-directional road network

A bi-directional network, on the other hand, separates travel in each direction. For example, if there are two nodes that connect a segment (node A and node B), then there are typically two segments for travel in each direction (from node A to node B, and from node B to node A). In this representation, a one-way street is simply a segment that does not have a reciprocal pair (i.e., there is only a node A to node B segment, and not the reverse).

Most transportation agencies use bi-directional road networks for their travel demand modeling. The reason is that multiple attributes can be assigned to each direction separately, a feature that simplifies the building of a realistic network. Thus, speeds for different time periods can be assigned as separate fields on each segment (or, what is usually, done there are separate networks for the different travel periods that are modeled). Travel volumes can be assigned to each segment which, in turn, allows the creation of a *vehicle miles traveled* (VMT) field (length x volume). VMT, in turn, can be combined with travel speed to produce an estimate of travel time (e.g., VMT divided by speed - in miles per hour, times 60 to produce minutes traveled). Further, one-way streets are automatically handled since each direction is a separate segment (i.e., there just won't be a reciprocal pair in the opposite direction).

In short, a bi-directional network allows more flexibility in the creation of a network and the ability to distinguish travel in different directions as well as travel time by direction and time of day. It is not surprising, therefore, that most travel demand models use a bi-directional representation. Note, one can add these attributes to a single-directional network, but this requires many additional fields.

A further strength of a bi-directional network is that it is usually quite up-to-date and connectivity has been ensured. Most transportation agencies spend a lot of time cleaning and updating the network. While there are always errors in a network representation, the accuracy of most modeling networks is very good.

There is a downside to bi-directional networks, however. Typically, most bi-directional networks model only the larger roadways, those that contribute to regional travel. Thus, all freeways, principal arterial roads, minor arterial roads, and collector roads are included. However, most neighborhood streets are not included. The reason this is done is because the travel demand model is aimed at estimating regional and sub-regional travel patterns. Very localized travel is not of importance (and, in fact, is typically intra-zonal in nature). The result is a very efficient network because it's a lot smaller. But, there may be some error by using a 'skeleton' network. In particular, local travel might be distorted with such a simplified network. For example, if a neighborhood is bounded by four arterial roads, but with no internal streets, according to the model a crime event that originates from within the neighborhood (i.e., the offender lives inside the neighborhood) could take any of the four arterial roads to leave the network. In reality, the offender will probably take a particular route rather than necessarily the arterial that is closest to the offender's address. This can be handled, but it requires additional coding.[4]

As an example, for Baltimore County and the City of Baltimore, figure 12.7 shows the 49,015 segments in the TIGER representation of these two jurisdictions while figure 12.8 shows the 11,045 segments that are used by the Baltimore Metropolitan Council in their travel demand modeling.[5] Further, since most of the streets in the modeling representation are two-way streets, in effect, there are only about 5,000-6,000 actual streets. In other words, the TIGER network is 4.4 times larger than the modeling network. This makes calculation a lot slower than with a simplified network.[6] As we shall see in chapter 16, the accuracy of a network is essential for a more realistic modeling of actual travel routes by offenders.

Bus Network

A bus network, on the other hand, is a specialized road network that follows the actual routes used by buses. The general road network is useful for modeling driving, walking and bicycle trips. But, it cannot be used for bus trips. The reason is simply that buses don't use every street but only the larger arterial roads. Further, travel along many bus routes is variable. That is, a full route might be used during the peak rush hours, but a shortened route might be used during the off-peak hours. Similarly, the frequency of buses (what is called *headway* by transit agencies) varies by time of day; again, in the rush hours, buses are more frequent (though slower) than during the off-peak hours.

Figure 12.7:

TIGER Street Network
49,015 Road Segments

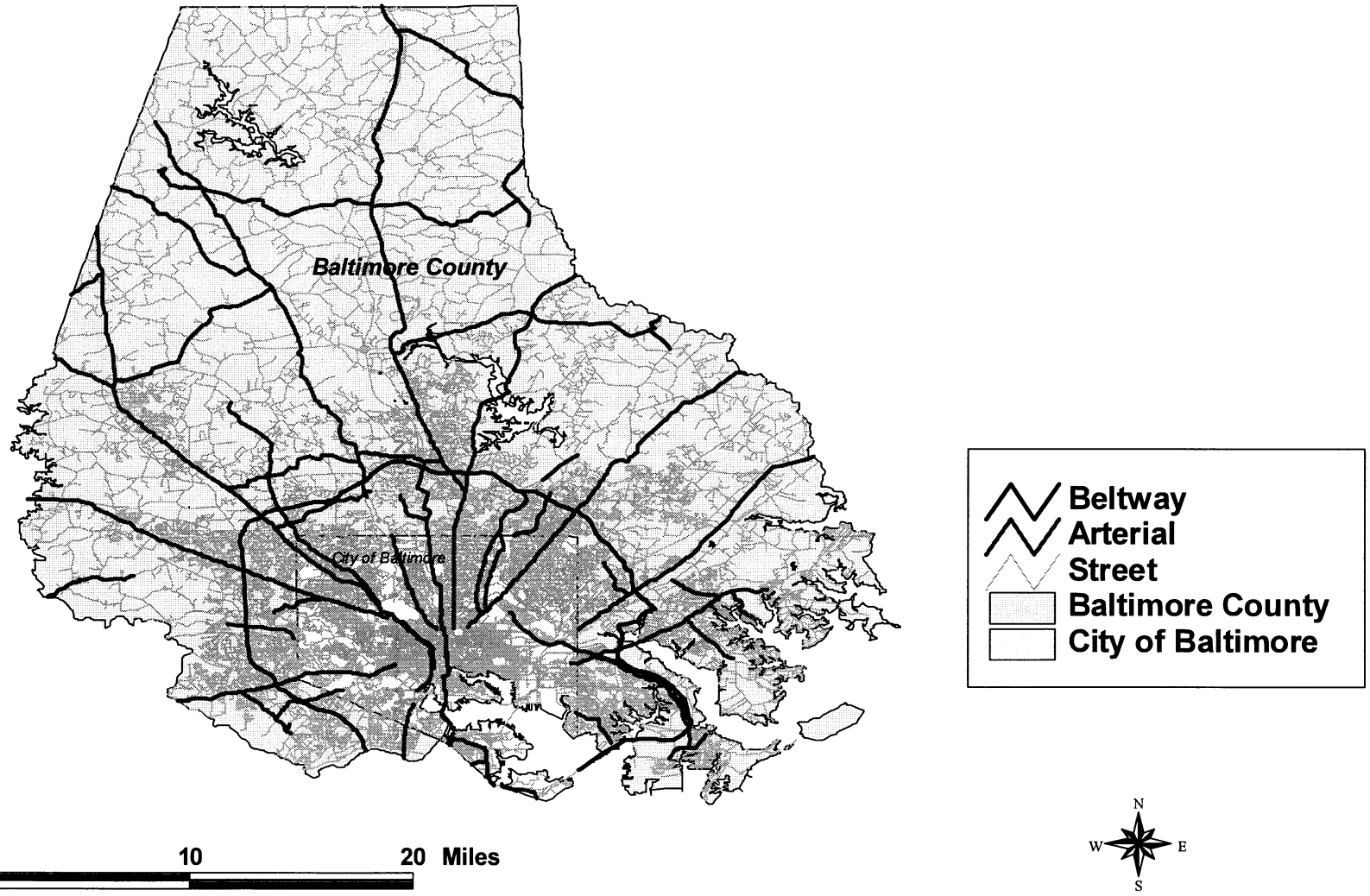
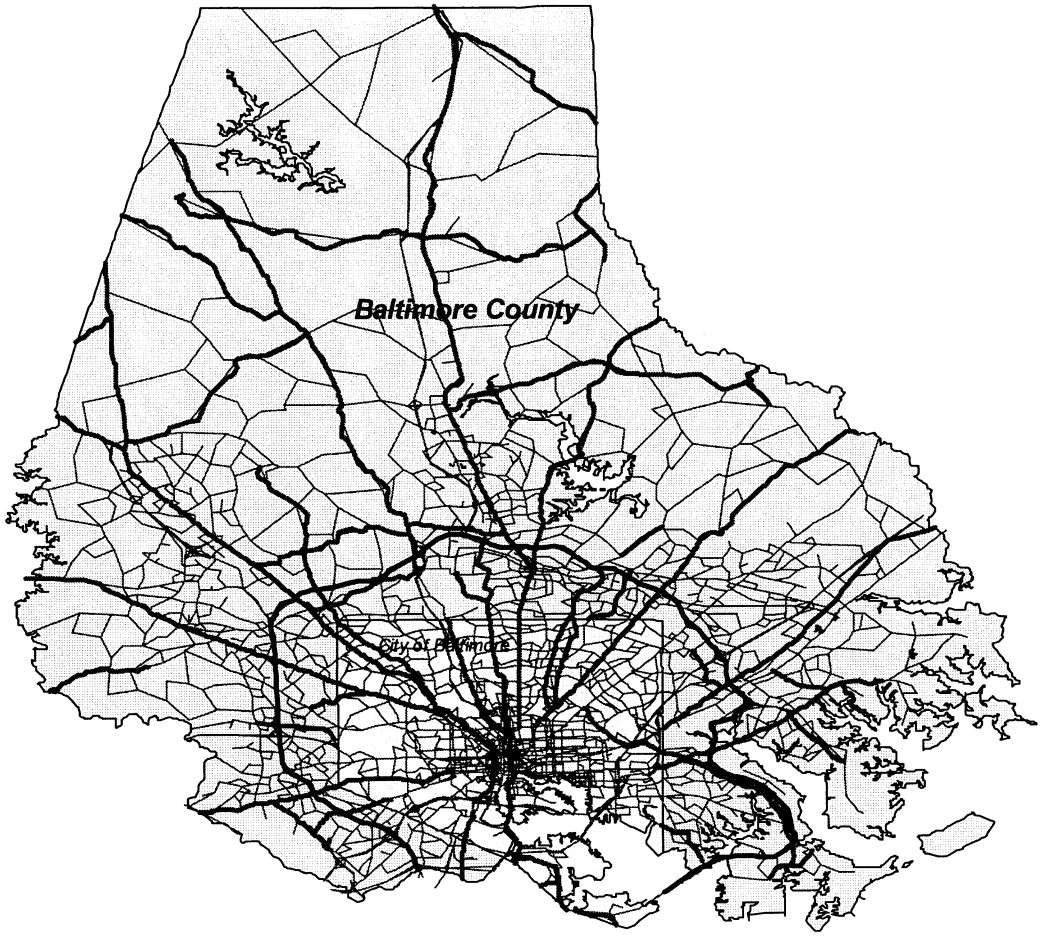


Figure 12.8:

Modeled Street Network
11,045 Road Segments



Beltway
Arterial
Modeled network
Baltimore County
City of Baltimore

0 10 20 Miles

N
W E
S

A bus network, therefore, is essential for modeling bus trips during both the mode split stage (when trips between zones are split into separate travel modes) and during the actual network assignment.

There are two components of a bus network that are needed, one of which is essential and the other is more optional. The first is a representation of the segments used in a bus network. Essentially, this is a network that shows where the buses travel. Bus travel can only occur along this network. As with road travel, the bus network can be represented either as a single-directional or as a bi-directional network though, again, most transportation modelers and transit agencies represent bus routes as single directions.

The second component is the location where access to the buses is allowed (i.e., the bus stops). Without explicitly indicating where there are loading and unloading points, a network routine would simply find the shortest distance from the origin to the bus route and 'add' the trip at that location. In practice, for most transit agencies, the degree of error in allowing direct access anywhere on the route is small since most bus routes stop very frequently (every couple of blocks). Thus, it may not be that important to actually code the bus stops since the amount of modeling error will be insignificant. However, for express buses and for those routes where there is a sizeable distance between bus stops, it is important to code the actual bus stops. In chapter 16, there is a more extensive discussion of coding bus routes. Figure 12.9 illustrates the bus network for Baltimore County and Baltimore City.

Train network

In those metropolitan areas that have intra-urban train travel, it is important to also obtain a rail network. An offender cannot travel on a train except by using the existing rail system. Further, unlike the bus network, it is impossible to 'enter' the train except at explicit station locations. Thus, it is critical to obtain both the network and the station locations. Figure 12.10 illustrates the intra-urban rail system in Baltimore County and Baltimore City.

Where to obtain network data?

There are many more choices in obtaining network data than with socioeconomic or land use data. Road networks can be obtained from the U.S. Census Bureau (for the TIGER system) or from vendors who improve on the TIGER system. For a modeling network, however, about the only choice is the Metropolitan Planning Organization (MPO). Since MPO's are set up to model regional travel, most agencies in a metropolitan area will defer to them for that activity. Transit networks can also be obtained from MPOs though the transit agencies will have their own networks that are usually more comprehensive than those of the MPO. As with all data, the MPO might charge for the data set, though policies vary widely.

Figure 12.9:

Baltimore Bus Network
Bus Routes

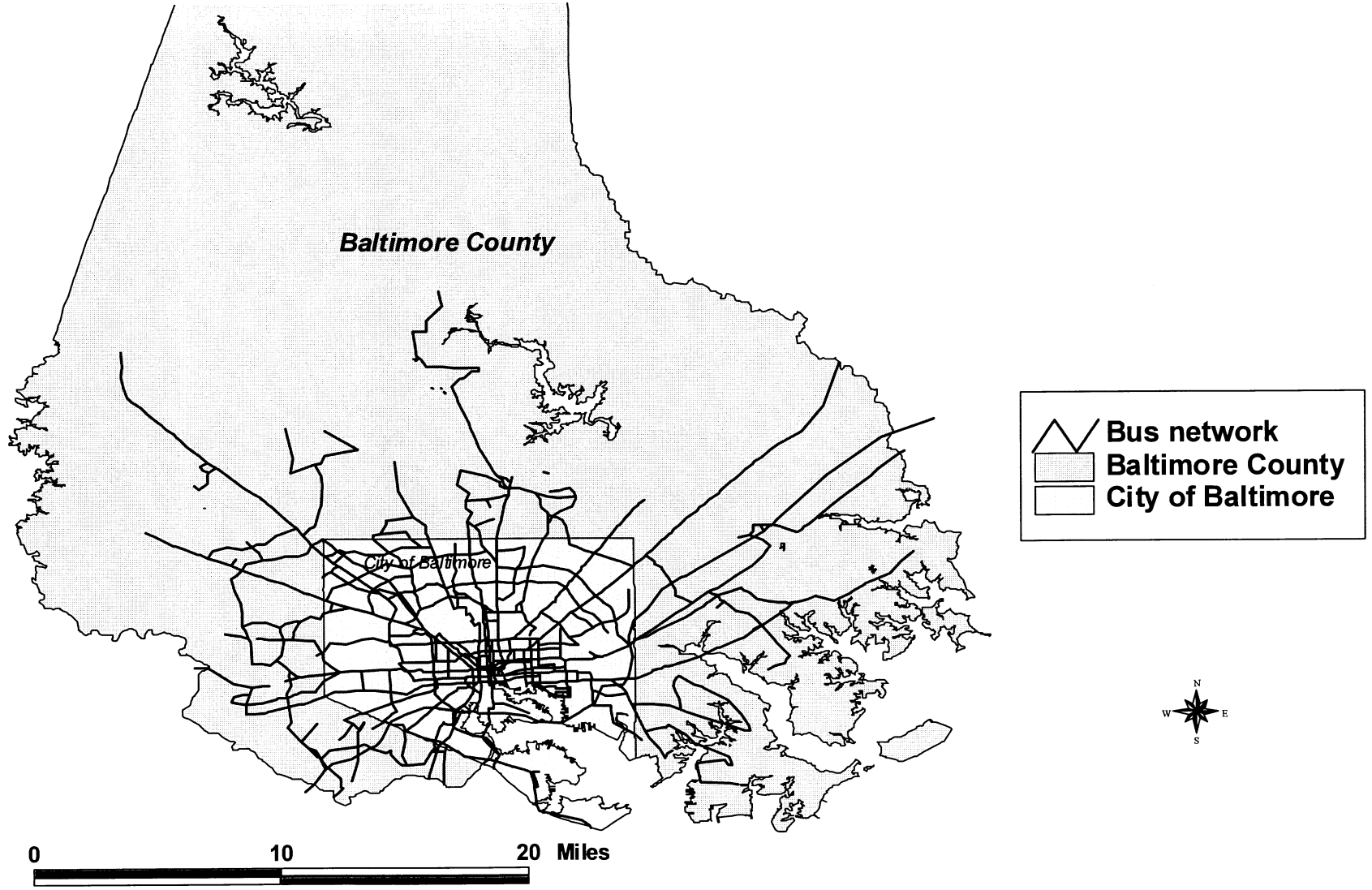
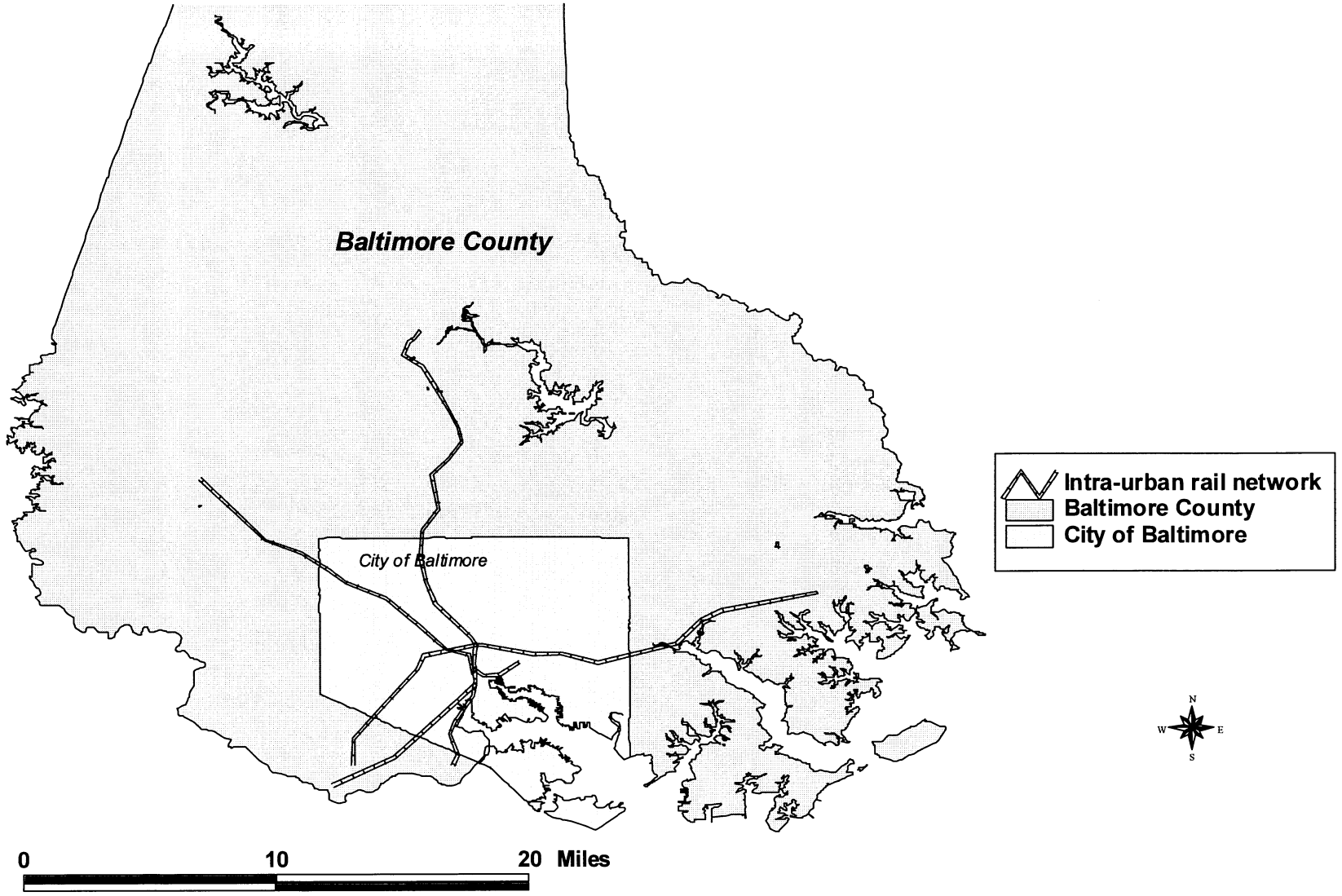


Figure 12.10:

Baltimore Intra-Urban Rail Network



Setting Up the Network

Whichever network is selected, the networks have to be set up for analysis. The user must specify the network that is to be used. There are two choices. First, if a network was defined on the Measurement parameters page (Data setup), that network can be used to calculate the shortest path. Second, whether a network has been defined on the Measurement parameters page or not, an alternative network can be selected on the network assignment page.

Network on the measurement parameters page

Check the 'Network on Measurement parameters page' box to use that network. All the parameters will have been defined for that setup (see Measurement parameters page).

Alternative network on the network assignment page

If an alternative network is to be used, it must be defined. Check the 'Alternative network' box and click on the 'Parameters' button.

Type of file

The network file can be either a shape file (line, polyline, or polylineZ file) or another file, either dBase IV 'dbf', Microsoft Access 'mdb', Ascii 'dat', or an ODBC-compliant file. The default is a shape file. If the file is a shape file, the routine will know the locations of the nodes. All the user needs to do is identify a weighting variable, if used and, possibly, a code (flag) for a one-way street. For a dBase IV or other file, the X and Y coordinate variables of the end nodes must be defined. These are called the "From" node and the "To" node, though there is no particular order. An optional weight variable is allowed for both a shape or dbf file. The routine identifies nodes and segments and finds the shortest path. If there are one-way streets in a bi-directional file, the flag fields for the "From" and "To" nodes should be defined.

Type of network

Specify whether the network is bi-directional or single directional.

One-way and two-way segments

One-way segments can be identified in a bi-directional file by a 'flag' field (it is not necessary in a single directional file). The 'flag' is a field for the end nodes of the segment with values of '0' and '1'. A '0' indicates that travel can pass through that node in either direction whereas a '1' indicates that travel can only pass from the other node of the segment (i.e., travel cannot occur from another segment that is connected to the node). The default assumption is for travel to be allowed through each node (i.e., there is a '0' assumed for each node). For bi-directional files, one-way streets can be indicated by defining a 'flag' for each end node (e.g., From one-way flag, To one-way flag) and giving the value '1' to the

node for which travel cannot pass from another segment. Flag fields which either have a '0' or are blank are assumed to allow travel to pass in either direction.

Type of coordinate system

The type of coordinate system for the network file is the same as for the primary file.

Measurement unit

By default, the shortest path is in terms of distance. However, each segment can be weighted by travel time, travel speed, or generalized cost. For travel time, the units are minutes, hours, or unspecified cost units. For speed, the units are miles per hour and kilometers per hour. In the case of speed as a weighting variable, it is automatically converted into travel time by dividing the distance of the segment by the speed, keeping units constant.

Network graph limit

Finally, the number of graph segments to be calculated is defined as the network limit. The default is 50,000 segments. Be sure that this number is slightly greater than the number of segments in your network.

Conclusion

In summary, a quite extensive collection of data is needed to run the crime travel demand model. Crime data, socioeconomic data, land use data, policy intervention scenarios, and network data must be obtained and prepared prior to running the models. Further, in practice, a lot of editing and 'cleaning' of data will be required during the modeling phase in order to improve the predictions.

Nevertheless, once the data are obtained, the model can be developed quite quickly. In the next chapter, we will examine the first stage of the crime travel demand model - trip generation.

Endnotes for Chapter 12

1. CATS was used as the prototype by the Federal Highway Administration for developing the original travel demand model. The grid was used because it minimized errors due to irregular size and shape. Nevertheless, that model has not been followed by planning agencies in the U.S.

2. If the actual origin was an intermediate location between the home and the crime location, then with a large sample of crimes and offenders the idiosyncracies of one offender's crime travel pattern is not going to effect the coefficients of the prediction model to any great extent. If *all* offenders from a particular zone committed crimes from an intermediate location which was always the same, then that condition might effect the coefficients (assuming one could obtain the data). But, it is highly unlikely that all offenders will commit crimes in the same destination zone using the same intermediate zone as an origin.

3. In the usual travel demand modeling conducted by transportation planners, the origins are assumed to be more accurate than the destinations. The origins are identified typically from census and other population enumerations whereas the destinations are estimated from surveys and employment databases. In the case of crime travel, however, the destinations are known with much greater accuracy since those locations are documented in police reports.

4. For example, transportation modelers often put in *centroid connectors*. These are pseudo-segments that connect a zone centroid with an arterial. It is possible to add pseudo-roads to the modeling network to force travel to follow a particular route. But, it does take a lot of editing to do this.

5. The modeling network was obtained from the Baltimore Metropolitan Council and, with their permission, is illustrated here.

6. As an example of the efficiency of a modeling network compared to a TIGER network, the network assignment routine was run in about 5 hours with the TIGER network for Baltimore City and Baltimore County, but was finished in about 50 minutes with the modeling network. See chapter 16 on network assignment for more information about the rules for network travel.

Chapter 13
Trip Generation

Background

In this chapter, the theory and mechanics of the trip generation stage will be explained. *Trip generation* is a model of the number of trips that originate and end in each zone for a given jurisdiction. Given a set of N destination zones and M origin zones (which include all the destination zones and, possibly, zones from adjacent jurisdictions), separate models are produced of the number of crimes originating and ending in each of these zones. That is, a separate model is produced of the number of crimes originating in each of the M origin zones, and another model is produced of the number of crimes ending in each of the N destination zones. The first is a *crime production* model while the second is a *crime attraction* model.

Two points should be emphasized. First, the models are predictive. That is, the result of the models are a prediction of both the number of crime trips originating in each zone and the number of crime tripss ending in each zone (i.e., crimes occurring in a zone). Because the models are a prediction, there is always error between the actual number and that predicted. As long as the error is not too large, the model can be a useful tool for both analyzing the correlates of crime as well as being useful for forecasting or for simulating policy interventions.

Second, because the number of crimes attracted to the study jurisdiction will usually be greater than the number of crimes predicted for the origin zones, due primarily to crime trips coming from outside the origin areas, it is necessary to balance the productions and attractions. This is done in two steps. One, an estimate of trips coming from outside the study area (external trips) is added to the predicted origins as an 'external zone'. Two, a statistical adjustment is done in order to ensure that the total number of origins equals the total number of destinations. This is called *balancing* and is essential as an input into the second stage of crime travel demand modeling - trip distribution.

In the following discussion, first, the logic behind trip generation modeling is presented, including the calibration of a model, the addition of external trips in making a model, and the balancing of predicted origins and predicted destinations. Second, the mechanics of conducting the trip generation model with *CrimeStat* is discussed and illustrated with data from Baltimore County.

Modeling Trip Generation

The process of modeling trip generation is fairly well developed, at least with respect to ordinary trips. It proceeds through a series of logical steps that make up the aggregate trip generation model.

Trip Purpose

Trip generation modeling starts with the reasons behind travel. At an individual level, people make trips for a reason - to go to work, to go shopping, to go to a medical appointment, to go for recreation, or, in the case of offenders, to commit a crime. These are called *trip purposes*. Since there are a very large number of trip purposes, usually these are categorized into a few major groupings. In the case of the usual travel demand forecasting, the distinctions are *home-to/from-work* (or home-based work trips), *home-to/from-non-work* (or home-based non-work trips, e.g., shopping), and a *non-home trip* where neither the origin nor the destination are at the traveler's residence location (non-home-based trips).

Since the model has aggregated trips to a zone, the trip purposes are collections of trips from each origin zone to each destination zone. Thus, each zone produces a certain number of home-work trips, home-non-work trips, and non-home trips and each zone attracts a certain number of home-work trips, home-non-work trips, and non-home trips. This is the usual distinction that most transportation modeling organizations make. The trip purposes are documented during a large travel survey that asks individuals to fill out travel diaries for one or two days of travel. In the travel diaries, detailed information about each trip is documented - time of day, destination of trip, purpose of trip, travel modes used in making the trips, accompanying passengers, route taken, and time to complete the trip.

Crime Trip Groupings

For crime trips, however, these distinctions are not very meaningful. There is very little information on how offenders make trips. One cannot just take a sample of offenders and ask them to complete a travel diary about how, when, and where the trip took place. With arrested offenders, it might be possible to produce such a diary, but both memory problems as well as legal concerns quickly make this an unreliable source of information. Therefore, as indicated in chapter 11, a decision has been made to reference all trips with respect to the residential home location. All crime trips are analyzed as *home-crime* trips.

However, other distinctions can be made. The most obvious is by type of crime. There are robbery trips, burglary trips, vehicle theft trips, and so forth. Similarly, distinctions can be made by travel time such as afternoon trips or evening trips. As mentioned in chapter 12, though, the sample size will decrease with greater distinctions. Logically, one can divide a sample into a very large number of important distinctions (e.g., afternoon burglary trips involving two or more offenders). However, this reduces the sample size and increases the error in estimation, particularly at the trip distribution and subsequent stages.

An important point that distinguishes the aggregate demand types of travel demand models, as is being implemented here, and the newer generation of activity-based trips is that there are no *linked trips* with the aggregate approach (FHWA, 2001a). If an offender first steals a car, then uses the car to rob a grocery store followed by a burglary, the

aggregate approach models this as three separate trips, rather than as a series of three linked crime trips (which the activity-based models do). This is a deficiency with the aggregate travel demand model. In order to make the aggregate models work, each trip is considered independent of any other trip. While this is not realistic behaviorally, since we know that many crimes are committed in sequence as part of a single journey (or tour), the zonal approach does limit the underlying logic of crime trips. Nevertheless, the aggregate approach can be very useful as long as it implemented consistently. With the current state of activity-based modeling, there is not yet any evidence that they produce more accurate predictions than the cruder, aggregate approach (FHWA, 2001a).

Correlates of Crime

Any trip has contextual correlates associated with it. It is well documented that the likelihood of making a trip (crime or otherwise) is not equal across areas of a metropolitan region. There are age correlates of travel, socioeconomic correlates of travel, and land use correlates of travel; the latter are usually associated with trip purposes (e.g., retail areas attract shopping trips).

The trip generation model being implemented in this version of *CrimeStat* is an aggregate model. Thus, the predictors are aggregate, rather than behavioral, in nature, as discussed in chapter 11. They are correlates of trips, not necessarily the *reasons* for the trips. For example, typically population is the best predictor of trips. Zones with many persons will produce, on average, more crime trips than zones with fewer persons. The observation is not a reason, but is simply a by-product of the size of the zone. Similarly, low-income zones will tend to produce, on average, more crime trips than wealthier zones; again, this is not a reason, but a correlate of the characteristics that might contribute to individual likelihoods for committing crimes.

As mentioned in chapter 12, there are a number of different variables that could be used for prediction, although population (or a proxy for population, such as households), income or poverty, and land use variables would be the most common (NCHRP, 1998).

Theoretical Relevance of the Variables

In general, the variables that are selected should be empirically stable and theoretically meaningful. That is, they should be stable variables that do not change dramatically from year to year. They should be reliably measured so that an analyst can depend on their values. Finally, they should be meaningful in some ways. That is, they should be plausible enough that both crime analysts and researchers and informed outsiders should agree that the relationship is plausible. The variables either should have been demonstrated to be predictors in earlier research or else to be so correlated with known factors as to be considered meaningful proxies.

Spurious correlates

On the other hand, if a variable is either a correlate of a known predictor or idiosyncratic, then it is liable not be believed. For example, the number of taxis usually correlates with the amount of employment since taxis tend to ply commercial areas for their trade. Adding the number of taxis in a predictive model is liable to produce significant statistical effects in predicting crime destinations. However, few persons are going to believe that this is a real factor since it is understood to be a correlate of a more structural variable.

Idiosyncratic variables are those that appear in unique situations. For example, in some cities, adjacency to a freeway is a correlate of crime origins (e.g., in Baltimore County where low income populations live) whereas in other cities, it is a correlate of crime destinations (e.g., in Houston where there are frontage roads with major commercial strips that attract crimes). The variables may be real predictors. However, the analyst or researcher will have difficulty persuading others to believe in the model, at least until the results can be replicated.

In other words, what is required for the model is a set of reasonable correlates of crime trips that would be plausible and stable over time. It is an ecological model, not a behavioral one.

Social Disorganization Variables

There is a very large literature on the predictors of crime, typically following from the social disorganization literature (for example, Park and Burgess, 1924; Thrasher, 1927; Shaw and McKay, 1942; Newman, 1972; Ehrlich, 1975; Cohen and Felson, 1979; Wilson and Kelling, 1982; Stack, 1984; Messner, 1986; Chiricos, 1987; Kohfeld and Sprague, 1988; Bursik and Grasmick, 1993; Hagan, J. & R. Peterson, 1994; Fowles and Merva, 1996; Bowers and Hirschfield, 1999 among many other studies). Much of this literature identifies correlates that are associated with crime incidents. Among the factors that have been associated with crime and delinquency are poverty, low income households, overcrowding, substandard housing, low education levels, single-parent households, high unemployment, minority and immigrant populations.

Multicolinearity among the independent variables

There are two statistical problems associated with using these variables as predictors. The first is the high degree of overlap between the variables. Zones that have high poverty levels typically also have low household income levels, higher population densities, substandard housing, a high percentage of renters, and higher proportion of minority and immigrant populations. In a regression model, this overlap causes a condition known as *multicolinearity*. Essentially, the independent variables correlate so highly among themselves that they produce ambiguous, and sometimes strange, results in a regression model. For example, if two independent variables are highly correlated, frequently one will have a positive coefficient with the dependent variable while the other

will have a negative coefficient; conversely, they sometimes can cancel each other out. Thus, in spite of the correlates with crime levels, in a model it is usually best to eliminate *co-linear* variables. The result is that simple variables usually end up being the most straightforward to use (population, median household income) with many of the subtle, but theoretically relevant, variables typically dropping out of the equation.

Failure to distinguish origins from destinations

Second, in much of this literature, however, there is not a clear distinction between origin predictors and destination predictors. That is, in most cases, the correlates of crimes were identified but it is often unclear whether these correlates are associated with the neighborhoods of the offenders (origins) or the locations where the crimes occur (destinations). This can result in a set of vague correlates without clear direction about whether the variables are associated with producing or attracting conditions. In fact, in much of the early literature on social disorganization, it was implicitly assumed that crimes are produced in the neighborhoods where the offenders lived, a linkage that is increasingly becoming disconnected. For modeling crime trips, however, it is essential that the predictors of origins be kept separate from the predictors of destinations.

Accuracy and Reliability

A trip generation model should be accurate and reliable. *Accuracy* means that the model should replicate as closely as possible the actual number of trips originating or ending in zones and that there should be no bias (which is a systematic under- or over-estimating of trips). *Reliability* means that the amount of error is minimized.

These criteria have two implications which are somewhat at odds. First, we have to choose models that replicate as closely as possible the number of trips originating or ending in a zone. In general, this would be a model that had the highest overall predictability. But, second, we have to choose models that minimize total prediction errors. This allows a model to replicate the number of trips for as many zones as possible. The two criteria are somewhat contradictory because crime trips are highly skewed. That is, a handful of zones will have a lot of crimes originating or ending in them while many zones will have few or no crimes. The zones with the most crimes will have a disproportionate impact on the final model. Thus, a model that obtains as high a prediction as possible (i.e., highest log-likelihood or R^2) may actually only predict accurately for a few zones and may be very wrong for the majority.

The strategy, therefore, is to obtain a model that balances high predictability but by keeping the total prediction error low.

Count Model

Another element of the model is that the trip generation model is for *counts* (or volumes), not for rates. The model predicts the number of crimes originating in each origin zone and the number of crimes occurring in each destination zone. The model could be

constructed to predict rates, but normally it is not done. For most travel demand modeling, as mentioned in chapter 11, the model predicts the *number* of trips originating or ending in a zone. Thus, there is a *crime production* model that predicts the number of crimes originating in each zone and a *crime attraction* model that predicts the number crimes

Approaches Towards Trip Generation Modeling

Trip Tables

There are two classic approaches to trip generation modeling. The first uses a *trip table* (sometimes called a cross-classification table or a category analysis). A trip table is a cross-classification matrix. Several predictive variables are divided into categories (e.g., three level of household income; four levels of vehicle ownership; three levels of population density) and a mean number of trips is estimated, usually from a survey. For example, a survey of household income might show the relationship between household income and the number of trips taken by individuals of the households. Based on a sample, estimates of the *average number of trips per person* can be obtained for each income level (e.g., 3.4 trips per day for persons from low income households; 4.5 trips per day for persons from median income households; 6.7 trips per day for persons from high income households). These variables are further subdivided into two-way or three-way cross-tabulation tables (e.g., low income and medium vehicle ownership; low income and high vehicle ownership). Table 13.1 illustrates a *possible* trip table model involving two variables. In practice, three or four variables are used.

The main reason that trip tables are used in a trip generation model is because of the non-linear nature of trips. Predictive variables are usually not linear in their effects on the number of trips. Thus, unless a sophisticated non-linear model is used, sizeable error can be introduced in a prediction. It is usually safer to use a trip table approach (Ortuzar and Willumsen, 2001). There are some major handbooks on the topic (Henscher and Button, 2002; ITE, 2003). In fact, the Institute of Transportation Engineers publishes a large handbook that gives extensive trip production and trip attraction tables by detailed land uses (ITE, 2003). These tables are often used in formal environmental review processes for site analysis and are frequently accepted by courts in litigation. They are not without their problems, however, and there have been numerous critiques of the tables (Shoup, 2002; NCHRP, 1998). They also cannot be used in a travel demand model and will produce erroneous results.

The problem for crime analysis, however, is that it is impossible to obtain these data. One cannot ask a sample of offenders how many crimes they undertake each day in order to estimate the mean expectations for a table. Thus, one has to adopt a more indirect approach in modeling crime productions and attractions.

A second problem with the trip table approach is its use with zonal data. While it could be applied to zonal data (e.g., using median household income and average vehicle ownership in table 13.1 instead of individual household income and vehicle ownership),

such an approach requires interpretation and some degree of arbitrariness. For example, how does one subdivide median household income? One person might interpret it slightly

Table 13.1

Illustration of Possible Trip Table Approach to Trip Generation
Average Trips Per Adult, Age 16+

		Household income		
		Low	Medium	High
Vehicle Ownership	0-1	3.2	4.6	6.7
	2+	5.4	7.8	8.1

differently than another; unlike simple numerical counts (e.g., 0 vehicle ownership; 1 vehicle ownership; 2 vehicle ownership), there is too much variability in categorizing variables at the zonal level.[1]

OLS Regression Modeling

The second approach is to use a *regression* framework. In this approach, the number of crimes either originating or ending in each zone are estimated from zone characteristics using a regression model. This can be written in an equation:

$$Y_i = f(X_1, X_2, X_3, \ldots X_k) + \epsilon \qquad (13.1)$$

The mean number of crimes, Y_i (either originating or ending in zone I), is a function of a number of independent variables, $X_1, X_2, X_3, \ldots X_k$ for these zones; there are *k* independent variables, including any constants. There is also an error term which represents the discrepancy between the actual observation and what the model predicts. This is sometimes called *residual error* since it is the difference between the observed and predicted values ($O_i - Y_i$). The function is unspecified and can be non-linear.[2]

The traditional approach to regression modeling assumed that the independent variable are linear in their effect on the dependent variable. Thus,

$$Y_i = \alpha + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 \ldots + \beta_k X_k + \epsilon \qquad (13.2)$$

In this model, there are K independent variables and one constant term (α) that needs to be estimated. For each zone, I, each of the independent variables has a weight associated with

it (the coefficients, β). The product of the value of the independent variable times its weight represents its *effect*. The individual effects of each of the K independent variables are summed to produce an overall estimate of the dependent variable, Y.

The method for estimating this equation usually minimizes the sum of the squares of the residual errors. Hence, the procedure is called *Ordinary Least Squares* (or OLS). If the equation is correctly specified (i.e., all relevant variables are included), the error term, ϵ, will be normally distributed with a mean of 0 and a constant variance, σ^2.

Problems with OLS Regression Modeling

However, there are a number of major problems associated with OLS regression modeling.

Skewness of crime events

First, crime events are extremely statistically skewed. Some locations have a much higher likelihood of a crime event (either an origin or a destination) than others. Figure 13.1 below shows the number of crimes from 1993 to 1997 in Baltimore County that occurred at each location. That is, the graph shows the number of incidents that occurred at every location, plotted in decreasing order of frequency. Thus, there were 7,965 locations where only one crime occurred between 1993 and 1997. There were 2,878 locations where two crimes occurred in that period. There were 1,138 locations where three crimes occurred in that period. At the other end of the spectrum, there were 332 locations that had 10 or more crimes during the period and there were 97 locations that had 30 or more crimes occur. If we add to this the very large number of locations that had no crimes occur, the unequal likelihoods of crime by location is even more dramatic. In other words, the data are highly skewed with respect to the frequency of crimes. Most locations either had no crimes occur or very few, while a few locations had many crimes occur.

Aggregating crimes into zones tends to reduce *some* of the skewness. For example, grouping the crimes by origin traffic analysis zone (TAZ) reduced it a little bit. Nineteen of the 525 origin zones in Baltimore County and Baltimore City did not have any crimes occur in them while 15 zones had only one crime occur. Six zones had two crimes originate from them while 8 zones had three crimes originate from them. At the other end, 1 zone had 738 crimes originate from it and another zone had 533 originate from it. Of the 525 origin zones, 155 had 100 or more crime events. Similar results are found for the destination zones. Figure 13.2 graphs the distribution of origins and destinations by TAZ's in bins of 50 incidents each.

Skewness in the dependent variable usually makes the final model biased and unreliable. Particularly if the skewness is positive (i.e., a handful of cases have very large values), the resulting regression coefficients will reflect the cases with the highest values rather than represent all the cases with approximately equal weights. These so-called 'outliers' can overwhelm a regression equation. In an extreme case, a very large outlier may totally determine the model. For example, an experiment with 100 cases was created with a

Figure 13.1:

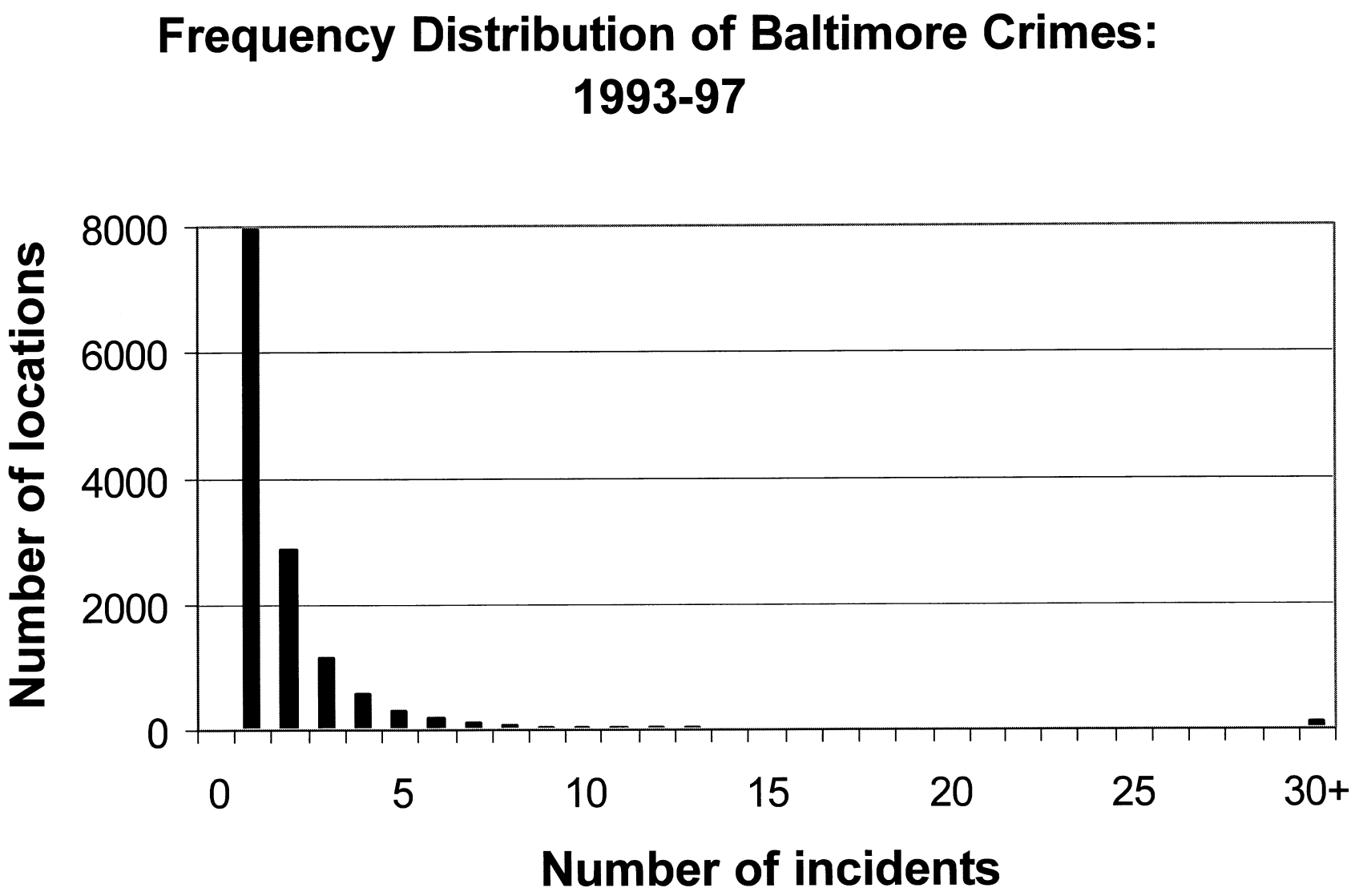
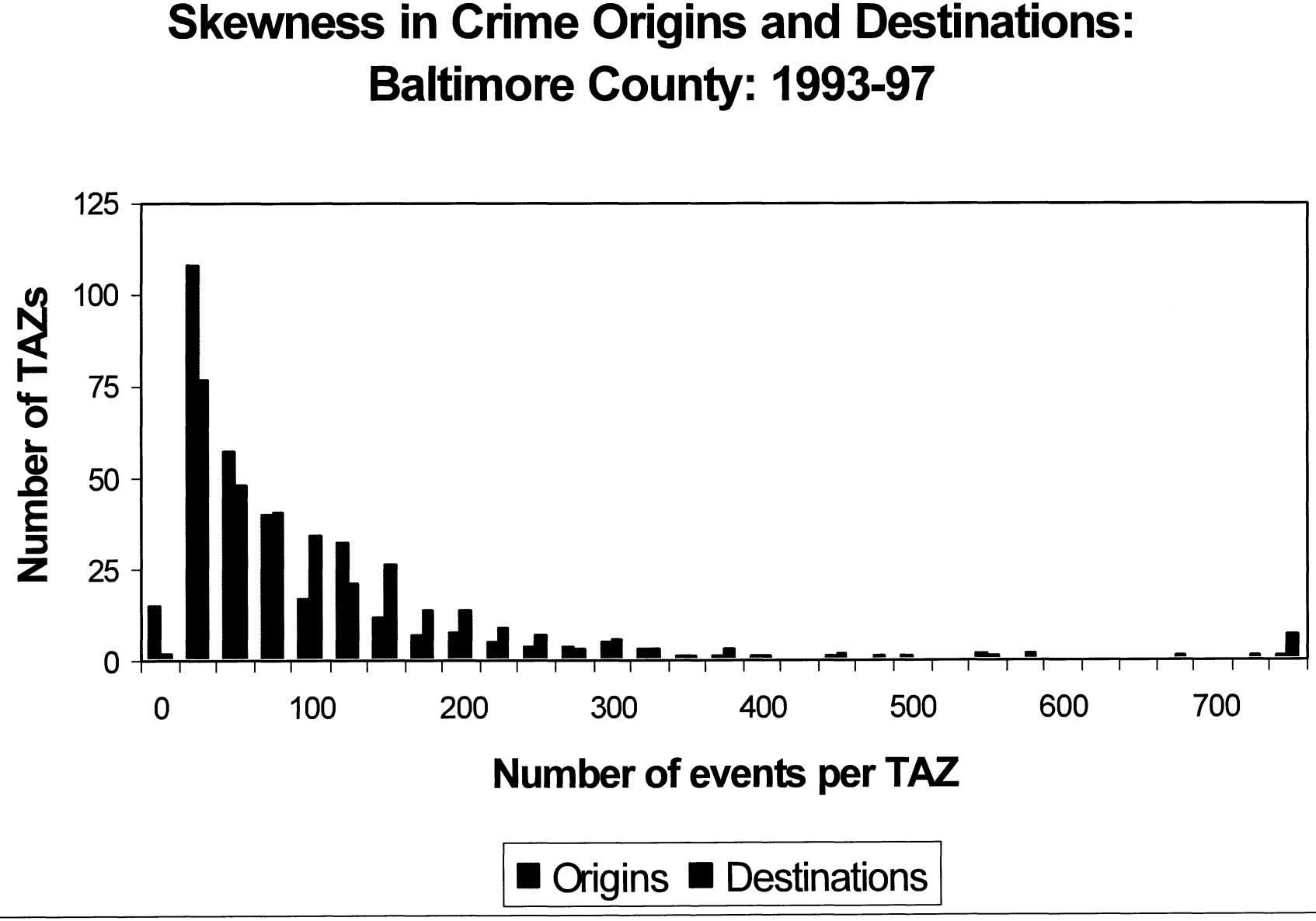


Figure 13.2:



progressing dependent variable and a **random** independent variable (i.e., the independent variable had its value selected randomly). The dependent variable progressed from 1 to 100. For the first 99 cases, the independent variable took values from 0.12 to 9.9, randomly assigned. The correlation between these two variables for the first 49 cases was 0.04. However, for the 100th case, the independent variable was given a value of 100. The correlation between the two variables now shot up to 0.17. Even though the F-test for this was not significant, it represented a sizeable jump. Replacing one other independent value with a 50 caused the correlation to jump to 0.23, which was statistically significant. In other words, two outliers caused a random series to appear significant!

Skewness makes prediction difficult. The OLS model assumes that each independent variable contributes to the dependent variable at an arithmetic rate; there is a constant slope such that a one unit change in the independent variable is associated with a constant change in the dependent variable. With skewness, on the other hand, such a relationship will not be found. Large changes in the independent variable will be necessary to produce small changes in the dependent variable, but the effect is not constant. In other words, the OLS model typically cannot explain the non-linear changes in the dependent variable.[3]

Negative predictions

A second problem with OLS is that it can have negative predictions. With a count variable, such as the number of crimes originating or ending in a zone, the minimum number is zero. That is, the count variable is always *positive*, being bounded by 0 on the lower limit and some large number on the upper limit. The OLS model, on the other hand, can produce negative predicted values since it is additive in the independent variables. This clearly is illogical and is a major problem with data that are very skewed. If the most common value is close to zero, it is very possible for an OLS model to predict a negative count.

Non-consistent summation

A third problem with the OLS model is that the sum of the input values do not necessarily equal the sum of the predicted values. Since the estimate of the constant and coefficients is obtained by minimizing the sum of the squared residual errors, there is no balancing mechanism to require that they add up to the same as the input values. For a trip generation model in which the number of predicted origins has to equal the number of predicted destinations (after adding in the number of predicted external trips), this can be a big problem. In calibrating the model, adjustments can be made to the constant term to force the sum of the predicted values to be equal to the sum of the input values. But in applying that constant and coefficients to another data set, there is no guarantee that the consistency of summation will hold. In other words, the OLS method cannot guarantee a consistent set of predicted values.

Non-linear effects

A fourth problem with the OLS model is that it assumes the independent variables are linear in their effect. If the dependent variable was normal or relatively balanced, then a linear model might be appropriate. But, when the dependent variable is highly skewed, as is seen with these data, typically the additive effects of each component cannot usually account for the non-linearity. Independent variables have to be transformed to account for the non-linearity and the result is often a complex equation with non-intuitive relationships.[4] It is far better to use a non-linear model for a highly skewed dependent variable.

Greater residual errors

The final problem with an OLS model and a skewed dependent variable is that the model tends to over- or under-predict the correct values, but rarely comes up with the correct estimate. With skewed data, typically an OLS equation produces non-constant residual errors. That is, one of the major assumptions of the OLS model is that all relevant variables have been included. If that is the case, then the errors in prediction (the residual errors - the difference between the observed and predicted values) should be uncorrelated with the predicted value of the dependent variable. Violation of this condition is called *heteroscedasticity* because it indicates that the residual variance is not constant. The most common type is an increase in the residual errors with higher values of the predicted dependent variable. That is, the residual errors are greater at the higher values of the predicted dependent variable than at lower values (Draper and Smith, 1981, 147).

A highly skewed distribution tends to encourage this. Because the least squares procedure minimizes the sum of the squared residuals, the regression line balances the lower residuals with the higher residuals. The result is a regression line that neither fits the low values or the high values. For example, motor vehicle crashes tend to concentrate at a few locations (crash hot spots). In estimating the relationship between traffic volume and crashes, the hot spots tend to unduly influence the regression line. The result is a line that neither fits the number of expected crashes at most locations (which is low) nor the number of expected crashes at the hot spot locations (which are high). The line ends up over-estimating the number of crashes for most locations and under-estimating the number of crashes at the hot spot locations.

Poisson Regression Modeling

Poisson regression is a non-linear modeling method that overcomes some of the problems of OLS regression. It is particularly suited to count data (Cameron and Trivedi, 1998). In the model, the number of events is modeled as a Poisson random variable with a probability of occurrence being

$$\text{Prob}\,(Y_i) = \frac{e^{-\lambda}\,\lambda^{Y_i}}{Y_i!} \qquad (13.3)$$

where Y_i is the count for one group or class, i, λ is the mean count over all groups, and **e** is the base of the natural logarithm. The distribution has a single parameter, λ, which is both the mean and the variance of the function.

The "law of rare events" assumes that the total number of events will approximate a Poisson distribution *if* an event occurs in any of a large number of trials but the probability of occurrence in any given trial is small (Cameron and Trivedi, 1998). Thus, the Poisson distribution is very appropriate for the analysis of rare events such as crime incidents (or motor vehicle crashes or rare diseases or any other rare event). The Poisson model is not particularly good if the probability of an event is more balanced; for that, the normal distribution is a better model as the sampling distribution will approximate normality with increasing sample size. Figure 13.3 illustrates the Poisson distribution for different expected means.

The mean can, in turn, be modeled as a function of some other variables (the independent variables). Given a set of observations on dependent variables, X_{ki} (X_1, X_2, X_3,...,X_K), the *conditional mean* of Y_i can be specified as an exponential function of the X's:

$$E(Y_i / X_{ki}) = \lambda_i = e^{Xki\,\beta} \tag{13.4}$$

where X_{ki} is a set of independent variables, β is a set of coefficients, and **e** is the base of the natural logarithm.. Now, the conditional mean (the mean controlling for the effects of the independent variables) is non-linear. Equation 13.4 is sometimes written as

$$Ln\,(\lambda_i) = X_{ki}\,\beta \tag{13.5}$$

and is known as the *loglinear* model. In more familiar notation, this is

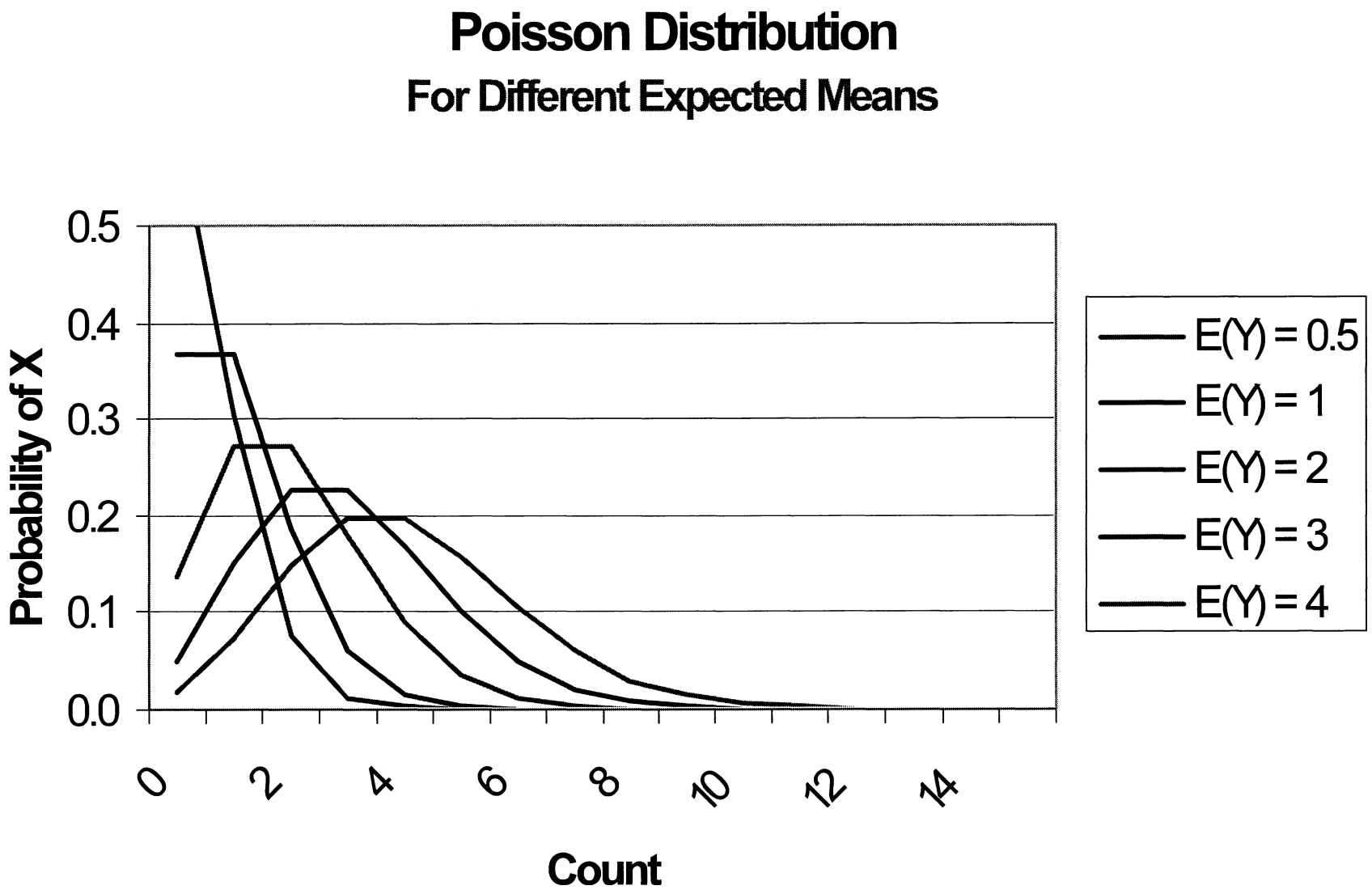
$$Ln\,(\lambda_i) = \alpha + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 X_{3i} + \ldots\ldots\ldots + \beta_k X_{ki} \tag{13.6}$$

That is, the natural log of the mean is a function of K random variables.

Note, that in this formulation, there is not a random error term. The data are assumed to reflect the Poisson model. There can be "residual errors", but these are assumed to reflect an incomplete specification (i.e., not including all the relevant variables. Also, since the variance equals the mean, it is expected that the residual errors should increase with the conditional mean. That is, there is inherent heteroscedasticity (Cameron and Trivedi, 1998). This is very different than an OLS where the residual errors are expected to be constant.

The model is estimated using a maximum likelihood procedure, typically the Newton-Raphson method. In Appendix C, Luc Anselin presents a more formal treatment of both the OLS and Poisson regression models, including the methods by which they are estimated.

Figure 13.3:



Advantages of the Poisson Regression Model

The Poisson model overcomes some of the problems of the OLS model. First, the Poisson model has a minimum value of 0. It will not predict negative values. This makes it ideal for a distribution in which the mean or the most typical value is close to 0. Second, the Poisson is a fundamentally skewed model; that is, it is non-linear with a long 'right tail'. Again, this model is appropriate for counts of rare events, such as crime incidents.

Third, because the Poisson model is estimated by a maximum likelihood method, the estimates are adapted to the actual data. In practice, this means that the sum of the predicted values is virtually identical to the sum of the input values, with the exception of very slight rounding off error. In the subsequent balancing of the predicted origins and the predicted destinations, this leads to a more stable estimate since the only difference between the predicted origins and predicted destinations is the number of trips that come from outside the study area (external trips). Since the external trips are added to the predicted origins, the balancing operation is less prone to adjustment error.

Fourth, compared to the OLS model, the Poisson model generally gives a better estimate of the number of crimes for each zone. The problem of over- or under-estimating the number of incidents for most zones with the OLS model is usually lessened with the Poisson, at least for crime and other rarer events. When the residual errors are calculated, generally the Poisson has a lower total error than the OLS.

In short, the Poisson model has some desirable statistical properties that make it very useful for predicting crime incidents (origins or destinations).

Problems with the Poisson Regression Model

On the other hand, the Poisson model is not perfect. The primary problem is that count data are usually *over-dispersed*.

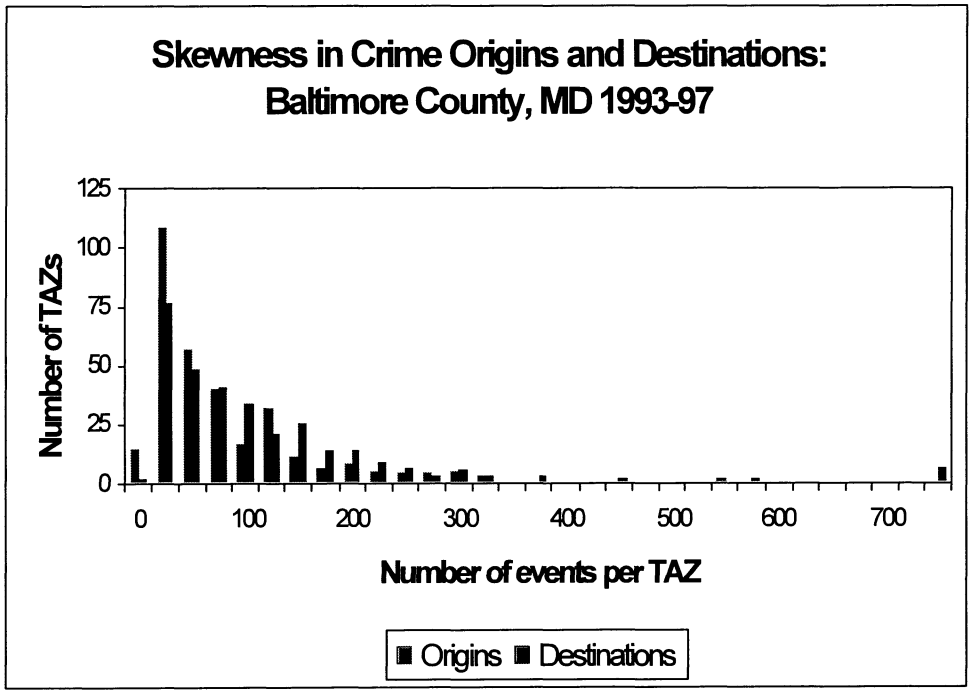
Over-dispersion in the residual errors

In the Poisson distribution, the mean equals the variance. In a Poisson regression model, the mathematical function, therefore, equates the conditional mean (the mean controlling for all the predictor variables) with the conditional variance. However, most real data are over-dispersed; the variance is generally greater than the mean. Figure 13.4 shows the distribution of Baltimore County and Baltimore City crime origins and Baltimore County crime destinations by TAZ (repeat of figure 13.2) and also indicates the variance-to-mean ratio of each variable. For the origin distribution, the ratio of the variance to the mean is 14.7; that is, the variance is 14.7 times that of the mean! For the destination distribution, the ratio is 401.5!

In other words, the variance is many times greater than the mean. Most real-world count data are similar to this; the variance will usually be a lot greater than the mean. What this means in practice is that the residual errors - the difference between the observed

Figure 13.4:

Over-dispersion



Origins:

Mean = 75.8
Variance = 7848.8
Ratio of variance to mean = 14.7

Destinations:

Mean = 129.1
Variance = 51,849.1
Ratio of variance to mean = 401.5

and predicted values for each zone, will be greater than what is expected. The Poisson model calculates a standard error as if the variance equals the mean. Thus, the standard error will be underestimated using a Poisson model and, therefore, the significance tests (the coefficient divided by the standard error) will be greater than it really should be. This would have the effect of identifying variables as being more statistically significant in a model than what they actually should be. In other words, in a Poisson multiple regression model, we would end up selecting variables that really should not be selected because we think they are statistically significant when, in fact, they are not.

Another problem with the Poisson, which is true for most of the common regression methods, is the lack of a spatial predictor component. As mentioned in chapter 12, in the crime travel demand model, spatial interaction is handled during the second stage of the model - trip distribution. Thus, any errors introduced in the first stage - trip generation, are usually compensated for during the second. Nevertheless, the inclusion of a spatial component in a regression model would generally improve the prediction. For this version of *CrimeStat*, non-spatial methods are used for the first stage.

Dispersion Correction Parameter

There are a number of methods for correcting the over-dispersion in a count model. Most of them involve modifying the assumption of the conditional variance equal to the conditional mean. For example, the negative binomial model assumes a Poisson mean but a gamma-distributed variance term (Cameron and Trivedi, 1998, 62-63; Venables and Ripley, 1997, 242-245). That is, there is an unobserved variable that affects the distribution of the count. The model is then of a Poisson mean but with a 'longer tail' variance function. As another example, the zero-inflated Poisson model assumes a Poisson function combined with a degenerate function with a probability of 1 for zero counts (Hall, 2000). Such mixed function models are a current topic of research. In general, though, they are complicated and require estimating several parameters.

There is a simple correction for over-dispersion that usually works (Cameron and Trivedi, 1998, 63-65). The model proceeds in two steps. In the first, the Poisson model is fitted to the data and the degree of over- (or under-) dispersion is estimated. The dispersion parameter is defined as:

$$\Phi = \frac{1}{N - K - 1} \sum \left\{ \frac{(Y_i - P_i)^2}{P_i} \right\} \qquad (13.7)$$

where N is the sample size, K is the number of independent variables, Y_i is the observed number of events that occur in zone I, and P_i is the predicted number of events for zone I. The test is similar to an average chi-square in that it takes the square of the residuals (Y_i - P_i) and divides it by the predicted values, and then averages it by the degrees of freedom. The dispersion parameter is a standardized number. A value greater than 1.0 indicates over-dispersion while a value of less than 1 indicates under-dispersion (which is rare, though possible). A value of 1.0 indicates *equidispersion* (or the variance equals the mean).

In the second step, the Poisson standard error is multiplied by the square root of the dispersion parameter to produce an *adjusted standard error*:

$$SE_{adj} = SE * SQRT[\Phi] \quad (13.8)$$

The new standard error is then used in the t-test to produce an adjusted t-value. This adjustment is found in most Poisson regression packages using a Generalized Linear Model (GLM) approach, such as SAS (McCullagh and Nelder, 1989, 200). Cameron and Trivedi (1998) have shown that this adjustment produces results that are virtually identical to that of the negative binomial, but involving fewer assumptions.

Diagnostic Tests

There are a number of diagnostics tests that are used in a regression framework, whether OLS, Poisson, or other methods.

Skewness Tests

First, there are tests of skewness in the dependent variable. As mentioned above, the OLS model cannot be applied to data that are highly skewed. If they are skewed, a non-linear model, such as the Poisson, must be used. Therefore, it is essential to evaluate the degree of skewness.

A commonly used measure of skewness is the *g* statistic (Microsoft, 2000):

$$\text{Skewness (g)} = \frac{n}{(n-1) * (n-2)} \sum_{I=1}^{n} [(X_i - \text{MeanX})/s]^3 \quad (13.9)$$

where n is the sample size, X_i is observation I, MeanX is the mean of X, and s is the sample standard deviation (corrected for degrees of freedom):

$$s = SQRT\left[\sum_{I=1}^{n} \frac{(X_i - X)^2}{(n-1)} \right] \quad (13.10)$$

The standard error of skewness (SES) can be approximated by (Tabachnick and Fidell, 1996):

$$SES = SQRT\left[\frac{6}{n} \right] \quad (13.11)$$

An approximate Z-test can be obtained from:

$$Z(g) = \frac{g}{SES} \qquad (13.12)$$

Thus, if Z is greater than +1.96 or smaller than -1.96, then the skewness is significant at the $p \leq .05$ level.

As an example, for the data on the origins of crimes by TAZ in Baltimore County:

$\bar{X} = 75.108$
$s = 96.017$
$n = 325$
$\sum_{I=1}^{n} [(X_i - MeanX)/s]^3 = 898.391$

Therefore,

$$g = \frac{325}{324*323} * 898.391 = 2.79$$

$$SES = SQRT \left[\frac{6}{325} \right] = 0.136$$

$Z(g) = 20.51$

The Z of the g value shows the data are highly skewed as we, of course, knew.

Likelihood Ratio Test

Second, there are tests of the overall model. In a maximum likelihood framework, the first test is of the *log-likelihood* function. A *likelihood* function is the joint density of all the observations, given a value for the parameters, β, and the variance, σ^2. The log-likelihood is the natural log of this product, or the sum of the logs of the individual densities. For the OLS model, the log-likelihood is:

$$L = - (N/2) \ln(2\pi) - (N/2) \ln(\sigma^2) - (\tfrac{1}{2} \sigma^2) - (\tfrac{1}{2}) \left[\frac{(Y_i - X_{ki} \beta_k)^2}{\sigma^2} \right] \qquad (13.13)$$

where N is the sample size, σ^2 is the variance, Y_i is the observed number of events for zone I, and $X_{ki}\beta_k$ is a series of K independent predictors multiplied by their coefficients.

In the Poisson model, the log-likelihood is:

$$L = \Sigma [-\lambda_i + Y_i X_{ki}\beta_k - \ln Y_i!] \quad (13.14)$$

where λ_i is the conditional mean for zone I, Y_i is the observed number of events for zone ii, and $Y_i X_{ki}\beta_k$ is a cross-product of the observed events times the K independent predictors multiplied by their coefficients. As mentioned above, Luc Anselin provides a more detailed discussion of these functions in Appendix C.

Since the maximum likelihood method achieves the model with the highest log-likelihood, the log-likelihood is a negative number. Even though the model with the highest log-likelihood is considered 'best', it is not an intuitive number. Consequently, the *Likelihood Ratio* compares the log-likelihood of the regression model with the log-likelihood that would be obtained if only the mean number of counts was taken. This latter log-likelihood is:

$$L_R = -N (MeanY) +[\ln(MeanY) (\Sigma Y_i)] - \Sigma \ln Y_i! \quad (13.15)$$

The Likelihood Ratio test is:

$$LR = 2(L - L_R) \quad (13.16)$$

where L is the model log-likelihood and L_R is the log-likelihood of the mean count. The Likelihood Ratio is twice the difference between log-likelihood values of the regression and mean models respectively. It follows a $\chi 2$ distribution with K degrees of freedom (where K is the number of independent variables).[5]

Adjusted likelihood ratio

The Likelihood Ratio is a more intuitive index since it is a chi-square test. However, it is prone to the problem of all regression methods of over-fitting - the more independent variables are added to the model, the higher is the Likelihood Ratio. Consequently, there are several methods that adjust for the number of parameters fit. One is the Akaike Information Criterion (AIC) which is defined as:

$$AIC = -2L + 2 (K+1) \quad (13.17)$$

where L is the log-likelihood and K is the number of independent variables. A second one is the Schwartz Criterion (SC), which is defined as:

$$SC = 2L+[(K+1)\ln(N)] \quad (13.18)$$

These two measures adjust the log-likelihood for degrees of freedom, and flip the sign around. The model with the highest AIC or SC values are 'best'.

R-square Test

The most familiar test of an overall model is the R-square (or R^2) test. This is the percent of the total variance of the dependent variable accounted for by the model. More formally, it is defined as:

$$R^2 = 1 - \frac{\Sigma (Y_i - P_i)^2}{\Sigma (Y_i - MeanY)^2} \qquad (13.19)$$

where Y_i is the observed number of events for a zone, I, P_i is the predicted number of events given a set of K independent variables, and MeanY is the mean number of events across zones. The R-square value is a number from 0 to 1; 0 indicates no predictability while 1 indicates perfect predictability.

For an OLS model, R-square is a very consistent estimate. It increases in a linear manner with predictability and is, therefore, a good indicator of how effective one model is compared to another. As with all diagnostic tests, the value of the R-square increases with more independent variables. Consequently, R-square is usually adjusted for degrees of freedom:

$$R_a^2 = 1 - \frac{[\Sigma (Y_i - P_i)^2] / (N-K+1)}{\Sigma (Y_i - MeanY)^2 / (N - 1)} \qquad (13.20)$$

where N is the sample size and K is the number of independent variables.

R-square for Poisson model

With the Poisson model, however, the R-square value (whether adjusted or not) is not necessarily a good measure of overall fit. While the Poisson R-square varies from 0 to 1, similar to the OLS, it is not monotonic. That is, the addition of a new variable to an equation often has unpredictable effects; sometimes it will increase substantially and sometimes it will increase only a little independent of how strong is a variable's association with the dependent variable (Miaou, 1996). This inconsistency comes from the decomposition of the total sum of squares:

$$\Sigma (Y_i - MeanY)^2 = \Sigma(Y_i - P_i)^2 + \Sigma(P_i - MeanY)^2 + 2\Sigma(Y_i - P_i)(P_i - MeanY) \qquad (13.21)$$

The first term in the equation is the residual sum of squares (or error term) while the second term is the explained sum of squares. In an OLS model, the third term is zero if an intercept is included (Cameron and Trivedi, 1998, 153). Hence, the total sum of squares is broken into two parts - that which is explained and that which is unexplained. However, for the Poisson

and other non-linear regression methods, the last term is not zero. Consequently, a test that compares the explained sum of squares to the total sum of squares will not produce consistent results.

Consequently, alternative R-square measures are sometimes used. One of these is *Deviance R-square.* It is defined as:

$$R_D^2 = 1 - \frac{\Sigma[Y_i * \mathrm{Ln}\{P_i / \mathrm{Mean}Y\} - (Y_i - P_i)]}{\Sigma [Y_i * \mathrm{Ln}\{Y_i / \mathrm{Mean}Y} \quad (13.22)$$

where Y_i is the observed number of events for each zone, I, P_i is the predicted number of events for each zone based on K independent predictors, and MeanY is the mean number of events across all zones.

The Deviance R-square measures the reduction in the Likelihood Ratio due to the inclusion of predictor variables. It produces a slightly different R-square, one that is typically higher than the traditional R-square. Whereas the traditional one might not show a large increase upon the introduction of an independent variable, the Deviance R-square often does show the increase.

Nevertheless, it has problems, too. Miaou (1996) argues that there is not a single R-square index that is perfectly consistent and suggests that users need to use multiple ones. There are other R-square values that have been proposed, but these two are sufficient for now. In short, a user must look at both as an indicator of how good is a model compared to another model.

Dispersion Parameter

Finally, in the Poisson model only, the dispersion parameter indicates the extent to which the variance is different from the mean. This was defined in equation 13.7 above.

Coefficients, Standard Errors, and Significance Tests

The second type of diagnostic tests are those for the individual predictors in the model. In both the OLS and Poisson models, there are three tests:

1. The coefficient. This indicates the change in the dependent variable associated with the change in the independent variable. In the case of the OLS, it is a linear term (i.e., the value of the dependent variable is multiplied by the coefficient) while in the Poisson model, it has to be converted by raising the product to an exponential term (i.e., $e^{\beta x}$).

2. The standard error. Each estimated coefficient in a model accounts for some of the variance in the dependent variable. This variance is the contribution of

the particular independent variable to the variance of the dependent variable. The square root of that variance is the *standard error*.

3. The significance level. The ratio of the coefficient to the standard error produces a significance test of the coefficient. In the OLS model, it is a t-test with N-K-1 degrees of freedom whereas in the Poisson model it is an asymptotic t-test, which is effectively a Z-test. The appropriate tables (t-test or standard normal) produce approximate probability levels of a Type I error (the likelihood of falsely rejecting a true null hypothesis of no relationship).

Testing for Multicolinearity

One of the major problems with any regression model, whether OLS or Poisson, is multicolinearity among the independent variables. In theory, each independent variable should be statistically independent of the other independent variables. Thus, the amount of variance for the dependent variable that is accounted for by each independent variable should be a unique contribution. In practice, however, it is rare to obtain completely independent predictive variables. More likely, two or more of the independent variables will be correlated. The effect is that the estimated standard error of a predictor variable is no longer unique since it shares some of the variance with other independent variables. The greater the *communality* of the variances, the more ambiguous the predicted effects. If two variables are highly correlated, it is not clear what contribution each makes towards predicting the dependent variable. In effect, multicolinearity means that variables are measuring the same effect.

Multicolinearity among the independent variables can produce very strange effects in a regression model. Among these effects are: 1) If two independent variables are highly correlated, but one is more correlated with the dependent variable than the other, the stronger one will usually have a correct sign while the weaker one will sometimes get flipped around (e.g., from positive to negative, or the reverse); 2) Two variables can cancel each other out; each coefficient is significant when it alone is included in a model but neither are significant when they are together; 3) One independent variable can inhibit the effect of another correlated independent variable so that the second variable is not significant when combined with the first one; and 4) If two independent variables are virtually perfectly correlated, many regression routines break down because the matrix cannot be inverted.

All these effects indicate that there is non-independence among the independent variables. Aside from producing confusing coefficients, multicolinearity can overstate the amount of prediction in a model. Since every independent variable accounts for some of the variance of the dependent variable, with multicolinearity, the overall model will appear to improve when it probably hasn't.

Tolerance test

A user has to be aware of the problem of multicolinearity and seek to minimize it. The simplest solution is to drop variables that are co-linear with other independent variables

already in the equation. A relatively simple test for assessing this is called *tolerance.* Tolerance is defined as *lack of predictability* of each independent variable by the other independent variables, or:

$$\text{Tol} = 1 - (R_{ijk..l})^2 \qquad (13.23)$$

where $(R_{ijk..l})^2$ is the R-square of an equation where independent variable I is predicted by the other independent variables, j, k, l, and so forth. That is, each independent variable in turn is regressed against the other independent variables in the equation. The R^2 associated with that model is subtracted from 1. The higher the tolerance level, the less a particular independent variable shares its variance with the other independent variables.

Fixed Model vs. Stepwise Variable Selection

There are several strategies designed to reduce multicolinearity in a model. One is to start with a defined model and eliminate those variables that have a low tolerance. The total model is estimated and the coefficients for each of the variables are estimated at the same time. This is sometimes called a *fixed model.* Then, variables that are co-linear are removed from the equation, and the model is re-run.

Another strategy is to estimate the coefficients a step at a time, a procedure known as *stepwise* regression. There are several standard stepwise procedures. In the first procedure, variables are added one at a time (a *forward selection* model). The independent variable having the strongest linear correlation with the dependent variable is added first. Next, the independent variable from the remaining list of independent variables with the highest correlation with the dependent variable, *controlling for* the one variable already in the equation, is added next and the model is re-estimated. In each step, the independent variable with the highest correlation with the dependent variable controlling for the variables already in the equation is added to the model, and the model is re-estimated. This proceeds until either all the independent variables are added to the equation or else a stopping criterion is met. The usual criterion is only variables with a certain significance level are allowed to enter (called a *p-to-enter*).

A *backward elimination* procedure works in reverse. All independent variables are initially added to the equation. The variable with the weakest coefficient (as defined by the significance level) is removed, and the model is re-estimated. Next, the variable with the weakest coefficient in the second model is removed, and the model is re-estimated. This procedure is repeated until either there are no more independent variables left in the model or else a stopping criterion is met. The usual criterion is that all remaining variables pass a certain significance level (called a *p-to-remove*).

There are combinations of these procedures, for example adding a variable in a forward selection manner but then removing any variables that are no longer significant or using a backward elimination procedure but allowing new variables to enter the model if they suddenly become significant.

There are advantages to each approach. A fixed model allows specified variables to be included. If either theory or previous research has indicated that a particular combination of variables is important, then the fixed model allows that to be tested. A stepwise procedure might drop one of those variables. On the other hand, a stepwise procedure usually can obtain the same or higher predictability than a fixed procedure (whether predictability is measured by a log-likelihood or an R-square).

Within the stepwise procedures, there are also advantages and disadvantages to each method, though the differences are generally very small. A forward selection procedure adds variables one at a time. Thus, the contribution of each new variable can be seen. On the other hand, a variable that is significant at an early stage could become not significant at a later stage because of the unique combinations of variables. Similarly, a backward elimination procedure will ensure that all variables in the equation meet a specified significance level. But, the contribution of each variable is not easily seen other than through the coefficients. In practice, one usually obtains the same model with either procedure, so the differences are not that critical.

A stepwise procedure will not guarantee that multicolinearity will be removed entirely. However, it is a good procedure for narrowing down the variables to those that are significant. Then, any co-linear variables can be dropped manually and the model re-estimated. In the *CrimeStat* trip generation, both a fixed model and a backward elimination procedure are allowed.

Alternative Regression Models

There are a number of alternative methods for estimating the likely value of a count given a set of independent predictors. The negative binomial has already been mentioned. There are a number of variations of these involving different assumptions about the dispersion term. There are also a number of different Poisson-type models. Among these are the zero-inflated Poisson (or ZIP; ; Hall, 2000), the Weibul function, the Cauchy function, and the lognormal function (see NIST 2004 for a list of common non-linear functions).

There are also a set of spatial regression type models that correct for spatial autocorrelation in the dependent variable, such as geographically-weighted regression using a Poisson function (Fotheringham, Brunsdon, and Charlton, 2002), a hierarchical Bayesian model (Clayton and Kaldor, 1987), and a Markov Chain Monte Carlo simulation method (Miouw, Song, and Balilick, 2003).

In future versions of *CrimeStat*, several of these methods will be introduced. For the time being, though, the Poisson model is available as it is the most commonly used functional model for fitting count data.

Adding Special Generators

In a travel demand model, there are *special generators*. These are unique land uses or environments that produce an extra large number of trips. For regular travel demand

modeling, stadiums, airports, train stations, large parks, and 'mega-malls' generate more than their share of trips, or at least than what would be predicted by the amount of employment at those locations. They are usually attractors, not producers. In a normal transportation travel demand model, these zones are excluded from the cross-classification and independent estimates are made of them.

For crime trips, there are also special generators. Typically, these are zones that have more crimes being attracted to them than are expected on the basis of the population and employment at those locations.

Since we are using a regression model to estimate the productions and attractions, a simple way to model a special generator is to create a simple *dummy* variable. This is a variable where zones with the special generator get a value '1' and zones without the special generator get a '0'. Essentially, the variable is a cross-classification of the special generator versus every other zone.

One has to be cautious is doing this, however. Typically, special generators are identified by having a greater number of crimes being attracted to a zone than is predicted by the model. In other words, they have a greater positive residual error (observed - predicted) and are 'outliers' in the residual error distribution. By adding a variable to explain those cases, the residual error decreases.

But, in doing so, we aren't really explaining why the zone has more crimes than expected, but simply have accounted for it by putting in an empirical variable. In re-running the model, there will be, usually, new outliers that have a greater positive residual error. If this logic is to be repeated, then we would create new special generators for those zones and re-estimate the model. If continued without limits, eventually there would not be a model anymore but just a collection of dummy variables, one for each zone.

Therefore, a user should be cautious in introducing special generators. It is generally alright to introduce a few for the truly exceptional zones. These are zones where it is logical to treat them as special generators and where one would expect continuity over time. In other words, they should be used if the special generator status is expected to last over time. For example, a stadium or an airport or a train station is liable to remain at its location for many years. A particular shopping mall, on the other hand, may attract crimes at one particular point in time but not necessarily in the future. Unless it is a mall that is so much larger than any other mall in the region (a 'mega-mall'), it shouldn't be given a special generator status.

Adding External Trips

External trips are, by definition, trips that come from outside the region. They are part of the origin/production model in that these are trips that are not accounted for by the model. There are also trips that originate within the study area, but end outside the area; however, those are usually not modeled since the focus will be on the study area itself. In the usual travel demand framework, external trips are those coming from major corridors into

the region. Estimates of the travel on these corridors are obtained by *cordon counts*, counts of vehicles coming into the region and leaving the region (net inflow). Estimates of future growth of those external trips has to based on expectations of future population growth the metropolitan region and in nearby regions.

For crime trips, external trips are defined as trips that originate outside the study area. But they must be estimated by the difference between the total number of crimes occurring in the destination study area and the total originating in the origin zones. That is, of all the crimes occurring in the study area, the origin zones are modeled. Those trips that originate from outside the origin zones are external trips. They must be added to the predicted number of origin trips to produce an adjusted estimate of total origins, or:

$$O_j = O_{pi} + O_e \qquad (13.24)$$

where O_j is the total number of crime origins for crimes committed in study area, j, O_{pi} is the total number of crimes originating in the origin zones, I, and O_e is the total number of crimes originating outside the region, e.

In other words, for the production (origin) model **only**, we add an external zone to account for crime trips that originated outside the modeled region. If we don't do that, in the balancing step, we'll overestimate the number of crimes originating in each zone because the predicted origins will be multiplied by a factor to ensure that the total number of origins equals the total number of destinations.

Not including the external trips can lead to bias in the model. If the number of external trips is a sizeable percentage of all crime origins occurring in the study area, then the coefficients of the origin model could be misleading. In practice, most travel demand modelers assume that if the percentage of external trips is not greater than 5%, there usually is little bias introduced (Ortuzar and Willumsen, 2001). If it is greater than 5%, then origin zones from adjacent jurisdictions need to be included in the origin model.

Balancing Predicted Origins and Predicted Destinations

The trip generation 'model' is actually two separate models: 1) a model of trips produced by every zone and 2) a model of trips attracted to every zone. Since a trip has an origin and a destination (by definition), then the total number of productions must equal the total number of attractions:

$$\sum_{I=1}^{n} O_i = \sum_{j=1}^{n} D_j \qquad (13.25)$$

where *O* is a trip origin, *D* is a trip destination, and *I* and *j* are zone numbers.

To ensure that this equality is true, a balancing operation is conducted. Essentially, this means multiplying either the number of predicted origins in each origin zone or the

number of predicted destinations in each destination zone by a constant which is the ratio of either the total destinations to the total origins (to multiply the number of predicted origins) or the ratio of the total origins to the total destinations (to multiply the number of predicted destinations).

With crime analysis, the number of destinations would generally be considered a more reliable data set than the number of origins. Because crimes are enumerated where they occur, the number of crimes occurring at any one location is more accurate than the location of the offenders. Thus, we adjust the predicted origins so that they equal the predicted destinations.[6]

Summary of the Trip Generation Model

In summary, the trip generation model is estimated in four steps:

1. A model of the predictors of the number of crimes origins (a crime production model);
2. A model of predictors of the number of crime destinations (a crime attraction model);
3. External trips are estimated and added to the number of predicted origins as an external zone; and
4. The total number of predicted crime origins is balanced to be equal to the total number of predicted crime destinations.

The *CrimeStat* Trip Generation Model

In this section, we describe the trip generation model implemented in *CrimeStat*. As mentioned above, this step involves calibrating a regression model against the zonal data. Two separate models are developed, one for trip origins and one for trip destinations. The dependent variable is the number of crimes originating in a zone (for the trip origin model) or the number of crimes ending in a zone (for the trip destination model). The independent variables are zonal variables that may predict the number of origins or destinations.

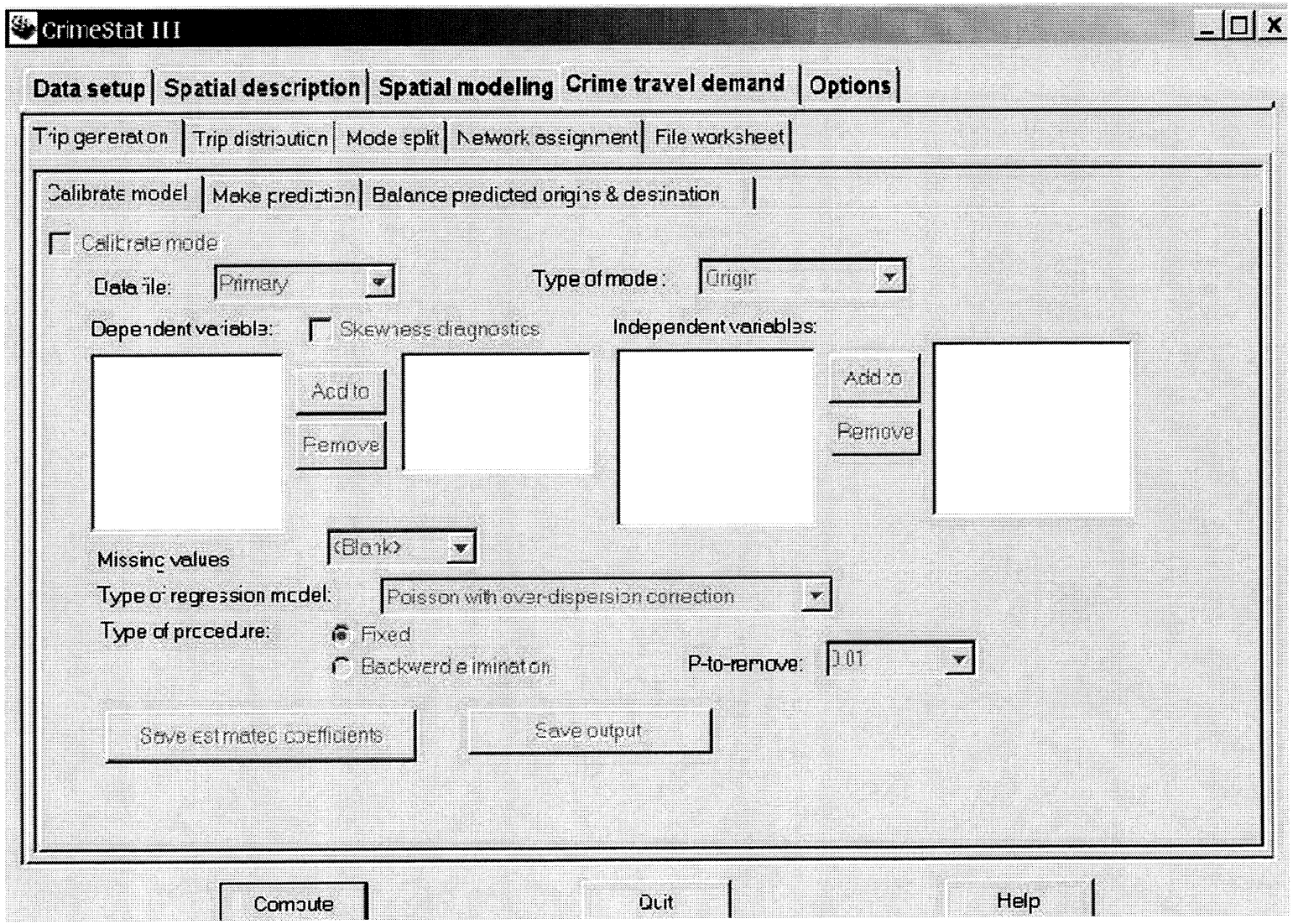
There are three steps to the model, each corresponding to a separate tab in *CrimeStat*:

1. Calibrate the model
2. Make a prediction
3. Balance the predicted origins and the predicted destinations

Figure 13.5 shows an image of the trip generation model page within *CrimeStat*. The trip generation model is made up of three separate pages (or tabs):

Figure 13.5:

Trip Generation Module



1. A *Calibrate model* page in which a regression model can be run to estimate either an origin (production) model or a destination (attraction) model;

2. A *Make prediction* page in which the estimated coefficients can be applied to the same or a different data set and in which the external trips can be added to the predicted origins; and

3. A *Balance predicted origins & destinations* page in which the total predicted origins can be adjusted to equal the total predicted destinations.

Calibrate Model

In the first step, models are calibrated using the input data. There is a model for the origin zones and another model for the destination zones. The user should indicate what type of model is being run in order to make the output more clear (it is not essential but can minimize confusion from mislabeling).

Data File

The data file is input as either the primary or secondary file. Specify whether the data file is the primary or secondary file.

Type of Model

Specify whether the model is for origins or destinations. This will be printed out on the output header.

Dependent Variable

Select the dependent variable from the list of variables. There can be only one dependent variable per model.

Skewness Diagnostics

If checked, the routine will test for the skewness of the dependent variable. The output includes:

1. The "g" statistic
2. The standard error of the "g" statistic
3. The Z value for the "g" statistic
4. The probability level of a Type I error for the "g" statistic
5. The ratio of the sample variance to the saple mean

Error messages indicate whether there is probable skewness in the dependent variable. If there is skewness, use a Poisson regression model.

Independent variables

Select independent variables from the list of variables in the data file. Up to 15 variables can be selected.

Missing values

Specify any missing value codes for the variables. Blank records will automatically be considered as missing. If any of the selected dependent or independent variables have missing values, those records will be excluded from the analysis.

Type of Regression Model

Specify the type of regression model to be used. The default is a Poisson regression with over-dispersion correction. Other alternatives are a Poisson regression and an Ordinary Least Squares regression.

Type of Regression Procedure

Specify whether a fixed model (all selected independent variables are used in the regression) or a backward elimination stepwise model is used. The default is a fixed model. If a backward elimination stepwise model is selected, choose the P-to-remove value (default is .01). The backward elimination starts with all selected variables in the model (the fixed procedure). However, it proceeds to drop variables that fail the P-to-remove test, one at a time. Any variable that has a significance level in excess of the P-to-remove value is dropped from the equation.

Save Estimated Coefficients/Parameters

The estimated coefficients of the final model can be saved as a 'dbf' file. Specify e a file name. This would be useful in order to repeat the regression while adding in external trips to the predicted origins (see Make trip generation prediction below) or to apply the coefficients to another dataset (e.g., future values of the independent variable).

Save Output

The output is saved as a 'dbf' file under a different file name. The output includes all the variables in the input data set plus two new ones: 1) the predicted values of the dependent variable for each observation (with the name PREDICTED); and 2) the residual error values, representing the difference between the actual /observed values for each observation and the predicted values (with the name RESIDUAL).

Poisson output

The output of the Poisson regression routines includes 13 fields for the entire model:

1. The dependent variable
2. The type of model
3. The sample size (N)
4. The degrees of freedom (N - # dependent variables – 1)
5. The type of regression model (Poisson, Poisson with over-dispersion correction)
6. The log-likelihood value
7. The Likelihood Ratio
8. The probability value of the Likelihood Ratio
9. The Akaike Information Criterion (AIC)
10. The Schwartz Criterion (SC)
11. The Dispersion Multiplier
12. The approximate R-square value
13. The deviance R-square value

and 5 fields for each estimated coefficient:

14. The estimated coefficient
15. The standard error of the coefficient
16. The pseudo-tolerance value of the coefficient (see below)
17. The Z-value of the coefficient
18. The p-value of the coefficient.

OLS output

The output of the Ordinary Least Square (OLS) routine includes 9 fields for the entire model:

1. The dependent variable
2. The type of model
3. The sample size (N)
4. The degrees of freedom (N - # dependent variables – 1)
5. The type of regression model (Norma/Ordinary Least Squares)
6. Squared multiple R
7. Adjusted squared multiple R
8. F test of the model
9. p-value of the model

and 5 fields for each estimated coefficient:

10. The estimated coefficient
11. The standard error of the coefficient
12. The tolerance value of the coefficient (see below)
13. The t-value of the coefficient
14. The p-value of the coefficient.

Multicolinearity Among the Independent Variables

To test multicolinearity, a tolerance test is run (see equation 13.23 above). There is not a simple test of whether a particular tolerance is meaningful or not. In *CrimeStat*, several qualitative categories are used and error messages are output:

1. If the tolerance value is 0.80 or greater, then there is little multicolinearity (No apparent multicolinearity);

2. If the tolerance is between 0.50-0.79, there is some multicolinearity (possible multicolinearity);

3. If the tolerance is between 0.25-0.49, there is probable multicolinearity (probable multicolinearity. Eliminate variable with lowest tolerance and re-run); and

4. If tolerance is less than 0.25, there is definite multicolinearity. (Definite multicolinearity. Results are not reliable. Eliminate variable with lowest tolerance and re-run).

Graph

While the output page is open, clicking on the graph button will display a graph of the residual errors (on the Y axis) against the predicted values (on the X axis).

Make Trip Generation Prediction

This routine applies an already-calibrated regression model to a data set. This would be useful for several reasons: 1) if external trips are to be added to the model (which is normally preferred); 2) if the model is applied to another data set; and 3) if variations on the coefficients are being tested with the same data set. The model will need to be calibrated first (see Calibrate trip generation model) and the coefficients saved as a parameters file. The coefficient parameter file is then re-loaded and applied to the data.

Data File

The data file is input as either the primary or secondary file. Specify whether the data file is the primary or secondary file.

Type of Model

Specify whether the model is for origins or destinations. This will be printed out on the output header.

Trip Generation Coefficients/Parameters File

This is the saved coefficient parameter file. It is an ASCII file and can be edited if alternative coefficients were being tested (be careful about editing this without making a backup). Load the file by clicking on the Browse button and finding the file. Once loaded, the variable names of the saved coefficients are displayed in the "Matching parameters" box.

Independent Variables

Select independent variables from the list of variables in the data file. Up to 15 variables can be selected.

Matching Parameters

The selected independent variables need to be matched to the saved variables in the trip generation parameters file in the same order. Add the appropriate variables one by one in the order in which they are listed in the matching parameters box. It is essential that the order by the same otherwise the coefficients will be applied to the wrong variables.

> Hint: With your cursor placed in the list of independent variables, typing the first letter of the matching variable name will take you to the first variable that starts with that letter. Repeating the letter will move down the list to the second, third, and so forth until the desired variable is reached.

Missing Values

Specify any missing value codes for the variables. Blank records will automatically be considered as missing. If any of the selected dependent or independent variables have missing values, those records will be excluded from the analysis.

Add External Trips

External trips are trips that start outside the modeled study area. Because they are crimes that originate outside the study area, they were not included in the zones used for the origin model. Therefore, they have to be independently estimated and added to the origin zone total to make the number of origins equal to the number of destinations. Click on the "Add external trips" button to enable this feature.

Number of external trips

Add the number of external trips to the box. This number will be added as an extra origin zone (the External zone).

Origin ID

Specify the origin ID variable in the data file. The external trips will be added as an extra origin zone, called the "External" zone. Note: the ID's used for the destination file zones should be the same as in the origin file. This will be necessary in subsequent modeling stages.

Type of Regression Model

Specify the type of regression model to be used. The default is a Poisson regression and the other alternative is a Normally-distributed/Ordinary Least Squares regression.

Save Predicted Values

The output is saved as a 'dbf' file under a different file name. The output includes all the variables in the input data set plus the predicted values of the dependent variable for each observation (with the name PREDICTED). In addition, if external trips are added, then there should be a new record with the name EXTERNAL listed in the Origin ID column. This record lists the added trips in the PREDICTED column and zeros (0) for all other numeric fields.

Output

The tabular output includes summary information about file and lists the predicted values for each input zone.

Balance Predicted Origins & Destinations

Since, by definition, a 'trip' has an origin and a destination, the number of predicted origins must equal the number of predicted destinations. Because of slight differences in the data sets of the origin model and the destination model, it is possible that the total number of predicted origins (including any external trips – see Make trip generation prediction above) may not equal the total number of predicted destinations. This step, therefore, is essential to guarantee that this condition will be true. The routine adjusts either the number of predicted origins or the number of predicted destinations so that the condition holds. The trip distribution routines will not work unless the number of predicted origins equals the number of predicted destinations (within a very small rounding-off error).

Predicted Origin File

Specify the name of the predicted origin file by clicking on the Browse button and locating the file.

Origin variable

Specify the name of the variable for the predicted origins (e.g., PREDICTED).

Predicted Destination File

Specify the name of the predicted destination file by clicking on the Browse button and locating the file.

Destination variable

Specify the name of the variable for the predicted origins (e.g., PREDICTED).

Balancing method

Specify whether origins or destinations are to be held constant. The default is 'Hold destinations constant'.

Save Predicted Origin/Destination File

The output is saved as a 'dbf' file under a different file name. The output includes all the variables in the input data set plus the adjusted values of the predicted values of the dependent variable for each observation. If destinations are held constant, the adjusted variable name for the predicted trips is ADJORIGIN. If origins are held constant, the adjusted variable name for the predicted trips is ADJDEST.

Output

The tabular output includes file summary information plus information about the number of origins and destinations before and after balancing. In addition, the predicted values of the dependent variable are displayed.

Example Trip Generation Model

To illustrate this model, let's run through these procedures using an example from Baltimore County. In the case of Baltimore County, traffic analysis zones (TAZ) were used for the zonal geography. Two data sets are produced, one for the crime origins and one for the crime destinations. For Baltimore County, the origin data set has 532 zones covering both Baltimore County and the City of Baltimore with the total number of crime origins for each zone (sub-divided into different crime types - robberies, burglaries, vehicle theft) and a number of possible predictor variables (population, retail and non-retail employment, median household income, poverty levels, and vehicle ownership). Similarly, the destination data set has 325 zones with the number of crime destinations for each zone (again, sub-divided into different crime types) and number of possible predictor variables (population, retail and non-retail employment, median household income, and several land use categories - acreage allocated for retail, residential, office space, and conservation uses). Sample data sets are provided on the *CrimeStat* download page.

Setting Up the Origin Model

In the first step, an origin model is created. Figure 13.6 shows the selection of the dependent variable and some possible independent variables. The type of model is an ordinary Poisson regression. The dependent variable is the number of crimes occurring between 1993 and 1997 in each origin zone (BCORIG). Eight possible independent variables have been selected: the 1996 population of each zone (POP96), the median household income of the zone relative to the zone with the highest median household income (INCEQUAL), the number of 1996 non-retail employees in each zone (NONRET96), the number of 1996 retail employees in each zone (RETEMP96), the total linear miles of arterial roads in each zone (ARTERIAL), a dummy variable for whether the Baltimore Beltway (I-695) passed through the zone or not (BELTWAY), the linear distance of the zone from Baltimore harbor in the CBD (DISTANCE), and the number of households without automobiles (ZEROAUTO - this cannot be seen in the image).

The model is set up to run a Poisson regression without an over-dispersion correction. It is a fixed model in which all independent variables are included. The coefficients are saved under "Save estimated coefficients" dialogue box and the output (the predicted values) are saved under the "Save output" dialogue box. Both boxes ask for a file name.

Table 13.2 shows the results. Key statistics are highlighted. The overall model is highly significant. The Likelihood Ratio is highly significant and the R-squares are reasonably high (0.50 for the R-square and 0.42 for the deviance R-square). The coefficients for each of the variables are significant.

However, there are two major problems. First, the dispersion multiplier (parameter) is very large (37.087), indicating that the conditional variance is more than 37 times greater than the conditional mean. Second, while all of the coefficients are significant, several show sizeable multicolinearity as evidenced by the pseudo-tolerance value (POP96, DISTANCE, ZEROAUTO). This indicates that these variables are essentially measuring the same thing.

Restructuring the Origin Model

Consequently, the model is restructured in three ways (figure 13.7). First, the over-dispersion correction is applied. Second, the co-linear variables DISTANCE and ZEROAUTO are dropped from the model. Third, a stepwise backward elimination procedure is used with the probability for keeping a variable in the equation (p-to-remove) being 0.01; that is, unless the probability that a coefficient could be obtained by chance is less than 1 in 100, the variable is dropped.

The result is now a model with the Likelihood Ratio and R-squares being almost as high as in the first model and in which all the coefficients are significant, but there is very little multicolinearity. (Table 13.3). The dispersion multiplier is now 1.0 since the coefficient standard errors have been corrected for the original over-dispersion (see equation 13.8 above).

Figure 13.6:

Origin Poisson Model Setup

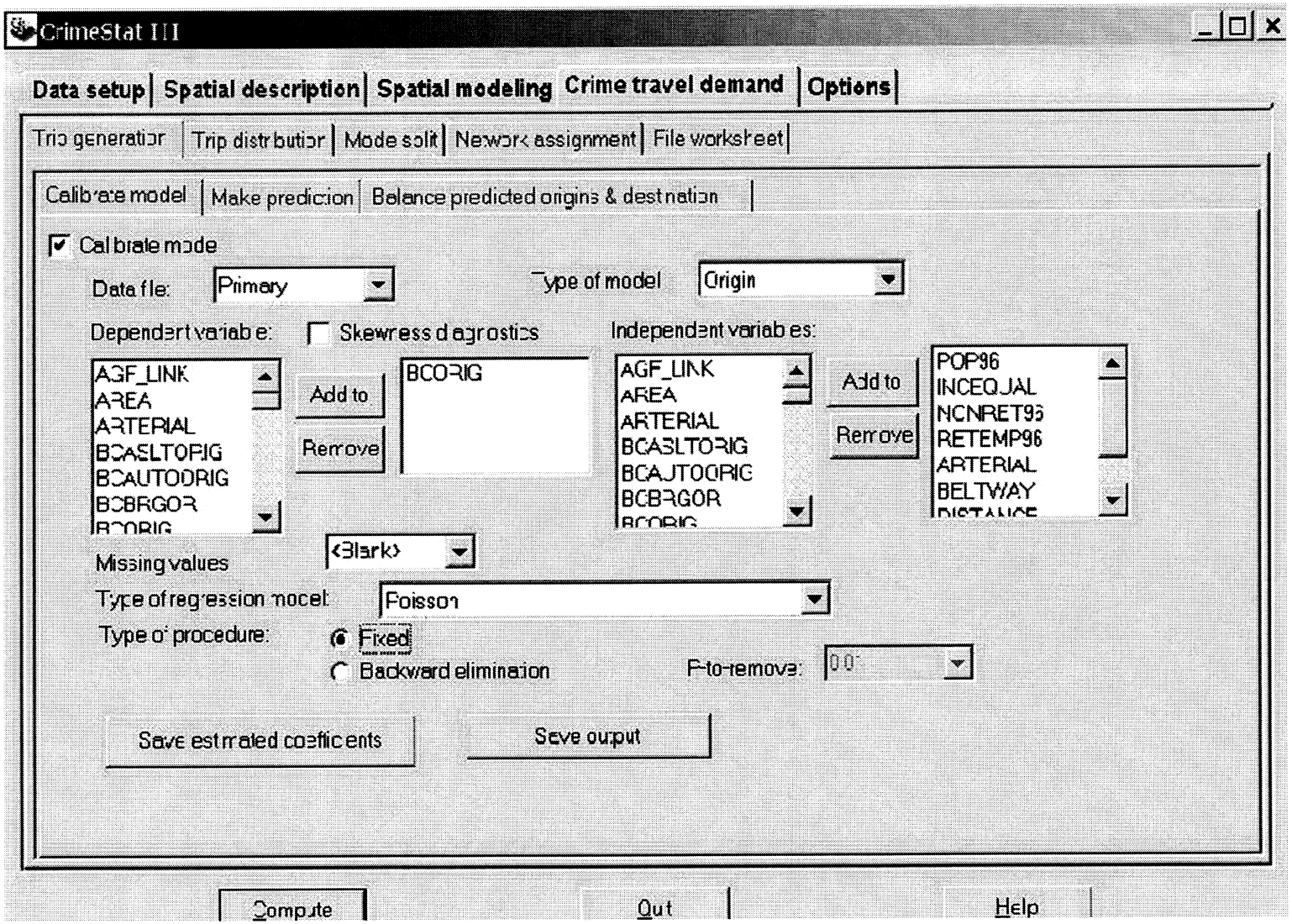


Figure 13.7:

Origin Poisson Model with Over-dispersion Correction

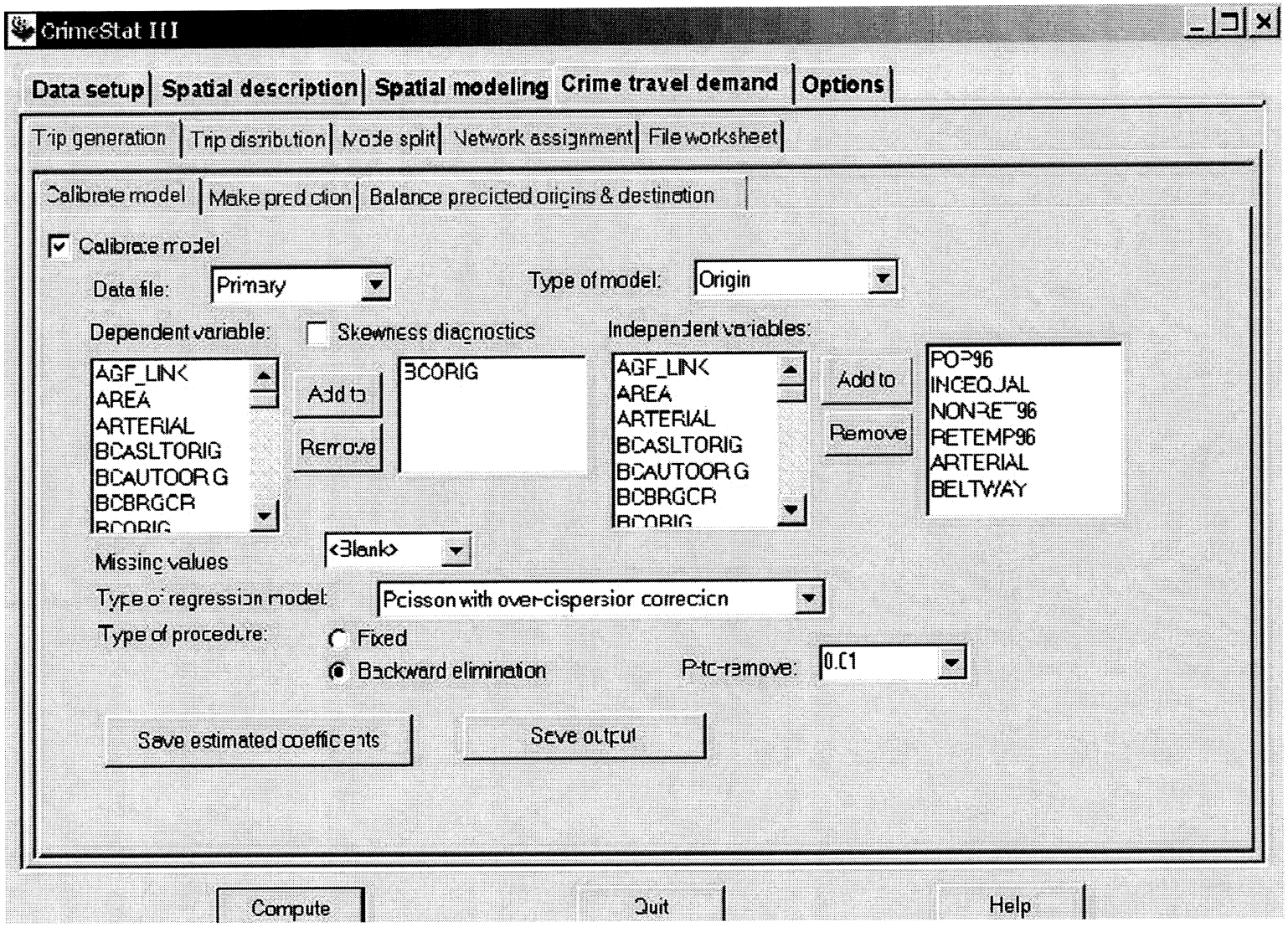


Table 13.2

Results of First Origin Model Run

Model result:

Data file:	BaltOrigins.dbf
Type of model:	Origin
DepVar:	**BCORIG**
N:	532
Df:	523
Type of regression model:	Poisson with over-dispersion correction
Log Likelihood:	-10678.051687
Likelihood ratio(LR):	**25609.182621**
P-value of LR:	**0.0001**
AIC:	21374.103373
SC:	21412.593165
Dispersion multiplier:	37.086973
R-square:	**0.499539**
Deviance r-square:	0.420031

Predictor	DF	Coefficient	Stand Error	Pseudo-Tolerance	z-value	p-value
CONSTANT	1	**0.887266**	0.037707	.	23.530608	0.001
POP96	1	**0.000337**	0.000016	**0.463218**	21.665568	0.001
INCEQUAL	1	**-0.033017**	0.001226	0.608346	-26.926013	0.001
NONRET96	1	**-0.000173**	0.000028	0.842042	-6.082943	0.001
RETEMP96	1	**-0.000364**	0.000117	0.960564	-3.107357	0.010
ARTERIAL	1	**-0.108257**	0.025888	0.771634	-4.181834	0.001
BELTWAY	1	**0.150967**	0.036047	0.958973	4.188082	0.001
DISTANCE	1	**0.034289**	0.007842	**0.491906**	4.372170	0.001
ZEROAUTO	1	**-0.000462**	0.000141	**0.355510**	-3.283930	0.010

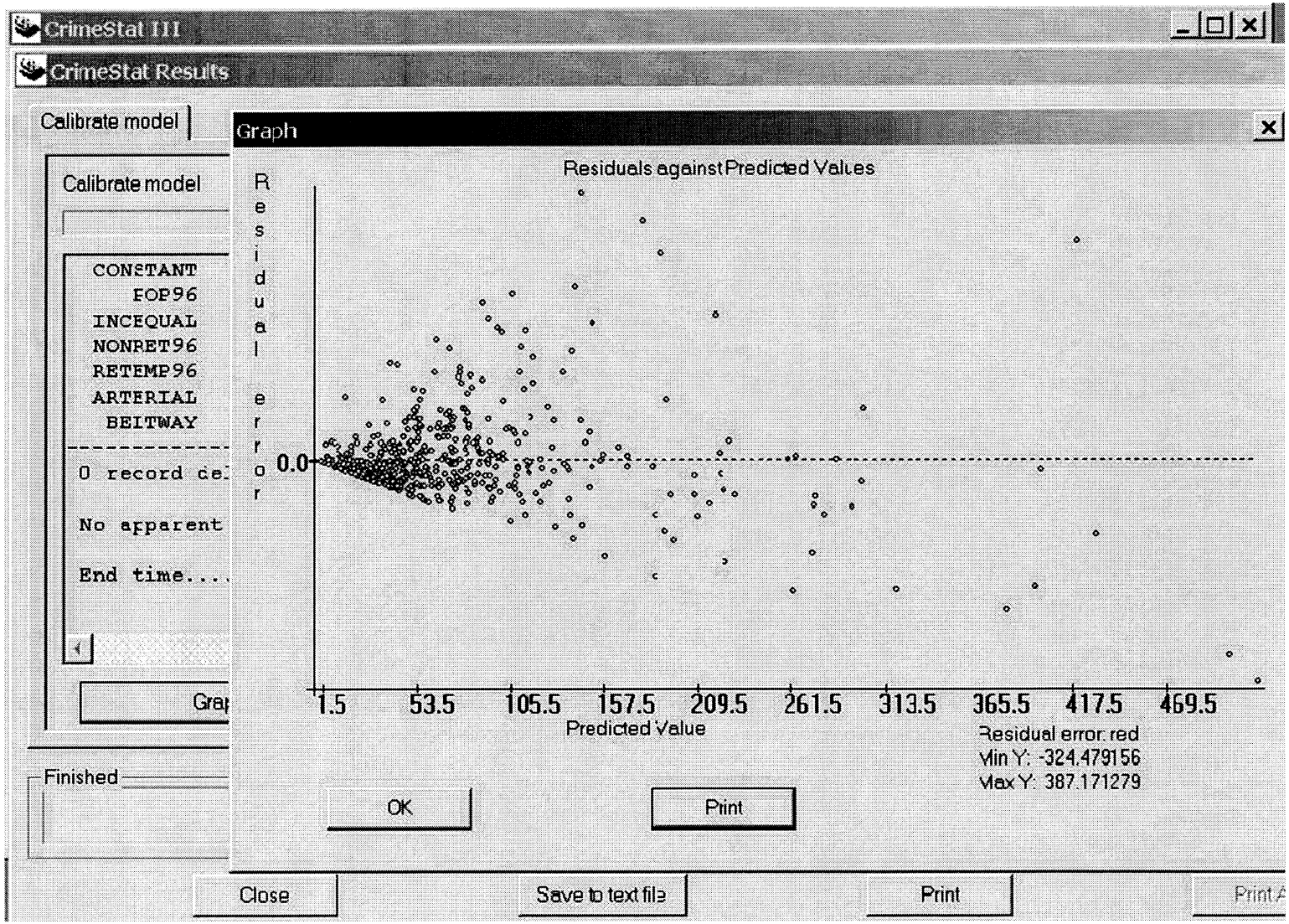
Looking at the model, we see six variables that significantly predict the number of crime origins. Population is the strongest, as indicated by its Z-test. Relative income equality is the next strongest, but this is a negative coefficient; that is, zones with high relative income equality produce fewer crime origins whereas zones with low relative income equality produce more crime origins. The third and fourth strongest variables are non-retail and retail employment respectively, but, again, the coefficients are negatives; zones with less employment have more crimes originate from them. Finally, the two roadway variables show significant effects. Zones in which the Baltimore Beltway passes through them have a higher number of crimes originating (as might be expected) and also zones with fewer miles of arterial have more crimes originating; with the latter variable, it's possible that we are picking up the lack of commercial employment opportunities since retail firms tend to locate on arterial roads rather than local streets.

Residual Analysis of Origin Model

The *CrimeStat* output includes a graph of the residual errors (actual values minus the predicted values) on the Y-axis by the predicted values on the X-axis. It is important to examine the residual errors as these can indicate outliers, problems in the data, and violation of assumptions. Figure 13.8 shows an image of the residual graph screen. As seen,

Figure 13.8:

Plot of Residual Errors and Predicted Values



the errors increase with the value of the predicted dependent variable. With the Poisson model, this is expected and does not indicate the violation of the independent errors assumption, as it does with the OLS. The errors are reasonably symmetrical and do not indicate differences in over- and under-estimation across the band of the predicted values.

There are some outliers; there are two zones that predicted substantially more crimes than actually originated in those zones and there is one zone that had more crimes originate from it than was predicted by the model. But, in general, the model appears to be reasonably balanced.

Table 13.3

Results of Second Origin Model

Model result:	
Data file:	BaltOrigins.dbf
Type of model:	Origin
DepVar:	**BCORIG**
N:	532
Df:	525
Type of regression model:	Poisson with over-dispersion correction
Log Likelihood:	-11262.292156
Likelihood ratio(LR):	**24440.701682**
P-value of LR:	**0.0001**
AIC:	22538.584312
SC:	22568.520816
Dispersion multiplier:	1.000000
R-square:	**0.455630**
Deviance r-square:	0.446502

Predictor	DF	Coefficient	Stand Error	Pseudo-Tolerance	z-value	p-value
CONSTANT	1	**2.286699**	0.039339	.	58.127787	0.001
POP96	1	**0.000284**	0.000013	0.943426	22.473451	0.001
INCEQUAL	1	**-0.018525**	0.001026	0.849679	-18.048743	0.001
NONRET96	1	**-0.000186**	0.000030	0.866522	-6.139941	0.001
RETEMP96	1	**-0.000353**	0.000125	0.960769	-2.820286	0.010
ARTERIAL	1	**-0.085070**	0.027006	0.938167	-3.150019	0.010
BELTWAY	1	**0.123109**	0.037868	0.970051	3.251004	0.010

Setting Up the Destination Model

The same logic is applied for the destination model. In this case, the destination file has data on 325 zones within Baltimore County only. Similar possible predictor variables are included in the file. Aside from population, retail and non-retail employment, and the roadway variables, more detailed analysis on land uses were included (acreage of commercial, residential, office space, recreational, and conservation lands). The model that was run was a Poisson with an over-dispersion correction. Again, a backward elimination procedure was adopted. Once a final model was selected, it was re-run as a fixed model to ensure that the coefficients were consistently estimated. Table 13.4 presents the results.

Five variables ended up in the final model. Again, population was significantly related to the number of crimes attracted to a zone, but was not the strongest predictor as indicated by the Z-test. The strongest relationship was for the number of retail employees. This suggests that retail/commercial areas attract many crimes. This is supported by one of the land use variables - the acreage associated with very large malls; in other words, there are additional crimes attracted to very large malls above-and-beyond the number of retail employees in those zones. Two other variables are in the equation. Relative income equality was, again, negatively related to crime destinations/attractions; zones with low income tend to attract more crimes. Also, there was a negative association with distance from the CBD. The farther away from the CBD, the lower the number of crimes. Overall, the model suggests that zones with commercial activities, particularly with large malls, but which are closer to the city center and which have households with relatively lower incomes are those that attract the most crimes.

The overall model was highly significant, as indicated by the Likelihood Ratio and the R-square. There was a discrepancy between the R-square statistic and the Deviance R-square, making it unclear about how strong is the model (the R-square would suggest that it's strong whereas the Deviance R-square would not). Nevertheless, the overall predictability is reasonable. The amount of multicolinearity is tolerable.

Table 13.4

Results of First Destination Model

Model result:

Data file:	BCDestinations.dbf
Type of model:	Destination
DepVar:	**BCDEST**
N:	325
Df:	319
Type of regression model:	Poisson with over-dispersion correction
Log Likelihood:	-10347.872494
Likelihood ratio(LR):	**41708.925054**
P-value of LR:	**0.0001**
AIC:	20707.744988
SC:	20730.447939
Dispersion multiplier:	1.000000
R-square:	**0.596921**
Deviance r-square:	0.310251

Predictor	DF	Coefficient	Stand Error	Pseudo-Tolerance	z-value	p-value
CONSTANT	1	**5.485851**	0.218977	.	25.052182	0.001
POP96	1	**0.000190**	0.000027	0.928694	6.935850	0.001
INCEQUAL	1	**-0.017176**	0.005464	0.903130	-3.143462	0.010
RETEMP96	1	**0.001018**	0.000062	0.717076	16.297855	0.001
VERYLRGMLACR	1	**0.006446**	0.000974	0.740927	6.616423	0.001
DISTANCE	1	**-0.115709**	0.017069	0.876461	-6.778875	0.001

Residual Analysis of Destination Model

As with the origin model, an analysis was conducted of the residual errors. This time, the output 'dbf' file was brought into Excel and a nicer graph created (figure 13.9). Unlike the best origin model, the dispersion of the residuals is not symmetrical. There are several major outliers, both on the negative end of the residuals (over-estimation of crime attractions) and on the positive end (under-estimation of crime attractions). In particular, there are two zones that seem to stand out. Both of them have shopping malls (Golden Ring Mall and Eastpoint Mall). But the amount of crime in those zones is much greater than the model predicts. This is seen as high positive residuals (i.e., there were more actual crimes than predicted). They both are older malls, but are located in relatively high crime areas. Golden Ring Mall was demolished several years ago, but after the data that are being analyzed in this example were collected.

Adding in Special Generators

Since the number of crime incidents (attractions) in those two zones were much higher than was expected, they were treated as 'special generators'. Keeping in mind the caution that one doesn't want to over-use this category, we can still demonstrate how it works. Two new variables were created for the data set. One was for the Golden Ring Mall and one was for the Eastpoint Mall. For the Golden Ring Mall, the zone that included it received a '1' for this variable while all other zones received a '0'. Similarly, for the Eastpoint Mall variable, the zone in which it occurred received a '1' while all other zones received a '0'. These *dummy* variables were then included in the model (Table 13.5).

Adding the two special generators increases the predictability substantially. The Likelihood Ratio jumps as does the R-square; the Deviance R-square statistic, however, actually drops, suggesting that it is not a reliable indicator with these data. The coefficients for the two zones, treated as special generators, are both highly significant and, in fact, are the strongest variables in the equation. All other variables have the same relationships as in the first run. There does not appear to be substantial multicolinearity.

This brings up an issue over the status of a special generator. In this example, the two zones were treated as special generators in the model. While the predictability increased substantially, one has to wonder whether this was a meaningful operation? That is, if this model were applied to data for a later time period (e.g., 2002-2004 crime data), would the relationships still hold? In the case of the Golden Ring Mall, it wouldn't since that mall has since been demolished.

The value of a special generator is that it identifies a land use that would be expected to be relatively permanent (e.g., a stadium or a train station or an airport). In the case of a shopping mall, it may or may not. If it's a high visibility 'regional' mall, then treating it as a special generator is probably a good idea. If it's a smaller, older mall, on the other hand, the analysis is guessing that the mall will maintain its status as a high crime attraction location. Clearly, judgment and knowledge of the particular mall is essential.

Figure 13.9:

Residual Errors for Crime Destinations

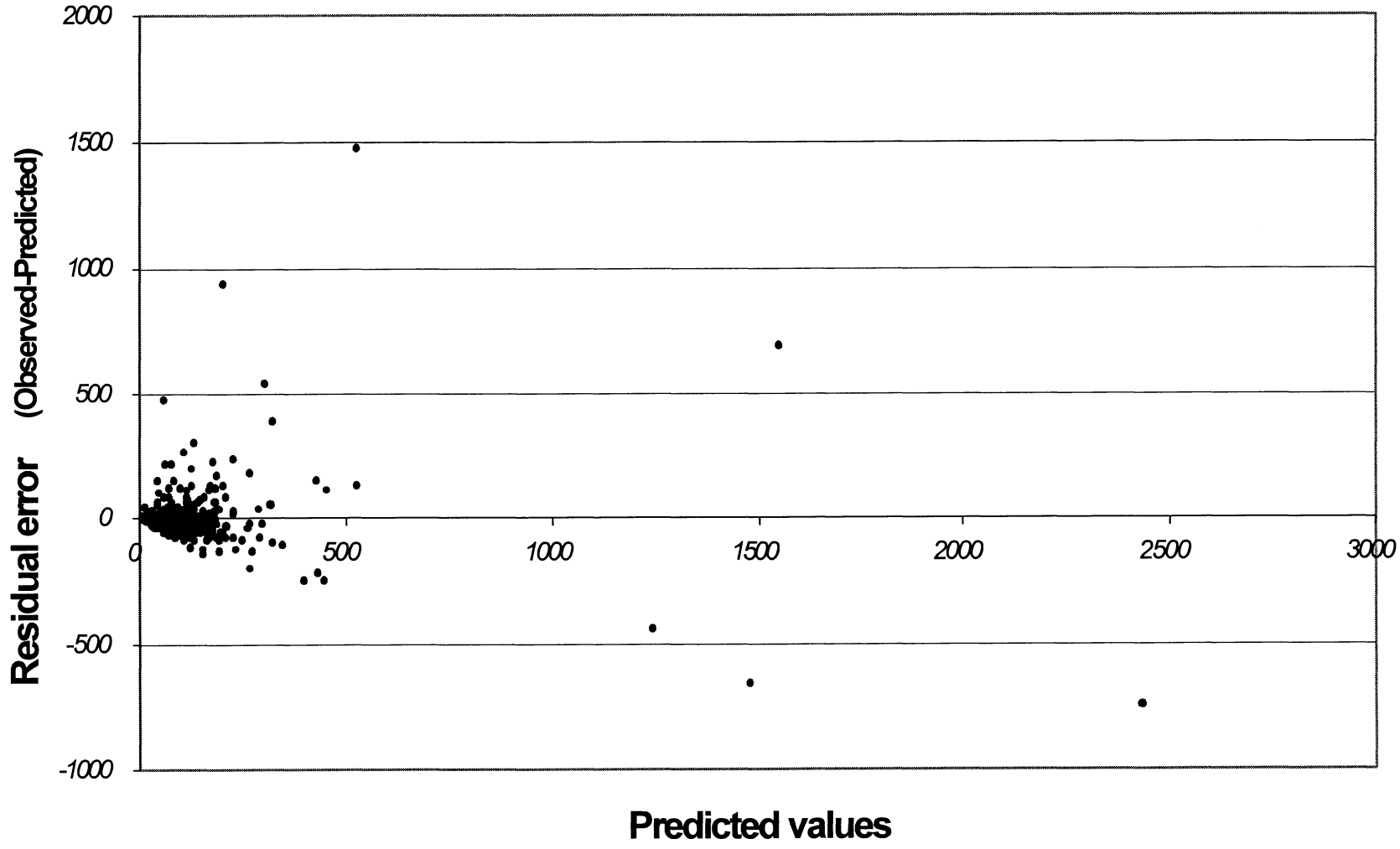


Table 13.5

Results of Second Destination Model

Model result:

Data file:	BcDestinations.dbf
Type of model:	Destination
DepVar:	**BCDEST**
N:	325
Df:	317
Type of regression model:	Poisson with over-dispersion correction
Log Likelihood:	-7852.238456
Likelihood ratio(LR):	**46700.193131**
P-value of LR:	**0.0001**
AIC:	15720.476911
SC:	15750.747513
Dispersion multiplier:	1.000000
R-square:	**0.784194**
Deviance r-square:	0.227710

Predictor	DF	Coefficient	Stand Error	Pseudo-Tolerance	z-value	p-value
CONSTANT	1	**5.182117**	0.067867	.	76.356923	0.001
INCEQUAL	1	**-0.020797**	0.003942	0.902950	-5.276135	0.001
RETEMP96	1	**0.000995**	0.000051	0.700294	19.338957	0.001
VERYLRGMLACR	1	**0.006590**	0.000869	0.716299	7.582758	0.001
POP96	1	**0.000238**	0.000020	0.921456	12.164552	0.001
DISTANCE	1	**-0.087826**	0.012462	0.872535	-7.047735	0.001
GOLDENRING	1	**1.933321**	0.069636	0.969044	27.763123	0.001
EASTPOINT	1	**1.602000**	0.067934	0.943548	23.581751	0.001

Comparing Different Crimes Types

With or without special generators, a trip generation model is an ecological model that predicts crime origins and crime destinations. A point was made in chapter 11 that these models are not behavioral, but are correlates of crimes. That is, the variables that end up predicting the number of crimes are not *reasons* (or explanations) for the crimes. Population almost always enters the equation because, all other things being equal, zones with larger numbers of persons will have more crimes, both originating and ending in them. Similarly, low income status is frequently associated with high crime areas. It doesn't follow that low income persons will be more prone to commit crimes; it may be true but these models don't test that proposition. These are only correlates with crime in those environments. As was mentioned earlier, these variables are often correlated with many specific conditions that *may* be predictors of individual crime - poverty, drug use, substandard housing, and lack of job opportunities.

To see this, three separate models of specific crime types were run for robbery, burglary, and vehicle theft. For each crime type, the general model was tested for both the origin and the destination models. If a variable was not significant, it was dropped and the model was re-run. The results of the origin model for the three crime types are seen in table 13.6 while the results of the destination model are seen in table 13.7.

Table 13.6

Models for Specific Crime Types: Origin Model

	All Crimes	Robbery	Burglary	Vehicle Theft
CONSTANT	2.286699	-0.652291	1.621546	-0.800759
INCOME EQUALITY	-0.018525	-0.023964	-	-0.019620
NON-RETAIL EMPLOYMENT	-0.000186	-0.000237	-0.000239	-0.000188
RETAIL EMPLOYMENT	-0.000353	-	-	
POPULATION	0.000284	0.000297	0.000242	0.000342
BELTWAY	0.123109	-	-	-
MILES OF ARTERIAL	-0.085070	-	-	-0.180966

Table 13.7

Models for Specific Crime Types: Destination Model

	All Crimes	Robbery	Burglary	Vehicle Theft
CONSTANT	5.485851	3.284488	3.246183	2.610299
INCOME EQUALITY	-0.017176	-0.027946	-0.034598	-0.012910
RETAIL EMPLOYMENT	0.001018	0.000844	-	0.000507
VERY LARGE MALL ACREAGE	0.006446	0.004332	-	-
POPULATION	0.000190	0.000223	0.000309	0.000247
DISTANCE FROM CBD	-0.115709	-0.096330	-0.038715	-0.096088

The population variable appears in every single model. As mentioned, all other things being equal, the larger the number of persons in a zone, the more crime events will occur whether those events are crime productions (origins) or crime attractions

(destinations). Similarly, relative income equality appears in five of the six models with the coefficient always being negative. In general, zones with relatively lower incomes will have more robberies, burglaries, and vehicle thefts. The only model for which income equality did not appear was as an origin variable for burglaries; apparently, burglars come from zones with various income levels, at least in Baltimore.

The other general variables have more limited applicability. Retail employment predicts both total crime origins and total crime destinations, but only predicts specifically robbery destinations and vehicle theft destinations; the latter tend to occur more in commercial areas than not. On the other hand, non-retail employment appears to be important only as a crime origin variable; zones with less non-retail employment tend to produce more offender trips. Distance from the CBD only appears as a destination variable; the closer a zone is to the metropolitan center, the higher the number of crimes being attracted to that zone; this variable was not important in the origin model.

In other words, these models are measuring general conditions associated with crime, not causes *per se*. They capture the general contextual relationships associated with crime productions and attractions. But, they don't necessarily predict individual behavior. Nevertheless, the models can be used for prediction since the conditions appear to be quite general.

Adding External Trips

After an origin and destination model has been developed, the next step is to add any crime trips that came from outside the modeling area (external trips). In this case, these would be trips that came from areas that were not in either Baltimore County or the City of Baltimore (the modeling area).

A simple estimate of external trips is obtained by taking the difference between the total number of crimes occurring in the study area (Baltimore County destinations) and the total number of crimes originating in the modeling area (table 13.8).

The difference between the number of crime enumerated within Baltimore County and that originating from both Baltimore County and the City of Baltimore is 1,627. This is 3.9% of the total Baltimore County crimes. In general, it is important that the external trips be as small as possible. Ortuzar and Willumsen (2001) suggest that this percentage be no greater than 5% in order to minimize potential bias from not including those cases in the origin model. It's not an absolute percentage, but more like a rule of thumb; in theory, any external trips could bias the origin model. But, in practice, the error will be small if external crime trips are a small percentage of the total number enumerated in the destination county.

In this case, the condition holds. For the three types of crime modeled, the percentage of external trips was also less than 5%: robbery (4.0%), burglary (4.5%), and vehicle theft (1.4%). On the other hand, if the percentage of external trips is greater than approximately 5%, a user would be advised to widen the origin study area to include more zones in the model.

Table 13.8

Estimating External Crime Trips in Baltimore County

Number of crimes ending in 325 Baltimore County zones:	41,969
Number of crimes originating in 532 Baltimore County/City zones:	40,342
Crimes from outside the modeling area:	1,627

Note: external trips are only added to the origin model since they are crime trips that originate outside the modeling area. They are not relevant for the destination model.

Predicting External Trips

If a model is being applied to another data set from which it was initially estimated, a problem emerges about how to estimate the number of external trips. It is one thing to apply simple arithmetic in order to determine how many trips originated outside the modeling area (as in table 13.8). It is another to know how to calculate external trips when the model is being applied to other data. For the modeled zones, the coefficients are applied to the variables of the model (see "Make Prediction" below). But, the external trips have to be estimated independently.

There is not a simple way to estimate external crime trips. Unlike regular trips that can be estimated through cordon counts, crime trips are not detectable while they are occurring (i.e., one cannot stand by a road and count offenders traveling by). Thus, they have to be estimated.

A simple method is to calculate the number of external trips for two time periods. For example, external trips could be calculated from a 2000 data set by subtracting the total number of crimes occurring in the modeling region from the total number of crimes occurring in the study area (e.g., as in table 13.8 above). If a similar calculation was made for, say, 2002, then the difference (the 'trend') could be extrapolated. To take our example, between 1993 and 1996, there were 1,627 external trips. If the number of external trips turned out to be 1,850 for 1997-2000, then the difference (1,850 - 1,627 = 223) could be applied for future

years. Essentially, a slope is being calculated and applied as a linear equation:

$$Y_i = 1850 + 223*X_i$$

where Y_i is the number of crime origins during a four year period, I, and X_i is an integer for a four year period starting with the next period (i.e., the base year, 1997-2000, has integer value of 0). In other words, a linear trend is being extrapolated.

How realistic is this? For short time periods, linear extrapolation is probably as good a method as any. But for longer time periods, it can lead to spurious conclusions (e.g., crime trips from outside the region will always increase). Short of developing a sophisticated model that relates crime trips to the growth of the metropolitan area and to other metropolitan areas within, say, 500 miles, a linear extrapolation is one of the few methods that one can apply.[7]

Make Prediction

In *CrimeStat*, external trips are added on the second page of the trip generation - Make prediction. This is a page where the modeled coefficients and any external trips are applied to a data set. There are two reasons why this is a separate page from the "Calibrate model" page where the model was calibrated. First, the coefficients might be applied to another data than that from which it was calibrated. For example, one might calibrate the model with a data set from 1998-2000 and then apply to a data set covering 2001-2003. Similarly, one might take future year forecasts (e.g., 2025) and apply the model. In effect, the model would be predicting the number of future crimes *if* the same conditions hold over the time frame.

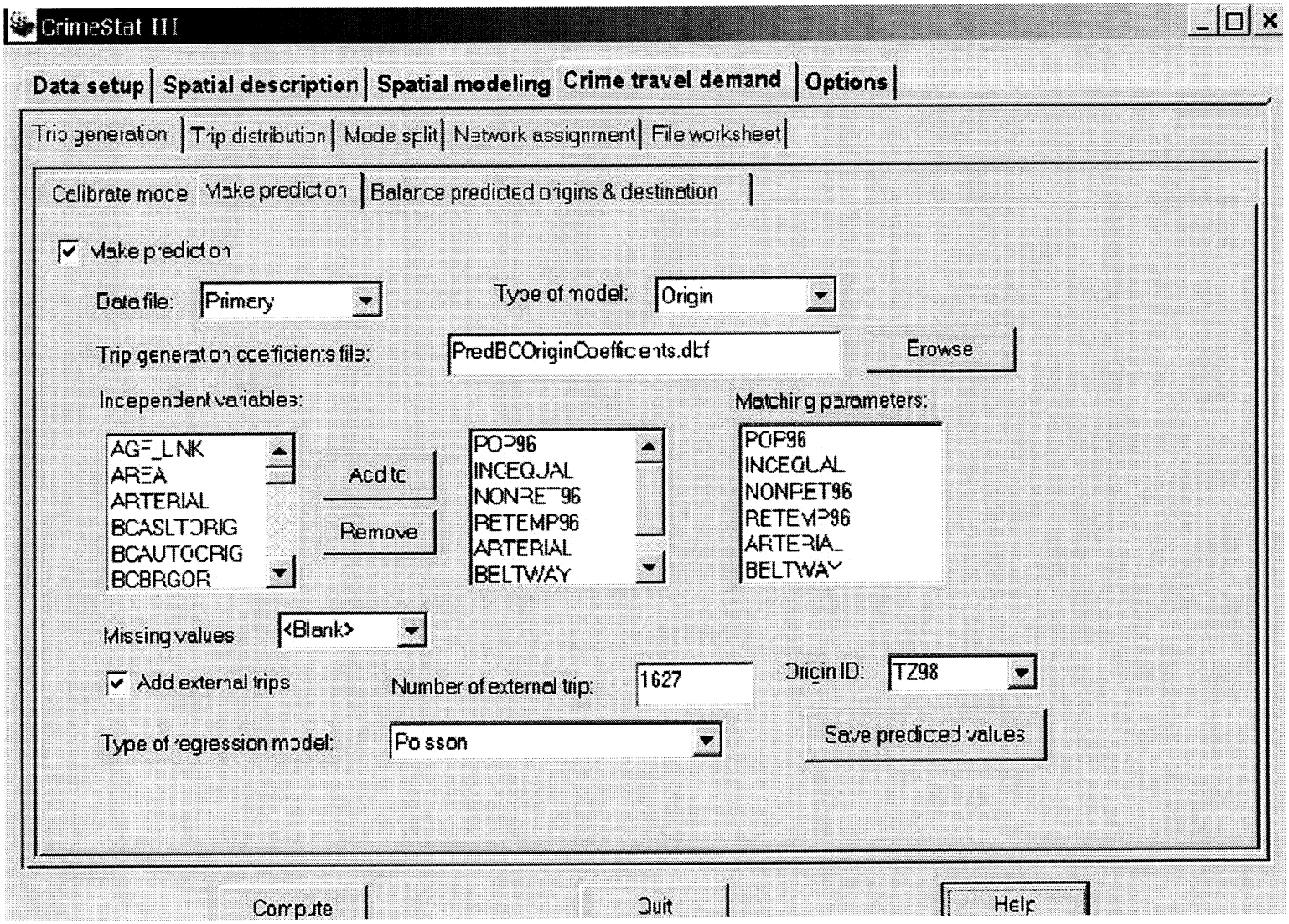
A second reason for separating the calibration and application pages is to add external trips to the origin zones. As mentioned above, external trips are, by definition, those that were not modeled in the calibration. They have to be calculated independently of the model and then added in.

Thus, the "Make prediction" page allows these operations to occur. Figure 13.10 shows the page. There are several steps that have to be implemented for this page to be operative.

1. The data file has to be input as either the primary or secondary file (not shown in the image). In this example, the same data set is being used as was used for the calibration. But, if it's a different data set, that will need to be input in the Data Setup section. Whether the input data set is a primary file (the usual occurrence) or a secondary file needs to be specified. Also, indicate whether the applied model is to be an origin or destination model. In figure 13.10, it is specified as an origin file.

Figure 13.10:

"Make Prediction" Setup Page



2. A trip generation coefficients file needs to be input. These were the estimated coefficients from the calibration stage. Inputting this file brings in the coefficients in the order in which they were saved. They are listed in the "Matching parameters" dialogue box on the right side of the page.

3. On the left side of the page are listed all the variables in the input data set (primary or secondary file). In the middle box, the variables are added in the **same order** as in the matching parameters box. That is, each independent variable needs to be matched to the variable from the coefficients file, one for one. **This is very important**. The names do not have to be the same (e.g., if the model was calibrated with data set and applied to another, the variable names may not be identical). But the content and order of the variables needs to be the same. In the example, the first variable in the coefficients file is INCEQUAL. The selected variable in the middle box has to be the income equality variable (whatever its name). In the example, the same data set is being used so the names are identical. This is repeated for each of the independent variables in the coefficients file.

4. Next, any missing value codes are specified in the missing values box. Any records with a missing value for *any* of the selected independent variables will be dropped from the calculation. In the example, there are no missing value codes applied other than the default blank field.

5. If external trips are to be added, the external trips box must be checked. External trips could be applied in an origin model, but not in a destination model. If they are to be added, the number of trips should be specified in the "Number of external trips" box and the zone ID field for the file indicated; in the example, 1627 is added as external trips and the TAZ field is specified as the ID variable (TZ98).

6. The type of model to be applied is indicated in the "Type of regression model" box. There are only two choices: Poisson (the default) and Normal (OLS). Since the coefficients are being applied to the data, no over-dispersion correction is necessary (since it was probably used in calibrating the model).

7. Finally, the output file name is defined in the "Save predicted values" box.

For each zone, the routine will then take the appropriate variable from the input data set and apply the matching coefficient from trip generation coefficients file to produce a predicted estimate of the number of trips. To calculate this value, for the OLS model, the routine will use equation 13.2 above while for the Poisson model, the routine will use equation 13.6 above; for the latter, it will then raise the predicted log value to the power, e, to produce a prediction for the expected number of crime trips:

$$\lambda_i = e^{]Ln\,(\lambda i)\,]} \qquad (13.26)$$

If external trips are added, a new zone is created called EXTERNAL in the ID field that was indicated on the page. Then, the specified number of external trips is simply placed in that field with zeros being placed for the values of all the remaining variables in the file. By default, the output name for the predicted number of crimes will be called PREDORIG for an origin model and PREDDEST for a destination model. An example data set is available on the *CrimeStat* download page.

> Note: for a destination model, this "Make prediction" operation is not necessarily needed if the same data set is used for calibration and prediction. This step is primarily for the origin file

Balancing Predicted Origins and Destinations

After the origin model and destination model are calibrated and applied to a data set, the final step in trip generation is to ensure that the number of predicted origins equals the number of predicted destinations. This is necessary for the next stage of crime travel demand modeling - trip distribution. Since a trip has both an origin and a destination, the total number of origins has to equal the total number of destinations. This is an absolute condition for the trip distribution model to work; the routine will return an error message if the number of origins does not equal the number of destinations.

If the Poisson model is used for calibration, the routine ensures that the number of predicted trips equals the number of input trips. Further, if the calculation of external trips has been obtained by subtracting the total number of predicted origins from the total number of predicted destinations, and if the external trips are then added to the predicted origins, then most likely the total number of origins will equal the total number of destinations. However, because of rounding-off errors and inconsistent external trip estimates, it is possible that the sums are not equal.

Consequently, it is important to balance the predicted origins and destinations to ensure that no problems will occur in the trip distribution model. There are two ways to do this in *CrimeStat*. First, the number of predicted destinations is held constant and the number of predicted origins is adjusted to match this number. This is the default choice. Second, the number of predicted origins is held constant and the number of predicted destinations is adjusted to match this number.

The calculation is essentially a multiplier that is applied to each zone. If destinations are to be held constant, the multiplier is defined as:

$$M_j = \Sigma \text{ (Crimes by destinations, j)} / \Sigma \text{ (Crimes by origins, I)} \qquad (13.27)$$

and the predicted number of origins is multiplied by M_j. If, on the other hand, the origins are to be held constant, the multiplier is defined as:

$$M_i = \Sigma \text{ (Crimes by origins, I)} / \Sigma \text{ (Crimes by destinations, j)} \qquad (13.28)$$

and the predicted number of destinations is multiplied by M_i. The multiplication simply ensures that the sums of the predicted origins and predicted destinations are equal.

The third page in the trip generation model is the "Balance predicted origins & destinations" page. Figure 13.11 shows the setup for this page. The steps are as follows:

1. The box is checked indicating that it is a balancing operation.

2. The predicted origin file is input and the predicted origin variable is identified. In the example, the predicted origin file is called "PredictedOrigins.dbf" and the field with the predicted numbers was called PREDORIG.

3. The predicted destination file is input and the predicted destination variable is identified. In the example, the predicted destination file is called "PredictedDestinations.dbf" and the field with the predicted numbers was called PREDDEST.

Note that these files are input on this page and not on the primary or secondary file pages.

4. Next, the type of balancing is specified - Holding destinations constant (the default) or holding origins constant. In the example, the destinations are to be held constant.

5. Finally, the output file is specified. If the origins are to be adjusted, then only the origin file is saved. If the destinations are to be adjusted, then only the destination file is saved. In other words, the adjustment is applied to only one of the two predicted crime files. In the example, the file was named "AdjustedPredictedOrigins.dbf" (not shown) since the origin file was adjusted.

The output produces a new column with the adjusted values. Table 13.9 shows the origin output for the Baltimore data of the first 11 records. Once the balancing has been completed, the trip generation model is finished and the user can go on to the trip distribution model. In other words, the output file ensures that both the predicted origin file (crime productions) and predicted destination file (crime attractions) are balanced.

Strengths and Weaknesses of Regression Modeling of Trips

As mentioned earlier, the use of regression for producing the trip generation model has its strengths and weaknesses. The advantages are that, first, the approach is applicable to crime incidents. Unlike regular travel behavior, crime trips have to be inferred from police reports; one cannot conduct a household survey of offenders asking them about their crime travel. Thus, starting with counts of the number of crimes occurring in each zone and the number of crimes that originate from each zone, a model can be constructed.

Figure 13.11:

Balance Predicted Origins and Destinations Setup

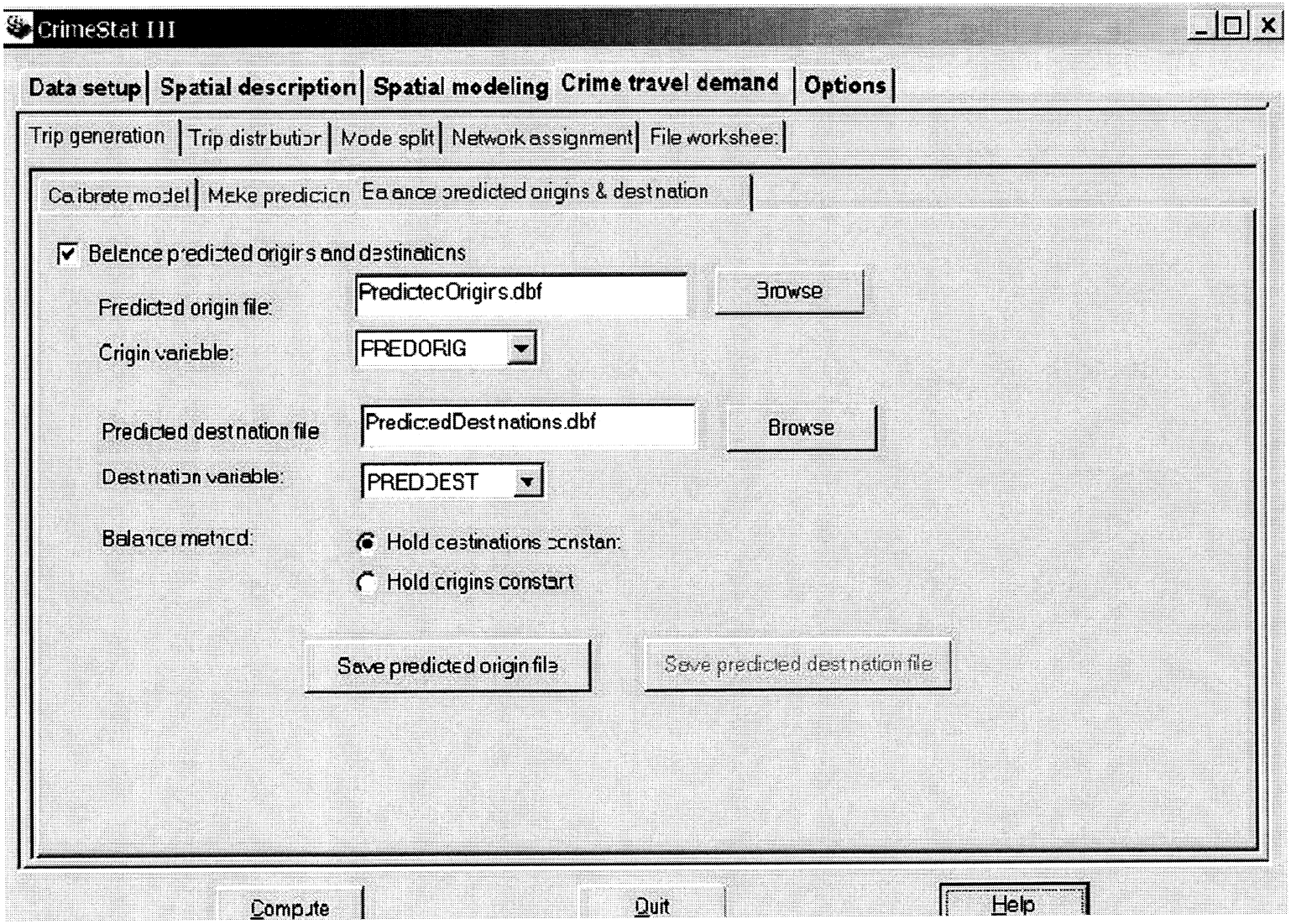


Table 13.9

Adjusted Data Should Have These Fields

Zone	PREDICTED	ADJORIGIN
0001	225.818482	225.850955
0002	187.527819	187.554785
0003	320.877458	320.923600
0004	75.096631	75.107430
0005	44.981775	44.988243
0006	32.574758	32.579442
0007	107.334835	107.350270
0008	74.683931	74.694671
0009	76.425236	76.436226
0010	34.183846	34.188762
0011	66.975803	66.985434
etc	etc	etc

Second, the use of a non-linear model, such as the Poisson, allows more complex fitting of crime counts. In the early 1970s when trip generation models were starting to be implemented in Metropolitan Planning Organizations around the U.S., the major type of regression modeling available was OLS. At that time, researchers could not demonstrate that this method was reliable in terms of predicting travel; we've discussed those reasons earlier in this chapter. However, with the availability of software for conducting Poisson and other non-linear models, that criticism is no longer applicable. The Poisson model is very 'well behaved' with respect to count data. It does not produce negative estimates. It requires high levels of an independent variable to produce a slight effect in the dependent variable, but that the level increases as the values of the independent variable increase. It maintains constancy between the sum of the input counts and the sum of the predicted counts. Non-linear models are much more realistic for modeling trips than OLS.

Third, the use of a multiple regression model allows multiple independent variables to be included. In our example, there were six and five variables respectively in the general origin and destination models. Trip tables, on the other hand, typically only have three or four independent predictors; it becomes too complicated to keep track of multiple conditions of predictor variables. Thus, a more complex and sophisticated model can be produced with a regression framework.

Fourth, and finally, a regression framework allows for complex interactions to be estimated. For example, the log of an independent variable can be defined. An interaction between two of the independent variables can be examined (e.g., median household income for those zones having a sizeable amount of retail employment). In the trip table approach, these interactions are implicit in the cell means. Thus, overall, the regression framework allows for a more complex model than is available with a trip table approach.

On the other hand, there are potential problems associated with a regression framework. First, the regression coefficients can be influenced by zone size. Since the model is estimating differences between zones (i.e., differences in the number of crimes as a function of differences in the values of the independent variables), zone size affects the level of those differences. With small zone sizes, there will be substantial differences between zones in both the independent and dependent variables. Conversely, large zone sizes will minimize within-zone differences, but will usually increase the estimate of the between-zone differences. The result could be an exaggeration of the effect of a variable that would not be seen with small zone geography. As we argued in chapter 12, one should choose the smallest zone geography that is practical in order to minimize this problem.

Second, a point that has been repeated again and again, these models are not behavioral explanations. They represent ecological correlations with crime trips. It's important to not try to convert these models into explanations of offender behavior. Too often, researchers have jumped to conclusions about individuals based on the relationships with environments and neighborhoods. It is important to not do this. This criticism, incidentally, applies both to the trip table as well as the regression approach to trip generation modeling.

The new generation of travel demand models is specifically behavioral and involves modeling the behavior of specific individuals. Probabilities are calculated based on individual choice and a micro-simulation routine can apply these probabilities to a large metropolitan area (RDC, 1995; Pas, 1996; Recker, 2000; Shifton et al, 2003). While this approach offers some definite theoretical advantages and is the subject of much current research, to date there has not been a demonstration that this approach is more accurate at predicting trips than the tradition trip-based travel demand model.

Summary

In summary, the trip generation model is a valuable tool for predicting the number of crimes that originate in each zone and the number of crimes that end in each zone. Even if the model is not behavioral, the model can be stable and useful for many years in the future. It is best thought of as a *proxy model* in which the variables in the models are proxies for conditions that are generating crimes, either in terms of environments that produce offenders or in terms of locations that attract them.

In the next chapter, we will examine the second stage in the travel demand model - trip distribution. In that stage, the predicted crime origins and the predicted crime destinations are linked to produce crime trips.

Endnotes for Chapter 13

1. There is also subjectivity in subdividing variables at an individual level. For example, household income levels can be subdivided in different ways. However, with aggregate data, all variables have to be subdivided arbitrarily whereas with individual level data, typically only income is done this way.

2. Some statisticians often refer to the number of *parameters* that have to be estimated in an equation, not just the number of independent variables. In an OLS model, for example, there are K+1 parameters that are estimated - coefficients for the K independent variables and a constant term. In this text, K refers to the number of independent variables, not estimated parameters.

3. It is possible to transform the independent variable into a non-linear predictor, for example by taking the log of the independent variable or raising it to some power (e.g., X^2). However, this won't solve the other problems associated with OLS, namely negative and non-summative predictions.

4. For example, to account for the skewed dependent variable, one or more of the independent variables have to be transformed with a non-linear operator (e.g., log or exponential term). When more than one independent variable is non-linear in an equation, the model is no longer easily understood. It may end up making reasonable predictions for the dependent variable, but it is not intuitive and not easily explained to non-specialists.

5. Note, Luc Anselin uses K for the number of parameters (coefficients + intercept) in Appendix C whereas we use it for the number of independent variables. Readers should be aware of this difference.

6. In the usual travel demand modeling, on the other hand, modelers usually adjust the predicted destinations since the origin data is more reliable. These numbers are obtained from the census or from the sample of households who are interviewed to produce a sample from which data on destinations are obtained.

7. An alternative might be to use cordon counts from major highways coming into the region and assume that crime trips represent a constant proportion of those trips. Thus, if the total number of estimated external highway trips increases by 5%, one could assume that the external trips also increase by 5%. While this is plausible, it is not necessarily an accurate estimate. Talk to your Metropolitan Planning Organization or the State Department of Transportation if you are interested in developing this type of model as you will need their estimates of external trips.

Chapter 14
Trip Distribution

In this chapter, the mechanics of the second crime travel demand modeling stage - trip distribution - is explained. *Trip distribution* is a model of the number of trips that occur between each origin zone and each destination zone. It uses the predicted number of trips originating in each origin zone (trip production model) and the predicted number of trips ending in each destination zone (trip attraction model). Thus, trip distribution is a model of travel between zones - trips or links. The modeled trip distribution can then be compared to the actual distribution to see whether the model produces a reasonable approximation.

Theoretical Background

The theoretical background behind the trip distribution module is presented first. Next, the specific procedures and tests are discussed with the model being illustrated with data from Baltimore County.

Logic of the Model

Trip distribution usually occurs through an allocation model that splits trips from each origin zone into distinct destinations. That is, there is a matrix which relates the number of trips originating in each zone to the number of trips ending in each zone. Figure 14.1 illustrates a typical arrangement. In this matrix, there are a number of origin zones, M, and a number of destination zones, N. The origin zones include ***all*** the destination zones but may also include some additional ones. The reasons that there would be different numbers of zones for the origin and destination models are that crime data for other jurisdictions are not available but that a sizeable number of crimes that occurred in the study jurisdiction are committed by individuals who lived those other jurisdictions. If it were possible to obtain crime data for the City of Baltimore, then it would be preferable to have the same number of zones for both the origin file and the destination file.

For example, with the Baltimore County data that are being used to illustrate the model, there are 325 destinations zones for Baltimore County while the origin zones include both the 325 in Baltimore County and 207 more from the adjacent City of Baltimore. As chapter 12 pointed out, the study area should extend beyond the modeling area until the origins of at least 95% of all trips ending in the study area are counted.

Each cell in the matrix indicates the number of *trips* that go from each origin zone to each destination zone. To use the example in figure 14.1, there were 15 trips from zone 1 to zone 2, 21 trips from zone 1 to zone 3, and so forth. Note that the trips are asymmetrical; that is, trips in one direction are different than trips in the opposite direction. To use the table, there were 15 trips from zone 1 to zone 2, but only 7 trips from zone 2 to zone 1.

The trips on the diagonal are *intra-zonal* trips, trips that originate and end in the same zone. Again, to use the example below, there were 37 trips that both originated and ended in zone 1, 53 trips that both originated and ended in zone 2, and so forth.

In such a model, constancy is maintained in that the number of trips originating from all origins zones must equal the number of trips ending in all destination zones. This is the fundamental balancing equation for a trip distribution. In equation form, it is expressed as:

$$\sum_{I=1}^{M} O_i = \sum_{j=1}^{N} D_j \qquad (14.1)$$

where the origins, O_i, are summed over M origin zones while the destinations, D_j, are summed over N destination zones. To use the example below, the total number of origins is equal to the total number of destinations, and is equal to 43,240

Figure 14.1

Example Crime Origin-Destination Matrix

Crime origin zone	1	2	3	4	5	.....	N	Σ
1	**37**	15	21	4	3	.....	12	346
2	7	**53**	14	0	4	.....	15	1050
3	12	9	**81**	7	6	.....	33	711
4	4	10	6	**12**	1	.....	0	84
5	8	7	28	2	**24**	.....	14	178
⋮	⋮	⋮	⋮	⋮	⋮		⋮	
M	12	5	43	3	10	.....	**92**	1466
Σ	153	276	1245	99	110		812	**43,240**

Crime destination zone

The balancing equation is implemented in a series of steps that include modeling the number of crimes originating in each zone, adding in trips originating from outside the study area (external trips), and statistically balancing the origins and destinations so that

equation 14.1 holds. This was done in the trip generation stage. But, it is essential that the step should have been completed for the trip distribution to be implemented.

Observed and Predicted Distributions

There are two trip distribution matrices that need to be distinguished. The first is the *observed* (or empirical) distribution. This is the actual number of trips that are observed traveling between each origin zone and each destination zone. In general, with crime data, such an empirical distribution would be obtained from an arrest record where the residence (or arrest) location of each offender is listed for each crime that the offender was charged with. In this case, the residence/arrest location would be considered the origin while the crime location would be considered the destination.

In chapter 12, it was mentioned that there is always uncertainty as to the true origin location of a crime incident, whether the offender actually traveled from the residence location to the crime location or even whether the offender was actually living at the residence location. But absent any alternative evidence, a meaningful distribution can still be obtained by simply treating the residence location as an approximate origin.

The observed distribution is calculated by simply enumerating the number of trips by each origin-destination combination. This is sometimes called a *trip link* (or trip pair). There are not any special statistics other than a simple two-way cross-classification table.

The second distribution, however, is a model of the trip distribution matrix. This is usually called the *predicted* distribution. In this case, a simple model is used to approximate the actual empirical distribution. The trips originating in each origin zone are allocated to destination zones usually on the basis of being directly proportional to attractions and inversely proportional to costs (or impedance).

Thus, a model of the trip distribution is produced that approximates the actual, empirical distribution. There are a number of reasons why this would be useful - to be able to apply the model to a different data set from which it was calibrated, to use the model for evaluating a policy intervention, or to use the model for forecasting future crime trip distribution. But, whatever the reason, it has to be realized that the model is not the observed distribution. There will always be a difference between the observed distribution from which a model is constructed and the resulting predicted distribution of the model. It is useful to compare the observed and predicted model because this allows a test of the validity of the impedance function. But, rarely, if ever, will the predicted distribution be identical to the empirical distribution.

Another way to think of this is that the actual distribution of crime trips is complex, representing a large number of different decisions on the part of offenders who do not necessarily use the same decision logic. The model, on the other hand, is a simple allocation on the basis of three or, sometimes, four variables. Almost by definition, it will be much simpler than the real distribution. Still, the simple model can often capture the most important characteristics of the actual distribution. Hence, modeling can be an

extremely useful analytical exercise that allows other types of questions to be asked that are not possible with just the observed distribution.

The Gravity Model

A model that is usually used for trip distribution is that of the *gravity function*, an application of Newton's fundamental law of attraction (Oppenheim, 1980; Field and MacGregor, 1987; Ortuzar and Willumsen, 2001). Much of the discussion below is also repeated in chapter 9 on journey to crime since there is a common theoretical basis. In the original Newtonian formulation, the attraction, F, between two bodies of respective masses M_1 and M_2, separated by a distance D, will be equal to

$$F = g \frac{M_1 M_2}{D^2} \qquad (14.2)$$

where g is a constant or scaling factor which ensures that the equation is balanced in terms of the measurement units (Oppenheim, 1980). As we all know, of course, g is the gravitational constant in the Newtonian formulation. The numerator of the function is the *attraction* term (or, alternatively, the attraction of M_2 for M_1) while the denominator of the equation, d^2, indicates that the attraction between the two bodies falls off as a function of their *squared* distance. It is an *impedance* (or resistance) term.

Social Applications of the Gravity Concept

The gravity model has been the basis of many applications to human societies and has been applied to social interactions since the 19th century. Ravenstein (1895) and Andersson (1897) applied the concept to the analysis of migration by arguing that the tendency to migrate between regions is inversely proportional to the squared distance between the regions. Reilly's 'law of retail gravitation' (1929) applied the Newtonian gravity model directly and suggested that retail travel between two centers would be proportional to the product of their populations and inversely proportional to the square of the distance separating them:

$$I_{ij} = \alpha \frac{P_i P_j}{D_{ij}^2} \qquad (14.3)$$

where I_{ij} is the interaction between centers I and j, P_i and P_j are the respective populations, D_{ij} is the distance between them raised to the second power and α is a balancing constant. In the model, the initial population, P_i, is called a *production* while the second population, P_j, is called an *attraction*.

Stewart (1950) and Zipf (1949) applied the concept to a variety of phenomena (migration, freight traffic, information) using a simplified form of the gravity equation

$$I_{ij} = K \frac{P_i P_j}{D_{ij}} \tag{14.4}$$

where the terms are as in equation 14.3 but the exponent of distance is only 1. Given a particular pattern of interaction for any type of goods, service or human activity, an optimal location of facilities should be solvable.

In the Stewart/Zipf framework, the two P's were both population sizes. However, in modern use, it is not necessary for the productions and attractions to be identical units (e.g., P_i could be population while P_j could be employment).

Trips as Interactions

It should be obvious that this interaction equation can be applied to trips from one area (zone) to another. Changing the symbols slightly, the total volume of trips from a particular origin zone, i, to a single location, j, is directly proportional to the product of the productions at *i* and the attractions at *j*, and inversely proportion to the impedance (or cost) of travel between the two zones

$$T_{ij} = \frac{\alpha P_i \beta A_j}{D_{ij}} \tag{14.5}$$

where P_i are the productions for zone I, A_j are the attractions zone j, α is a production constant, β is an attraction constant, and D_{ij} is the impedance (cost) of travel between zone ii and zone j.

Over time, the concept has been generalized and applied to many different types of travel behavior. For example, Huff (1963) applied the concept to retail trade between zones in an urban area using the general form of

$$A_{ij} = \alpha \frac{S_j^{\lambda}}{D_{ij}^{\rho}} \tag{14.6}$$

where A_{ij} is the number of purchases in location *j* by residents of location *I*, S_j is the attractiveness of zone j (e.g., square footage of retail space), D_{ij} is the distance between zones *I and j*, α is a constant, λ is the exponent of S_j, and ρ is the exponent of distance (Bossard, 1993). $D_{ij}^{-\rho}$ is sometimes called an *inverse distance* function. This differs from the traditional gravity function by allowing the exponents of the production from location *I*, the attraction from location *j*, and the distance between zones to vary.

Equation 14.6 is a *single constraint* model in that only the attractiveness of a commercial zone is constrained, that is the sum of all attractions for *j* must equal the total

attraction in the region. Again, it can be generalized to all zones by, first, estimating the total trips generated from one zone, *i*, to another zone, *j*,

$$T_{ij} = \alpha \frac{P_i^{\lambda} A_j^{\tau}}{D_{ij}^{\rho}} \qquad (14.7)$$

where T_{ij} is the interaction between two locations (or zones), P_i is productions of trips from zone I, A_j is the attractiveness of zone j, D_{ij} is the distance between zones I and j, λ is the exponent of P_i, τ is the exponent of A_j, ρ is the exponent of distance, and α is a constant.

Second, the total number of trips generated by a location, I, to all destinations is obtained by summing over all destination locations, j:

$$T_i = \alpha P_i^{\lambda} \Sigma (A_j^{\tau}/D_{ij}^{\rho}) \qquad (14.8)$$

and generalizing this to all zones, we get:

$$T_{ij} = \frac{\alpha P_i^{\lambda} \beta A_j^{\tau}}{D_{ij}^{\rho}} \qquad (14.9)$$

where α is a constant for the productions, P_i^{λ}, but β is a constant for the attractions, A_j^{τ}. This type of function is called a *double constraint* model because the equation has to be constrained by the number of units in both the origin and destination locations; that is, the sum of P_i over all locations must be equal to the total number of productions while the sum of A_j over all locations must be equal to the total number of attractions. Adjustments are usually required to have the sum of individual productions and attractions equal the totals (usually estimated independently).

Negative Exponential Distance Function

One of the problems with the traditional gravity formulation is in the measurement of travel impedance (or cost). For locations separated by sizeable distances in space, the gravity formulation can work properly. However, as the distance between locations decreases, the denominator approaches infinity. Consequently, an alternative expression for the interaction uses the negative exponential function (Hägerstrand, 1957; Wilson, 1970).

$$T_{ji} = A_j^{\beta} e^{(-\alpha D ij)} \qquad (14.10)$$

where T_{ji} is the attraction of location j for residents of location I, A_j is the attractiveness of location j, D_{ij} is the distance between locations i and *j*, β is the exponent of S_j, and *e* is the base of the natural logarithm (i.e., 2.7183...). Derived from principles of *entropy*

maximization, the latter part of the equation is a negative exponential function that has a maximum value of 1 (i.e., $e^{-0} = 1$) (Wilson, 1970). This has the advantage of making the equation more stable for interactions between locations that are close together. For example, Cliff and Haggett (1988) used a negative exponential gravity-type model to describe the diffusion of measles into the United States from Canada and Mexico. It has also been argued that the negative exponential function generally gives a better fit to urban travel patterns, particularly by automobile (Foot, 1981; Bossard, 1993;). Figure 14.2 shows a typical negative exponential function and one recommended for home-based work trips by the Transportation Research Board as a default value (NCHRP, 1995).

Note that by moving the distance term to the numerator, strictly speaking it no longer is an impedance term since impedance increases with distance. Rather it is a *discount* factor (or *disincentive*); the interaction is discounted with distance. Nevertheless, the term 'impedance' is still used, primarily for historical reasons.

There are other distance functions, as well. Chapter 9 explored some of these. For example, we are finding that, for crime trips, these other functions may produce better results than the negative exponential (e.g., the lognormal), primarily because many crimes are committed at short-to-moderate distances.

Travel Impedance

One of the biggest advances in this type of model has been to increase the flexibility of the denominator. In the traditional gravity model, the denominator is distance. This is a proxy for a *discount factor* (or cost); the farther two zones are from each other, the less likely there is to be interaction between them, all other things being equal. Conversely, the closer two zones are, the more likely there is to be interaction, all other things being equal.

Travel Time

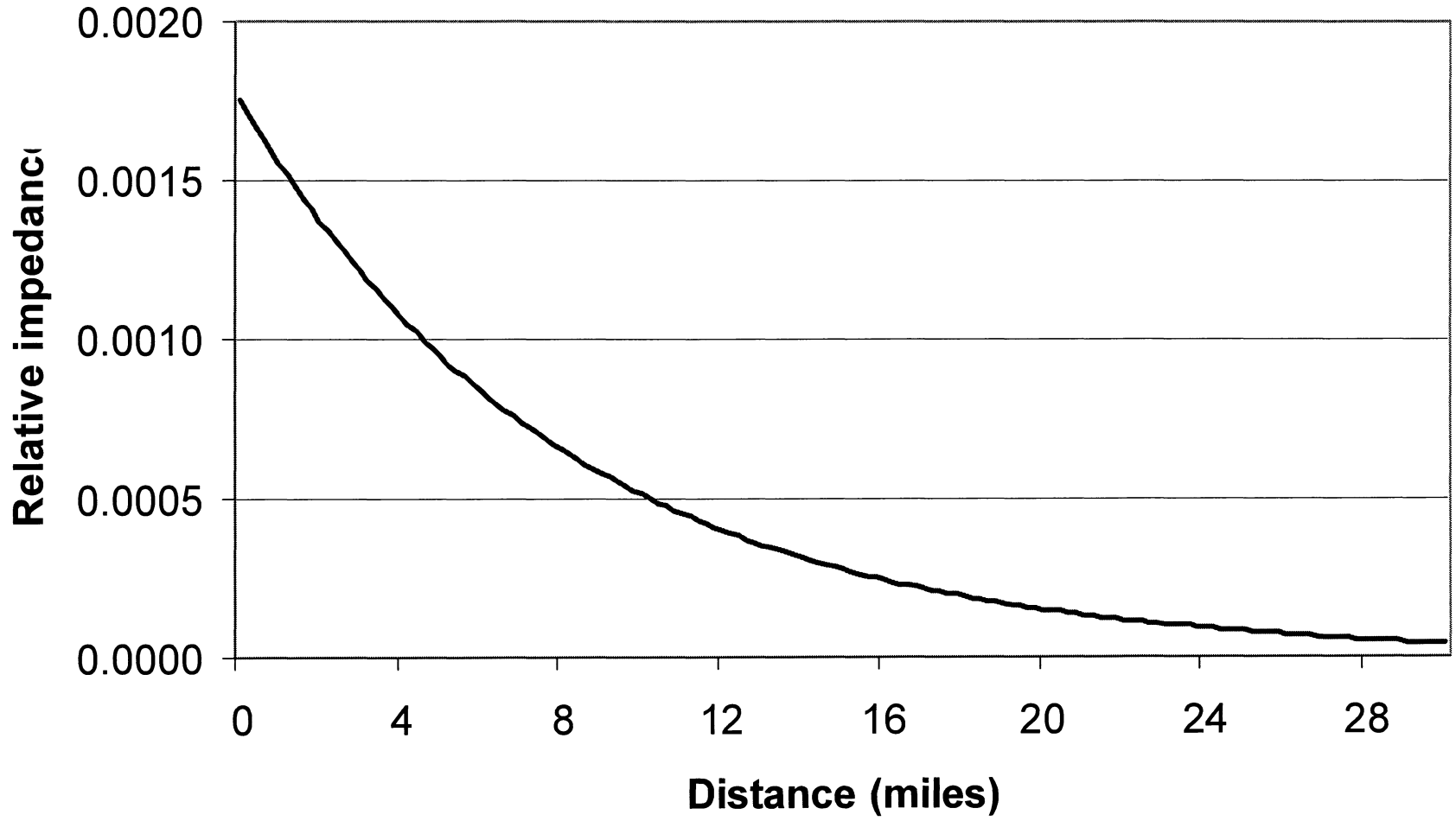
It has been realized, however, that distance is only an approximation to impedance. In real travel, travel time is a much better indicator of the *cost* of travel in that time varies by the time of day, day of week, and other factors. For example, travel across town in any metropolitan area is generally a lot easier at 3 in the morning, say, than at the peak afternoon rush period. The difference in travel time can vary as much as two-to-three times between peak and off-peak hours. Using only distance, however, these variations are never picked up because the distance between locations is invariant.

This realization has led to the concept of *travel impedance* which, in turn, has led to the concept of *travel cost*. 'Impedance' is the resistance (or discounting) in travel between two zones. Using travel time as an impedance variable, the longer it takes to travel between two zones, the less likely there will be interaction between them, all other things being equal. Conversely, a shorter travel time leads to greater interaction between zones, again, all other things being equal. Similarly, a travel route that shortens travel time will

Figure 14.2:

Default Home-Based Work Trip Impedance

(Source: National Cooperative Highway Research Program 365, 1995)



generally be selected over one that takes longer even if the first one is longer in distance. For example, it's been documented that people will change work locations that are farther from their home if traveling to the new work location takes less time (e.g., traveling in the 'opposite' direction to the bulk of traffic; Wachs, Taylor, Levine & Ong, 1993).

If travel time is a critical component of travel, why then don't offenders commit more crimes at, say, 3 in the morning than at the peak afternoon travel times? Since the impedance is less at 3 in the morning than at, say, 5 in the afternoon, wouldn't the model predict more trips occurring in the early morning hours than actually occur in those hours? The answer has to do with the numerator of the gravity equation and not just the denominator. At 3 in the morning, yes, it is easier to travel between two locations, at least by personal automobile (not by bus or train when those services are less frequent). But the attraction side of the equation is also less strong at 3 in the morning. For a street robber, there are fewer potential 'victims' on the street at 3 in the morning than in the late afternoon. For a residential burglar, there is more likely to be someone at home while they burgle at night than in the afternoon. The travel time component is only one dimension of the likelihood of travel between two locations. The distribution of opportunities and other costs can alter the likelihood considerably.

Nevertheless, shifting to an impedance function allows a travel model to better replicate actual travel conditions. Most travel demand models used by transportation planners use an impedance function, rather than a distance function.[1] Distance would only be meaningful if the standards were invariant with respect to time (e.g., a model calculated over an entire year, 24 hours a day). As will be demonstrated in chapter 16 on network assignment, a travel time calculation leads to a very different network allocation than a distance calculation. For example, if distance is used as an impedance variable, then the shortest trips will rarely take the freeways because travel to and from a freeway usually makes a trip longer than a direct route between an origin and a destination. But as most people understand, taking a freeway to travel a sizeable distance is usually a lot quicker than traversing an urban arterial system with many traffic lights, stop signs, crossing pedestrians, cross traffic from parking lots and shopping malls, and other urban 'obstacles'. Today, the use of distance in travel demand modeling has virtually been dropped by most transportation planners.

Travel Cost

An even better concept of impedance is that of *travel cost* (sometimes called *generalized cost*) which incorporates real and perceived costs of travel between two locations. Travel time is one component of travel cost in that there is an implicit cost to the trip (e.g., an hourly wage or price assigned). In this case, two different individuals will value the time for a trip differently depending on their hourly 'wage'. For example, for an individual who prices his/her travel at $100 an hour, the per minute cost is $1.67. For another individual who prices his/her travel at $12 an hour, the per minute cost is 20¢. These relative prices assigned to travel will substantially affect individual choices in travel modes and routes. For instance, these two hypothetical individuals will probably use a

different travel mode in getting from an airport to a hotel on a trip; the former will probably take a taxi whereas the latter will probably take a bus or train (if available).

But cost involves other dimensions that need to be considered. There are real operating costs in the use of a vehicle - fuel, oil, maintenance, insurance. Many travel studies have suggested that drivers incorporate these costs as part of their implicit hourly travel price (Ortuzar and Willumsen, 2001; 323-327). But, there are also real, 'out-of-pocket' costs such as parking or toll costs. Parking is particularly a major expense for intra-urban driving behavior. In many built-up business districts, parking costs can be considerable, for example as much as $40 a day in major metropolitan centers. In most busy commercial areas, there are some parking costs, if only at on-street parking meters. Thus, a travel cost model needs to incorporate these real costs as the out-of-pocket costs may overwhelm the implicit value of the travel time. For example, an offender who lives 10 minutes from the downtown area by car would probably not drive into the downtown to commit a robbery since that individual will have to bear the price of parking. There are lots of well known stories that circulate about bank robbers who are caught because they incur parking tickets while committing their crime. How often this has occurred is not known from any study that I'm aware, but the story line is cognizant of the actual costs of travel that must be incurred as part of travel.

In addition to real costs are perceived costs. For transit users, particularly, these perceived costs affect the ease and time of travel. One of the standard questions in travel surveys for transit users is the time it takes to walk from their home to the nearest bus stop or intra-urban rail system (if available) and from the last transit stop to their final destination; the longer it takes to access the transit system, the less likely an individual will use it. Similarly, transfers between buses or trains decrease the likelihood of travel by that mode, almost in proportion to the number of transfers. The reason is the difficulty in getting out of one bus or train and into another. But, the time between trains adds an implicit travel cost; the longer the wait between buses, the less likely that mode will be used by travelers. In short, ease of access and convenience are positive incentives in using a mode or a route while difficulty in accessing it, lack of convenience, and even fear of being vulnerable to crime will decrease the likelihood of using that mode or route.[2]

If the concept is expanded to that of an offender, there are other perceived costs that might affect travel. One obvious one is the likelihood of being caught. It may be easy for one offender to travel to a upscale, high visibility shopping area, but if there are many police and security guards around, the individual is more likely to be caught. Hence, that likelihood (or, more accurately, an assumption that the offender makes about that likelihood since he/she doesn't really know what is the real likelihood) is liable to affect the choice of a destination and, possibly, even a route.

Another perceptual component affecting a likely choice is the reliability of the transportation mode. Many offenders are poor and don't have expensive, well maintained vehicles. If the vehicle is not capable of higher speeds or is even likely to break down while an offence is being committed, that vehicle is not liable to be used in making a trip or the choice of destination may be altered. It is well known that many offenders steal vehicles

for use in a crime. Fears about not being identified are clearly a major factor in those decisions, but the reliability of their own vehicles may also be a factor.

Thus, in short, a more realistic model of the incentive or disincentive to make a trip between two locations requires a complex function that weights a number of factors affecting the cost of travel - the travel time, implicit operating costs, out-of-pocket costs, and perceived costs. Many travel demand models used by Metropolitan Planning Organizations use such a function, usually under the label of 'generalized cost'. The more complex the pricing structure for parking and travel within a metropolitan area, the more likely a generalized cost function will provide a realistic model of trip distribution.

Travel Utility

The final concept that is introduced in defining impedance is that of *travel utility*. 'Utility' is an individual concept, rather than a zonal one. Also, it is the flip side of cost (i.e., higher cost is associated with less utility). A generalized cost function calculates the objective and average perceived costs of travel between two zones. But the utility of travel for an individual is a function of both those real costs and a number of individual characteristics that affect the value placed on that travel. Thus, two individuals living in the same zone (perhaps even living next door to each other) who travel to the same destination location may 'price' their trip very differently. Aside from income differences which affects the average hourly 'wage', there may be differences due to convenience, attractiveness, or a host of other factors. Other factors are more idiosyncratic. For example, a trip by a gang member into another gang's 'turf' might be expected to increase the perceived costs to the individual of traveling to that location, above and beyond any objective cost factors. Alternatively, a trip to a location where a close friend or relative is located might decrease the perceived cost of travel to that zone. In other words, there are both objective costs as well as subjective costs in travel between two zones.

The concept of utility may be less useful for crime analysis than for general travel behavior. For one thing, since the concept is individual, it can only be identified by individual surveys (Domencich and McFadden, 1975). For crime analysis, this makes it virtually impossible to use since it is very difficult to interview offenders, at least in the United States. In addition, the mathematics required for articulating a utility function are difficult since utility functions have a very complex form, usually involving the binomial or multinomial logit function with a Weibull error term. In the next chapter, brief mention is made of this type of model.

But, for completeness sake, we need to understand that the likelihood or disincentive to travel between two locations is a function of individual characteristics as well as objective travel cost components.

Impedance Function

Thus, for a zonal type model, we can leave the gravity function as a generalized impedance function. For travel between any one zone and all other zones, we have:

$$T_i = \alpha P_i^{\lambda} \Sigma (A_j^{\tau}/I_{ij}) \quad (14.11)$$

where the number of trips from zone i to all other zones, j, is a function of the productions at zone i and the relative attraction of any one zone, j, to the impedance of that zone for i, I_{ij}. The impedance function, I_{ij} , is some declining function of cost for travel between two zones. It does not have to be any particular form and can be (and usually is) a non-linear function. The costs can be in terms of distance, travel time, speed (which is converted into travel time) or general costs. The greater the separation between two zones (i.e., the higher the impedance), the less likely there will be a trip between them. Generalizing this to all zones, we get:

$$T_{ij} = \alpha P_i^{\lambda} \beta A_j^{\tau} I_{ij} \quad (14.12)$$

where P_i is the production capacity of zone I, A_j is the attraction of zone j, I_{ij}, is a generalized function that discounts the interaction with increasing separation in distance, time, or cost, α and β are constants that are applied to the productions and attractions respectively, and λ and τ are 'fine tuning' exponents of the productions and attractions respectively. This is the gravity function that we will estimate in the *CrimeStat* trip distribution model.

Alternative Models: Intervening Opportunities

There are alternative allocations procedures to the gravity model. One well known one is that of *intervening opportunities*. Stouffer (1940) modified the simple gravity function by arguing that the attraction between two locations was a function not only of the characteristics of the relative attractions of two locations, but of intervening opportunities between the locations. His hypothesis "..assumes that there is no necessary relationship between mobility and distance... that the number of persons going a given distance is directly proportional to the number of opportunities at that distance and inversely proportional to the number of intervening opportunities" (Stouffer, 1940, p. 846). This model was used in the 1940s to explain interstate and inter-county migration (Bright and Thomas, 1941; Isbell, 1944; Isard, 1979). Using the gravity type formulation, this can be written as:

$$A_{ji} = \alpha \frac{S_j^{\beta}}{\Sigma(S_k^{\xi})\ D_{ij}^{\lambda}} \quad (14.13)$$

where A_{ji} is the attraction of location j by residents of location I, S_j is the attractiveness of zone j, S_k is the attractiveness of all other locations that are *intermediate* in distance between locations I and j, D_{ij} is the distance between zones *I* and *j*, β is the exponent of S_j, ξ is the exponent of S_k, and λ is the exponent of distance. While the intervening opportunities are implicit in equation 14.7 in the exponents, β and λ, and coefficient, α, equation 14.13 makes the intervening opportunities explicit. The importance of the concept

is that travel between two locations becomes a complex function of the spatial environment of nearby areas and not just of the two locations.

In practice, in spite of its more intuitive theoretical model, the intervening opportunities model does not improve prediction much beyond that of the gravity model since it includes the attractions associated with the destination zones. Also, it is a more difficult model to estimate since the attractions of all other zones must be considered for each zone pair (origin-destination combination). Consequently, it is rarely used in actual practice (Ortuzar and Willumsen, 2001; Zhao et al, 2001).

Another alternative method was conducted by Porojan (2000) in applying the gravity model to international trade flow. He added a spatial autocorrelation component in addition to impedance and obtained a slightly better fit than the pure gravity function. However, whether this approach would improve the fitting of intra-regional crime travel patterns in still unknown. Nevertheless, this and other approaches might improve the predictability of a gravity function for intra-urban crime travel.

Method of Estimation

The *CrimeStat* trip distribution model implements equation 14.12. The specific details are discussed below, but the model is iterative. The steps are as follows:

1.Depending on whether a singly constrained or doubly constrained model is to be estimated, it starts with a initial guess of the values for α or β (or both for a doubly constrained model). Table 14.1 illustrates the three models.

Table 14.1

Three Methods of Constraining the Gravity Model

Single constraint

Constrain origins

$$T_{ij} = \alpha P_i^{\lambda} A_j^{\tau} I_{ij}$$

Constrain destinations

$$T_{ij} = P_i^{\lambda} \beta A_j^{\tau} I_{ij}$$

Double constraint

Constrain both origins and destinations

$$T_{ij} = \alpha P_i^{\lambda} \beta A_j^{\tau} I_{ij}$$

2. The routine proceeds to estimate the value for each cell in the origin-destination matrix (see figure 14.1 above) using the existing estimates for α and β.

3. The routine then sums the rows and columns in the matrix. Then, depending on whether a single- or double-constraint model is to be estimated and, if a single-constraint, whether origins or destinations are to be held constant, it then calculates the ratio of the summed value (row or column or both) to the initial row or column sum. The inverse of that ratio is the subsequent estimate for α or β (or both for a double-constrained model).

4. The routine repeats steps 2 and 3 until the changes from one iteration to the next are very small.

5. The last estimate of α or β (or both for a double-constrained model) is taken as the final values of these parameters.

6. Once the parameters have been estimated, the model can be applied to the calibration data set or to another data set. Note that the parameters are row or column specific (or both). That is, in the 'constrain origins' model, there is a separate coefficient for each row. In the 'constrain destinations' model, there is a separate coefficient for each column. In the 'constrain both origins and destinations', there is a separate coefficient for each cell (row-column combination).

7. A comparison can be made between the observed distribution and the predicted (modeled) distribution. Because most origin-destination matrices are very large, the vast majority of cells will have zero in them. Thus, a chi-square test would be inappropriate. Instead, a comparison of the *trip length* distribution is made using two different statistics - a coincidence ratio and the Komologorov-Smirnov Two-sample statistic. Details are provided below.

CrimeStat III Trip Distribution Routines

Next, we examine the actual tools that are available in the *CrimeStat* trip distribution module. The tools are illustrated with examples from Baltimore County.

The *CrimeStat* trip distribution module includes one setup screen and five routines that implement the model:

1. **Calculate observed origin-destination distribution**. If there is a file available with the coordinates for individual origins and destinations (e.g., an arrest record), this routine will calculate the empirical trip distribution matrix;

2. **Calibrate impedance function**. If there is a file available with the coordinates for individual origins and destinations, this routine will calibrate an empirical impedance function.

3. **Setup origin-destination model**. This screen allows the user to define the parameters of a trip distribution (origin-destination) model with either a mathematical or empirical impedance function.

4. **Calibrate origin-destination model**. This routine calibrates the parameters of the trip distribution model (equation 14.12) using the parameters defined on the setup page.

5. **Apply predicted origin-destination model**. This routine applies the estimated parameters to a data set. The data set can be either the calibration file or another file.

6. **Compare observed and predicted origin-destination trip lengths.** This routine compares the trip lengths from the observed (empirical) trip distribution with that predicted by the model. Comparison are made graphically, by a coincidence ratio, by the Komologorov-Smirnov Two-Sample test, and by a Chi square test on the most frequent trip links.

Each of these routines are described in detail below. Figure 14.3 shows a screen shot of the trip distribution module.

Describe Origin-Destination Trips

An empirical description of the actual trip distribution matrix can be made if there is a data set that includes individual origin and destination locations. The user defines the origin location and the destination location for each record and a set of zones from which to compare the individual origins and destinations. The routine matches up each origin location with the nearest zone, each destination location with the nearest zone, and calculates the number of trips from each origin zone to each destination zone. This is an *observed* distribution of trips by zone.

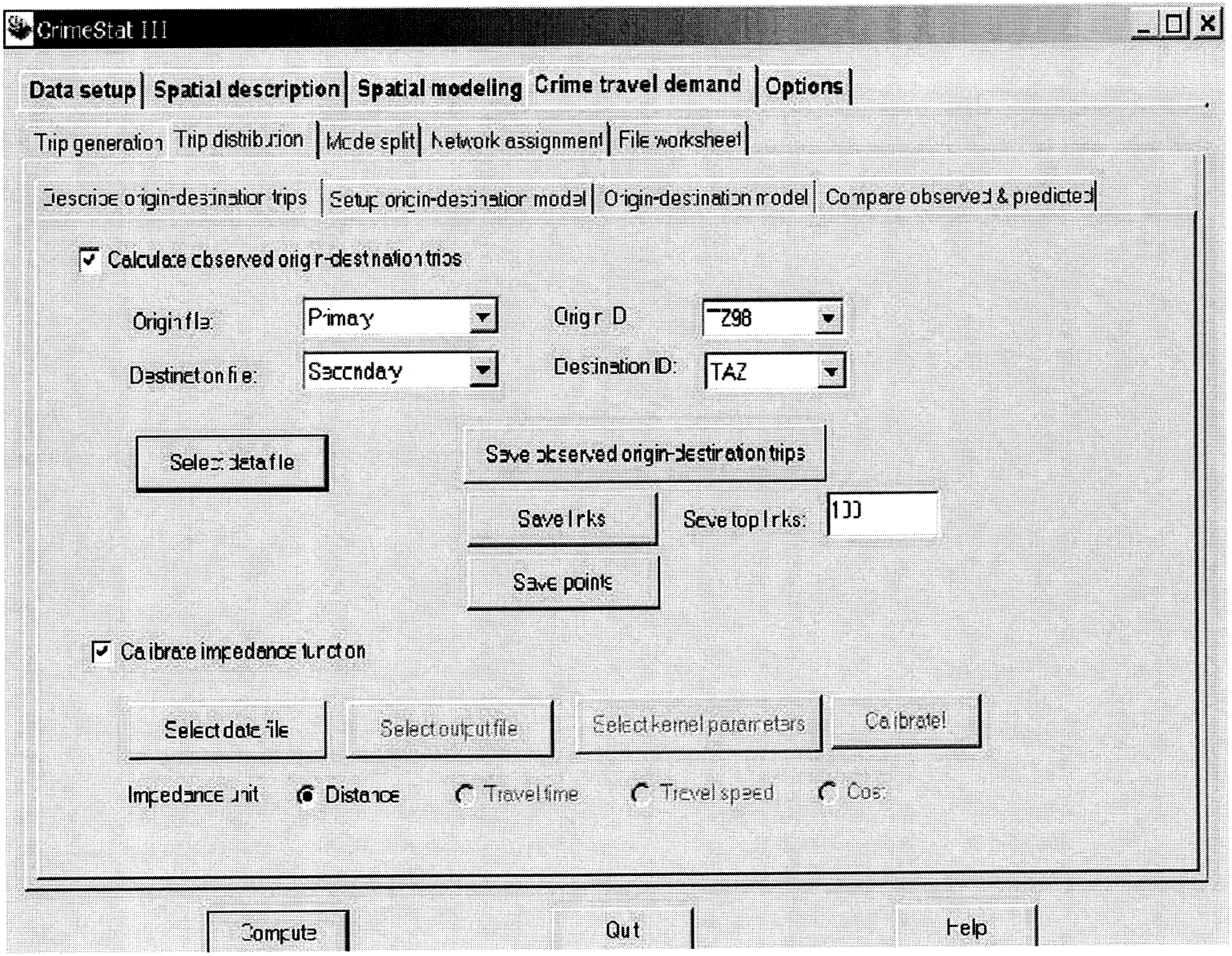
The steps in running the model are as follows:

1. **Calculate observed origin-destination trips**. Check if an empirical origin-destination trip distribution is to be calculated.

2. **Origin file**. The origin file is a list of origin zones with a single point representing the zone (e.g., the centroid). It must be input as either the primary or secondary file. Specify whether the data file is the primary or secondary file.

Figure 14.3:

Trip Distribution Module



Origin ID. Specify the origin ID variable in the data file (e.g., CensusTract, Block, TAZ).

3. **Destination file.** The destination file is a list of destination zones with a single point representing the zone (e.g., the centroid). It must be input as either the primary or secondary file. Specify whether the data file is the primary or secondary file.Specify the destination ID variable in the data file (e.g., CensusTract, Block, TAZ).

Note: all destination ID's should be in the origin zone file and must have the same names and both should be character (string) variables.

4. **Select data file**. The data set must have individual origin and destination locations. Each record must have the X/Y coordinates of an origin location and the X/Y coordinates of a destination location. For example, an arrest file might list individual incidents with each incident having a crime location (the destination) and a residence or arrest location (the origin). Select the file that has the X and Y coordinates for the origin and destination locations. *CrimeStat* can read ASCII, dbase '.dbf', ArcView '.shp' and MapInfo 'dat' files. Select the tab and specify the type of file to be selected. Use the browse button to search for the file. If the file type is ASCII, select the type of data separator (comma, semicolon, space, tab) and the number of columns.

Variables. Define the file which contains the X and Y coordinates for both the origin (residence) and destination (crime) locations.

Column. Select the variables for the X and Y coordinates respectively for *both* the origin and destination locations (e.g., Lon, Lat, HomeX, HomeY, IncidentX, IncidentY.) Both locations must be defined for the routine to work.

Missing values. Identify whether there are any missing values for these four fields (X and Y coordinates for both origin and destination locations). By default, *CrimeStat* will ignore records with blank values in any of the eligible fields or records with non-numeric values (e.g.,alphanumeric characters, #, *). Blanks will always be excluded unless the user selects <none>. There are 8 possible options:

1. <blank> fields are automatically excluded. This is the default
2. <none> indicates that no records will be excluded. If there is a blank field, *CrimeStat* will treat it as a 0
3. 0 is excluded
4. –1 is excluded
5. 0 and –1 indicates that both 0 and -1 will be excluded
6. 0, -1 and 9999 indicates that all three values (0, -1, 9999) will be excluded

Any other numerical value can be treated as a missing value by typing it (e.g., 99)Multiple numerical values can be treated as missing values by typing them, separating each by commas (e.g., 0, -1, 99, 9999, -99).

Type of coordinate system and data units. The coordinate system and data units are listed for information. If the coordinates are in longitudes and latitudes, then a spherical system is being used and data units will automatically be decimal degrees. If the coordinate system is projected (e.g., State Plane, Universal Transverse Mercator – UTM), then data units could be either in feet (e.g., State Plane) or meters (e.g., UTM.).

5. **Table output**. The full origin-destination matrix is output as a table to the screen including summary file information and:

    1. The origin zone (ORIGIN)
    2. The destination zone (DEST)'
    3. The number of observed trips (FREQ)

6. **Save observed origin-destination trips.** If specified, the full origin-destination matrix output is saved as a 'dbf' file named by the user. The file output includes:

    1. The origin zone (ORIGIN)
    2. The destination zone (DEST)
    3. The X coordinate for the origin zone (ORIGINX)
    4. The Y coordinate for the origin zone (ORIGINY)
    5. The X coordinate for the destination zone (DESTX)
    6. The Y coordinate for the destination zone (DESTY)
    7. The number of trips (FREQ)

    Note: each record is a unique origin-destination combination. There are M x N records where M is the number of origin zones (including the external zone) and N is the number of destination zones.

7. **Save links.** The top observed origin-destination trip links can be saved as separate **line** objects for use in a GIS. Specify the output file format (*ArcView* '.shp', *MapInfo* '.mif' or *Atlas*GIS* '.bna') and the file name.

    Save top links. Because the output file is very large (number of origin zones x number of destination zones), the user can select a sub-set of zone combinations with the most observed trips. Indicating the top K links will narrow the number down to the most important ones. The default is the top 100 origin-destination combinations. Each output object is a line from the origin zone to the destination zone with an ODT prefix. The prefix is placed before the output file name. The line graphical output for each object includes:

1. An ID number from 1 to K, where K is the number of links output (ID)
2. The feature prefix (ODT)
3. The origin zone (ORIGIN)
4. The destination zone (DEST)
5. The X coordinate for the origin zone (ORIGINX)
6. The Y coordinate for the origin zone (ORIGINY)
7. The X coordinate for the destination zone (DESTX)
8. The Y coordinate for the destination zone (DESTY)
9. The number of observed trips for that combination (FREQ)
10. The distance between the origin zone and the destination zone.

Save points. Intra-zonal trips (trips in which the origin and destination are the same zone) can be output as separate **point** objects as an *ArcView* '.shp', *MapInfo* '.mif' or *Atlas*GIS* '.bna' file. Again, the top K points are output (default=100). Each output object is a point representing an intra-zonal trip with an ODTPOINTS prefix. The prefix is placed before the output file name.

The point graphical output for each object includes:

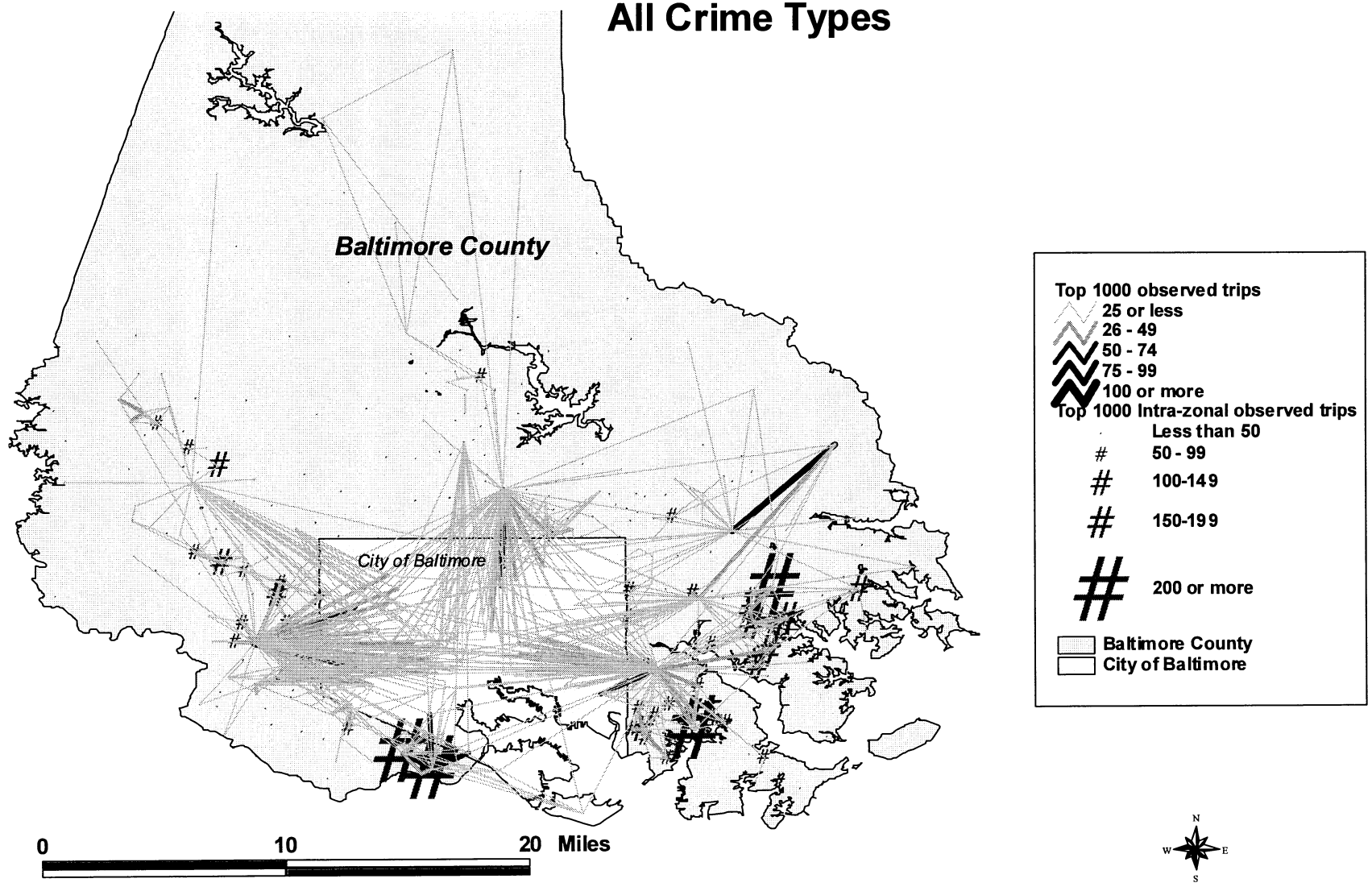
1. An ID number from 1 to K, where K is the number of links output (ID)
2. The feature prefix (POINTSODT)
3. The origin zone (ORIGIN)
4. The destination zone (DEST)
5. The X coordinate for the origin zone (ORIGINX)
6. The Y coordinate for the origin zone (ORIGINY)
7. The X coordinate for the destination zone (DESTX)
8. The Y coordinate for the destination zone (DESTY)
9. The number of observed trips for that combination (FREQ)

Example of Observed Distribution from Baltimore County

Figure 14.4 shows the output of the top 1000 links for the observed trip distribution from a sample of 41,974 records for incidents committed between 1993 and 1997. The zonal model used was that of traffic analysis zones (TAZ). These were discussed in chapter 12. Because there are a large number of links (532 origin zones by 325 destination zones), the top 1000 were taken. These accounted for 19,615 crime trips (or 46.7% of all trips). A larger number of links could have been selected, but the map would have become more cluttered. Of the 19,615 trips that are displayed in the map, 7,913 or 40.3% are intra-zonal trips. These were output by the routine as points and have been displayed as circles with the size proportional to the number of trips. The remaining 11,702 trip links were output by the routine as lines and are displayed with the thickness and strength of color of the line being proportional to the number of trips.

Figure 14.4:

Observed Baltimore County Crime Trips: 1993-1997
Top 1000 Links
All Crime Types



There are several characteristics of the trip pattern that should be noted. First, the intra-zonal trips tend to concentrate on the eastern part of Baltimore County. This is an area that is relatively poor with a high number of public housing projects. This suggests that there are a lot of intra-community crimes being committed in these locations. Second, the zone-to-zone pattern, on the other hand, tends to concentrate at five different locations relatively close to border with the City of Baltimore. These five locations are all major shopping malls. Third, the origins for those trips to the shopping mall tend to come from within the City of Baltimore. Fourth, in general, the locations with high intra-zonal trips do not have a large number of zone-to-zone trips. However, there is one exception in the southwest corner of the county.

In other words, the observed distribution of crime trips is complex, but with several patterns being shown. A lot of crime trips occur over very short distances. But there is also a convergence of many crime trips on major shopping malls in the County.

Calibrate Impedance Function

This routine allows the calibration of an approximate travel impedance function based on actual trip distributions. It is used to describe the travel impedance in distance only of an actual sample (the calibration sample). Unlike the remaining routines in this section, the "Calibrate impedance function cannot use travel time, or cost. A file is input which has a set of incidents (records) that include both the X and Y coordinates for the location of the offender's residence (origin) and the X and Y coordinates for the location of the incident that the offender committed (destination.)

The routine estimates a travel distance function using a one-dimensional kernel density method. See the details in chapter 9. Essentially, for each record, the separation between the origin location and the destination location is calculated and is represented on a distance scale. The maximum impedance is calculated and divided into a number of intervals; the default is 100 equal sized intervals, but the user can modify this. For each impedance point calculated, a one-dimensional kernel is overlaid. For each interval, the values of all kernels are summed to produce a smooth function of travel impedance. The results are saved to a file that can be used for the origin-destination model.

Note, however, that this is an empirical distribution and represents the combination of origins, destinations, and costs. It is not necessarily a good description of the impedance (cost) function by itself. Many of the mathematical functions produce a better fit than the empirical impedance function.

The steps in calculating an empirical impedance function are as follows:

1. **Select data file for calibration**. Select the file that has the X and Y coordinates for the origin and destination locations. *CrimeStat* can read ASCII, dbase '.dbf', ArcView '.shp' and MapInfo 'dat' files. Select the tab and select the type of file to be selected. Use the browse button to search for the

file. If the file type is ASCII, select the type of data separator (comma, semicolon, space, tab) and the number of columns.

Variables. Define the file which contains the X and Y coordinates for both the origin (residence) and destination (crime) locations

Columns. Select the variables for the X and Y coordinates respectively for *both* the origin and destination locations (e.g., Lon, Lat, HomeX, HomeY, IncidentX, IncidentY.) Both locations must be defined for the routine to work.

Missing values. Identify whether there are any missing values for these four fields (X and Y coordinates for both origin and destination locations). By default, *CrimeStat* will ignore records with blank values in any of the eligible fields or records with non-numeric values (e.g.,alphanumeric characters, #, *). Blanks will always be excluded unless the user selects <none>. There are 8 possible options:

1. <blank> fields are automatically excluded. This is the default
2. <none> indicates that no records will be excluded. If there is a blank field, *CrimeStat* will treat it as a 0
3. 0 is excluded
4. –1 is excluded
5. 0 and –1 indicates that both 0 and -1 will be excluded
6. 0, -1 and 9999 indicates that all three values (0, -1, 9999) will be excluded

Any other numerical value can be treated as a missing value by typing it (e.g., 99)Multiple numerical values can be treated as missing values by typing them, separating each by commas (e.g., 0, -1, 99, 9999, -99).

Type of coordinate system and data units. Select the type of coordinate system. If the coordinates are in longitudes and latitudes, then a spherical system is being used and data units will automatically be decimal degrees. If the coordinate system is projected (e.g., State Plane, Universal Transverse Mercator – UTM), then data units could be either in feet (e.g., State Plane) or meters (e.g., UTM.) Directional coordinates are not allowed for this routine.

2. **Select Kernel Parameters**. There are five parameters that must be defined.

Method of interpolation. There are fives types of kernel distributions that can be used to estimate point density:

1. The **normal** kernel overlays a three-dimensional normal distribution over each point that then extends over the area defined by the reference file. This is the default kernel function.
2. The **uniform** kernel overlays a uniform function (disk) over each point that only extends for a limited distance.
3. The **quartic** kernel overlays a quartic function (inverse sphere) over each point that only extends for a limited distance.
4. The **triangular** kernel overlays a three-dimensional triangle (cone) over each point that only extends for a limited distance.
5. The **negative exponential** kernel overlays a three dimensional negative exponential function ('salt shaker') over each point that only extends for a limited distance

The methods produce similar results though the normal is generally smoother for any given bandwidth.

***Choice of* bandwidth**. The kernels are applied to a limited search distance, called 'bandwidth'. For the normal kernel, bandwidth is the standard deviation of the normal distribution. For the uniform, quartic, triangular and negative exponential kernels, bandwidth is the radius of a circle defined by the surface. For all types, larger bandwidth will produce smoother density estimates and both adaptive and fixed bandwidth intervals can be selected.

1. ***Fixed bandwidth***. A fixed bandwidth distance is a fixed interval for each point. The user must define the interval, the interval size, and the distance units by which it is calculated (miles, nautical miles, feet, kilometers, meters.) The default bandwidth setting is fixed with intervals of 0.25 miles each. The interval size can be changed.

2. ***Adaptive bandwidth***. An adaptive bandwidth distance is identified by the minimum number of other points found within a symmetrical band drawn around a single point. A symmetrical band is placed over each distance point, in turn, and the width is increased until the minimum sample size is reached. Thus, each point has a different bandwidth size. The user can modify the minimum sample size. The default for the adaptive bandwidth is 100 points.

Specify Interpolation Bins. The interpolation bins are defined in one of two ways:

1. By the number of bins. The maximum distance calculated is divided by the number of specified bins. This is the default with 100 bins. The user can change the number of bins.

2. By the distance between bins. The user can specify a bin width in miles, nautical miles, feet, kilometers, and meters.

3. **Output (Areal) Units**. Specify the density units as points per mile, nautical mile, foot, kilometer, or meter. The default is points per mile.

4. **Calculate Densities or Probabilities**. The density estimate for each cell can be calculated in one of three ways:

   1. **Absolute densities**. This is the number of points per grid cell and is scaled so that the sum of all grid cells equals the sample size.

   2. **Relative densities**. For each grid cell, this is the absolute density divided by the grid cell area and is expressed in the output units (e.g., points per square mile)

   3. **Probabilities**. This is the proportion of all incidents that occur in the grid cell. The sum of all grid cells equals a probability of 1. Unlike the Jtc calibration routine, this is the default. In most cases, a user would want a proportional (probability) distribution as the relative differences in impedance for different costs are what is of interest.

      Select whether absolute densities, relative densities, or probabilities are to be output for each cell. The default is probabilities.

5. **Select Output File**. The output *must* be saved to a file. *CrimeStat* can save the calibration output to either a dbase 'dbf' or ASCII text 'txt' file.

6. **Calibrate!** Click on 'Calibrate!' to run the routine. The output is saved to the specified file upon clicking on 'Close'.

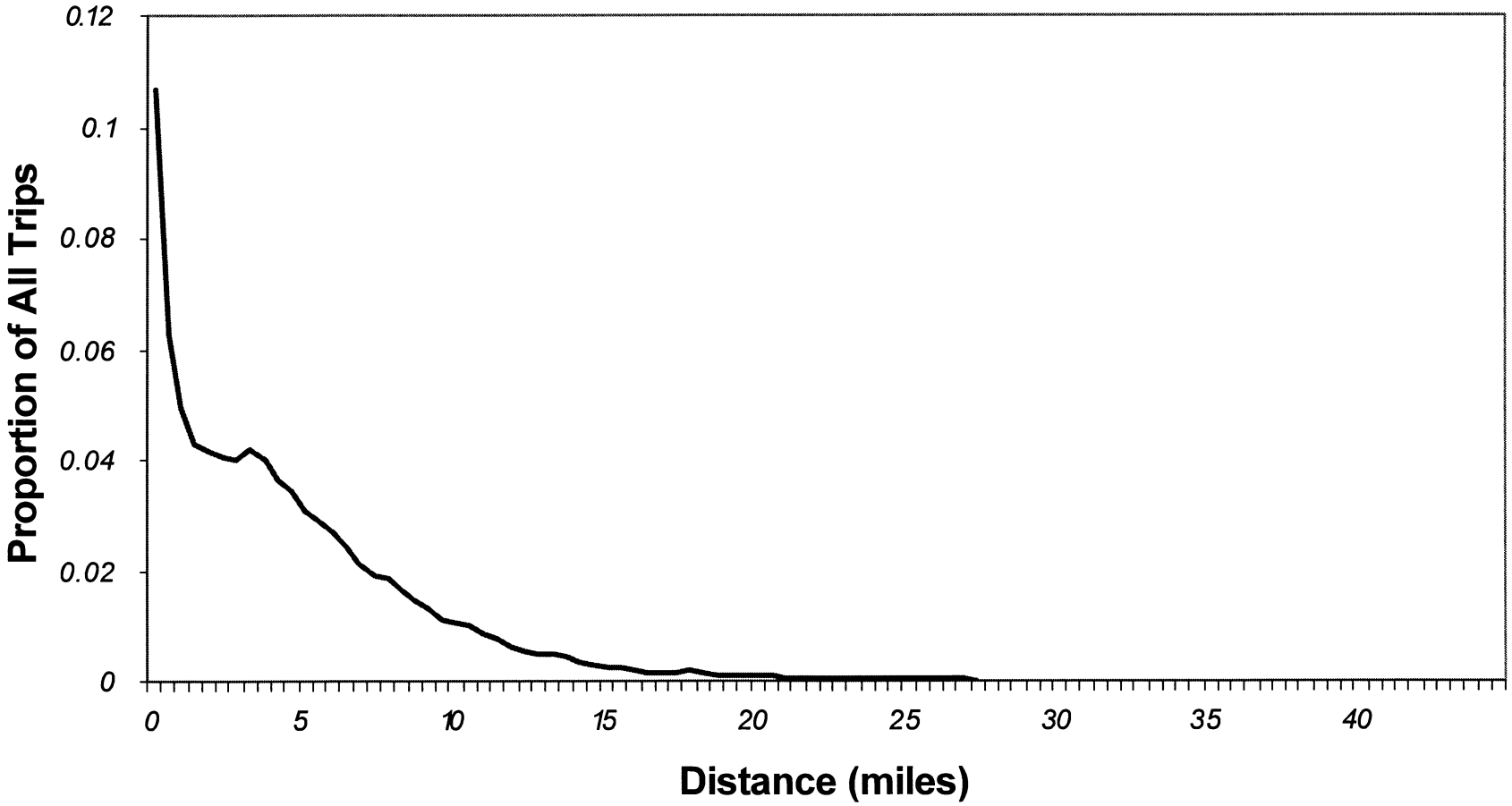
7. **Graphing the travel impedance function**. Click on 'View graph' to see the travel impedance function. The screen view can be printed by clicking on 'Print'. For a better quality graph, however, the output should be imported into a graphics or spreadsheet program.

Example of Empirical Impedance from Baltimore County

An example of an empirical impedance function from Baltimore County is seen in figure 14.5. This was derived from the 41,974 incidents in which both the crime location and the offender's origin location were known. As seen, the function looks similar to a

Figure 14.5:

Empirical Impedance Function: All Crimes



negative exponential function. But there is a little 'hitch' around 3 miles where the travel likelihood increases, rather than decrease. This could possibly be due to the City of Baltimore border which abuts much of the southern part of the County.

Whatever the reason, the empirical impedance function can be used as a proxy for travel 'cost' by offenders. As we shall see, however, it may not produce as good a fit in the gravity model as some of the mathematical functions, particularly the lognormal. The reason is that it is a behavioral description. Consequently, the pattern reflects both the existence of crime opportunities (attractions) as well as costs. While an empirical description is useful for guessing where a serial offender might live, for a trip distribution model it apparently does not cleanly estimate the real costs to an offender. Nevertheless, it is a tool that can be used.

Setup of Origin-Destination Model

The page is for the setup of the origin-destination model. All the relevant files, models and exponents are input on the page and it allows the trip distribution model to be calibrated and allocated. Figure 14.6 shows the setup screen. There are a number of parameters that have to be defined:

1. **Predicted origin file.** The predicted origin file is a file that lists the origin zones with a single point representing the zone (e.g., the centroid) and also includes the predicted number of crimes by origin zone. The file must be input as either the primary or secondary file. Specify whether the data file is the primary or secondary file.

   Origin variable. Specify the name of the variable for the predicted origins (e.g., PREDICTED, ADJORIGINS).

   Origin ID. Specify the origin ID variable in the data file (e.g., CensusTract, Block, TAZ).

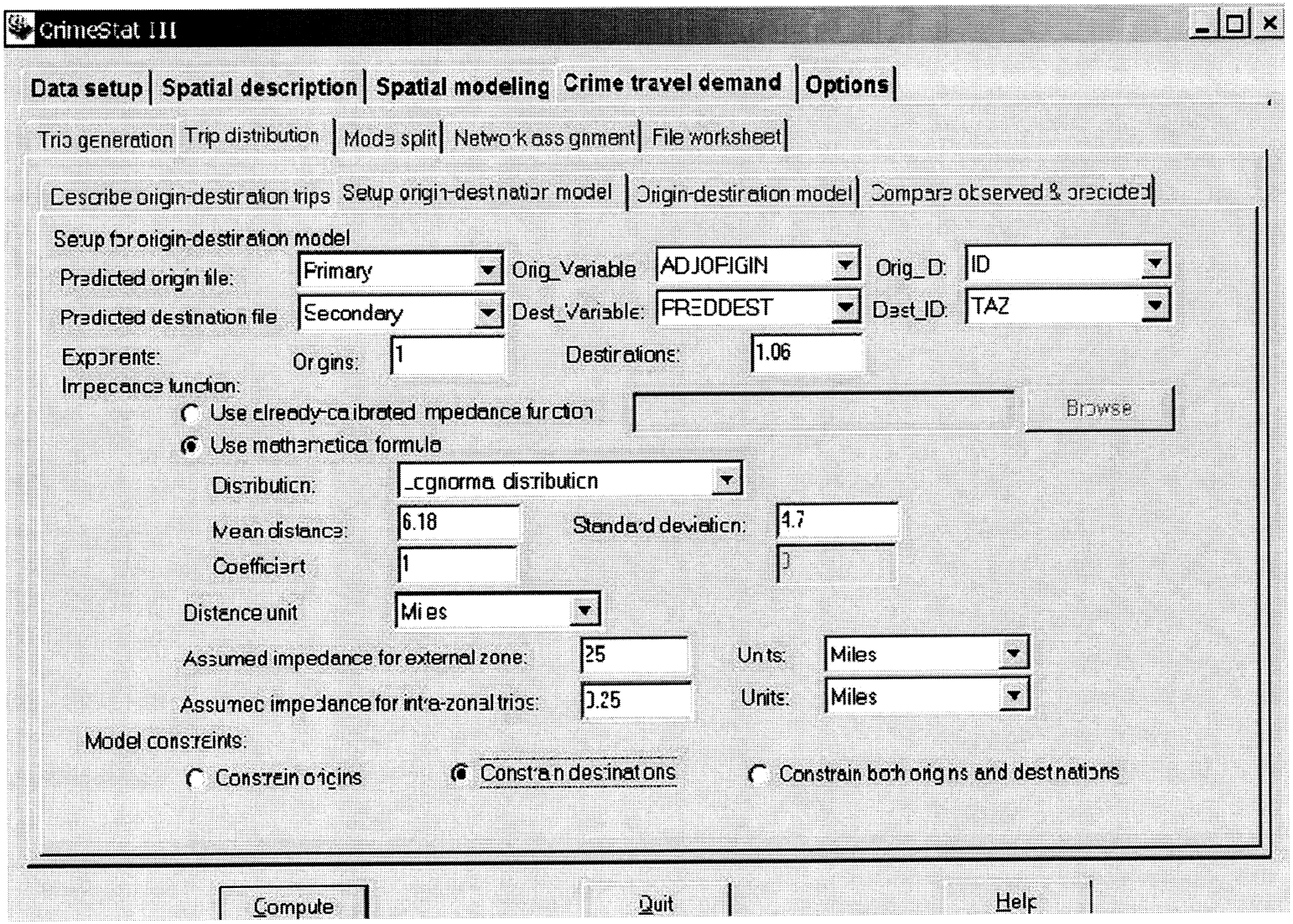
2. **Predicted destination file**. The predicted destination file is a list of destination zones with a single point representing the zone (e.g., the centroid) and also includes the predicted number of crimes by destination zone. It must be input as either the primary or secondary file. Specify whether the data file is the primary or secondary file.

   Destination variable. Specify the name of the variable for the predicted destination (e.g., PREDICTED, ADJDEST).

   Destination ID. Specify the destination ID variable in the data file (e.g., CensusTract, Block, TAZ).

Figure 14.6:

Trip Distribution Model Setup



Note: if M is the number of rows and N is the number of columns, then the total number of grid cells (M x N) cannot be greater than SQRT(RAM - 64)/56 where RAM is the available RAM. There is a maximum allowable of 4 Gb.

3. **Exponents**. The exponents are power terms for the predicted origins and destinations. They indicate the relative strength of those variables. For example, compared to an exponent of 1.0 (the default), an exponent greater than 1.0 will strengthen that variable (origins or destinations) while an exponent less than 1.0 will weaken that variable. They can be considered 'fine tuning' adjustments.

   Origins. Specify the exponent for the predicted origins. The default is 1.0.

   Destinations. Specify the exponent for the predicted origins. The default is 1.0.

4. **Impedance function**. The trip distribution routine can use two different travel distance functions:

   1) An already-calibrated distance function; and
   2) A mathematical formula (the default).

   Use an already-calibrated distance function. If a travel distance function has already been calibrated (see 'Calibrate impedance function' above), the file can be directly input into the routine. The user selects the name of the already-calibrated travel distance function. *CrimeStat* reads dbase 'dbf', ASCII text 'txt', and ASCII data 'dat' files.

   Use a mathematical formula. A mathematical formula can be used instead of a calibrated distance function. Similar to the Journey to crime module (see chapter 9), there are five mathematical functions. They measure a *separation* between two zones and estimate a likelihood value. 'Separation' can be in terms of distance, travel time, speed (which is converted into travel time), or travel costs.

   Mathematical functions

   Briefly, the five functions are:

1. **Linear**. The simplest type of distance model is a linear function. This model postulates that the likelihood of traveling to a zone from another by an offender declines by a constant amount with distance from the offender's home. It is highest near the offender's home but drops off by a constant amount for each unit of distance until it falls to zero. The form of the linear equation is

$$f(d_{ij}) = \alpha + \beta * S_{ij} \quad (14.14)$$

where $f(d_{ij})$ is the likelihood that the offender will travel from zone ii to a particular location, j, S_{ij} is the *separation* in distance, time or cost between the offender's residence, I, and location j, α is a slope coefficient which defines the fall off in distance, and β is a constant. It would be expected that the coefficient β would have a negative sign since the likelihood should decline with separation. The user must provide values for *A* and β. The default for A is 10 and for β is -1. When the function reaches 0 (the X axis), the routine automatically substitutes a 0 for the function. Figure 14.7 illustrates this function.

2. ***Negative Exponential***. A slightly more complex function is the negative exponential. In this type of model, the likelihood of travel also drops off with distance. However, the decline is at a constant *rate* of decline, thus dropping quickly near the offender's home until is approaches zero likelihood. The mathematical form of the negative exponential is:

$$f(d_{ij}) = \alpha * e^{-\beta * S_{ij}} \quad (14.15)$$

where $f(d_{ij})$ is the likelihood that the offender will travel from an origin zone, ii, to a destination zone, j, S_{ij} is the separation between the origin zone and the destination zone, e is the base of the natural logarithm, α is the coefficient and β is an exponent of e. The user inputs values for α - the coefficient, and β - the exponent. The default for α is 10 and for β is 1.

This function is the one most used by travel demand modelers. It has been recommended for use by the Federal Highway Administration (NCHRP, 1995). Figure 14.8 illustrates a typical negative exponential impedance function.

3. ***Normal***. A normal distribution assumes the peak likelihood is at some optimal distance from the offender's home base. Thus, the function rises to that distance and then declines. The rate of increase prior to the optimal distance and the rate of decrease from that distance is symmetrical in both directions. The mathematical form is:

$$Z_{ij} = \frac{(S_{ij} - \text{MeanD})}{\sigma_d} \quad (14.16)$$

$$f(d_{ij}) = \alpha * \frac{1}{\sigma_d * \text{SQRT}(2\pi)} * e^{-0.5 * Z_{ij}^2} \quad (14.17)$$

$$f(d_{ij}) = \alpha + \beta * S_{ij} \quad (14.14)$$

where $f(d_{ij})$ is the likelihood that the offender will travel from zone ii to a particular location, j, S_{ij} is the *separation* in distance, time or cost between the offender's residence, I, and location j, α is a slope coefficient which defines the fall off in distance, and β is a constant. It would be expected that the coefficient β would have a negative sign since the likelihood should decline with separation. The user must provide values for *A* and β. The default for A is 10 and for β is -1. When the function reaches 0 (the X axis), the routine automatically substitutes a 0 for the function. Figure 14.7 illustrates this function.

2. ***Negative Exponential***. A slightly more complex function is the negative exponential. In this type of model, the likelihood of travel also drops off with distance. However, the decline is at a constant *rate* of decline, thus dropping quickly near the offender's home until is approaches zero likelihood. The mathematical form of the negative exponential is:

$$f(d_{ij}) = \alpha * e^{-\beta * S_{ij}} \quad (14.15)$$

where $f(d_{ij})$ is the likelihood that the offender will travel from an origin zone, ii, to a destination zone, j, S_{ij} is the separation between the origin zone and the destination zone, e is the base of the natural logarithm, α is the coefficient and β is an exponent of e. The user inputs values for α - the coefficient, and β - the exponent. The default for α is 10 and for β is 1.

This function is the one most used by travel demand modelers. It has been recommended for use by the Federal Highway Administration (NCHRP, 1995). Figure 14.8 illustrates a typical negative exponential impedance function.

3. ***Normal***. A normal distribution assumes the peak likelihood is at some optimal distance from the offender's home base. Thus, the function rises to that distance and then declines. The rate of increase prior to the optimal distance and the rate of decrease from that distance is symmetrical in both directions. The mathematical form is:

$$Z_{ij} = \frac{(S_{ij} - \text{MeanD})}{\sigma_d} \quad (14.16)$$

$$f(d_{ij}) = \alpha * \frac{1}{\sigma_d * \text{SQRT}(2\pi)} * e^{-0.5 * Z_{ij}^2} \quad (14.17)$$

Figure 14.7:

Linear Impedance Function

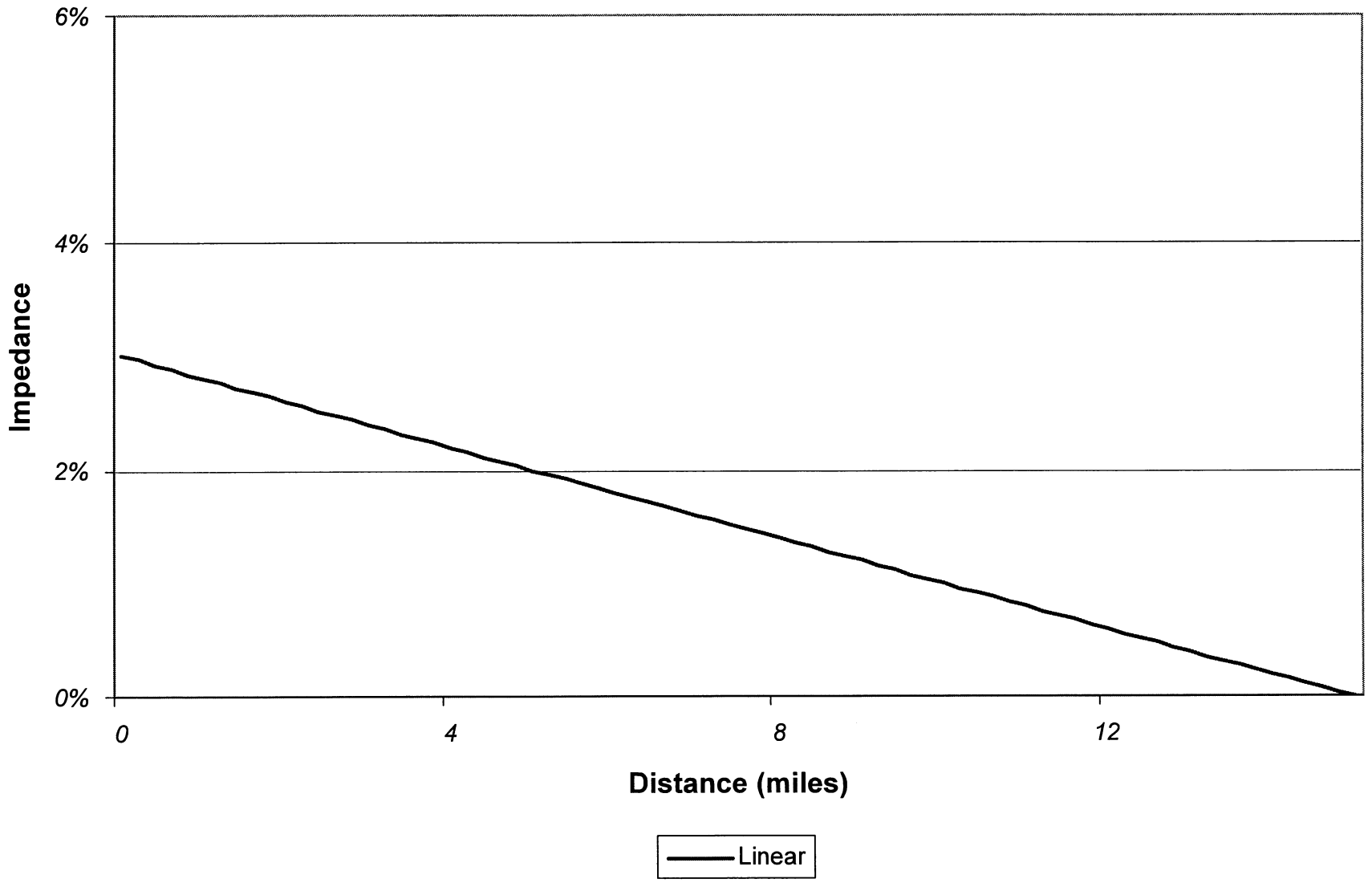
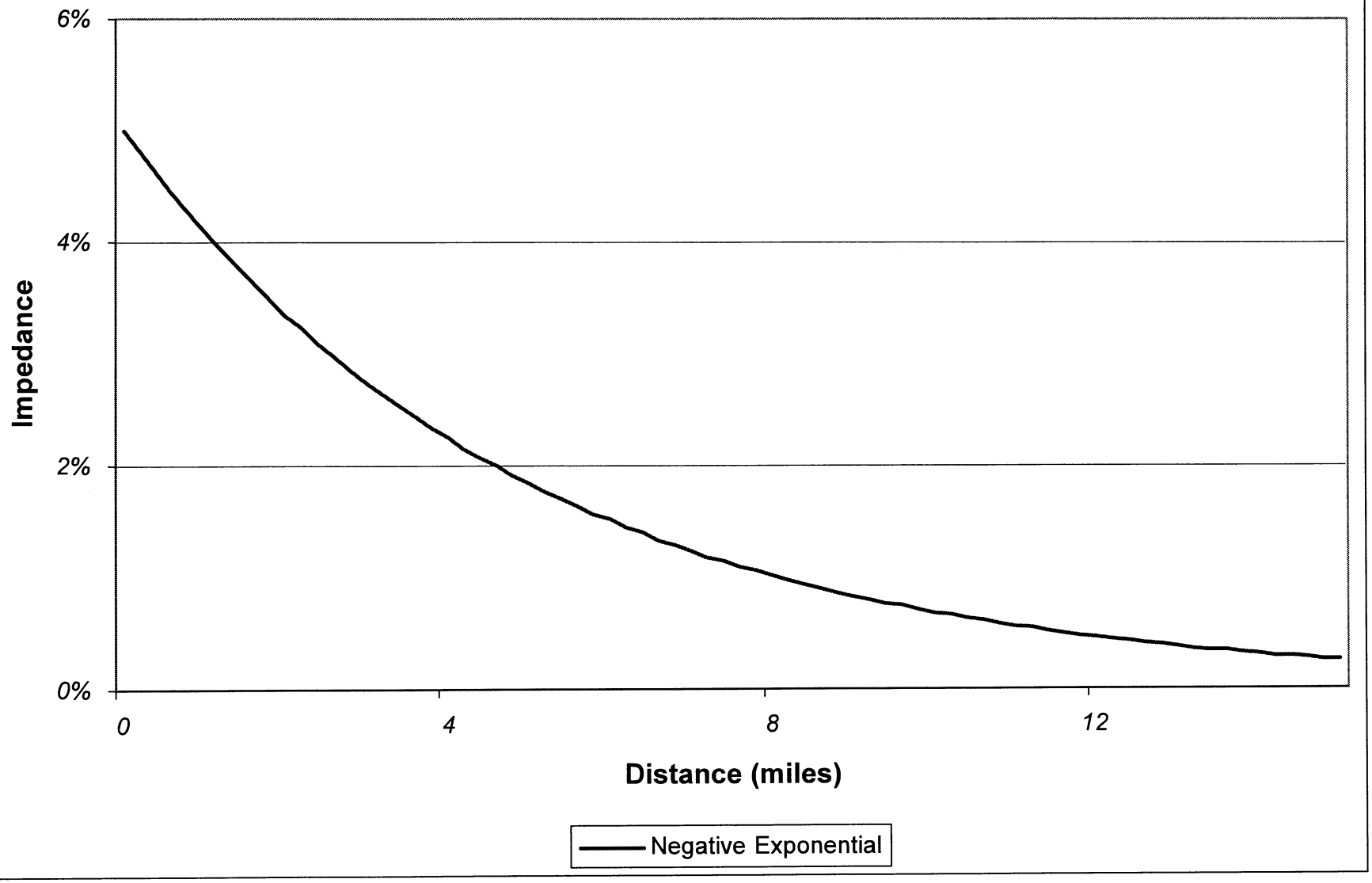


Figure 14.8:

Negative Exponential Impedance Function



where $f(d_{ij})$ is the likelihood that the offender will commit a crime at a particular location, I (defined here as the center of a grid cell), S_{ij} is the separation between each origins zone and destination zone, MeanD is the mean distance input by the user, σ_d is the standard deviation of distances, **e** is the base of the natural logarithm, and α is a coefficient. The user inputs values for MeanD, σ_d, and α. The default values are 1 for each of these parameters.

By carefully scaling the parameters of the model, the normal distribution can be adapted to a distance decay function with an increasing likelihood for near distances and a decreasing likelihood for far distances. For example, by choosing a standard deviation greater than the mean (e.g., MeanD = 1,S_d = 2), the distribution will be skewed to the left because the left tail of the normal distribution is not evaluated. Figure 14.9 illustrates a possible normal impedance function.

4. ***Lognormal***. The lognormal function is similar to the normal except it is more skewed, either to the left or to the right. It has the potential of showing a very rapid increase near the origin with a more gradual decline from a location of peak likelihood. The mathematical form of the function is:

$$f(d_{ij}) = \alpha * \frac{1}{S^2_{ij} * \sigma_d * \mathrm{SQRT}(2\pi)} * \mathbf{e}^{-[\ln(S^2_{ij})-\mathrm{MeanD}]^2/2*\sigma_d^2} \qquad (14.18)$$

where $f(d_{ij})$ is the likelihood that the offender will commit a crime at a particular location, I, defined here as the center of a grid cell, S_{ij} is the separation between the origin zone and the destination zone, MeanD is the mean separation input by the user, σ_d is the standard deviation of separation, **e** is the base of the natural logarithm, and α is a coefficient. The user inputs MeanD, σ_d , and α. The default values are 1 for each of these parameters. Figure 14.10 illustrates a log-normal impedance function that had wide utility in several studies that are discussed below.

5. ***Truncated Negative Exponential***. Finally, the truncated negative exponential is a joined function made up of two distinct mathematical functions - the linear and the negative exponential. For the near distance, a positive linear function is defined, starting at zero likelihood for distance 0 and increasing to d_p, a location of peak likelihood. Thereupon, the function follows a negative exponential, declining quickly with distance. The two mathematical functions making up this spline function are:

Linear: $f(d_{ij}) = 0 + \beta * S_{ij} = \beta * d_{ij}$ for $S_{ij} \geq 0$, $d_ij \leq d_p$ (14.19)

Negative Exponential: $f(d_{ij}) = \alpha * e^{-\xi * S_{ij}}$ for $X_i > S_p$ (14.20)

Figure 14.9:

Normal Impedance Function

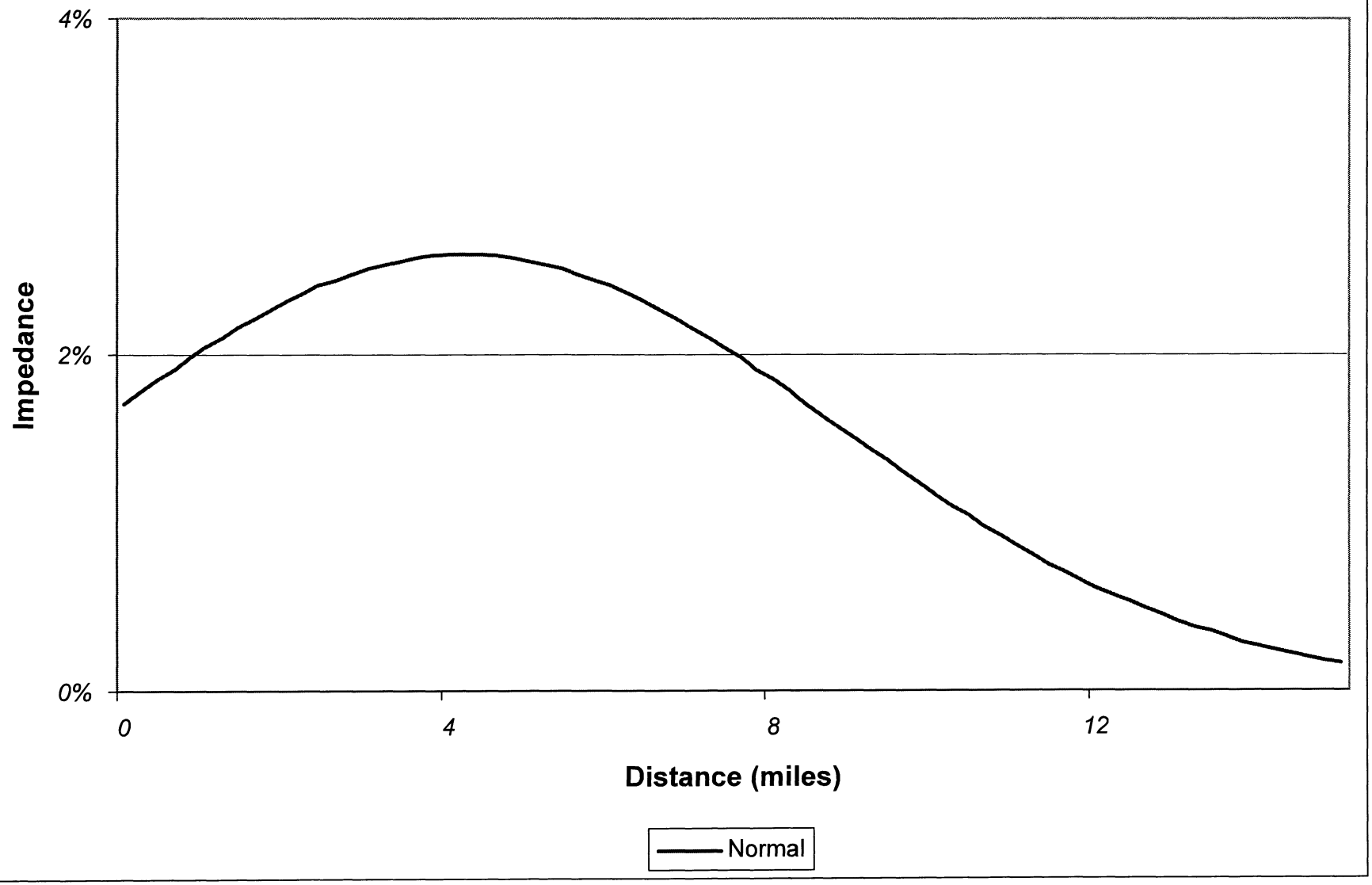


Figure 14.10:

Lognormal Impedance Function

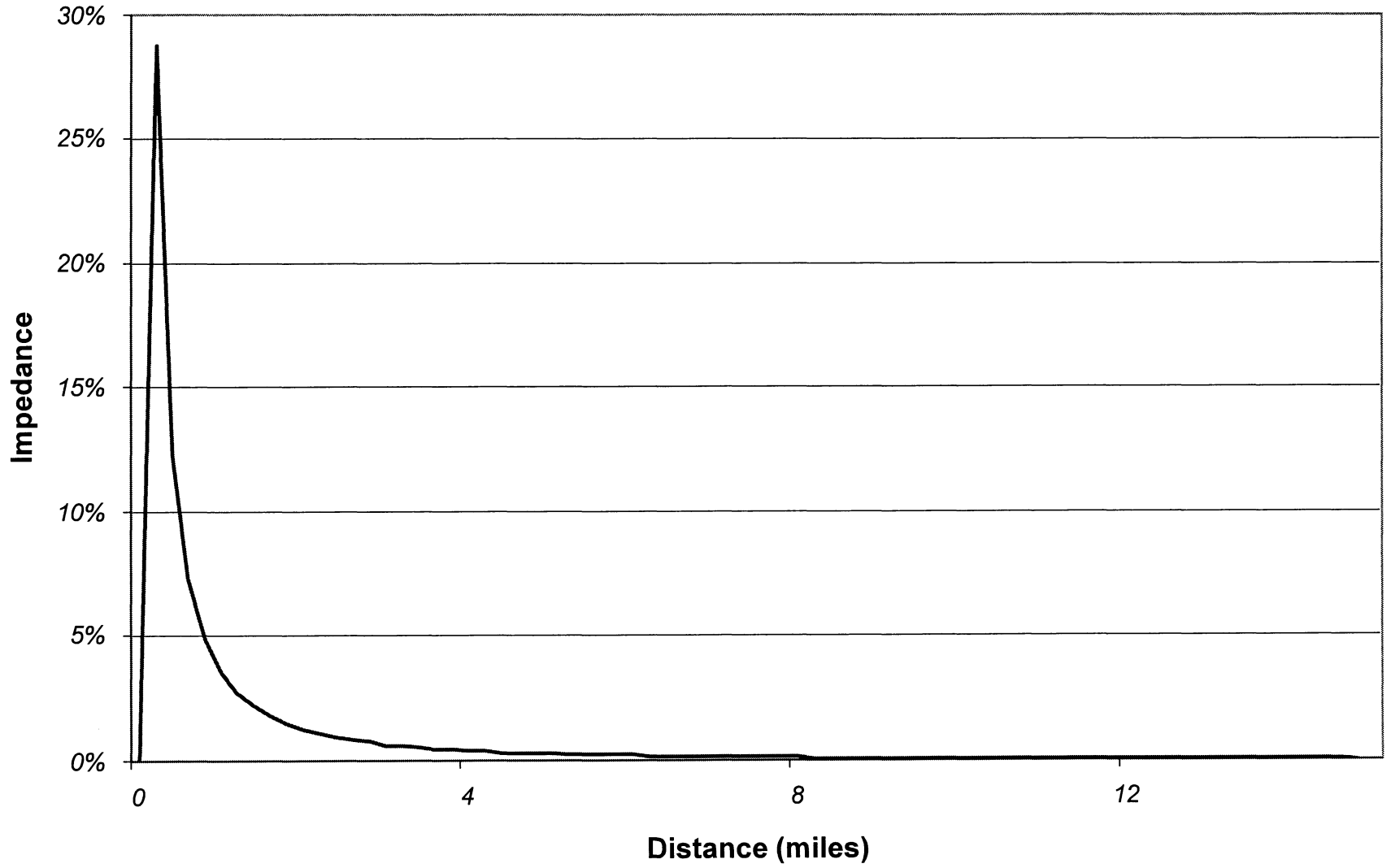
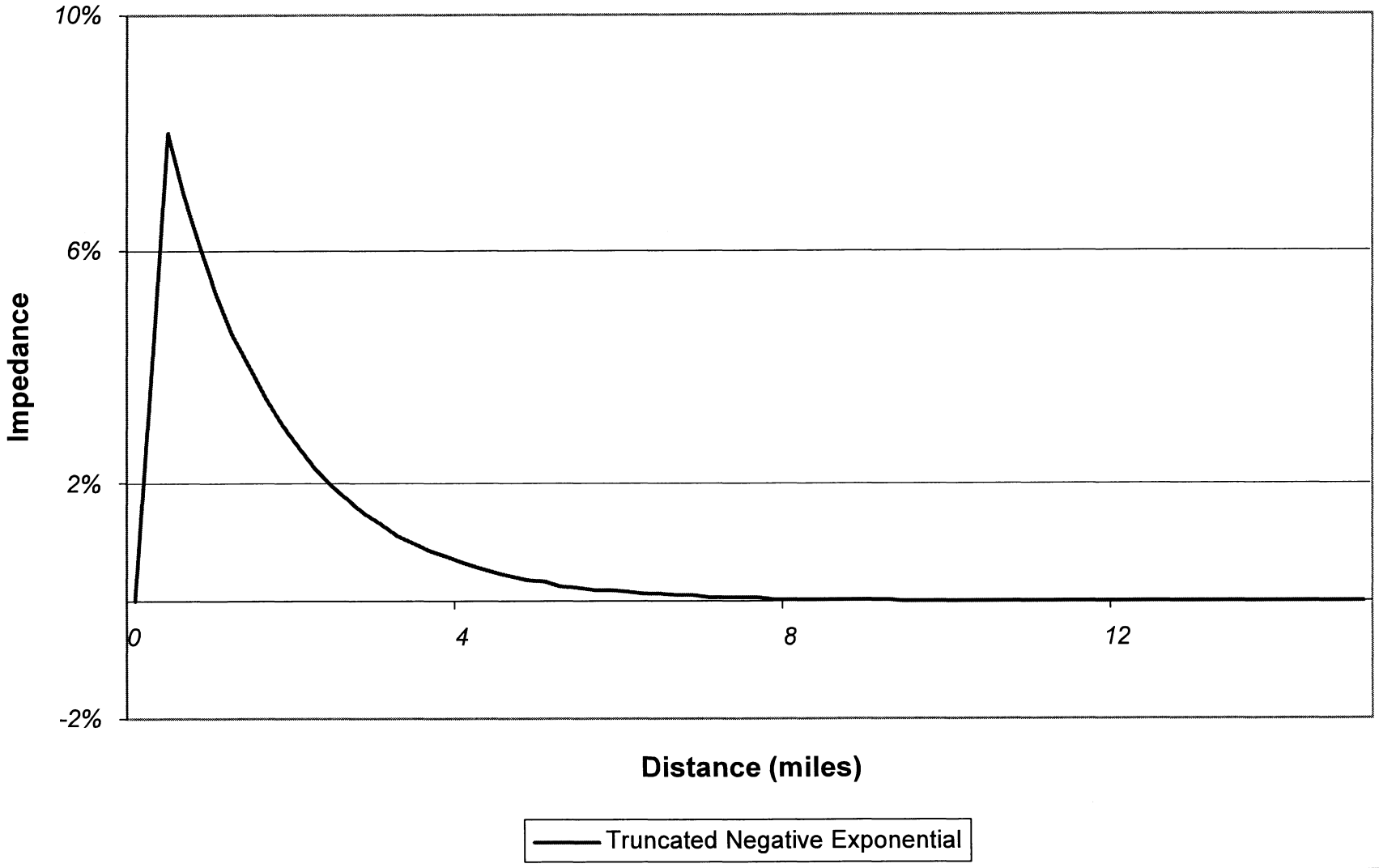


Figure 14.11:

Truncated Negative Exponential Impedance Function



where S_{ij} is the separation from the home base, β is the slope of the linear function (default=+1) and for the negative exponential function α is a coefficient and ξ is an exponent. Since the negative exponential only starts at a particular distance, $Maxd_{ij}$, α, is assumed to be the intercept if the Y-axis were transposed to that distance. Figure 14.11 illustrates a truncated negative exponential impedance function.

Model parameters. For each mathematical model, two or three different parameters must be defined:

1. For the negative exponential, the coefficient and exponent
2. For the normal distribution, the mean distance, standard deviation and coefficient
3. For lognormal distribution, the mean distance, standard deviation and coefficient
4. For the linear distribution, an intercept and slope
5. For the truncated negative exponential, a peak distance, peak likelihood, intercept, and exponent.

The parameters will be obtained either from a previous analysis or from an iterative process of experimentation. See the example below under "Compare observed and predicted trips".

5. **'Fine Tuning' Exponents**. In addition, for each function, exponents for the attraction and production terms can adjusted. This allows a 'fine tuning' of the impedance function to better fit the empirical distribution.

6. **Distance Units.** The routine can calculate impedance in four ways, by:

   1. Distance (miles, nautical miles, feet, kilometers, and meters)
   2. Travel time (minutes, hours)
   3. Speed (miles per hour, kilometers per hour). Speed is then converted into travel time, in minutes.
   4. General travel costs (unspecified units).

   These must be setup under 'Network Distance' on the Measurement Parameters page. In the Network Parameters dialogue, specify the measurement units. The default is distance in miles.

7. **Assumed Impedance for External Zones.** For trips originating outside the study area (external trips), specify the amount and the units that will be assumed for these trips. The default is 25 miles.

8. **Assumed Impedance for Intra-zonal Trips**. For trips originating and ending in the same zone (intra-zonal trips), specify the amount and the units that will be assumed for these trips. The default is 0.25 miles.

9. **Model Constraints.** In calibrating a model, the routine must constrain either the origins or the destinations (single constraint) or constrain both the origins and the destinations (double constraint). In the latter case, it is an iterative solution. The default is to constrain destinations as it is assumed that the destination totals (the number of crimes occurring in each zone) are probably more accurate than the number of crimes originating in each zone. . Specify the type of constraint for the model.

   Constrain origins. If 'constrain origins' is selected, the total number of trips from each origin zone will be held constant.

   Constrain destinations. If 'constrain destinations' is selected, the total number of trips from each destination zone will be held constant.

   Constrain both origins and destinations. If 'constrain both origins and destinations' is selected, the routine works out a balance between the number of origins and the number of destinations.

Fitting the Impedance Function

The impedance function is fit in an iterative manner. First, either an empirical impedance or a mathematical impedance is chosen. Second, the particular mathematical function is selected. For example, with the lognormal function, which has been found to produce the best fit for three different data sets, there are three parameters: 1) the mean distance; 2) the standard deviation of distance; and 3) the coefficient.

Third, initial values of the parameters are chosen; one suggestion is to use the defaults available in the *CrimeStat* routines. The "Compare observed and predicted trips" routine is used to evaluate the fit of the model. Fourth, the parameters are adjusted in small increments, one at a time, on both side of the initial guess in order to improve the fit. For example, with the lognormal function, the mean distance is fit first because it has the greatest impact on the overall fit. Then, after a "best" mean distance has been found, the standard deviation of distance is adjusted until it produces a "best" fit. Then, the coefficient is adjusted until it produces a "best" fit. Fifth, and finally, the 'fine tuning' exponents of the production and attraction functions are adjusted. Typically, these change the final fit only slightly. Hence, they represent a final adjustment.

This process is illustrated below in the discussion on the comparison of the observed and predicted trips. Essentially, the empirical (observed) distribution is being used as a calibration sample in order to find that impedance model and parameters that best approximate it.

The Origin-Destination Model

The trip distribution (origin-destination) model is implemented in two steps. First, the coefficients are calculated according to the exponents and impedance functions

specified on the setup page. Second, the coefficients and exponents are applied to the predicted origins and destinations resulting in a predicted trip distribution. Because these two steps are sequential, they cannot be run simultaneously.

Calibrate Origin-Destination Model.

In this routine, the row or column parameters (or both if double constraint is used) are estimated using a calibration file. The steps are as follows:

1. **Check** the 'Calibrate origin-destination model' box to run the calibration model.

2. **Save Modeled Coefficients (parameters).** The modeled coefficients are saved as a 'dbf' file. Specify a file name.

Apply Predicted Origin-Destination Model

In this routine, the coefficients that were calibrated in the above routine can be applied to a data set. The data set can be the same as the calibration file or a different one. The reason for separating the calibration from application steps is that the coefficients can be used for many different data sets. The steps are as follows:

1. **Check** the 'Apply predicted origin-destination model' box to run the trip distribution prediction.

2. **Modeled Coefficients File**. Load the modeled coefficients file saved in the 'Calibrate origin-destination model' stage.

3. **Assumed Coordinates for External Zone**. In order to model trips from the 'external zone' (trips from outside the study area), specify coordinates for this zone. These coordinates will be used in drawing lines from the predicted origins to the predicted destinations. There are four choices:

    1. Mean center (the mean X and mean Y of all origin file points are taken). This is the default.
    2. Lower-left corner (the minimum X and minimum Y values of all origin file points are taken).
    3. Upper-right corner (the maximum X and maximum Y values of all origin file points are taken).
    4. Use coordinates (user-defined coordinates). Indicate the X and Y coordinates that are to be used.

Because an arbitrary location is taken to represent the 'external zone', any lines that are shown from that zone will not necessarily represent any real travel behavior. However, if a very high proportion of all crime trips fall within the modeled origin zones

(i.e., 95% or more), then it is very unlikely that any of the top trip links will come from the 'external zone'.

4. **Table Output**. The table output includes summary file information and:

    1. The origin zone (ORIGIN)
    2. The destination zone (DEST)
    3. The number of predicted trips (PREDTRIPS)

5. **Save Predicted Origin-destination Trips**. Define the output file. The output is saved as a 'dbf' file specified by the user.

6. **File Output**. The file output includes:

    1. The origin zone (ORIGIN)
    2. The destination zone (DEST)
    3. The X coordinate for the origin zone (ORIGINX)
    4. The Y coordinate for the origin zone (ORIGINY)
    5. The X coordinate for the destination zone (DESTX)
    6. The Y coordinate for the destination zone (DESTY)
    7. The number of predicted trips (PREDTRIPS)

    Note: each record is a unique origin-destination combination and there are M x N records where M is the number of origin zones (including the external zone) and N is the number of destination zones.

7. **Save Links.** The top predicted origin-destination trip links can be saved as separate **line** objects for use in a GIS. Specify the output file format (*ArcView* '.shp', *MapInfo* '.mif' or *Atlas*GIS* '.bna') and the file name.

Save Top Links

Because the output file is very large (number of origin zones x number of destination zones), the user can select a sub-set of zone combinations with the most predicted trips. Indicating the top K links will narrow the number down to the most important ones. The default is the top 100 origin-destination combinations. Each output object is a line from the origin zone to the destination zone with an ODT prefix. The prefix is placed before the output file name.

The graphical output includes:

1. An ID number from 1 to K, where K is the number of links output (ID)
2. The feature prefix (ODT)
3. The origin zone (ORIGIN)
4. The destination zone (DEST)

5. The X coordinate for the origin zone (ORIGINX)
6. The Y coordinate for the origin zone (ORIGINY)
7. The X coordinate for the destination zone (DESTX)
8. The Y coordinate for the destination zone (DESTY)
9. The number of predicted trips for that combination (PREDTRIPS)
10. The distance between the origin zone and the destination zone.

8. **Save Points**

Intra-zonal trips (trips in which the origin and destination are the same zone) can be output as separate **point** objects as an *ArcView* '.shp', *MapInfo* '.mif' or *Atlas*GIS* '.bna' file. Again, the top K points are output (default=100). Each output object is a point representing an intra-zonal trip with an ODTPOINTS prefix. The prefix is placed before the output file name.

The graphical output for each includes:

1. An ID number from 1 to K, where K is the number of links output (ID)
2. The feature prefix (POINTSODT)
3. The origin zone (ORIGIN)
4. The destination zone (DEST)
5. The X coordinate for the origin zone (ORIGINX)
6. The Y coordinate for the origin zone (ORIGINY)
7. The X coordinate for the destination zone (DESTX)
8. The Y coordinate for the destination zone (DESTY)
9. The number of predicted trips for that combination (PREDTRIPS)

Example of the Predicted Trip Distribution from Baltimore County

The predicted origins and predicted destinations from Baltimore County were input into a trip distribution model and a predicted trip distribution was output. The impedance function was a lognormal distribution, which produced a good fit to the observed (empirical) distribution (see discussion below).

Figure 14.12 outputs the top 1000 links from the model. The top 1000 links account for 14,271.9 trips, or 34.0% of the total number of trips. Compared to the observed distribution, the top 1000 links account for a smaller proportion of the total trips (14,272 v. 19,615). This suggests that the actual distribution is slightly more concentrated than the model suggests. Like the observed distribution, however, a sizeable number of the top links are intra-zonal trips (5,428 or 12.9%). The intra-zonal trips have been displayed as circles in the figure.

Comparing the predicted trip distribution to the observed trip distribution, some similarities and differences are seen. Figure 14.13 compares the top 1000 zone-to-zone links for the predicted and observed distributions. The model has captured many of the

Figure 14.12:

Predicted Baltimore County Crime Trips: 1993-1997
Top 1000 Links
All Crime Types

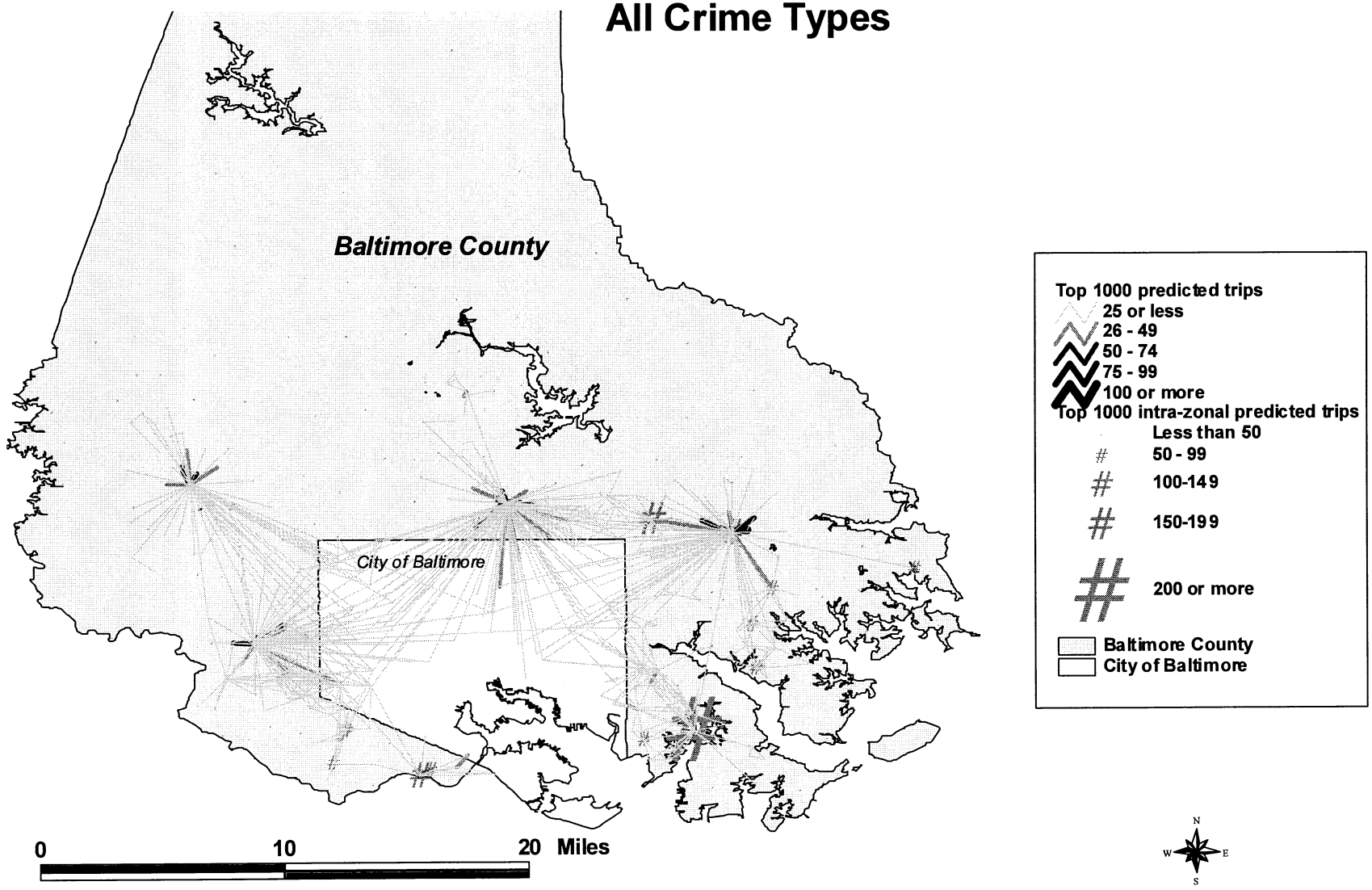
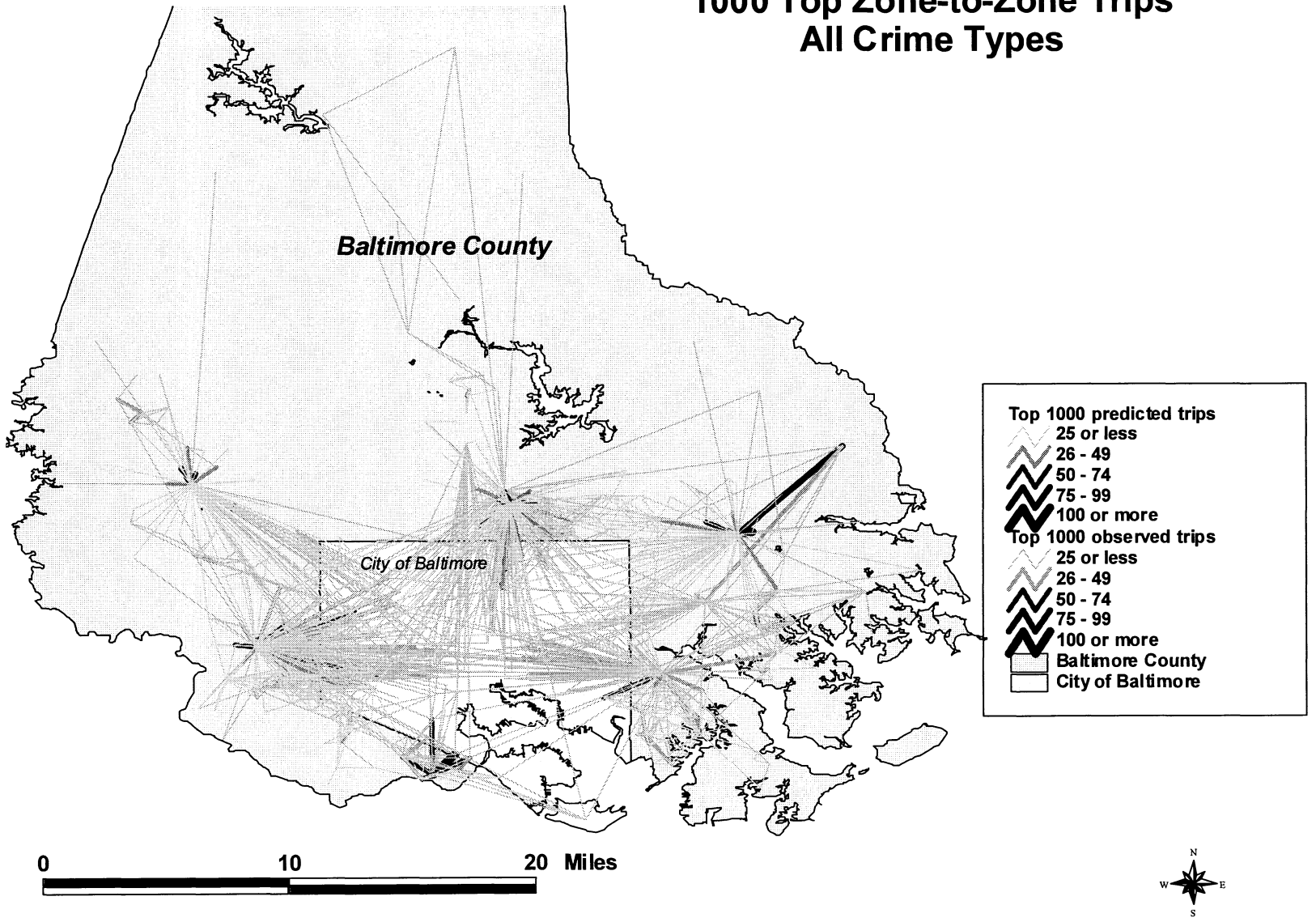


Figure 14.13:

Comparison of Predicted and Observed Crime Trips
1000 Top Zone-to-Zone Trips
All Crime Types



major links. For the five shopping malls that received many actual crime trips, the model has captured the majority of trips for three of them and some trips for a fourth. For the mall in the southeast corner of the county, on the other hand, the model has not allocated a large number of trips. Similarly, for a zone near the western edge of the county, the model has allocated more trips than actually occurred.

There are, of course, only 325 intra-zonal trip links (one for each destination zone). Looking at a comparison of the intra-zonal trips (figure 14.14), some similarities and differences are seen. Generally, the model captured the location of many intra-zonal trips, but it did not capture the quantity very accurately. Zones that had many intra-zonal trips are shown as having only some by the model and, conversely, the model predicts many intra-zonal trips for two zones which had only some.

In other words, the fit between the actual distribution and the model is not perfect. Considering that only 1000 of the 172,900 trip links (532 origin zones x 325 destination zones) are shown, the model has still done a reasonable job of capturing the major links

It is not surprising that the model is not perfect. The model is a simple analogue using only three variables (productions, attractions, impedance) whereas the actual distribution represents a very complex set of individual decisions made by offenders. What is perhaps remarkable is that the model has done a decent job of capturing some of these relationships at all.

This brings up an important point, namely that a model is not reality; it is only a simplified set of relationships that approximates reality (in this case, the observed distribution). It is important in developing any model to evaluate it relative to an observed set of facts, and this applies no less to the trip distribution model. One has to understand, however, that a good model will not capture all the relationships. Hopefully, it captures enough of them to make the model useful for prediction and evaluating policy options.

Comparing Observed & Predicted Trips

It is important to conduct a number of tests on the predicted model to ensure that it is capturing the most important elements of the observed distribution. These are conducted by comparing the predicted distribution with the observed (empirical) distribution.

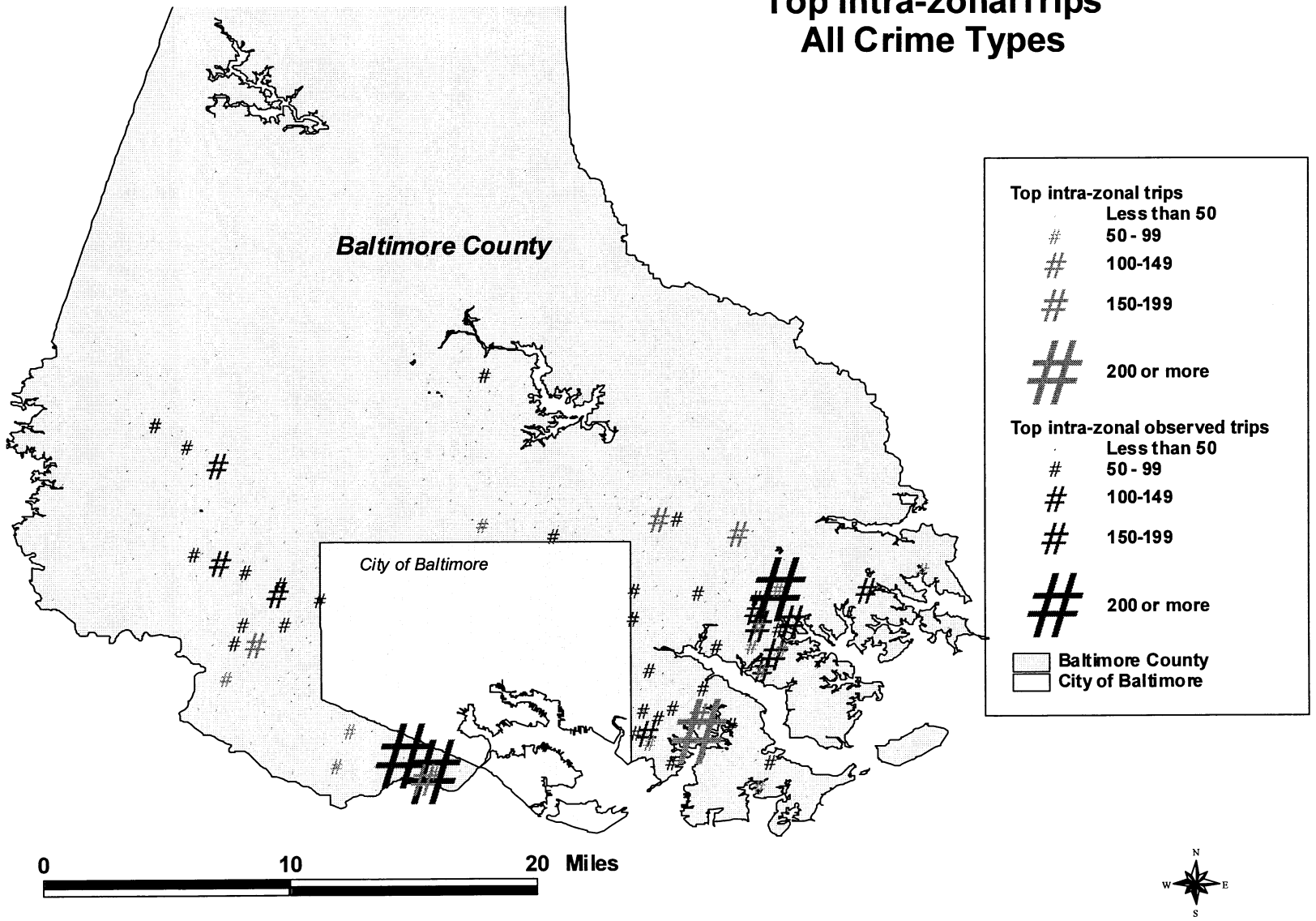
There are a number of tests that can be used to evaluate a model by comparing the predicted distribution with the observed one. *CrimeStat* includes three of these and the steps are as follows:

1. Estimate the parameters of the model and apply them to the calibration data set

2. Examine the intra-zonal trips to be sure that the predicted number corresponds to the observed number

Figure 14.14:

Comparison of Predicted and Observed Crime Trips
Top Intra-zonalTrips
All Crime Types



3. Compare the trip lengths of the observed and predicted distributions using two tests:

   A. The Coincidence Ratio
   B. The Komolgorov-Smirnov Two-sample Test

4. Compare the number of trips for the top links using a pseudo-Chi square test. That is, the number of trips for the most frequent links in the observed distribution are compared to the number predicted by the model for the same links.

Unfortunately, not one of these tests is sufficient to validate a model. Further, minimizing the discrepancy for only one of them may distort the others. It is very unlikely that there will be a model that minimizes the errors for all three tests. Consequently, the user will have to choose a model that balances these factors in a desirable way (an *optimum* model).

Estimating Impedance Parameters and Exponents of Gravity Model

While this is not strictly an evaluation test, this step is essential in estimating the particular impedance parameters that are used in the first place. Typically, an analyst will approximate an impedance function. Using a comparison between the observed and predicted models, the parameters can be adjusted to produce a better fit. The steps are as follows:

1. The model is estimated with a calibration data set. There is a file of predicted origins and another file of predicted destinations; typically, these are defined as the primary and secondary files respectively, though the order could be reversed or the same file used for both origins and destinations (if the number of origins zones was identical to the number of destination zones).

2. On the trip distribution setup page, select the type of impedance function that is to be used, already-calibrated (empirical) or mathematical. For the journey to crime routine, generally the empirical function led to better results than the mathematical. However, with a trip distribution function, a mathematical function may be as good, if not better. This was tested with three data sets for Baltimore County, Las Vegas, and Chicago and, in all cases, a mathematical function (the lognormal) gave a much better fit than an empirically-derived function (see chapter 17).

3. *If* a mathematical function is to be used, select the type of distribution. The default value is a lognormal, but the user can choose a negative exponential, a normal, a linear, or a truncated negative exponential function.

4. For the particular mathematical function, select initial guesses for the parameters. For each mathematical model, two or three different parameters must be defined:

    1. For the negative exponential, the coefficient and exponent
    2. For the normal distribution, the mean distance, standard deviation and coefficient
    3. For lognormal distribution, the mean distance, standard deviation and coefficient
    4. For the linear distribution, an intercept and slope
    5. For the truncated negative exponential, a peak distance, peak likelihood, intercept, and exponent.

5. In addition, there are exponents of the production and attraction side that can be made to 'fine tune' the model. In general, these exponents will only affect the results slightly, compared to the basic choices of the type of model and the selection of values for the main parameters.

6. Calibrate and apply the model to the calibration data set. Examine the three criteria discussed below to minimize the error between the actual distribution and that predicted by the model.

7. Modify the parameter values slightly.

8. Repeat steps 4 through 7 until a good fit is found between the actual and predicted distribution and in which the errors are minimized and optimized. The process by which this is done is discussed below.

Comparing Intra-zonal Trips

The first evaluation test is to compare the percentage of trips that occur within the same zone - intra-zonal trips. The Travel Model Improvement Program manual indicates that intra-zonal trips should represent typically no more than 5% of all trips for home-to-work trips; that is, commuting trips (FHWA, 1997, chapter 4). However, given that most crime trips are quite short, the proportion of trips that are intra-zonal is liable to be much higher. In Baltimore County, for example, 19.7% of all crime trips were intra-zonal. Ideally, the predicted model should also have 19.7% of all crime trips being intra-zonal.

The "Compare observed and predicted trip lengths" routine is discussed below. The routine outputs the number of trips that are intra-zonal in both the observed and predicted distributions. A good model should produce approximately the same number of intra-zonal trips in the predicted distribution as what actually occurred.

Illustration

For example, in the Baltimore County model displayed in figure 14.12 above, there were 8,272 intra-zonal trips in the actual distribution (out of 41,979). On the other hand, there were only 5,428 intra-zonal trips in the model. In other words, the predicted model assigned fewer intra-zonal trips than actually occurred.

It may be necessary to modify the model to produce a closer fit for the intra-zonal trips. A simple way to do this to increase or decrease the relative impedance in the model. So, to use the example, if the predicted model is assigning too few intra-zonal trips, then the cost function can be strengthened (i.e., making travel more expensive). In this case, in the original model the lognormal function was used with a mean distance of 6.18 miles. If the mean distance of the impedance function is reduced to 3.5, then the number of predicted intra-zonal trips increases to 8,275, almost the same number as occurred in the observed distribution.

In other words, by decreasing the mean distance for the lognormal function, the impedance function was strengthened (i.e., made more expensive) and a better fit was created between the observed and predicted distributions.

In and of itself, a mismatch for intra-zonal trips between the predicted model and what actually occurred doesn't necessarily require a modification of the gravity function. Other criteria must be considered, namely how well the predicted model fits the trip length distribution and how well the predicted models captures the most frequent inter-zonal (zone-to-zone) trip links. Later in the discussion, the issue of optimizing a model by balancing these different criteria will be described.

Compare Trip Length Distribution

The second evaluation test in comparing the observed with the predicted distribution is a calculation of the trip length distribution (see steps below). Because the trip distribution matrix will typically be very large, most cell values will be zero. Rarely will there be enough data to cover all the cells and, even if there was, the skewness in crime distributions will leave most cells with no data. For example, for the Baltimore County model, with 532 origin zones and 325 destination zones, there will be 172,900 cells (325 x 532). The calibration data set had only 41,974 cases. Thus, the number of cells is more than four times the sample size and it is not possible to fill all cells with a number.

Consequently, because of the large number of cells with zero counts, one cannot use the Chi square test to compare the observed and predicted distributions. The Chi square test assumes that, first, the distribution is relatively normal (which it is not since the data are highly skewed) and, second, that there are at least 5 cases per cell. The latter condition is impossible given the large number of cells.

Therefore, what is usually done is to compare the *trip length* distribution of the observed and predicted models. 'Trip length' is the length in distance, travel time, or cost

of each trip. It is measured by the actual length (or separation) between two zones times the number of cases for that zone pair. For example, in figure 14.1, there were 15 trips from zone 1 to zone 2 and 7 trips in the opposite direction (from zone 2 to zone 1). Let's assume that the distance between zone 1 and zone 2 is 1.5 miles. Thus, there are 22 trips that fall into a trip length of 1.5 miles (15 in the direction of zone 1 to zone 2 and 7 in the direction of zone 2 to zone 1).

If travel time is used, the calculations uses time rather than distance. For example, if a vehicle was traveling 30 miles per hour, then it would take 3 minutes to cover 1.5 miles (1.5 miles ÷ 30 miles per hour = 0.05 hours x 60 minutes per hour = 3 minutes). Thus, there are 22 trips that fall into a trip 'length' of 3 minutes. A similar logic would apply to travel cost categories.

This process is repeated for all cells and the distribution of trips is allocated to the distribution of trip lengths (in distance, travel time, or travel cost). In general, one uses many intervals (or bins) for trip length (25 or more). In *CrimeStat*, the default number of trip lengths is 25, but it is not unknown to use up to 100. The problem in using too many is that the distributions become unreliable and differences that appear may not be real.

Graphical fit

Once the trip length distribution is calculated for both the observed and predicted distributions, it is possible to compare them. *CrimeStat* outputs a graph showing the fit of the two distributions. In general, they should be very close. An examination of differences between the distributions can indicate at what trip lengths the model is failing. This might allow the parameters to be adjusted in order to improve the fit on the next iteration. Examples will be given below of the graphing of the two distributions. But, it's important to come up with a model in which the two distributions 'look' similar.

Coincidence ratio

The *coincidence ratio* compares the two trip length distributions by examining the ratio of the total area of those distributions that coincide (i.e., that are in common; FHWA, 1997, chapter 4). It is defined as:

$$\text{Coincidence} = \sum_{k=1}^{K} \min\left[\frac{f^O}{F^O}, \frac{f^P}{F^P}\right] \tag{14.13}$$

$$\text{Total} = \sum_{k=1}^{K} \max\left[\frac{f^O}{F^O}, \frac{f^P}{F^P}\right] \tag{14.14}$$

$$\text{Coincidence ratio} = \frac{\text{Coincidence}}{\text{Total}} \qquad (14.15)$$

The steps are as follows:

1. Essentially, the two distributions are broken into K bins (or intervals). That is, the number of trips in each bin is enumerated (see example above).

2. Each of the two distribution is converted into a proportion by dividing the bin count by the total number of trips in the distribution. This step is not absolutely essential as the test can be conducted of the raw counts. However, by converting into proportions, the two distributions are standardized.

3. A cumulative count is conducted of the *minimum* proportion in each interval. That is, starting at the lowest interval, the smaller of the two proportions is taken. At the next interval, the smaller of the two proportions is added to the count. This is repeated for all K bins. This is called the *coincidence* and measure the overlapping proportions over all intervals.

4. A similar cumulative count is conducted of the *maximum* proportion in each interval. That is, starting at the lowest interval, the larger of the two proportions is taken. At the next interval, the larger of the two proportions is added to the count. This is repeated for all K bins. This is called the *total* and measures the unique proportion over all intervals.

5. Finally, the coincidence ratio is defined as the ratio of the minimum count to the total count.

The coincidence ratio is a proportion from 0 to 1. It is analogous to the R^2 statistic in regression analysis in that it measures the 'explained' (or overlapping) variance. According to the Travel Model Improvement Program manual (FHWA, 1997, chapter 4), the higher the coincidence ratio, the better. A value of 0.9 would generally be considered good.

Komolgorov-Smirnov two-sample test

The Komolgorov-Smirnov Two-Sample Test is similar to the coincidence ratio, but it examines the maximum difference across all bins (Kanji, 1993). For each distribution, a cumulative sum is created. At each interval, the difference between the two cumulative sums is calculated. The maximum difference between the two distributions is taken as the test statistic:

$$D = | O_i - P_i | \qquad (14.16)$$

There are tables of critical values for the Komolgorov-Smirnov Two-Sample Test which are a function of the number of intervals, K (Smirnov, 1948; Massey, 1951; Siegel, 1956; Kanji, 1993).

Illustration

To illustrate the trip length comparison, figures 14.15 through 14.18 show the results for four different impedance models - an empirical impedance function, a negative exponential impedance function, a truncated negative exponential impedance function, and a lognormal impedance function. As seen, the fit of the empirical impedance function is not particularly good, but gets progressively better with the three different mathematical functions.

The best fit is clearly with the lognormal function. With these parameters (mean center = 6.0 miles, standard deviation = 4.7 miles, coefficient = 1, origin exponent = 1, and destination exponent = 1.06), the Coincidence Ratio was 0.93.

But, again, this is just one criteria, albeit one that fits most of the distribution matrix. As with the number of intra-zonal trips, minimizing the error for a trip length distribution will not necessarily minimize the error for the other two criteria (intra-zonal trips and the top links). But, it's important that the trip length comparison be reasonably close.

Comparing the Trips of the Top Links

The third evaluation test focuses on the top links. That is, it evaluates how well the predicted model captures the major trip links, both intra-zonal and inter-zonal. Since crime trips are very skewed (i.e., a handful of zones contribute to most crime origins and a handful of zones attract many crimes), capturing the most important links is essential for a good crime distribution model. This is particularly true since a model that produces the best fit for the overall trip length distribution may not capture the top links very well.

Therefore, simply comparing the trip length distribution may not adequately capture the top links. That is, on average a particular model may produce a good fit between the predicted and observed distributions, but may do this by minimizing error across the entire matrix of trip pairs without necessarily minimizing the error for the top links.

Consequently, it's important to also compare the fit of the model for the top links. One of the lines in the dialogue for the "Compare observed and predicted trip lengths" is "Compare top links". The user should specify the number of top links to be compared; the default is 100. The top links are the trip pairs that have the most number of actual trips, starting from the pair with the most trips and sorting in descending order. The routine calculates a pseudo-Chi square test on just those links. Since the top links will all have a sufficient number of trips, it is possible to calculate a Chi square statistic. However, since

Figure 14.15:

Comparing Observed and Predicted Crime Trip Lengths

Empirical Impedance Function

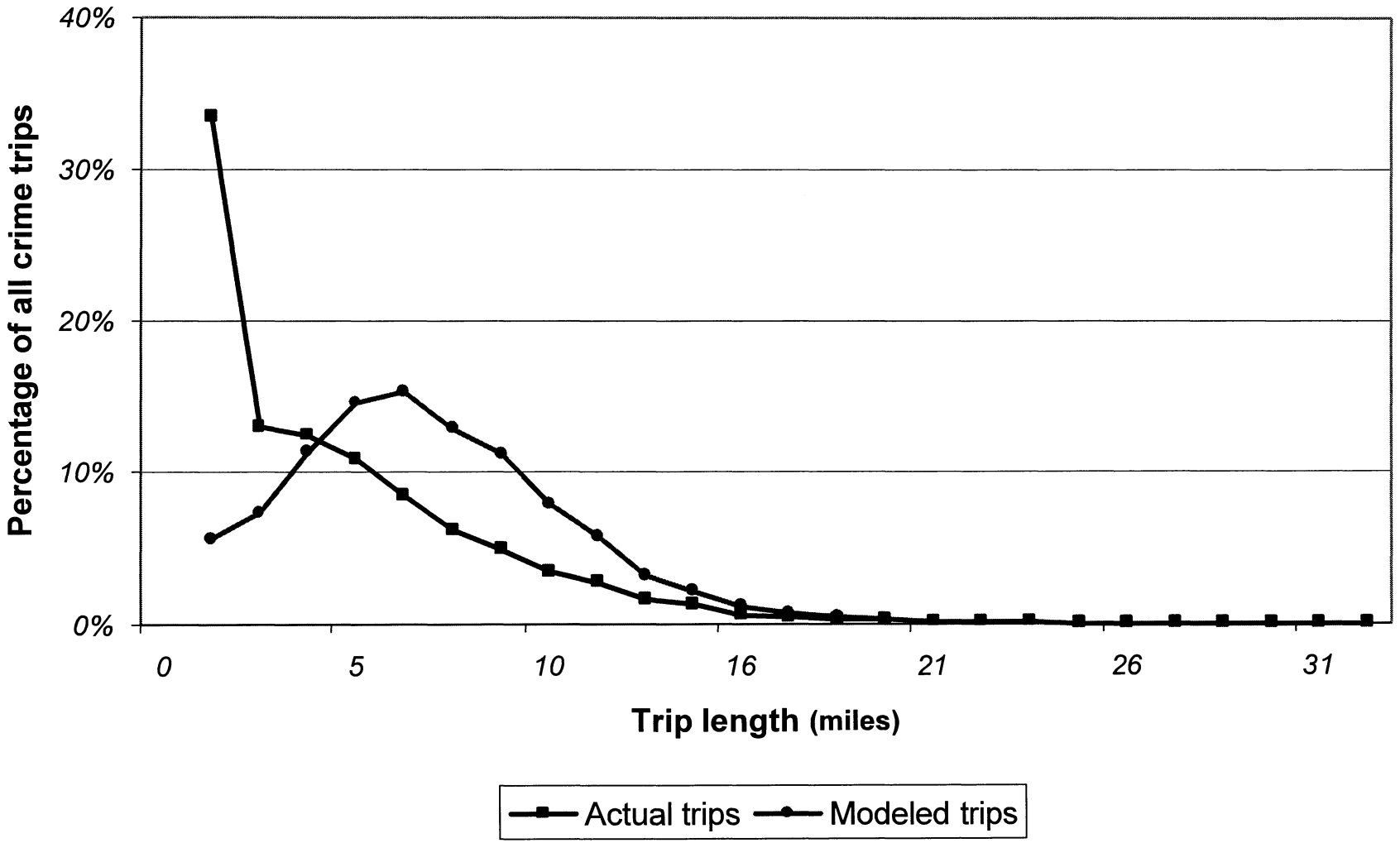


Figure 14.16:

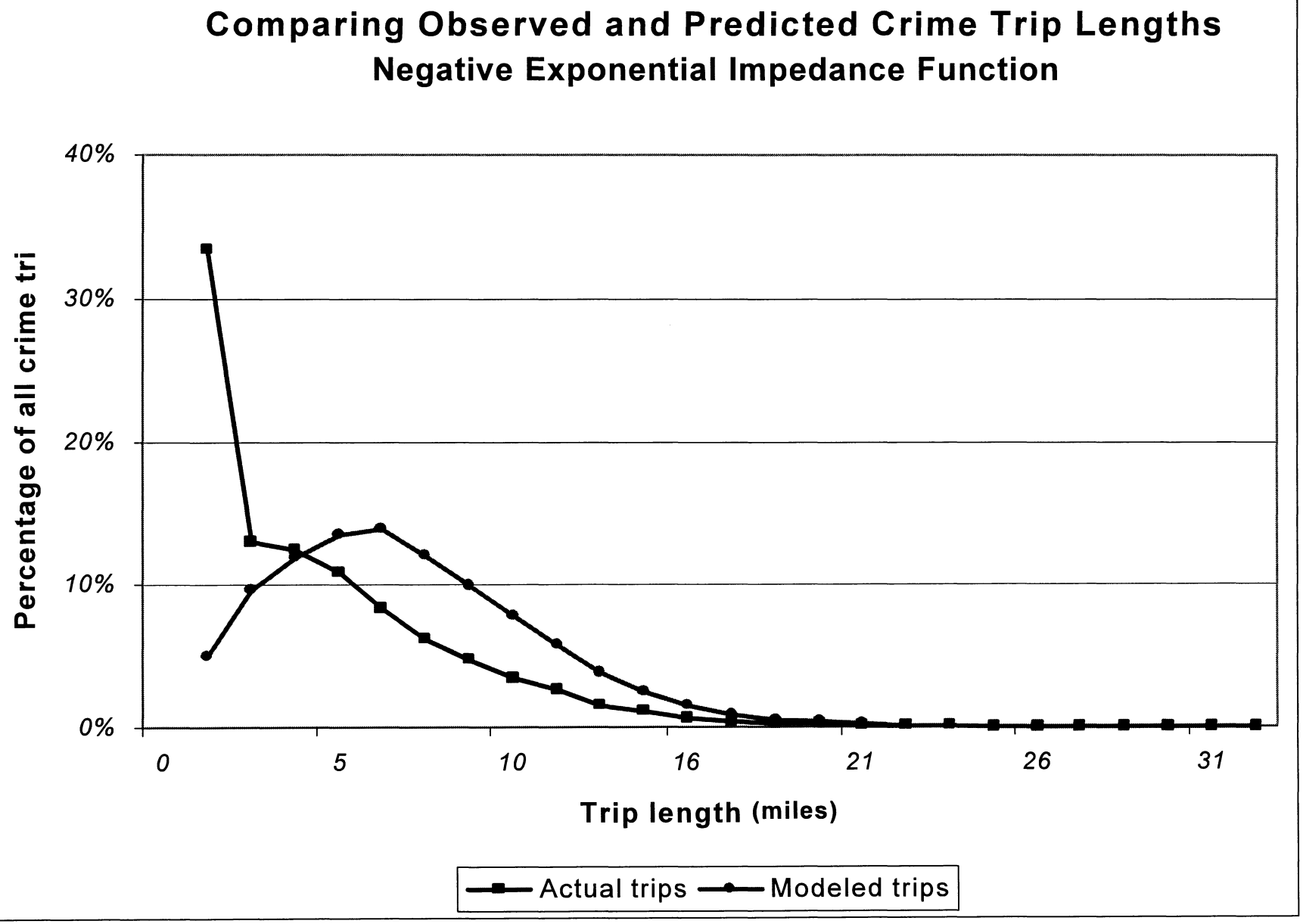


Figure 14.17:

Comparing Observed and Predicted Crime Trip Lengths
Truncated Negative Exponential Impedance Function

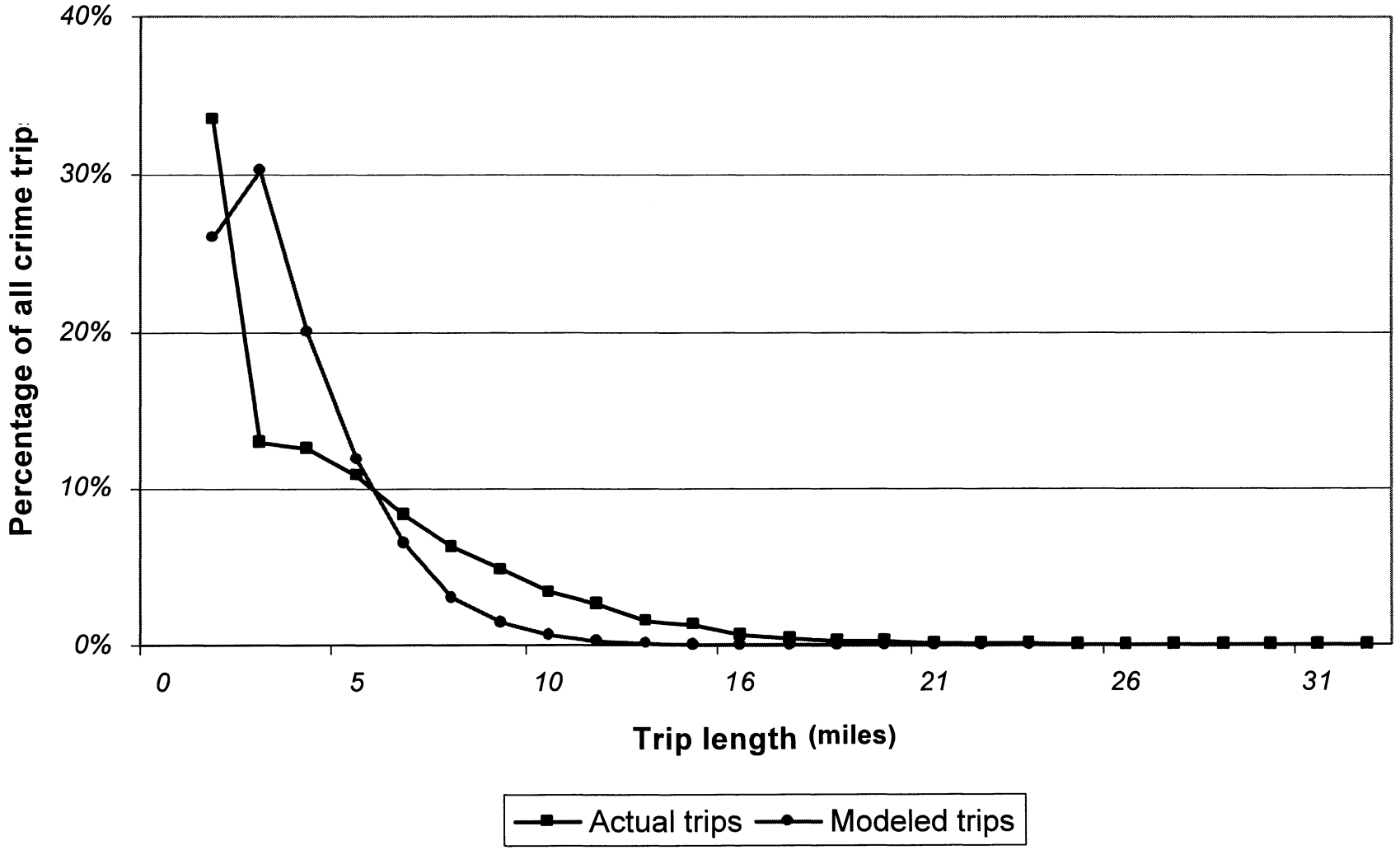
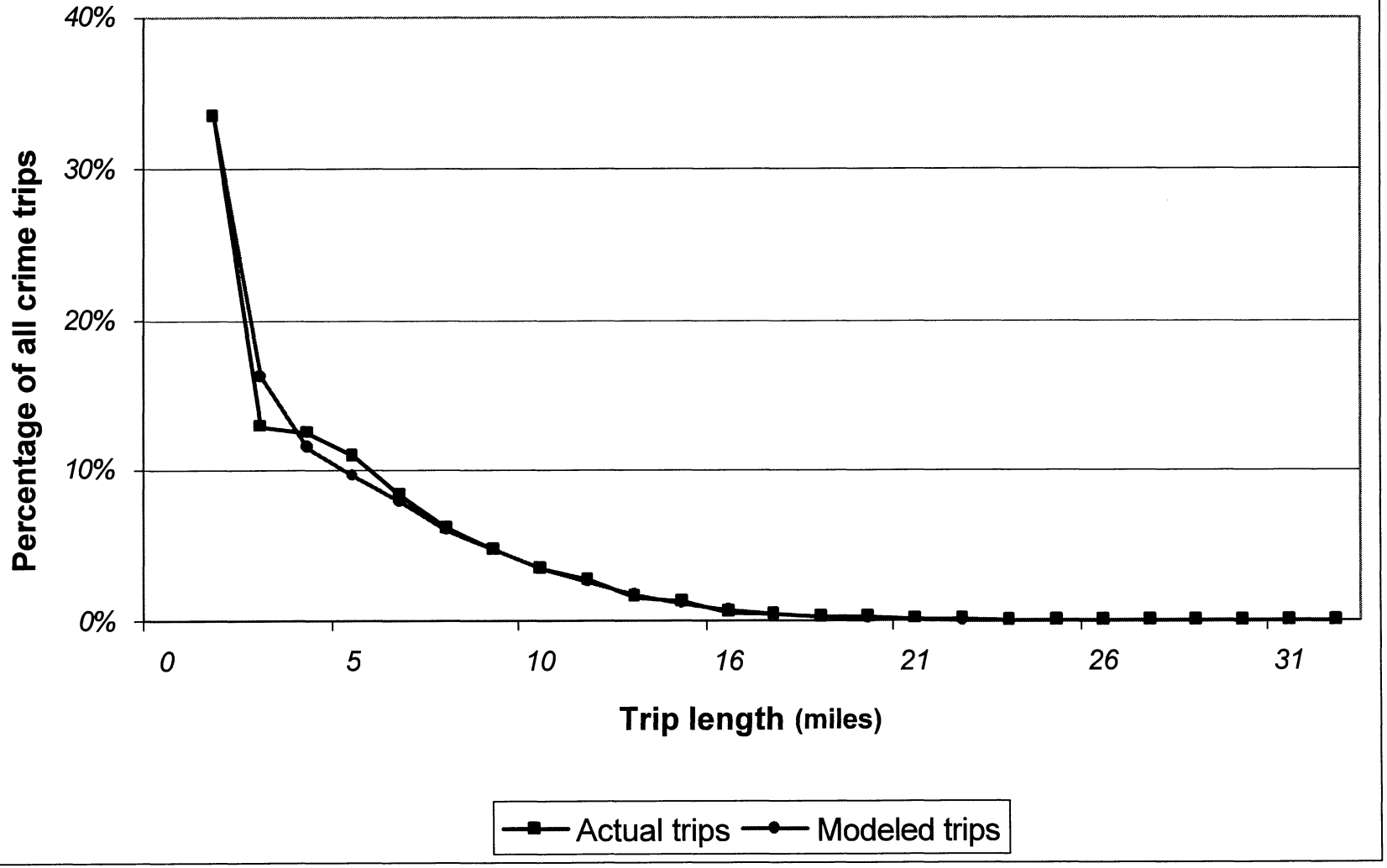


Figure 14.18:

Comparing Observed and Predicted Crime Trip Lengths
Lognormal Impedance Function



not all links are being considered in this test, a significance test of this statistic cannot be calculated since the sampling error is not known.

Using the observed (actual) links as the reference, the test calculates:

$$\text{Pseudo-chi square} = \sum_{i=1}^{K} \left[\frac{(P_i - O_i)^2}{O_i}\right] \qquad (14.17)$$

where P_i is the predicted number of trips for trip pair i, O_i is the observed (actual) number of trips for trip pair i, and i is the number of trip pairs that are compared up to K comparisons, where K is selected by the user.

Number of links to test

The number of top links that are to be compared depends on how skewed is the distribution. One good way to look at this is to plot the *rank size* distribution of the observed trips. Using the output 'dbf' file for the observed trip distribution (see "Calculate observed origin-destination trips" above), import the file into a spreadsheet. Sort the file in descending order of the trip frequency and create a new variable called "Rank order", which is simply the descending order of the trip frequencies. Then, plot the frequency of trips (FREQ) on the Y axis against the rank order of the trip pairs on the X axis.

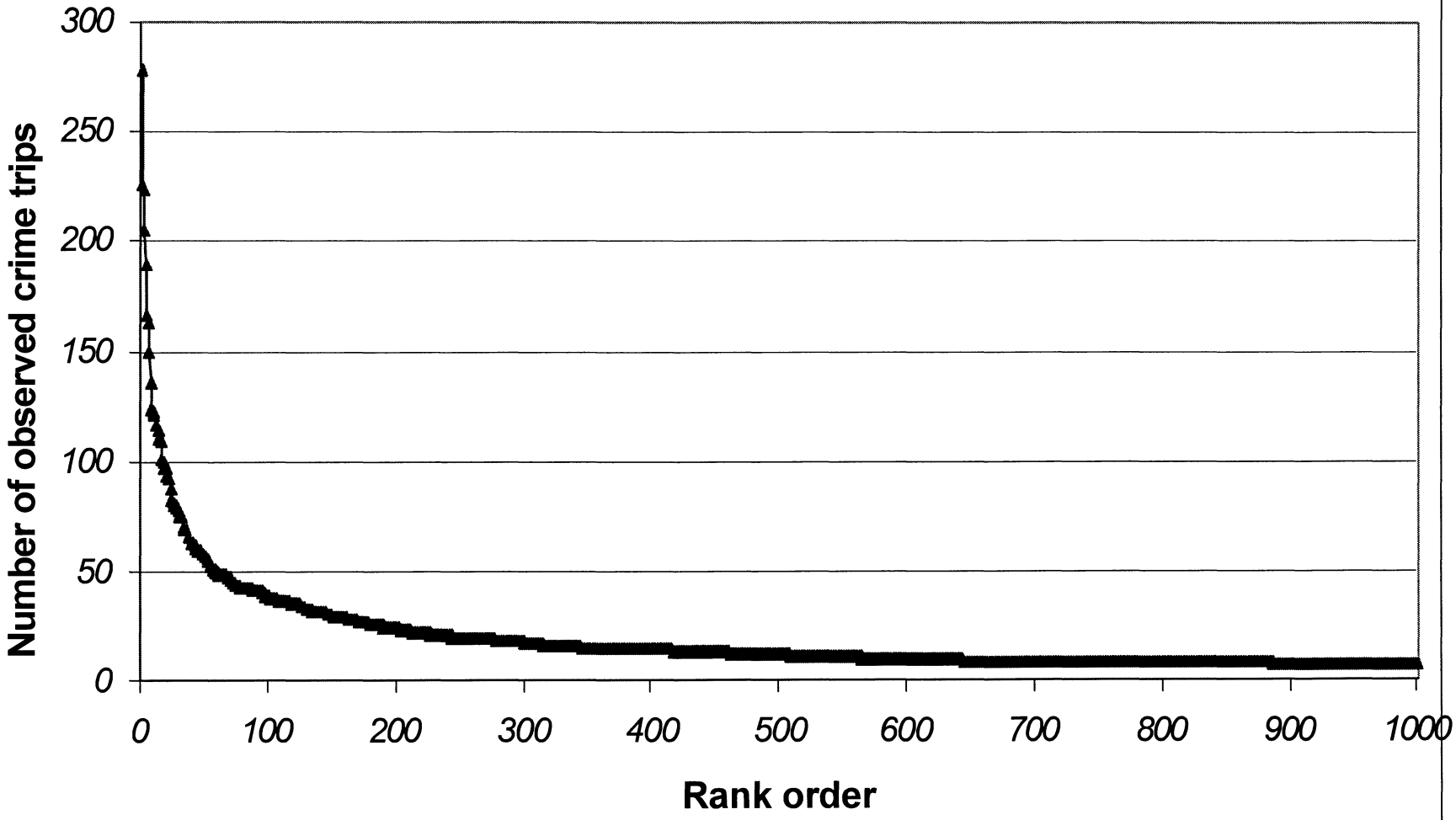
Figure 14.19 below shows the rank size distribution of the Baltimore County crime trips. Notice how the distribution is very skewed for the top crime trip pairs, but declines substantially after that. That is, the top trip link (which was an intra-zonal trip pair - zone 654 to itself) had 278 trips. The second top link (also an intra-zonal pair - zone 714 to itself) had 226 trips. The third had 223; the fourth had 205; and so forth. As mentioned above, the top 1000 trip links account for about 47% of all the trip in the matrix, but the first 176 account pairs account for half of that. In other words, if the top 150 to 200 trip pairs are examined, the highest volume links will be included and most of the skewness in the distribution will be accounted for. The remaining distribution, which is not fitted, will be less skewed.

Illustration

An illustration of how comparing the top links can modify a trip distribution model can be given. The same model as shown in figure 14.12 was run. The pseudo-Chi square test for the first 176 pairs was 5,832 (rounding-off to the nearest integer). However, by modifying the mean distance of the lognormal function a lower Chi square value was obtained. After several iterations, the lowest Chi square value was obtained for a mean distance of 5.2 miles ($\chi^2 = 5{,}448$).

Figure 14.19:

Rank Size Of Observed Trip Distribution



Again, the top links represents only one criteria out of the three mentioned. A good model should balance all three of these.

Optimizing the Three Evaluation Criteria

The ideal solution would be to have all three evaluation criteria minimized. That is, with an ideal model, there should be very little error between the predicted model and the observed distribution for the number of intra-zonal trips, the trip length distribution, and the top links.

In practice, it is unlikely that any one model will minimize all three types of errors. Thus, a balance (a compromise) must be obtained in order to produce an optimal solution. Since a balance can be obtained in different ways, there are multiple solutions possible.

> **Hint**: In CrimeStat, it is very easy to run through different models. The parameters are input on the "Setup origin-destination model page". The coefficients are calibrated in the "Calibrate origin-destination model" routine on the "Origin-Destination Model" page. The coefficient file which is output is then input into the "Apply predicted origin-destination model" routine on the same page. The comparison between the observed and predicted values is found in the "Compare observed and predicted origin-destination trip lengths" routine. Once set up, iterations of the models can be run very easily. A change is made on the setup page. The model is calibrated. It is then applied to the calibration data set. Finally, a comparison is made. Since the file names remain constant, an entire iteration takes less than a minute on a fast computer (1.6 Gb or faster).

To illustrate the multiple criteria, table 14.2 shows the best models for each of the three tests with variations on the mean distance in the model shown in figure 14.12. All other parameters were held constant. Many models were run to produce this table including testing other functions. These are the three best.

As seen, different models produce the lowest error for each of the criteria. For obtaining the closest fit to the number of intra-zonal trips, the mean distance of the lognormal function was 3.5 miles. For producing the best fit to the top 176 links, the mean distance for the best model was 5.2 models. For producing the best fit for the entire trip length distribution, the mean distance of the best model was 6.0 miles. The question is which one to use?

Table 14.2

Multiple Criteria in Selecting a Distribution Function

Lognormal function
Standard deviation = 4.7 miles
Coefficient = 1
Origin exponent = 1.0
Destination exponent = 1.06

Mean distance	Number of Intra-zonal Trips	Chi square for top 176 Links	Coincidence Ratio
Observed	8272	-	-
6.0	5463	5814	0.93
5.2	6296	5777	0.87
3.5	8275	5986	0.74

One solution for optimizing decisions

One possible solution is to optimize in the following way:

1. *If* the trip distribution matrix is highly skewed (which will occur with most crime data sets), then it's essential that the top links be replicated closely. This would take priority over the second criteria, which is minimizing the error for the trip length distribution, and the third criteria, which is minimizing the error in predicting intra-zonal trips.

2. Next fit the model to minimize the Chi square value for the top links. In the example above, this would be the top 176 pairs. Typically, the mean distance has the biggest impact for a lognormal or normal function and this would be adjusted first. For a negative exponential function, the exponent has the strongest impact. For a linear function, the slope has the strongest impact and for a truncated negative exponential, both the peak distance, for the near distance, and the exponent, for the far distance, has the biggest impacts (see chapter 9). Again, the aim is to produce the Chi square for the top links with the lowest value.

4. Then, while trying to maintain a Chi square value as close to this minimal value as possible, adjust the model to minimize the error in the trip length comparison. In this case, the model with the highest Coincidence Ratio is that which minimizes the error. For lognormal and normal functions, the standard deviation is the next parameter to adjust. For a negative exponential function, the coefficient should be adjusted next. For a linear function, the intercept would be adjusted next and for a truncated negative

exponential the slope would be adjusted next. Again, the aim should be to obtain the highest Coincidence Ratio without losing the fit for the top links.

5. Finally, if it is possible, adjust the exponents of the origins and destinations and the other parameters (e.g., the coefficient in the lognormal and normal distributions) to reduce the error in the total number of intra-zonal trips. Typically, however, these do not alter the results very much. They can be thought of as "fine tuning" adjustments.

Notice that this hierarchy fits the highest volume trip links first, then fits the overall trip length distribution, and finally fits the number of intra-zonal trips.

Illustration

To illustrate, we first start with the model that produced the lowest Chi square. That model used a lognormal function with a mean distance of 5.2 miles, a standard deviation of 4.7 miles, a coefficient of 1, an origin exponent of 1.0 and a destination exponent of 1.06. Varying the standard deviation of the lognormal function produced the following results (table 14.3).

Table 14.3

Minimizing the Second Criteria in Selecting a Distribution Function

Lognormal function
Mean distance = 5.2 miles
Standard Deviation = 4.6 miles
Coefficient = 1
Origin exponent = 1.0
Destination exponent = 1.06

Standard deviation	**Number of Intra-zonal Trips**	**Chi square for top 176 Links**	**Coincidence Ratio**
4.5	5809	5789	0.90
4.6	**6057**	**5779**	**0.88**
4.7 (baseline)	6296	5777	0.87
4.8	6526	5780	0.86
4.9	6746	5788	0.84

As the standard deviation was increased, the Coincidence Ratio decreased while the number of intra-zonal trips increased. Of these five different standard deviations, 4.5 produced the highest Coincidence Ratio, but also increased the Chi square statistic for the 176 top links. Since that criteria was set first, we don't want to loosen it substantially during the second adjustment. Consequently, a standard deviation of 4.6 was selected because this increased the Coincidence Ratio slightly while not substantially worsening the Chi square test (an increase of about 2).

Subsequent tests varying the coefficient of the lognormal function and the exponents of the origin and destination terms did not alter these values. Consequently, the final model that was selected is listed in table 14.4.

Table 14.14

Baltimore County Crime Trips: 1993-1997
Optimal Model Selected

Lognormal function
Mean distance = 5.2 miles
Standard deviation = 4.6
Coefficient = 1
Origin exponent = 1.0
Destination exponent = 1.06

The model was re-run with the new parameters used. The top 176 predicted trip links were output and were compared to the top 179 observed trip links (which exceeded 176 because of tied values). The top predicted 176 links accounted for 7,241 trips, or 17.3% of the total number of trips. The top observed 179 links accounted for 9,900 trip, or 23.6% of the total. Compared to the observed distribution, the top 176 predicted links accounted for a smaller proportion of the total trips.

However, the fit was generally better. Figure 14.20 shows the top predicted inter-zonal trip links and compares them to the top observed links while figure 14.21 shows the top predicted intra-zonal (local) trip links and compares them to the top observed intra-zonal links. Comparing these maps to figure 14.12 and 14.13 (which mapped the top 1000 links, not the top 176), the fit is a bit better for the major links, which is what we optimized. The fit is not perfect; it probably will never be. But, it is reasonably close.

Of course, this is not the only way to optimize and different users might approach it differently (e.g., minimizing the intra-zonal trips first, then the overall trip length distribution, and finally the top links). It has to be realized that optimizing in a different order will probably produce varying results; there is not, unfortunately, a single optimum solution to these three criteria. That is why it is important to explicitly define how an optimal solution will be obtained. In that way, users of the model can be cognizant of where the model is most accurate and where it is probably less accurate.

Implementing the Comparisons in *CrimeStat*

The mechanics of conducting the tests is fairly straightforward. The three tests are implemented in the "Compare Observed and Predicted Trip Lengths" routine on the last page of the Trip distribution module.

Figure 14.20:

Comparison of Predicted and Observed Crime Trips
Top Zone-to-Zone Trips from Optimized Model
All Crime Types

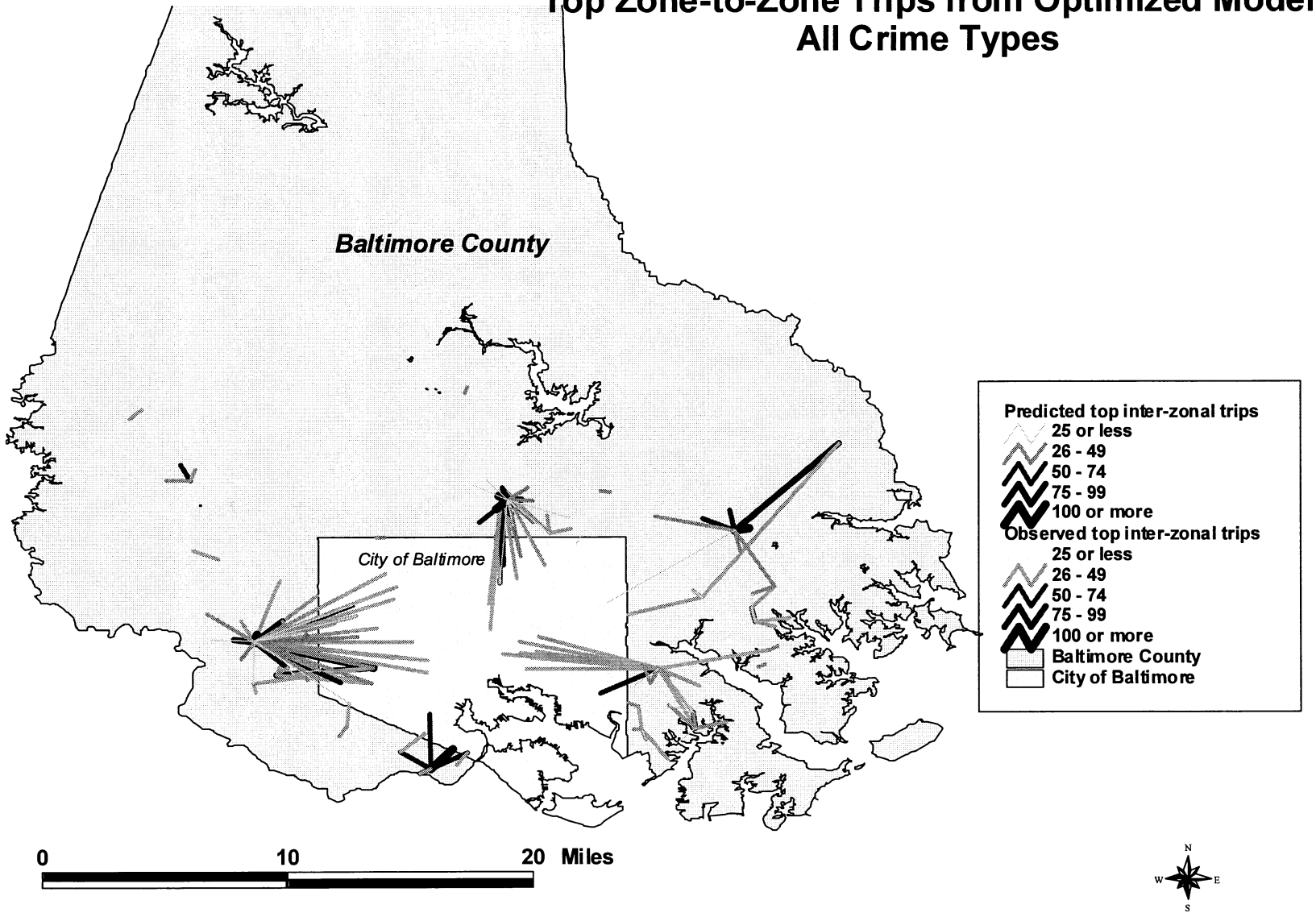
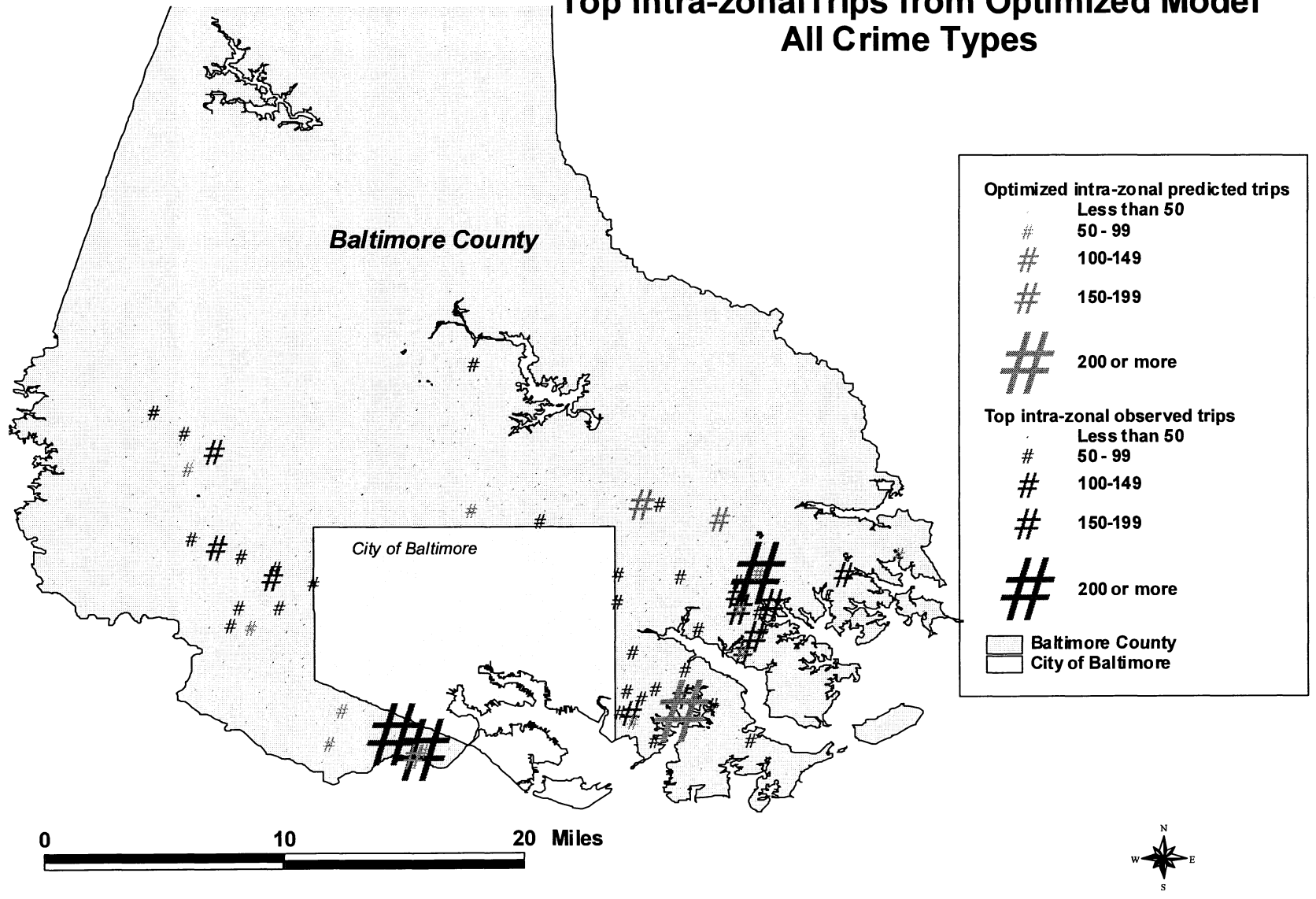


Figure 14.21:

Comparison of Predicted and Observed Crime Trips
Top Intra-zonalTrips from Optimized Model
All Crime Types



Observed trip file

Select the observed trip distribution file by clicking on the Browse button and finding the file.

Observed number of origin-destination trips

Specify the variable for the observed number of trips. The default name is FREQ.

Orig_ID

Specify the ID name for the origin zone. The default name is ORIGIN.

Note: the ID's used for the origin zones must be the same as in the destination file and the same as in the predicted trip file if the top links are to be compared.

Orig_X

Specify the name for the X coordinate of the origin zone. The default name is ORIGINX.

Orig_Y

Specify the name for the Y coordinate of the origin zone. The default name is ORIGINY.

Dest_ID

Specify the ID name for the destination zone. The default name is DEST.

Note: all destination ID's should be in the origin zone file and must have the same names and the same as in the predicted trip file if the top links are to be compared.

Dest_X

Specify the name for the X coordinate of the destination zone. The default name is DESTX.

Dest_Y

Specify the name for the Y coordinate of the destination zone. The default name is DESTY.

Predicted trip file

Select the predicted trip distribution file by clicking on the Browse button and finding the file.

Predicted number of origin-destination trips

Specify the variable for the observed number of trips. The default name is PREDTRIPS.

Orig_ID

Specify the ID name for the origin zone. The default name is ORIGIN.

Note: the ID's used for the origin zones must be the same as in the destination file and the same as in the observed trip file if the top links are to be compared.

Orig_X

Specify the name for the X coordinate of the origin zone. The default name is ORIGINX.

Orig_Y

Specify the name for the Y coordinate of the origin zone. The default name is ORIGINY.

Dest_ID

Specify the ID name for the destination zone. The default name is DEST.

Note: all destination ID's should be in the origin zone file and must have the same names and the same as in the observed trip file if the top links are to be compared.

Dest_X

Specify the name for the X coordinate of the destination zone. The default name is DESTX.

Dest_Y

Specify the name for the Y coordinate of the destination zone. The default name is DESTY.

Select bins

Specify how the bins (intervals) will be defined. There are two choices. One is to select a fixed number of bins. The other is to select a constant interval.

Fixed number

This sets a fixed number of bins. An interval is defined by the maximum distance between zone divided by the number of bins. The default number of bins is 25. Specify the number of bins.

Constant interval

This defines an interval of a specific size. If selected, the units must also be chosen. The default is 0.25 miles. Other distance units are nautical miles, feet, kilometers, and meters. Specify the interval size.

Compare top links

The "Compare top <value> links" dialogue implements a comparison of the top links. The user specifies the number of links to be compared. The default is 100. The routine calculates a Chi square statistic for these links.

> Note: in order to make the comparison, the origin and destination ID's must be the same for both the observed and predicted trip files.

Save comparison

The output is saved as a 'dbf' file specified by the user.

Table output

The table output includes summary information and:

1. The number of trips in the observed origin-destination file
2. The number of trips in the predicted origin-destination file
3. The number of intra-zonal trips in the observed origin-destination file
4. The number of intra-zonal trips in the predicted origin-destination file
5. The number of inter-zonal trips in the observed origin-destination file
6. The number of inter-zonal trips in the predicted origin-destination file
7. The average observed trip length
8. The average predicted trip length
9. The median observed trip length
10. The median predicted trip length
11. The Coincidence Ratio (an indicator of congruence varying from 0 to 1)
12. The D value for the Komolgorov-Smirnov two-sample test

13. The critical D value for the Komolgorov-Smirnov two-sample test
14. The p-value associated with the D value of Komolgorov-Smirnov two-sample test relative to the critical D value.
15. The pseudo-Chi square test for the top links

and for each bin:

16. The bin number
17. The bin distance
18. The observed proportion
19. The predicted proportion

File output

The saved file includes:

1. The bin number (BIN)
2. The bin distance (BINDIST)
3. The observed proportion (OBSERVPROP)
4. The predicted proportion (PREDPROP)

Graph

While the output page is open, clicking on the graph button will display a graph of the observed and predicted trip length proportions on the Y-axis by the trip length distance on the X-axis. This would produce a similar graph to that seen in figure 14.15 through 14.18 above.

Uses of Trip Distribution Analysis

There are a number of uses for the trip distribution analysis. First, for policing, an analysis of the actual (observed) trip distribution can be valuable. Second, the predicted model has value, above-and-beyond the analysis of the actual distribution.

Utility of Observed Trip Distribution Map

This information by itself can be very useful for police. Two applications will be discussed.

Crime prevention efforts

A major application is using the data shown in a trip distribution map to guide enforcement efforts. For example, in Baltimore County, with the crimes occurring at the five shopping malls, the origin locations can be more easily seen. This has utility for police. First, the police intervene more effectively on the routes leading from likely origin locations. They can patrol those routes more heavily and, perhaps, intervene more

frequently. By using the information from the trip distribution analysis, they make their enforcement efforts smarter. Second, they can conduct crime prevention efforts more effectively. By knowing the likely origin of offenders, intervention efforts in the origin zones may head off some of these incidents. Programs such as *weed-and-seed* and after-school programs depend on providing alternative facilities for youth, hoping to redirect them to more constructive activities. These facilities can be placed in locations where many crimes originate.

Improved Journey to crime analysis

A second application is in guessing the likely origin of a serial offender. In chapter 9, theories of travel behavior by a serial offender was discussed. The resulting analysis (geographic profiling, Journey to crime analysis) utilized information on the distribution of incidents committed by the offender. On the other hand, the trip distribution pattern seen in figure 14.4 provides a probability map of offender locations and gives more information than was evident in the Journey to crime model. That model assigned a likelihood of the offender living at a location (the origin) on the basis of the distribution of the incidents. There was no additional information used about likely origin locations. This trip distribution map, on the other hand, points to certain zones as being the likely origin for offenses committed at the major destination locations. There is more 'structure' in this analysis than in the Journey to crime logic.

One can think of this in terms of a quasi-Bayesian approach to guessing the likely origin of an offender. The geographic profiling/Journey to crime logic assumes no *prior probabilities*. The only information that is used is the distribution of crimes committed by a serial offender and a model of crime travel distance (essentially, an impedance function). The trip distribution map, on the other hand, points to certain locations as being the likely origin for incidents. Admittedly, this is based on a large sample of cases rather than one particular serial offender. But, the map points to certain prior probabilities for an origin location. If an analyst could combine those approaches - using a prior probability map along with the distribution of incidents committed by a serial offender, a more realistic and accurate guess about the offender's residence location could be obtained.

In other words, the empirical description of crime travel patterns is useful for policing, above-and-beyond any modeling that is developed.

Utility of Predicted Trip Distribution Map

The model also has a lot of utility for both policing and crime analysis. A number of examples will be given. First, it can be used for **forecasting**. By calibrating the model on one data set, it be applied to a future data set. As mentioned in chapter 12, much of the population and employment data that form the basis of a trip generation model comes from a Metropolitan Planning Organization (MPO). Most MPOs in the United States also make forecasts of future population and employment. Those forecasts can be, in turn, converted into forecasts of future crime origins and crime destinations. Thus, on the assumption that the distribution trends will remain the same over time, the trip distribution model can be

applied to the forecast set of origins and destinations. This could allow an examination of possible changes in the crime distribution (assuming that the future forecasts are correct and that the trip distribution coefficients remain constant).

Second, a model of crime trip distribution can be useful for modeling **changes in land uses**. For example, if a new shopping mall is being planned, one can take the existing trip generation model and adjust it to fit the planned situation (e.g., adding 500 retail jobs to the zone in which the mall is being developed). Then, the trip generation model is re-run with the new expected data, and the trip distribution model is applied to the predicted crime origins and crime destinations. The result would be a model of likely crime trips to the new shopping mall. This can be useful to the mall developers, to future businesses, and to the police. If it turns out that the model forecasts there will be a sizeable number of crime trips to that mall, then preventive actions can be developed before the mall is built (e.g., improving security design in the mall; improving the parking lot arrangement).

Third, a model of crime trip distribution can help in analyzing **future interventions**. For example, increasing police patrols in a high crime attraction area can be examined as to possible effectiveness before taking the trouble to reorganize deployment. Or, adding a new drug treatment center or a new youth center can be modeled as to its possible effectiveness in changing the nature of crime trips. Again, the input is at the data level, which affects the trip generation model. But the trip distribution model is applied to the new outputs from the trip generation model. The advantage of a model is that it explores a set of interventions without having to actually having to implement them; it's a 'thinking' tool for planning change.

Fourth, and finally, a crime trip distribution model is helpful in developing **crime theory**. As indicated in chapter 11, the theory of crime travel has been very elementary up to now. The primary focus of analysis has been only on the destinations and on the trip lengths as measured by distance traveled. A trip distribution model, on the other hand, analyzes both trip destinations and trip origins, and can include a more sophisticated measure of impedance than simple distance. Because the analysis is conducted over a larger area (a jurisdiction or a metropolitan area), the hierarchy of crime trips can be analyzed simultaneously and the interaction between origins and destinations can be examined. In short, a crime trip distribution model is a 'quantum leap' in sophistication and complexity compared to the usual Journey to crime types of models. Hopefully, it will generate even more sophisticated types of models.

At this point, we don't have any examples of the use of a crime trip distribution model for policy intervention or analysis of land use changes. The focus has been on putting the model together and ensuring that the routines work properly. But, it is hoped that these applications will be presented in future versions of the program.

The next chapter continues the travel demand model by examining how crime trip links are split into different travel modes. That is, the trip distribution model estimates

the number of trips flowing from each origin zone to each destination zone. The mode split model then breaks these trips into distinct travel modes.

Endnotes for Chapter 14

1. Distance can be used as a rough approximation for impedance, but is rarely a good predictor of actual travel behavior. For example, in the mode split mode that will be discussed in chapter 15, the distance between a location and the nearest bus or rail route can be used to quickly select trip pairs that might travel by transit. However, the actual prediction must be based on a network calculation of travel time or travel cost in traversing the system.

2. Most of the research on factors affecting use of transit were conducted in the 1960s and 1970s. These assumptions are more or less assumed by travel demand modelers, rather than documented *per se*. See Schnell, Smith, Dimsdale, and Thrasher, 1973; Roemer and Sinha, 1974; WASHCOG, 1974; Carnegie-Mellon University, 1975; Johnson, 1978; Levine and Wachs, 1986b for some examples.

Chapter 15
Mode Split

In this chapter, the third modeling step in the crime travel demand model is discussed, mode split. *Mode split* involves separating (splitting) the predicted trips from each origin zone to each destination zone into distinct travel modes (e.g., walking, bicycle, driving, train, bus).

This model has both advantages and disadvantages for crime analysis. At a theoretical level, it is the most developed of the four stages since there has been extensive research on travel mode choice. For crime analysis, on the other hand, it represents the 'weakest link' in the analysis since there is very little available information on travel mode by offenders. Since researchers cannot interview the general public in order to document crimes committed by respondents nor, in most cases, even interview offenders after they have been caught, there is very little information on travel mode by offenders that has been collected.[1] Consequently, we have to depend on the existing theory of travel mode choice and adapt it intuitively to crime data. The approach is solely theoretical and depends on the validity of the existing theory and on the intuitiveness of guesses. Hopefully, in the future, there will be more information collected that would allow the model to be calibrated against some real data. But, for the time being, we are limited in what can be done.

Theoretical Background

The theoretical background behind the mode split module is presented first. Next, the specific procedures are discussed with the model being illustrated with data from Baltimore County.

Utility of Travel and Mode Choice

The key aim of mode choice analysis is to distinguish the travel mode that travelers (or, in the case of crime, offenders) use in traveling between an origin location and a destination location. In the travel demand model, the choice is for travel between a particular origin zone and a particular destination zone. Thus, the trips that are distributed from each origin zone to each destination zone in the trip distribution module are further split into distinct travel modes.

With few exceptions, the assumption behind the mode split decision is for a two-way trip. That is, if an offender decides on driving to a particular crime location, we normally assume that this person will also drive back to the origin location. Similarly, if the offender takes a bus to a crime location, then that person will also take the bus back to the origin location. There are, of course, exceptions. A car thief may take a bus to a crime location, then steal a car and drive back. But, in general, without information to the contrary, it is assumed that the travel mode is for a round trip journey.

Underlying the choice of a travel mode is assumed to be a *utility function*. This is a function that describes the benefits and costs of travel by that mode (Ortuzar and Willumsen, 2001). This can be written with a conceptual equation:

$$\text{Utility} = F(\text{benefits, costs}) \tag{15.1}$$

where 'f' is some function of the benefits and the costs. The benefits have to do with the advantages in traveling to a particular destination from a particular origin while the costs have to do with the real and perceived costs of using a particular mode. Since the benefits of traveling a particular destination from a particular origin are probably equal, the differences in utility between travel modes essentially represent differences in costs. Thus, equation 15.1 breaks down to:

$$\text{Utility cost} = F(\text{costs}) \tag{15.2}$$

If different travel modes (e.g., driving, biking, walking) are each represented by a separate utility cost function, then they can be compared:

$$\text{Utility cost}_1 = F_1(\text{cost}_1 + \text{cost}_2 + \text{cost}_3 + ...+\text{cost}_k) \tag{15.3a}$$

$$\text{Utility cost}_2 = F_2(\text{cost}_1 + \text{cost}_2 + \text{cost}_3 + ...+\text{cost}_k) \tag{15.3b}$$

$$\text{Utility cost}_3 = F_3(\text{cost}_1 + \text{cost}_2 + \text{cost}_3 + ...+\text{cost}_k) \tag{15.3c}$$

.
.
.

$$\text{Utility cost}_L = F_L(\text{cost}_1 + \text{cost}_2 + \text{cost}_3 + ...+\text{cost}_k) \tag{15.3d}$$

where Utility cost_1 through Utility cost_L represents L distinct travel modes, cost_1through cost_k represent k cost components and are variables, and F_1 through F_L represent L different utility functions (one for each mode).

There are several observations that can be made about this representation. First, each of the cost components can be applied to all modes. However, the cost components are variables in that the values may or may not be the same. For example, if cost_1 is the operating cost of traveling from an origin to a destination, the cost for a driver is, of course, a lot higher than for a bus passenger since the latter person shares that cost with other passengers. Similarly, if cost_2 is the travel time from a particular origin zone to a particular destination zone, then travel by private automobile may be a lot quicker than by public bus. As mentioned in the last chapter, time differences can be converted into costs by applying some type of hourly wage/price to the time. To take one more example, for driving mode, there could be a cost in parking (e.g., in a central business district); for transit use, on the other hand, this cost component is zero. In other words, each of the travel modes has a different cost structure. The same costs can be enumerated, but some of them will not apply (i.e., they have a value of 0).

Second, the costs can be perceived costs as well as real costs. For example, a number of studies have demonstrated that private automobile is seen as far more convenient to most people than a bus or train (e.g., see Schnell, Smith, Dimsdale, and Thrasher, 1973; Roemer and Sinha, 1974; WASHCOG, 1974; Carnegie-Mellon University, 1975; Johnson, 1978; Levine and Wachs, 1986b). 'Convenience' is defined in terms of ease of access and effort involved in travel (e.g., how long it takes to walk to a bus stop from an origin location, the number of transfers that have to made to reach a final destination, and the time it takes to walk from the last bus stop to the final destination). While it is sometimes difficult to separate the effects of convenience from travel itself, it is clear that most people perceive this as dimension in travel choice. In turn, convenience can be converted into a monetary value in order to allow it to be calculated in a cost equation, for example how much people are willing to pay in time savings to yield an equivalent amount of convenience (e.g., asking how many more minutes in travel time by bus an individual would be willing to absorb in order to give up having to drive).

Third, these costs can be considered at an aggregate as well as individual level. At an aggregate level, they represent average or median costs (e.g., the average time it takes to travel between zone A and zone B by private automobile, bus, train, walking, or biking; the average dollar value assigned by a sample of survey respondents to the convenience they associate in traveling by car as opposed to bus).

On the other hand, at an individual level, the costs are specific to the individual. For example, travel time differences between car and bus can be converted into an hourly wage using the individual's income; someone making $100,000 a year is going to price that time savings differently than someone making only $25,000 a year.

Fourth, a more controversial point, the specific mathematical function that ties the costs together into a particular utility function may also differ. Typically, most travel demand models have assumed that a similar mathematical function is used for all travel modes; this is the negative exponential function described below (Domencich and McFadden, 1975; Ortuzar and Willumsen, 2001). However, there is no reason why different functions cannot be used. Thus, the equations above identify different functions for the modes, F_1 through F_L. One can think of this in terms of *weights*. Each of the different mathematical function weight the cost components differently.

It is an empirical question whether individuals apply different functions to evaluating the different modes. For example, most people would not drive just to travel one block (unless it was pouring rain or unless a heavy object had to be delivered or picked up). Even though it is convenient to get into a vehicle and drive the one block, most people see the effort involved (and, most likely, the fuel and oil costs) as not being worth it.

In other words, it appears that a different utility function is being applied to walking as opposed to driving (i.e., walk for distances up to a certain distance; drive thereafter). A strict utility theorist might disagree with this interpretation saying that the per minute cost of walking the one block and back was less than monetarized per minute cost of operating the vehicle (which may include opening a garage door, getting into the

vehicle, starting the vehicle, driving out of the parking spot, closing the garage door, and then driving the one block). In other words, it could be argued that the difference in behaviors has to do with the values of the different cost components, rather than the way they are *weighted* together (the mathematical function). In retrospect, one can explain any difference. We argue in this chapter, however, that crime trips appear to show different likelihoods by travel mode and that treating each of these functions as distinct allows more flexibility in the framework.

Discrete Choice Analysis

No matter how the utility functions are defined, they have to be combined in such a way as to allow a discrete choice. That is, an offender in traveling from zone A to zone B makes a discrete choice on travel mode. There may be a probability for travel by each mode, for example 60% by car and 40% by bus. But, for an individual, the choice is car or bus, not a probability. The probabilities are obtained by a sample of individuals, for example of 10 individuals 6 went by car and 4 went by bus. But, still, at the individual level, there is a distinct choice that was made.

Multinomial Logit Function

A common mathematical framework that used is for mode choice modeling at an aggregate level is the *multinomial logit function* (Domincich and McFadden, 1975; Stopher and Meyburg, 1975; Oppenheim, 1980; Ortuzar and Willumsen, 2001):

$$P_{ijL} = \frac{e^{(-\beta C_{ijL})}}{\sum_{L=1}^{P} [\, e^{(-\beta C_{ijL})}]} \qquad (15.4)$$

where P_{ijL} is the probability of using a mode for any particular trip pair (particular origin and particular destination) L is the travel mode, C_{ij} is the cost of traveling from origin zone i to destination zone j, e is the base of the natural logarithm, and β is a coefficient.

Several observations can be made about this function. First, each travel mode, L, has its own costs and benefits, and can be evaluated by itself. That is, there is a distinct utility function for each mode. This is the numerator of the equation, $e^{(-\beta C_{ijL})}$. However, the choice of any one mode is dependent on its utility value relative to other modes (the denominator of the equation). The more choices that are available, obviously, the less likely an individual will use that mode. But the value associated with the mode (the utility) does not change. As mentioned above, we generally assume that the benefit of traveling between any two zones is identical for all modes and, hence, any differences are due to costs.

Second, the mathematical form is the negative exponential. The exponential function is a growth function in which growth occurs at a constant *rate* (either positive - growth, or negative - decline). The use of the negative exponential assumes that the costs are related to the likelihood as a function that declines at a constant rate. It is actually a 'disincentive' or 'discount' function rather than a utility function, *per se*. That is, as the costs increase, the probability of using that mode decreases, all other things being equal. Still, for historical reasons, it is still called a utility function.

Third, for any one mode, the total cost is a logarithmic function of individual costs:

$$\text{Utility cost}_i = e^{(-\beta C_{ijL})} \tag{15.5}$$

$$\text{Ln(Utility cost}_L) = C_{ijL} = \alpha + \beta_1 X_1 + \beta_2 X_2 + \ldots\ldots + \beta_k X_k \tag{15.6}$$

where C_{ijL} is a cumulative cost made up of components X_1, X_2 through X_k, α is a constant, and β_1 through β_k are coefficients for the individual cost components. Thus, we see that the utility function is a loglinear model, as was seen in chapter 12. Thus, the utility function is Poisson distributed, declining at a constant *rate* with increasing cumulative costs. Domincich and McFadden (1975) suggest that the error terms are not Poisson distributed, but skewed in a Weibul function. As discussed in chapter 12, there are a variety of different models that incorporate skewed error terms (negative binomial, a simple linear correction of dispersion) so that the Weibul is but one of a number of possible descriptors. Nevertheless, the mean utility is a Poisson-type function.

Generalized Relative Utility Function

One can generalize this further to allow any type of mathematical function. While the Poisson has a long history and is widely used, allowing other non-linear functions allows greater flexibility. It is possible that individuals apply different *weighting* systems in evaluating different modes (e.g., a negative exponential for walking, but a lognormal function for driving). We certainly see what appear to be different functions when the actual travel behavior of individuals are examined (e.g., homeless individuals don't walk everywhere even though the cost of walking long distances is cheaper in travel time than taking a bus[2]; people don't drive or take a bus for very short distances, say a block or two). Therefore, if we allow that there are different travel functions for different modes, then more flexibility is possible than by assuming a single mathematical function.

We can, therefore, write a *generalized relative utility function* as:

$$P_{ijL} = \frac{F_L(-\beta C_{ijL})}{\sum_{L=1}^{P}[F_L(-\beta C_{ijL})]} = \frac{I_{ijL}}{\sum_{L=1}^{P}[I_{ijL}]} \tag{15.7}$$

where the terms are the same as in 15.4 except the function, F_L, is some function that is specific to the travel mode, L. The numerator is defined as the impedance of mode L in traveling between two zones i and j, while the denominator is the sum of all impedances.

Notice that the ratio of the cost function for one mode relative to the total costs is also the ratio of the impedance for mode L relative the total impedance. The total impedance was defined in chapter 14 as the disincentive to travel as a function of separation (distance, travel time, cost). We see that the share of a particular mode, therefore, is the proportion of the total impedance of that mode. This share will vary, of course, with the degree of separation. For any given separation, there will usually be a different share for each mode. For example, at low separation between zones (e.g., zones that are next to each other), walking and biking are much more attractive than taking a bus or a train and, perhaps even driving. At greater separation (e.g., zones that are 5 miles apart), walking and biking are almost irrelevant choices and the likelihood of driving or using public transit is much greater. In other words, the share that any one mode occupies is not constant, but varies with the impedance function.

Why then can't we estimate the mode split directly at the trip distribution stage? If the trip distribution function is

$$T_{ij} = \alpha P_i^{\lambda} \beta A_j^{\tau} I_{ij} \qquad \text{(14.12 repeat)}$$

and if these trips, in turn, are split into distinct modes using equation 15.7, couldn't 14.12 be re-written as

$$T_{ijL} = \alpha P_i^{\lambda} \beta A_j^{\tau} I_{ijl} \qquad (15.8)$$

where T_{ijL} is the number of trips between two zones, i and j, by mode L, P_i is the production capacity of zone i, A_j is the attraction of zone j, α and β are constants that are applied to the productions and attractions respectively, λ and τ are 'fine tuning' exponents of the productions and attractions respectively, and I_{ijL}, is the impedance of using mode L to travel between the two zones? The answer is, yes, it could be calculated directly. If I_{ijL} was a perfectly defined mode impedance function (with no error), then the mode share could be calculated directly at the distribution stage instead of separating the calculations into two distinct stages. The problem, however, is that the impedance functions are never perfect (far from it, in fact) and that re-scaling is required both to get the origins and destinations balanced in the trip distribution stage and to ensure that the probabilities in equation 15.7 add to 1.0. The effect of these adjustments generally throws off a model such as 15.8.[3] Consequently, the trip distribution and mode split stages are usually calculated as separate operations.

Measuring Travel Costs

The next question is what types of travel costs are there that define impedance? As mentioned above, there are real as well as perceived costs that affect a travel mode

decision. Some of these can be measured easily, while others are very difficult requiring detailed surveys of individuals. Among these costs are:

1. Distance or travel time. As mentioned throughout this discussion, distance is only a rough indicator of cost since it is invariant with respect to time. Actual travel time is a much better indicator because it varies throughout the day and can be easily converted into a *travel time value*, for example by multiplying by a unit wage.

2. Other real costs, such as the operating costs of a private vehicle (fuel, oil, maintenance), parking, and insurance. Some of these can be subsumed under travel time value by working out an hourly price for travel.

3. Perceived costs, such as convenience, fear of being caught by an offender, ease of escape from a crime scene, difficulties in moving stolen goods, and fear of retaliation by other offenders or gangs).

Some of these costs can be measured and some cannot. For example, the value of travel time can be inferred from the median household income of a zone for aggregate analysis or from the actual household income for individual-level analysis. Parking can be averaged by zone. Insurance costs can be estimated from zone averages *if* the data can be obtained.

Many perceived costs also can be measured. Convenience, for example, could be measured from a general survey. The fear of being caught can be inferred from the amount of surveillance in a zone (e.g., the number of police personnel, security guards, security cameras). Even though it may be a difficult enumeration process, it is still possible to measure these costs and come up with some average estimate.

Other perceived costs, on the other hand, may not be easily measured. For example, the fear an offender belonging to one gang has about retaliation from another gang is not easily measured. Similarly, the costs in moving stolen goods by a thief is not easily measured; one would need to know the location of the distributors of these goods.

In practice, travel modelers make simple assumptions about costs because of the difficulty in measuring many of them. For example, travel time is taken as a proxy for all the operating costs. Parking costs can be incorporated through simple assumptions about the distribution across zones (e.g., zones within the central business distract - CBD, are given an average high parking costs; zones that are central, but not in the CBD, are assigned moderate parking costs; zones that are suburban are assigned low parking costs). It would be just too time consuming to document each and every cost affecting travel behavior, particularly if we are developing a model of offender travel.

Nevertheless, theoretically, these are all potentially measurable costs. They are real and probably have an impact in the travel decisions that offenders make. As

researchers, we have to work towards articulating as many of these costs as possible in order to produce a realistic representation of offender travel.

Aggregate and Individual Utility Functions

One of the big debates in travel modeling is whether to use aggregate or individual utility functions to calculate mode share. The aggregate approach measures common costs for each zone, assuming an average value. The disaggregate approach (sometimes called 'second generation' models) measures unique costs for individuals, then sums upward to yield values for each zone pair. Even though the end result is an allocation of costs to each zone pair, the articulation of unique costs at the individual level can, in theory, allow a more realistic assessment of the utility function that is applied to a region.

The aggregate approach will measure costs by averages. Thus, a typical equation for driving mode might be:

$$\text{Total cost}_{ij} = \alpha + \beta_1 T_{ij} + \beta_2 P_j \qquad (15.9)$$

where T_{ij} is the average travel time between two zones, i and j, and P_{ij} is the average parking cost for parking in zone j. Notice that there are a limited number of variables in an aggregate model (in this case, only two) and that the assigned average is for an entire zone. Notice also that the parking cost is applied only to the destination zone. It is assumed that any traveler will pay that fee in that zone irrespective of which origin zone he/she came from.

A disaggregate approach can allow more cost components, if they are measured. Thus, a typical equation for driving mode might be:

$$\text{Total cost}_{ijk} = \alpha + \beta_1 T_{ijk} + \beta_2 P_j + \beta_3 C_{ijk} + \beta_4 CM_{ijk} + \beta_5 S_{ijk} \qquad (15.10)$$

where T_{ijk} is the travel time for individual *k* between two zones, i and j, P_{ij} is the average parking cost for parking in zone j, C_{ijk} is the convenience of traveling to zone j from zone i for individual *k*, CM_{ijk} is the comfort and privacy experienced by individual *k* in traveling from zone i to zone j, and S_{ijk} is the perceived safety experienced by individual *k* in traveling from zone i to zone j. Notice that there are more cost variables in the equation and that the model is targeted specifically to the individual, *k*. Two individuals who live next door to each other and who travel to the same destination may evaluate these components differently. If these individuals have substantially different incomes, then the value of the travel time will differ. If one values privacy enormously while the other doesn't, then the cost of driving for the first is less than for the second. Similarly, convenience is affected by both travel time and the ease of getting in and out of vehicle. Finally, the perception of safety may differ for these two hypothetical individuals. There are many studies that have documented the significant role played by safety in affecting, particularly, transit trips (Levine and Wachs, 1986b).

In other words, the aggregate approach applies a very elementary type of utility function whereas the disaggregate approach allows much more complexity and individual variability. Of course, one has to be able to measure the individual cost components, a difficult task under most circumstances.

There is also a question about which approach is more accurate for correctly forecasting actual mode splits. Historically, most Metropolitan Planning Organizations have used the aggregate method because it's easier. However, more recent research (Domincich and McFadden, 1975; Ben-Akiva and Lerman, 1985; McFadden, 2002) has suggested that the disaggregate modeling may be more accurate. At the very minimum, the disaggregate is more amenable to policy interpretations because it is more behavioral. *If* one could interview travelers with a survey, then it is possible to explore the variety of cost factors that affect a decision on both destination and mode split, and a more realistic (if not unique) utility function derived.

But, as mentioned above, with crime trips, this is very difficult, if not impossible, to do. Consequently, for the time being, we're stuck with an aggregate approach towards modeling the utility of travel by offenders.

Relative Accessibility

For this version of *CrimeStat*, an approximation to a utility function was created. The approach is to estimate a *relative accessibility* function and then apply that function to the predicted trip distribution. The relative accessibility function is a mathematical approximation to a utility function, rather than a measured utility function by itself. Because the cost components cannot be measured, at least for offenders, we use an inductive approach. Reasonable assumptions are made and a mathematical function is found that fits these assumptions.

It is a plausible model, not an analytical one. The plausibility comes by making reasonable assumptions about actual travel behavior. One can assume that walking trips will occur for short trips, say under two miles. Bicycle trips, on the other hand, could occur over longer distances, but will still be relatively short (also, there is always the risk of traffic on the safety of bicycle trips). Transit trips (bus and train) will be used for moderately long distances but require an actual transit network. Finally, driving trips are the most flexible because they can occur over any size distance and road network. They are less likely to be used for very short trips, on the other hand, due to reasons discussed above.

Hierarchical Approach to Estimating Mode Accessibility

Using this approach, specific steps can be defined to produce a plausible accessibility model. To help in establishing a model, an Excel spreadsheet has been developed for making these calculations (*Estimate mode split impedance values.xls*). It can also be downloaded from the *CrimeStat* download page. The spreadsheet has been defined with respect to distance, but it can be adapted for any type of impedance (travel time or

cost). A spreadsheet has been used because it is more flexible than incorporating it as a routine in *CrimeStat* to estimate the parameters. There is not a single solution to the parameters estimates, and the different choices can be seen more easily in a spreadsheet.

Define target proportions

First, define the *modes*. In the *CrimeStat* mode split routine, up to five different modes are allowed. These have default names of "Walk", "Bike", "Drive", "Bus", and "Train". The user is not required to use these names nor all five modes. Clearly, if there is not a train system in the study area, then the "Train" mode does not apply. Travel modelers use variations on these, such as "drive alone","carpool", "automobile", "motorcycle", and so forth.

Define target proportions

Second, define the *target proportions*. These are the expected proportions of travel for each mode. Where would such proportions come from? There have been many studies of driving and transit behavior, but relatively few studies of bicycle and pedestrian use (Turner, Shunk, and Hottenstein, 1998; Schwartz et al, 1999; Porter, Suhrbier and Schwartz, 1999). There are not simple tables that one can look up default values.

To solve this problem, examples were sought from different size metropolitan areas. Estimates of travel mode share for all trip purposes (work and non-work) were obtained from1) Ottawa (Ottawa, 1997); 2) Portland (Portland, 1999); and Houston[4]. Table 15.1 shows the estimated shares. The Houston data does not include walking and biking shares, and transit trips are not distinguished by mode in the Portland and Ottawa data.

Table 15.1
Estimated Mode Share for Three Metropolitan Areas
All Trip Purposes

	Ottawa	**Portland**	**Houston**
Population:	725 thousand (1995)	2.0 million (2001)	4.6 million (2000)
Percent of trips by:	(1995)	(1994)	(2025 forecast)
Driving	73.5%	88.6%	98.3%
Transit	15.2%	3.0	1.7% (bus 1.1%; rail 0.6%)
Walking	9.6%	4.6%	-
Bicycle	1.7%	1.0%	-
Other	-	2.8%	-

While it's difficult to generalize, walking is very much dependent on the existence of an extensive transit system. In Houston, the transit system is primarily a commuter

system whereas in Portland and Ottawa, it serves multiple purposes. Clearly, the more compact the urban area, the more likely that trips will occur by transit, walking or biking. But, even in the case of Ottawa where almost 10% of trips are by walking, the majority of trips are by private vehicle. In the United States and Canada, for metropolitan areas with extensive transit facilities (New York, Chicago, Boston, Montreal), a majority of regional trips are still by automobile.

Based on this, some default values were selected and put into the spreadsheet. The spreadsheet requires that they are entered as proportions (not percentages). The defaults values were (table 15.2):

Table 15.2
Default Mode Share Values
Proportions

Mode	**Share**
Walk	.04
Bicycle	.01
Driving	.90
Bus	.04
Train	.01

The user can modify these in the spreadsheet. It's important that a user contact the local Metropolitan Planning Organization to find out what would be reasonable values for the urban area. The default values are but guesses based on a limited amount of data.

An alternative approach is to use the Journey to Work data of the U.S. Census Bureau (2004). During every census, the Census Bureau documents home-to-work 'commute' trips and breaks down these data by mode share. They release these data under the title "Journey to Work". In the United States in 2000, 87.9% of all home-to-work trips were by private vehicle (automobile, van, truck), 4.7% were by public transit (bus 2.5%; rail 2.1%; other 0.1%), 2.9% were by walking, 0.4% were by bicycle, 0.1% were by motorcycle, 0.7% were by other means, and 3.3% worked at home.

National journey to work statistics for 1990 and 2000 and for metropolitan areas in 1990 can be found at http://www.census.gov/population/www/socdemo/journey.html. Data on metropolitan areas for 2000 can be found in McGuckin and Srinivasan (2003). In 2000, the home-to-work mode share for a sample of large metropolitan (including the 15 largest) areas is shown in Table 15.3. They are rank-ordered by the 2000 population of the metropolitan area.

As can be seen, the larger metropolitan areas generally have a higher share of transit use and walking than smaller metropolitan areas, but the differences are not that dramatic. Even the largest metropolitan areas have a majority of their home-to-work trips by private vehicle.

Table 15.3
Mode Share of Journey to Work Trips: 2000
(From McGuckin and Srinivasan, 2003)

Greater Metropolitan Area	2000 Pop (M)	Mode Share					
		Walk	Bicycle	Drive	Bus	Rail	Other*
New York	21.1	5.6%	0.3%	65.7%	6.8%	17.1%	4.5%
Los Angeles	16.4	2.6%	0.6%	87.6%	4.3%	0.3%	4.6%
Chicago	9.2	3.1%	0.3%	81.5%	4.6%	6.6%	3.9%
Washington DC	7.6	3.0%	0.3%	83.2%	4.1%	5.0%	4.4%
San Francisco	7.0	3.3%	1.1%	81.0%	5.7%	3.5%	5.4%
Philadelphia	6.2	3.9%	0.3%	83.6%	5.3%	3.3%	3.6%
Detroit	5.5	1.8%	0.2%	93.4%	1.7%	0.0%	2.9%
Boston	5.8	4.1%	0.4%	82.7%	3.2%	5.5%	4.1%
Dallas	5.2	1.5%	0.1%	92.7%	1.6%	0.1%	4.0%
Houston	4.7	1.6%	0.3%	91.3%	3.1%	0.0%	3.7%
Atlanta	4.1	1.3%	0.1%	90.6%	2.4%	1.1%	4.5%
Miami	3.9	1.8%	0.5%	90.1%	3.2%	0.5%	3.9%
Seattle	3.6	3.2%	0.6%	84.4%	6.2%	0.0%	5.6%
Phoenix	3.3	2.1%	0.9%	90.0%	1.9%	0.0%	5.1%
Minneapolis/St Paul	3.0	2.4%	0.4%	88.4%	4.4%	0.0%	4.4%
Cleveland	2.9	2.1%	0.2%	91.1%	3.1%	0.3%	3.2%
San Diego	2.8	3.4%	0.6%	86.9%	3.1%	0.2%	5.8%
St Louis	2.6	1.6%	0.1%	92.5%	2.1%	0.2%	3.5%
Denver	2.6	2.4%	0.7%	87.1%	4.2%	0.1%	5.5%
Pittsburgh	2.4	3.6%	0.1%	87.1%	6.0%	0.1%	3.1%
Portland	2.3	3.0%	0.8%	85.2%	5.1%	0.5%	5.4%
Cincinnati	2.0	2.3%	0.1%	91.4%	2.8%	0.0%	3.4%
Sacramento	1.8	2.2%	1.4%	88.9%	2.4%	0.3%	4.8%
Kansas City	1.8	1.4%	0.1%	93.2%	1.2%	0.0%	4.1%
Milwaukee	1.7	2.8%	0.2%	90.0%	3.9%	0.0%	3.1%
Indianapolis	1.6	1.7%	0.2%	93.3%	1.2%	0.0%	3.6%
Orlando	1.6	1.3%	0.4%	92.7%	1.6%	0.0%	4.0%
San Antonio	1.6	2.4%	0.1%	90.9%	2.8%	0.0%	3.8%
Norfolk	1.6	2.7%	0.3%	91.0%	1.7%	0.0%	4.3%
Las Vegas	1.6	2.4%	0.5%	89.5%	3.9%	0.0%	3.7%
Charlotte	1.5	1.2%	0.1%	93.8%	1.3%	0.0%	3.6%
New Orleans	1.3	2.7%	0.6%	87.7%	5.2%	0.0%	3.8%
Salt Lake City	1.3	1.8%	0.4%	90.3%	2.7%	0.3%	4.5%
Memphis	1.1	1.3%	0.1%	93.9%	1.6%	0.0%	3.1%
Rochester	1.1	3.5%	0.2%	90.9%	1.9%	0.0%	3.5%
Oklahoma City	1.1	1.7%	0.2%	93.8%	0.5%	0.0%	3.8%
Louisville	1.0	1.7%	0.2%	92.9%	2.2%	0.0%	3.0%

* Includes taxi, ferry, and working at home

The problem with these data, however, is that they only examine work trips. Nationally, home-to-work trips represent only about 15% of all daily trips (BTS, 2002). On the other hand, 45% of daily trips are for shopping and errands and 27% are social and

recreational. Further, non-work trips are even more likely to occur by automobile, and are generally shorter. For example, in Houston, for home-based non-work trips, only 1% of trips are by transit compared to 3.1% for home-to-work trips. These home-based non-work trips may be a better analogy to crime trips than work trips since they tend to be of similar trips lengths as crime trips.

Thus, unless the user is willing to assume that a crime trip is like a work trip (which is questionable), then the Journey to Work tables are probably not the best guide for the target proportions. Nevertheless, an examination of them is valuable to see how work trips are split among the various travel modes.

Select mode functions

Third, select mathematical functions that approximate accessibility utility. Again, some plausible assumptions need to be made. In *CrimeStat*, the user can select among five different mathematical functions (linear, negative exponential, normal, lognormal, truncated negative exponential). The default functions are (Table 15.4):

Table 15.4
Default Mode Share Functions

Mode	**Function**
Walk	Negative exponential
Bicycle	Negative exponential
Driving	Lognormal
Bus	Lognormal
Train	Lognormal

The reasoning behind this is that walking and biking are relatively short trips, whereas transit modes are used for intermediate length trips while driving can be used for any length trip. Thus, it's unlikely that an automobile will be used for very short trips (less than a quarter mile) and it's very unlikely that transit will be used for short trips (less than a half mile or more). Nevertheless, the user can modify these choices and examine the appropriate column in the spreadsheet.

Select model priorities

Fourth, select the priorities for modeling the target. Unfortunately, there may not be a single solution that will yield the target proportions. Therefore, a decision needs to be made on which **order** the spreadsheet will be calculated. The default order is (table 15.5):

Table 15.5
Default Mode Share Functions

Mode	Order of Iteration
Walk	1
Bicycle	2
Driving	3
Bus	4
Train	5

The reasoning is that the offender first makes a decision on the length of the trip (short, medium, long, or the equivalent in travel time). Then, within each category, makes a decision on which mode to choose. For very short trips, the default mode is walking. For intermediate to long trips, the default choice is driving. However, the user can change this order.

Iteratively estimate parameters

Fifth, in the spreadsheet, iteratively adjust the parameters until the target proportion is reached. Do this in the order selected in the above step. Again, there is not a single solution that will produce the target proportion. For example, each of the mathematical functions has two or three parameters that can be adjusted:

1. For the negative exponential, the coefficient and exponent
2. For the normal distribution, the mean distance, standard deviation and coefficient
3. For lognormal distribution, the mean distance, standard deviation and coefficient
4. For the linear distribution, an intercept and slope
5. For the truncated negative exponential, a peak distance, peak likelihood, intercept, and exponent.

The target proportion can be achieved by adjusting any or all of the parameters. For example, to achieve a target proportion of 0.05 (i.e., 5%) using the negative exponential, an infinite number of models can yield this, for example coefficient=0.0366, exponent=-2.63; coefficient=0.0459 or exponent=-5; coefficient=0.01966, exponent=-1; and so forth. Therefore, there must be additional criteria to constrain the choices.

One criteria is to set an approximate mean distance. For example, with walking trips, the mean distance can be set to a half mile or for driving, the mean distance can be set to 6 miles. Then, check the approximate mean distance of the selected function. Though rarely will the exact mean distance be replicated, the calculated mean distance should be close to the ideal. The one exception is for very short trips. Since the intervals in the spreadsheet are a half mile each, there is considerable error for very short distances.

Examine the graphs in the spreadsheet

Another is to examine the graph of the function in the spreadsheet (below the calculations). Does the typical trip approximate the expected mean distance? Does the selected function produce something that looks intuitive? Admittedly, these are subjective decisions. But, if the function looks strange, it can be caught and re-calculated.

In short, the aim should be to produce a function that not only captures the target proportion, but looks plausible. Several examples are shown below. Figure 15.1 shows the default walking model using a negative exponential. Figure 15.2 shows the default biking model, also using a negative exponential. Figure 15.3 shows the default driving mode using a lognormal function. Figure 15.4 shows the default bus mode, also using a lognormal function and figure 15.5 shows the default train mode using a lognormal function.

Figure 15.6 shows the cumulative results of the default values. This is also graphed in the spreadsheet, starting in cell I1. Notice how the relative accessibility function works. As distance increases, the mode proportions change. At very short distances, walking trips predominate with biking trips also getting a moderate share. As the distance increases, the proportions increasingly shift toward driving. Even though the likelihood of driving declines with distance, the other modes decline even faster. In other words, the relative accessibility function is estimating the relative shares of each mode as a function of the impedance (in this case, distance).

Adapting spreadsheet for travel time or travel cost

The illustrations to this point have used distance as an impedance unit. However, other impedance units, such as travel time and generalized travel cost, can also be used. These generally require a network (see below) in that weights have to be assigned to segments. Nevertheless, the same logic applies. For each travel mode, a specific impedance function is estimated and then applied to the trip distribution matrix.

Empirically estimating the mode-specific impedance

As mentioned at the beginning of this chapter, the lack of information about offender travel modes has necessitated the use of mathematical 'guesses' about travel behavior. However, if it were possible to obtain actual information on travel modes by offenders, then this information could be utilized directly to estimate a much more accurate impedance function. *If* this database existed, then two approaches are possible:

1. Fit the data with the various mathematical functions to see which ones fit best and to estimate the parameters.

2. Use the kernel density function to estimate a non-linear impedance value with the specific information.

Figure 15.1:

Negative Exponential Function: Walk Mode

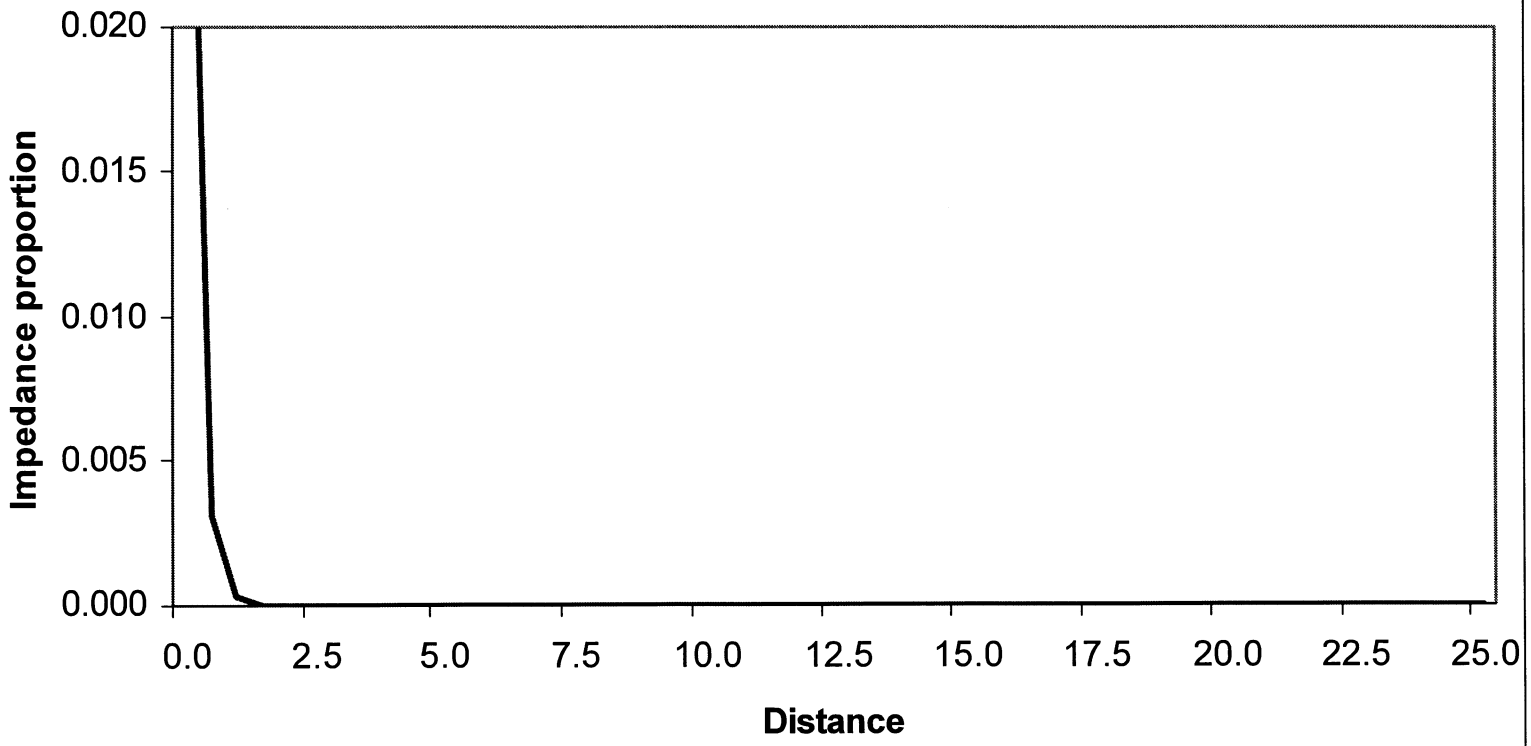


Figure 15.2:

Negative Exponential Function: Bike Mode

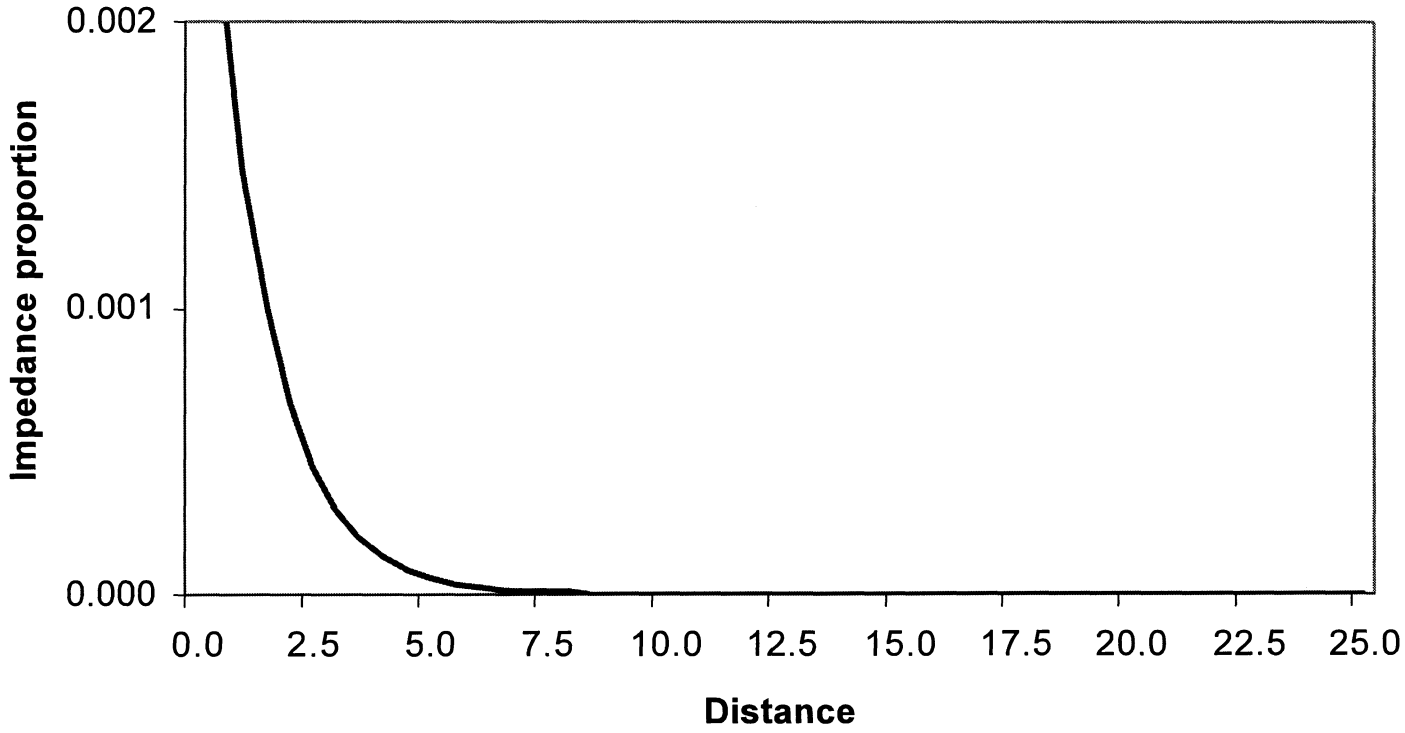


Figure 15.3:

Lognormal Function: Drive Mode

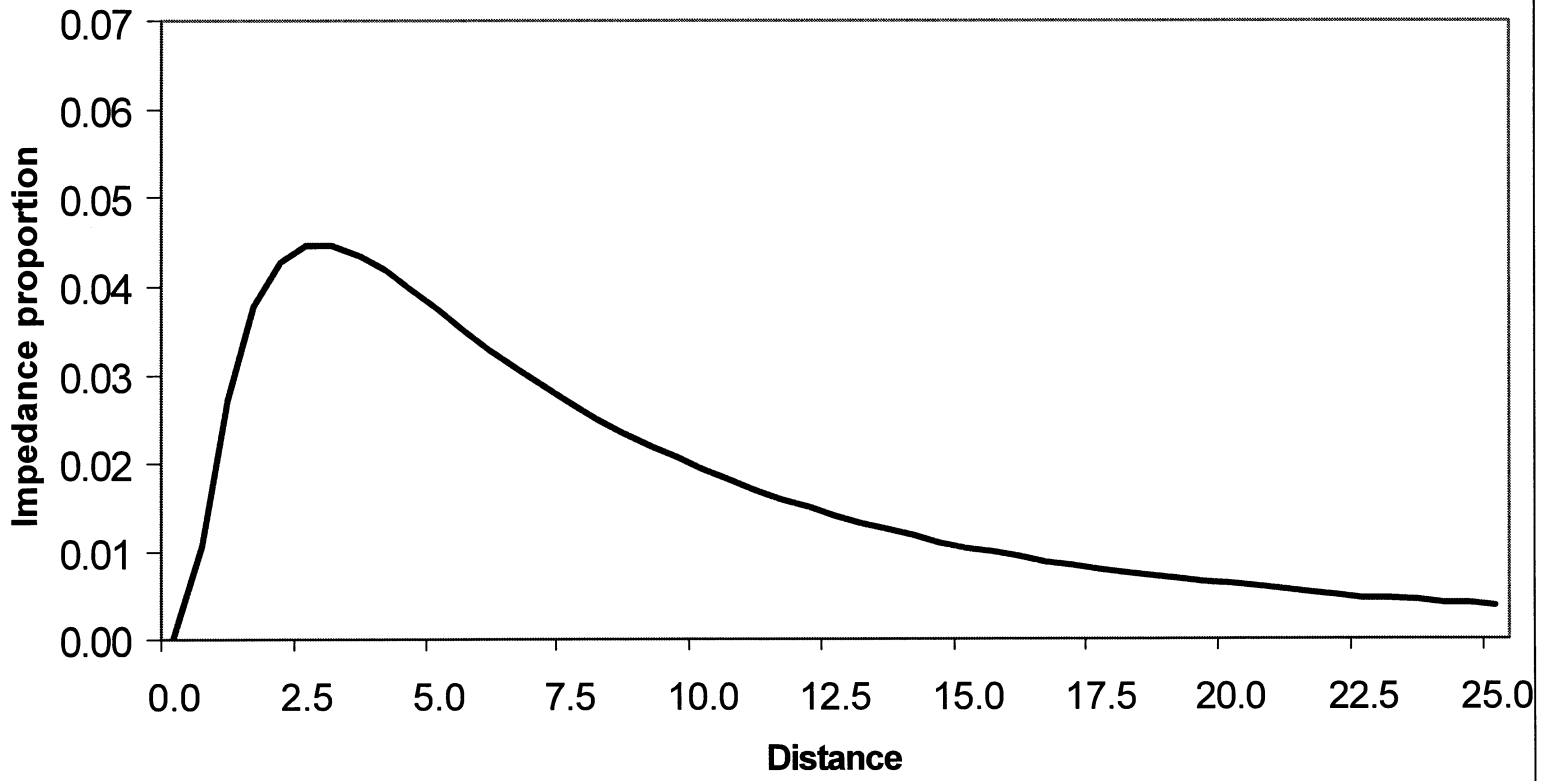


Figure 15.4:

Lognormal Function: Bus Mode

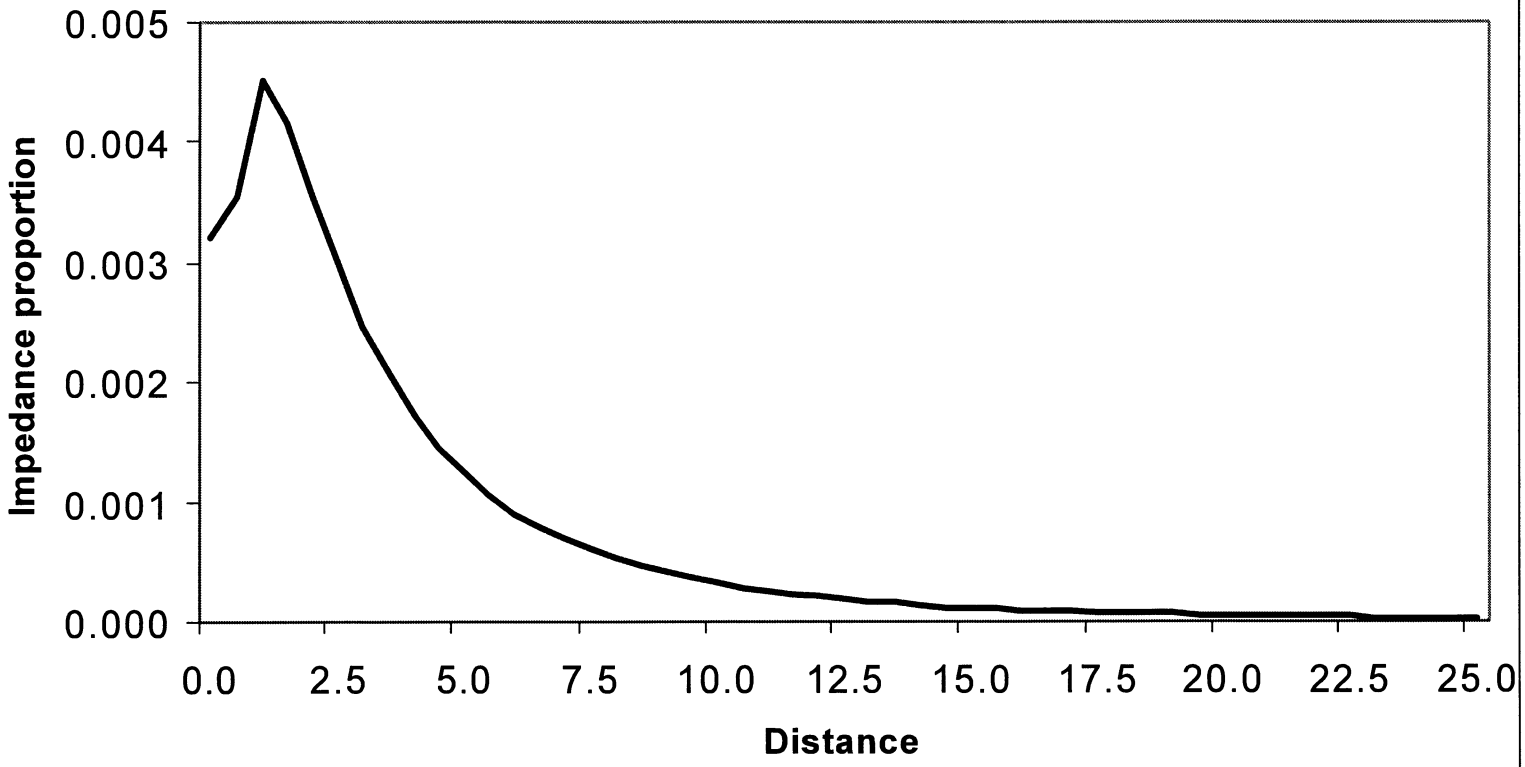


Figure 15.5:

Lognormal Function: Train Mode

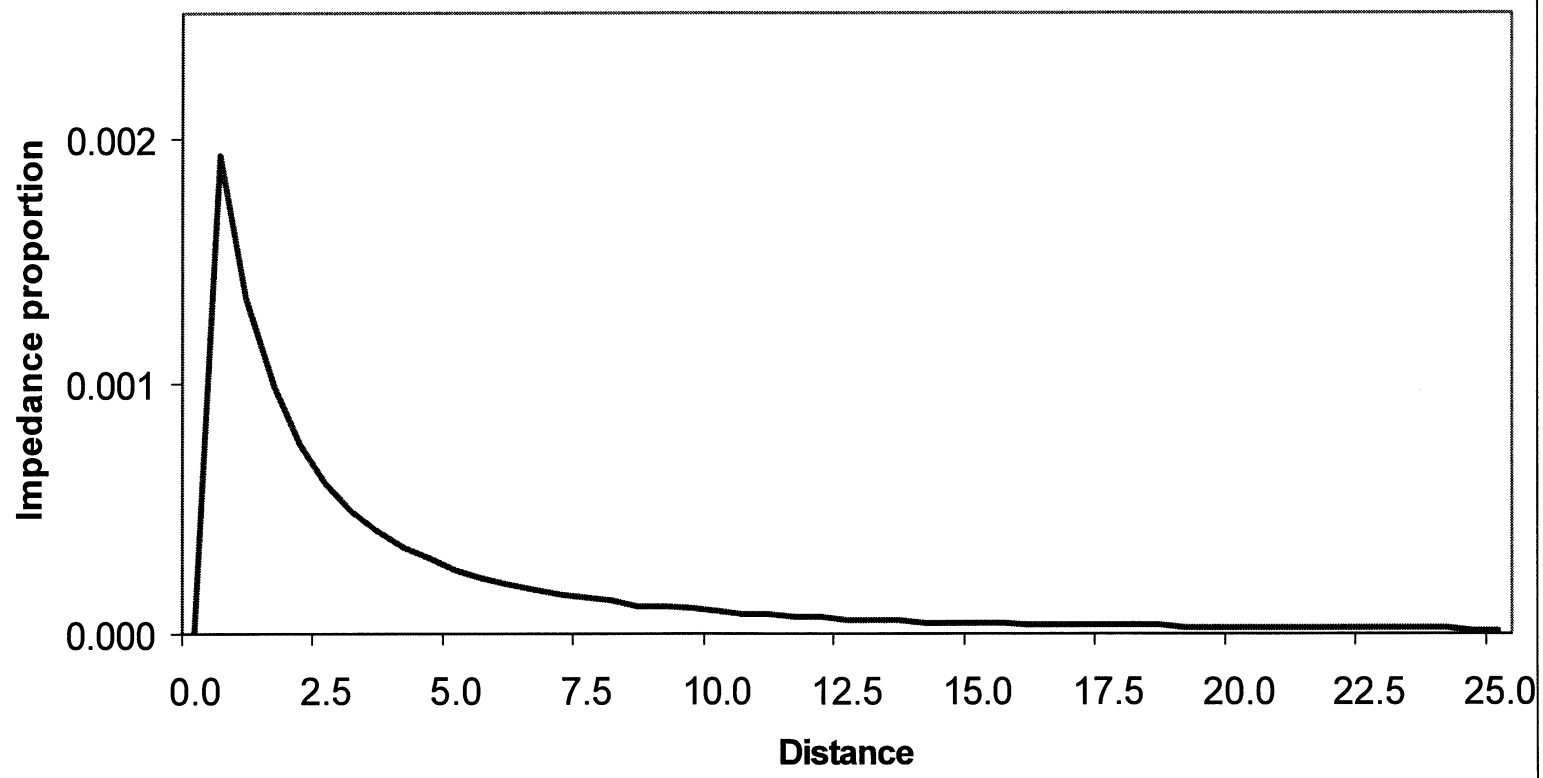
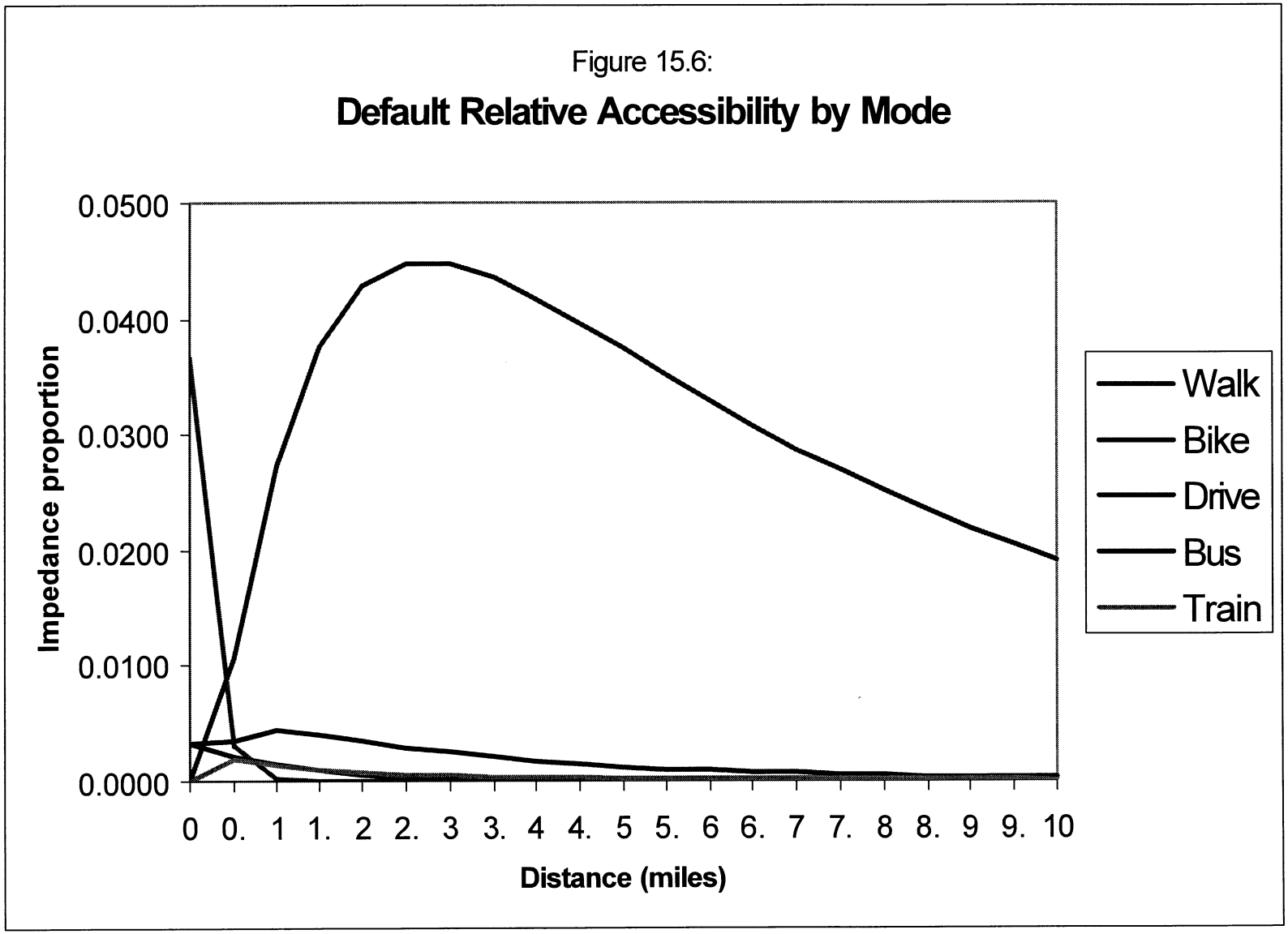


Figure 15.6:


Default Relative Accessibility by Mode
Impedance proportion
0.0500
0.0400
0.0300
0.0200
0.0100
0.0000
0 0. 1 1. 2 2. 3 3. 4 4. 5 5. 6 6. 7 7. 8 8. 9 9. 10
Distance (miles)
Walk
Bike
Drive
Bus
Train

These approaches were discussed in chapter 9 (Journey to Crime) and in chapter 14 (Trip Distribution). The "Calibrate impedance function" routine in the Trip Distribution module can be used for this purpose. The advantage would be enormous. Instead of guesses about likely impedance functions of specific travel modes, the user would have a function that was based on real data. There should be a substantial improvement in modeling accuracy that would result. However, these data have to be first collected.

CrimeStat III Mode Split Tools

The *CrimeStat* mode split module allows the relative accessibility function to be calculated. Figure 15.7 shows the setup page for the mode split routine and figure 15.8 shows the setup for modes 1 and 2, in the example "Walk" and "Bicycle". The setup for modes 3, 4, and 5 are similar.

Mode Split Setup

On the mode split setup page, the predicted origin and predicted destination files must be input as the primary and secondary files. If the origin and destination files are identical (i.e., all the origin zones are included in the destination zones), then the file must be input as the primary file.

In addition, the user must input a predicted origin-destination trip file from the trip distribution module. Finally, an assumed impedance value for trips from the "External zone" must be specified. The default is 25 miles. Choose a value that would represent a 'typical' trip from outside the study region.

For each mode, the user must provide a label for the name and define the mathematical function which is to be applied and specify the parameters. The first time the routine is opened, the default values are listed. However, the user can change these.

> Hint: Once the parameters are entered, they can be saved on the Options page. Then, they can be re-entered by loading the saved parameters file.

Constrain Choice to Network

The impedance will be calculated either directly or is constrained to a network. The default impedance is defined with the type of distance measurement specified on the Measurement Parameters page (under Data setup). On the other hand, if the impedance is to be constrained to a network, then the network has to be defined.

Default

The default impedance is that specified on the Measurement parameters page. If direct distance is the default distance (on the measurement parameters page), then all

Figure 15.7:

Mode Split Module

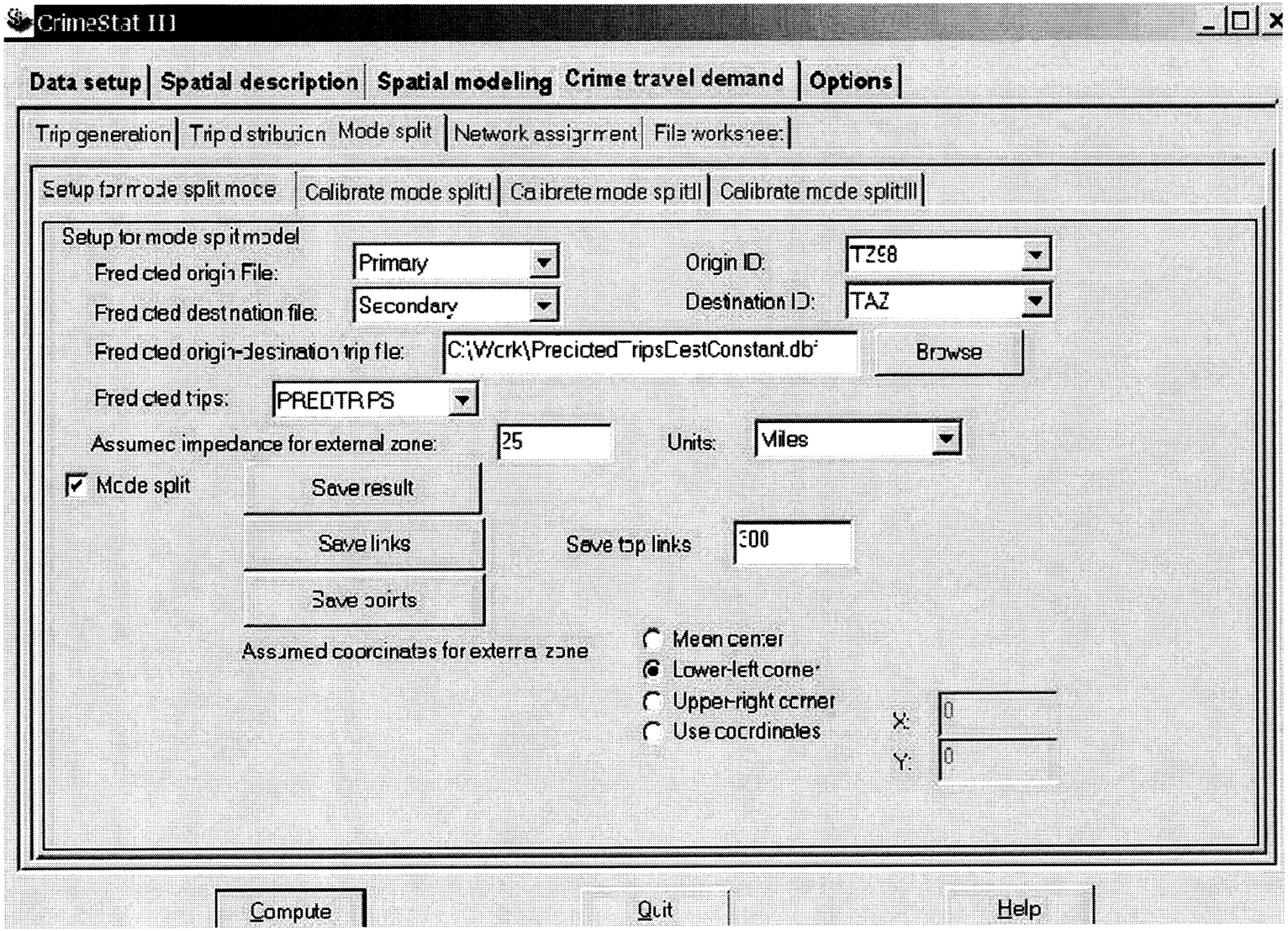
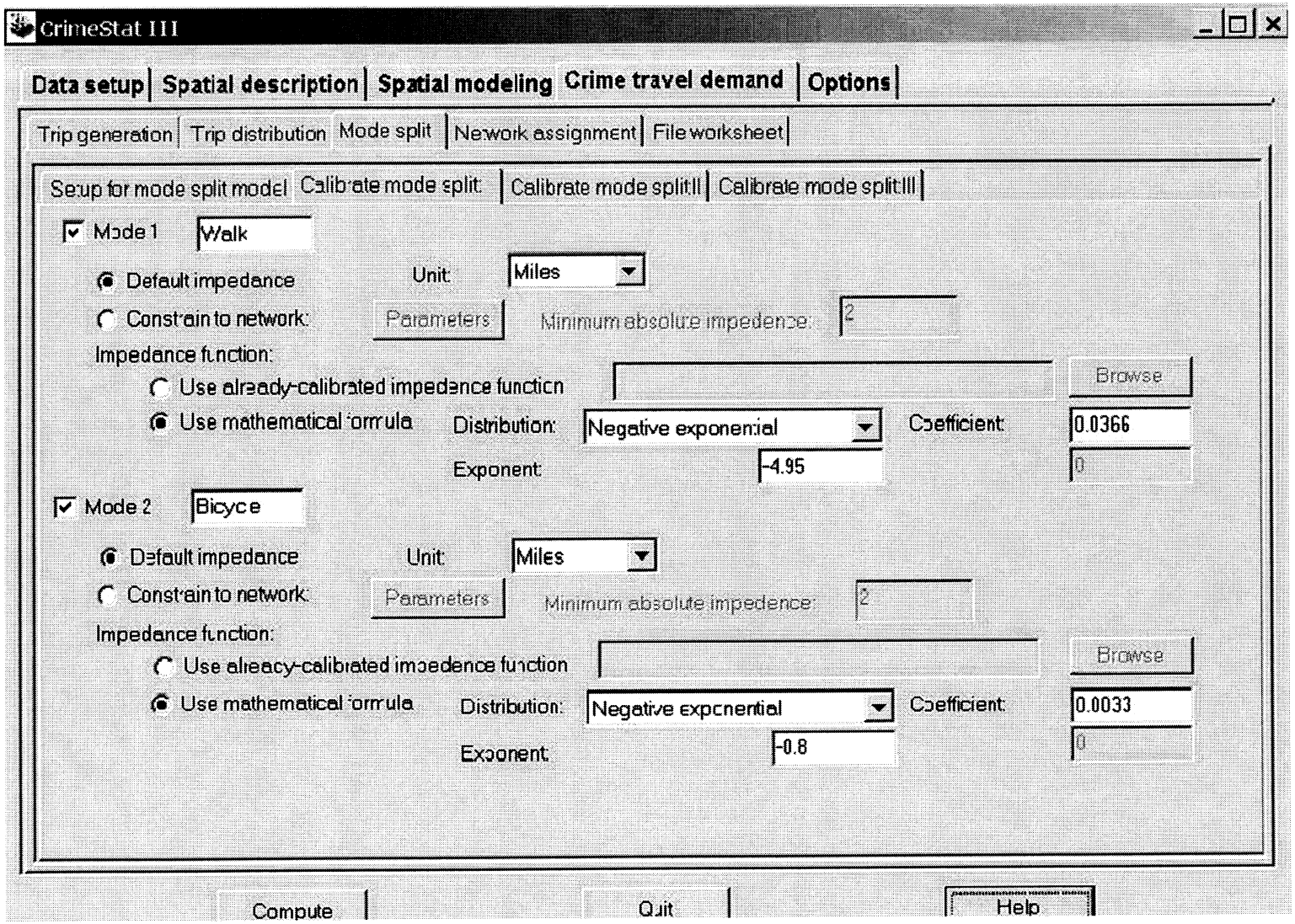


Figure 15.8:

Set Up for Individual Modes



impedances are calculated as a direct distance. If indirect distance is the default, then all impedances are calculated as indirect (Manhattan) distance. If network distance is the default, then all impedances are calculated using the specified network and its parameters; travel impedance will automatically be constrained to the network under this condition.

Constrain to network

An impedance calculation should be constrained to a network when there are limited choices. For example, a bus trip requires a bus route; if a particular zone is not near an existing bus route, then a direct distance calculation will be misleading since it will probably underestimate true distance. Similarly, for a train trip, there needs to be an existing train route. Otherwise, the routine will assign transit trips where those are not possible (i.e., it will assign train trips where there are no train stations and it will assign bus trips where there are no bus routes). The routine does not 'know' whether there are transit routes and must be told where they are. Even for walking, bicycling and driving trips, an existing network might produce a more realistic travel impedance than simply assuming a direct travel path.

If the impedance calculation is to be constrained to a network, then the network must be defined. A more extensive discussion of a network is provided in chapter 3 (under Type of distance measurement on the Measurement Parameters page) and in chapter 16 in the discussion of the Trip Assignment module. Essentially, a network is a series of connected segments that specify possible routes. Each segment has two end nodes (in *CrimeStat*, they are called 'FromNode' and 'ToNode). Depending on the type of network, the segments can be bi-directional (i.e., travel is allowed in either direction) or single directional (i.e., travel is allowed only from the "FromNode" to the "ToNode").

A critical component of a network for the mode split routine is that travel can only pass through nodes. This means that two segments that are connected can allow a trip to pass over those two segments whereas two segments that are not connected cannot allow a trip to pass directly from one to the other. From outside the network, a trip connects to it at a node. For a transit network, this can be critical. For a bus route, it may or may not be important. A precise bus network defines nodes by bus stops so that a trip can 'enter' or 'leave' the bus system at a real stop. A less precise bus network defines nodes by the ends of segments (e.g., the end nodes of a TIGER segment). The routine will not know whether the node it enters or leaves from is a real bus stop or not. In the case of bus routes, it probably doesn't matter since they generally make very regular stops (every two or three blocks).

Accurately defined transit networks

For train networks, however, it is absolutely critical that the network be defined accurately. The nodes must be legitimate stations; a trip can only enter or leave the train system through a station (i.e., it cannot enter or leave a train network at the end of an arbitrary segment node). Most travel demand models use very precise bus and train networks that have been carefully checked; where errors occur, the networks are edited

and updated. If the user does not have an edited transit network, one can be made in the trip assignment module. There is a "Create a transit network from primary file" routine that will draw segments between input primary file points; the user inputs the station or bus stop locations as the primary file and the routine creates a network from one point to the next in the *same* order as in the primary file (i.e., the primary file needs to be properly sorted in order to travel). See chapter 16 for more information about creating a transit network.

Entering the network parameters

The network is input by selecting "Constrain to network" and click on the 'Parameters' button. A dialogue is brought up that allows the user to specify the network to be used. The network file can be either a shape line or polyline file (the default) or another file, either dBase IV 'dbf', Microsoft Access 'mdb', Ascii 'dat', or an ODBC-compliant file. If the file is a shape file, the routine will know the locations of the nodes. All the user needs to do is identify a weighting variable, if used, and possible one way routes ('flags'). For a dBase IV or other file, the X and Y coordinate variables of the end nodes must be defined. These are called the "From" node and the "End" node, though there is no particular order.

An optional weight variable is allowed for both a shape or dbf file. The routine identifies nodes and segments and finds the shortest path. By default, the shortest path is in terms of distance though each segment can be weighted by travel time, travel speed, or generalized cost; in the latter case, the units are minutes, hours, or unspecified cost units.

Finally, the number of graph segments to be calculated is defined as the network limit. The default is 50,000 segments. This can be changed, but be sure that this number is greater than the number of segments in your network.

Minimum absolute impedance

If a mode is constrained to a network, an additional constraint is needed to ensure realistic allocations of trips. This is the minimum absolute impedance between zones. The default is 2 miles. For any zone pair that is closer together than the minimum specified (in distance, time interval, or cost), no trips will be allocated to that mode. This constraint is to prevent unrealistic trips being assigned to intra-zonal trips or trips between nearby zones.

CrimeStat uses three impedance components for a constrained network:

1. The impedance from the origin zone to the nearest node on the network (e.g., nearest rail station);

2. The impedance along the network to the node nearest to the destination; and

3. The impedance from that node to the destination zone.

Since most impedance functions for a mode constrained to a network will have the highest likelihood some distance from the origin, it's possible that the mode would be assigned to, essentially, very short trips (e.g., the distance from an origin zone to a rail network and then back again might be modeled as a high likelihood of a train trip even though such a trip is very unlikely).

For each mode that is constrained to a network, specify the minimum absolute impedance. The units will be the same as that specified by the measurement units. The default is 2 miles. If the units are distance, then trips will only be allocated to those zone pairs that are equal to or greater in distance than the minimum specified. If the units are travel time or speed, then trips will only be allocated to those zone pairs that are farther apart than the distance that would be traveled in that time at 30 miles per hour. If the units are cost, then the routine calculates the average cost per mile along the network and only allocates trips to those zone pairs that are farther apart than the distance that would be traveled at that average cost.

Applying the Relative Accessibility Function

To apply the relative accessibility function, the parameter choices for each mode are entered into the mode split routine. All transit modes are then constrained
Once the mode split setup has been defined and all transit modes have been constrained to a proper network, the mode split routine can be run.

Figure 15.9 shows the top 300 walking crime trips in Baltimore County estimated with the default accessibility functions. As seen, the vast majority of walking trips are intra-zonal (local). There are only a couple of inter-zonal walking trip links shown. The default impedance function assigned approximately 4% of the trips to this mode and the result is many intra-zonal trips.

Figure 15.10 shows the top 300 bicycle crime trips in Baltimore County. There are fewer trips by bicycle and they also tend to be quite local. The impedance function used for bicycle trips allocated approximately 1% of all trips to this mode. Thus, it's less frequent than walking mode. There are proportionately more inter-zonal trips among the top 300 than for walking trips, but these tend to be quite short (travel between adjacent zones).

On the other hand, driving is the predominant travel mode for the crime trips (Figure 15.11). The impedance function used allocated approximately 90% of the trips to driving. The pattern almost perfectly replicates the predicted trip distribution (see figures 14.12 and 14.20 in chapter 14). Further, the trips are a lot longer. Among the top 300 links, there were no intra-zonal driving trips. The use of a lognormal function minimized intra-zonal travel.

To allocate bus and train trips, however, it was necessary to constrain them to a network. Separate bus and train networks were obtained from the Baltimore Metropolitan Council. Figure 15.12 shows the Baltimore bus network and figure 15.13 shows the predicted bus trips superimposed over the bus network. Overall, about 4% of the total

Figure 15.9:
Mode Split: Walking Crime Trips

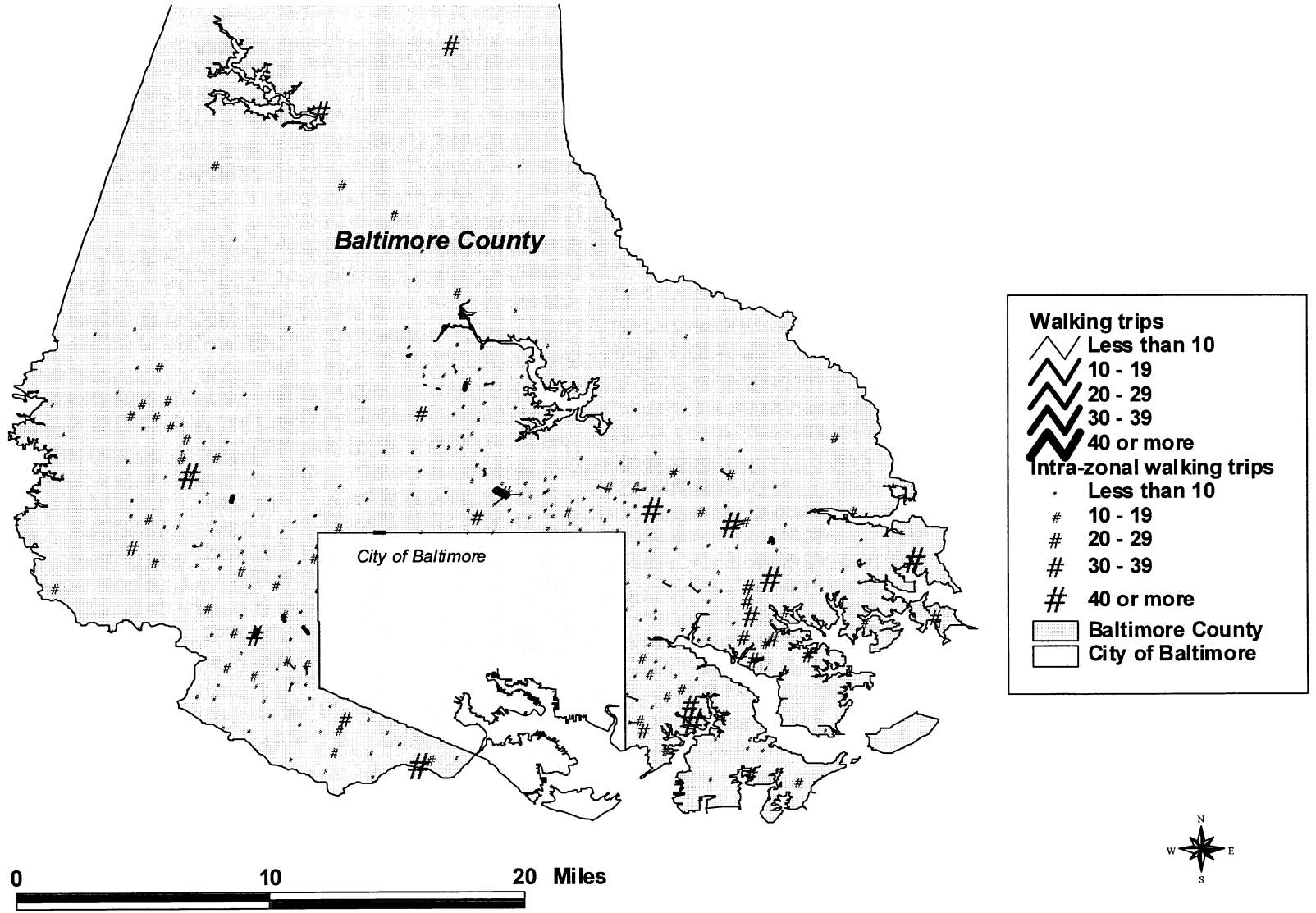


Figure 15.10:
Mode Split: Bicycle Crime Trips

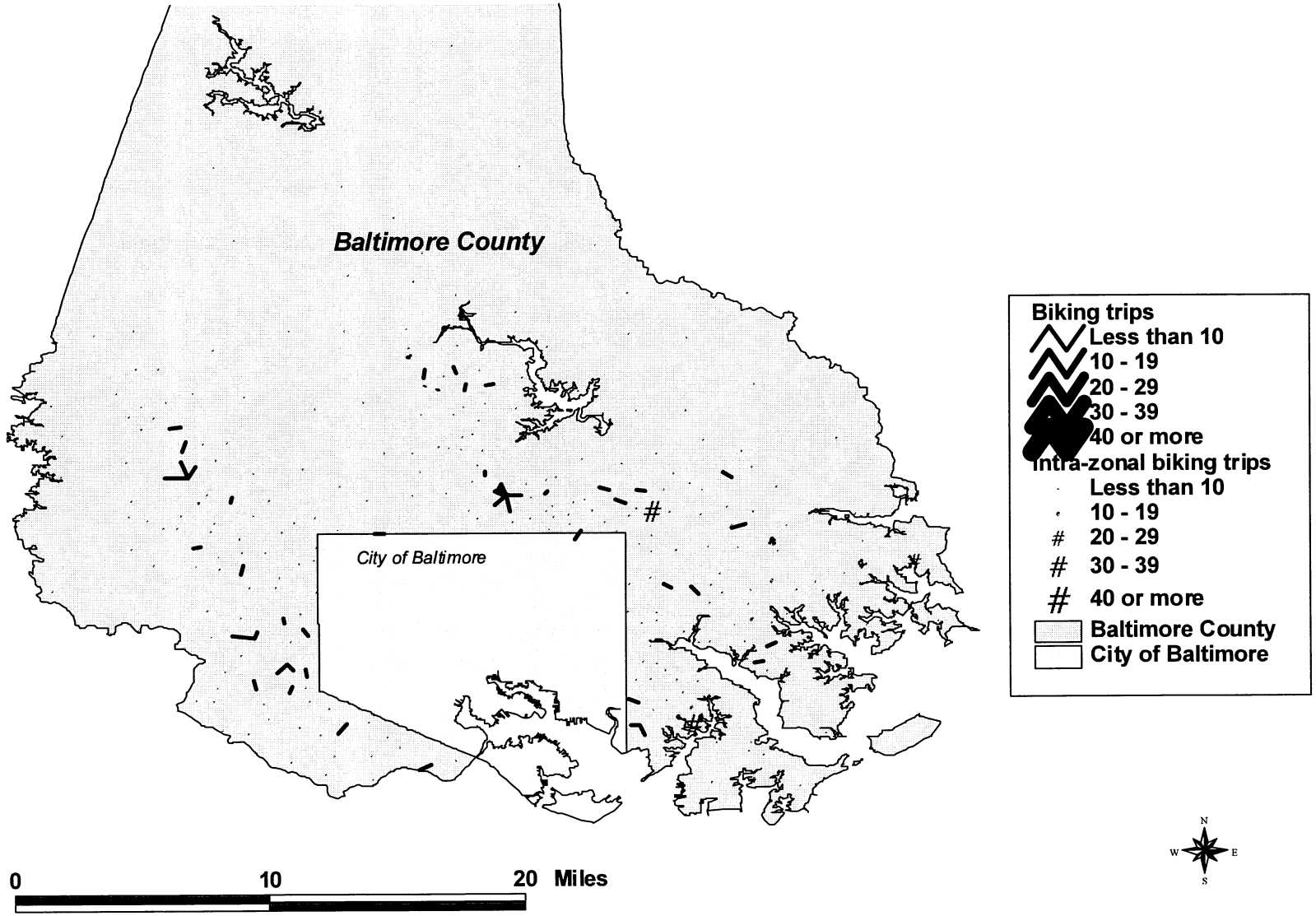


Figure 15.11:
Mode Split: Driving Crime Trips

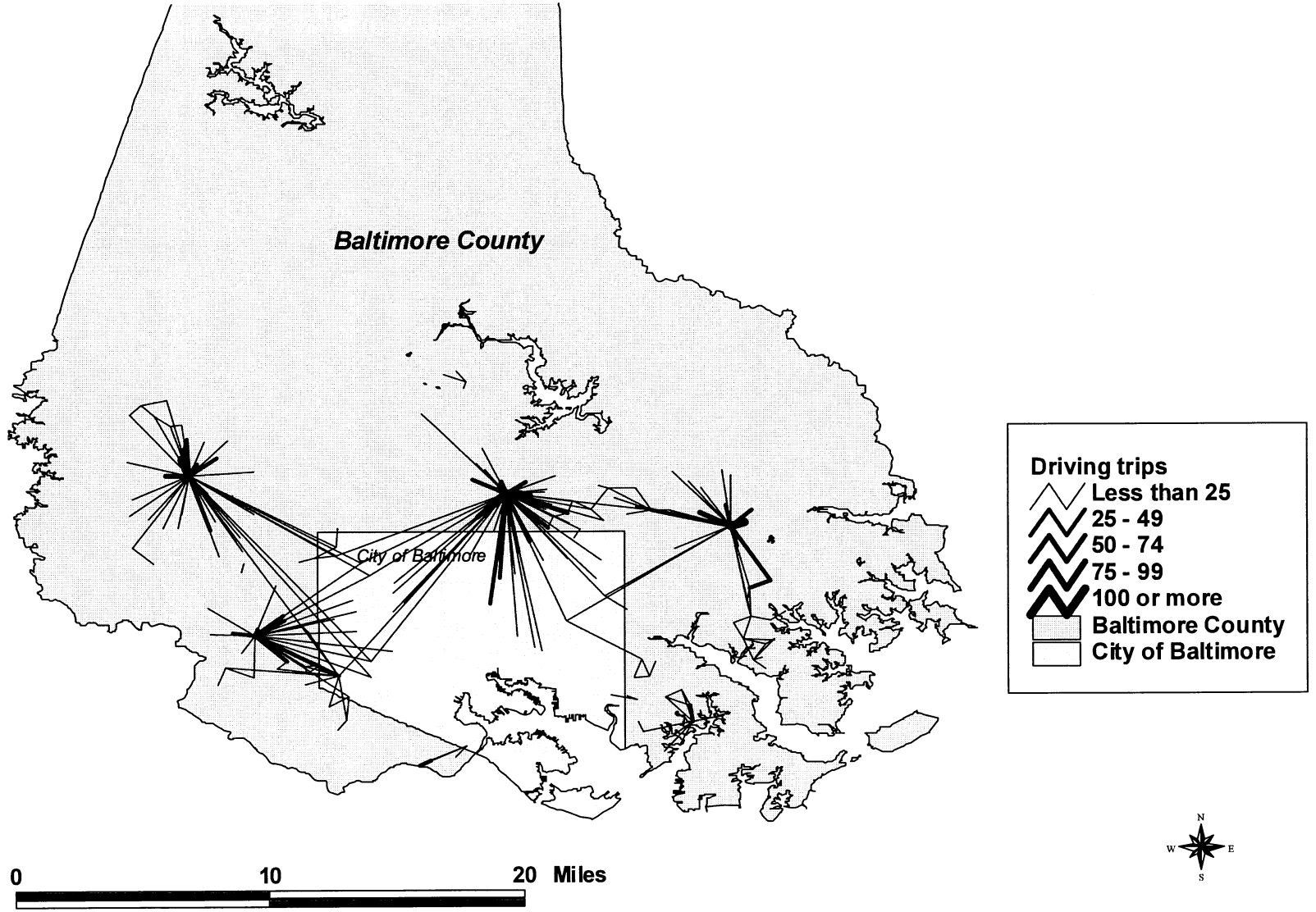


Figure 15.12:
Baltimore Bus Network

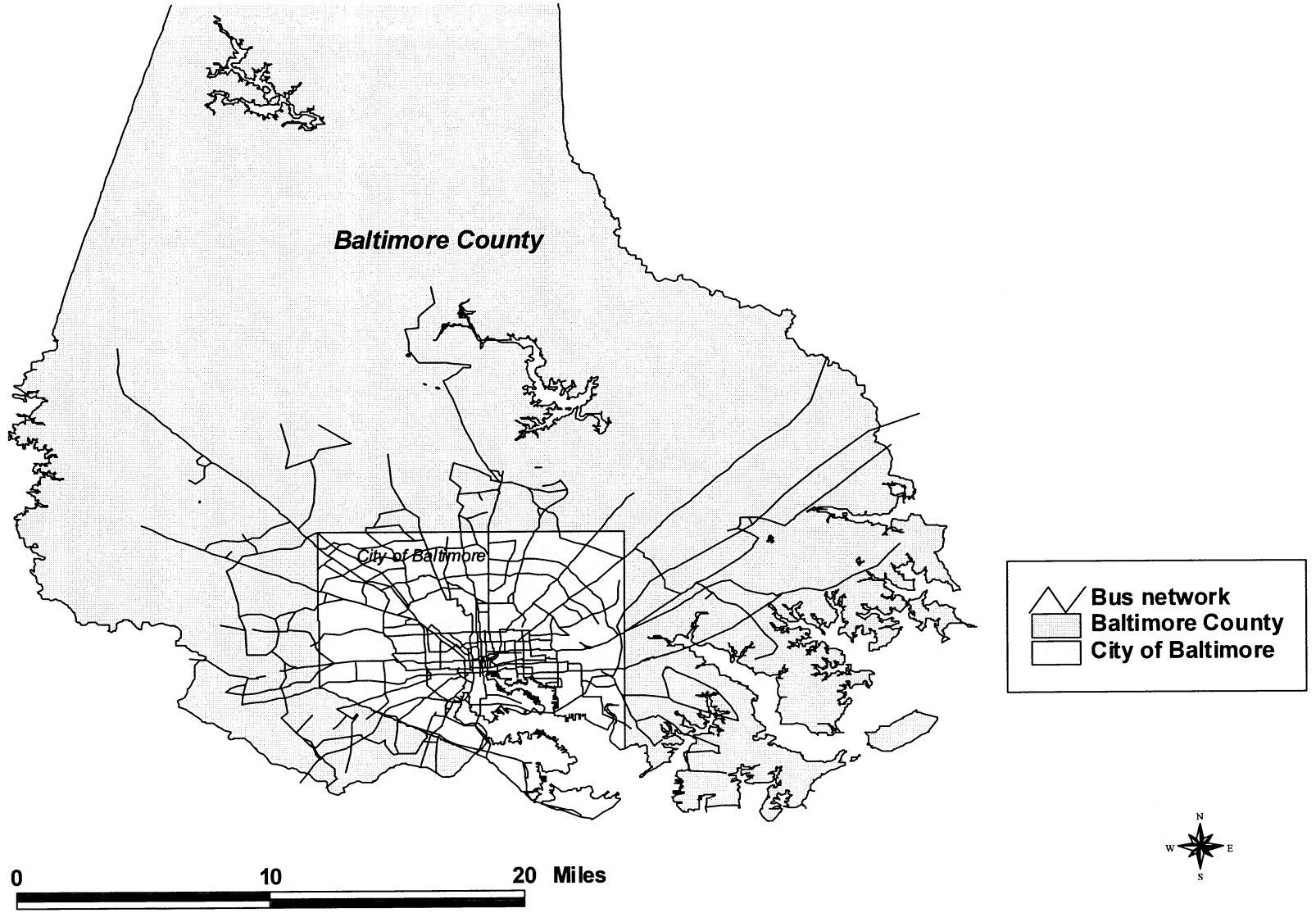
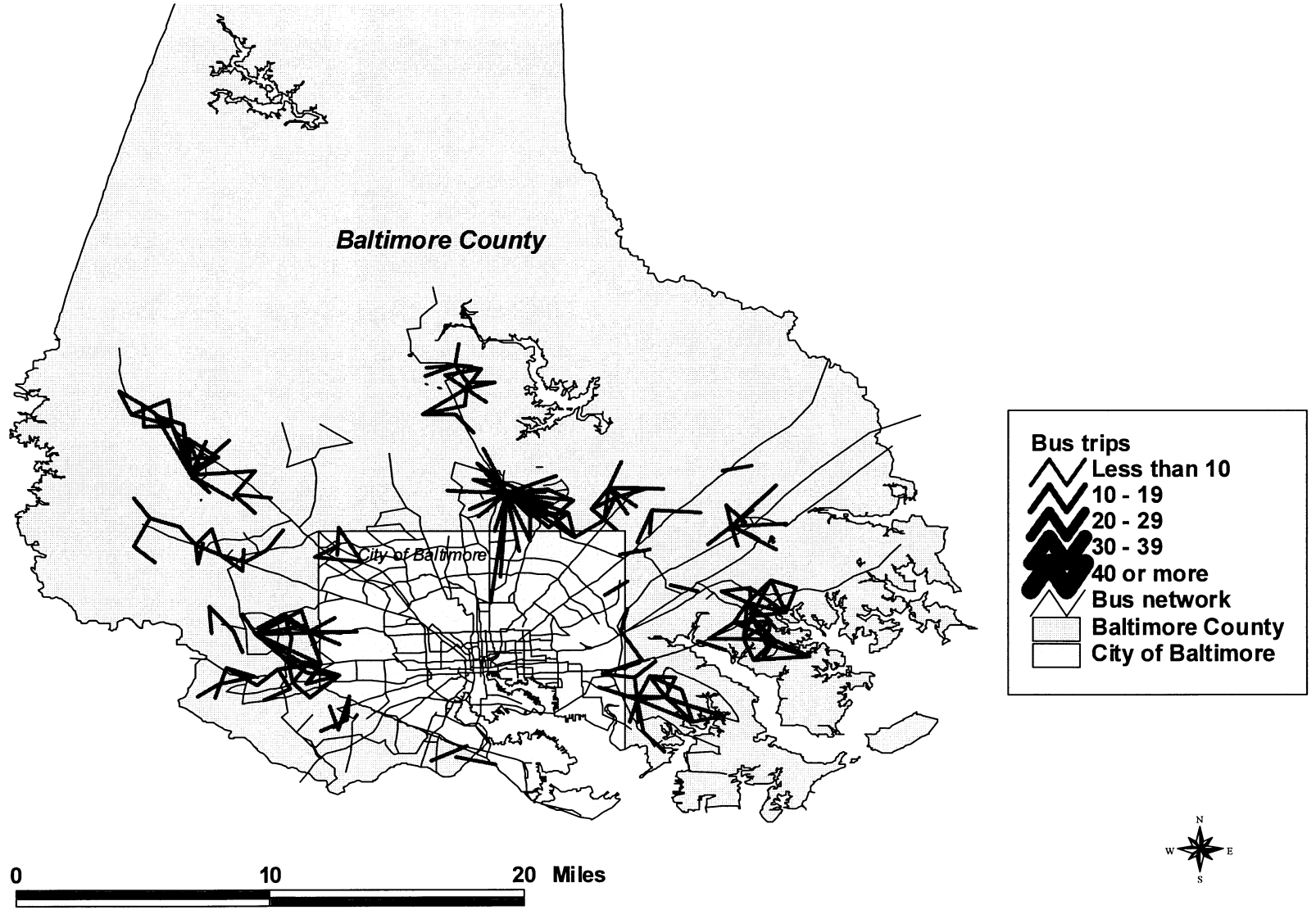


Figure 15.13:
Mode Split: Bus Crime Trips



trips were allocated to the bus mode by the accessibility function. As seen, the trips tend to be moderate distances and tend to be close to the bus network. Constraining these trips by the network decreases the likelihood that the routine would assign a particular trip link that was far from the bus work to a bus trip.

Finally, train crime trips were constrained to the train network. Figure 15.14 superimposes the assigned train trips over the intra-urban rail network. Overall, only 1% of the total trips were allocated to train mode. Therefore, the number of trips for any zone pair is quite small. The trips are generally longer than the bus trips, as might be expected, and they also tend to fall along the major rail lines. Some of the trips start quite far from the rail lines, so it's possible that these are not realistic representations. Keep in mind that this is a mathematical model and is far from perfect.

Overall, the mode split routine has produced a reasonable approximation to travel modes for crime trips. Since there was no data upon which to calibrate the functions, reasonable guesses were made about the accessibility function. The mathematical model produced a plausible, though not perfect, representation of these assumptions, generally fitting into what we know about crime travel patterns.

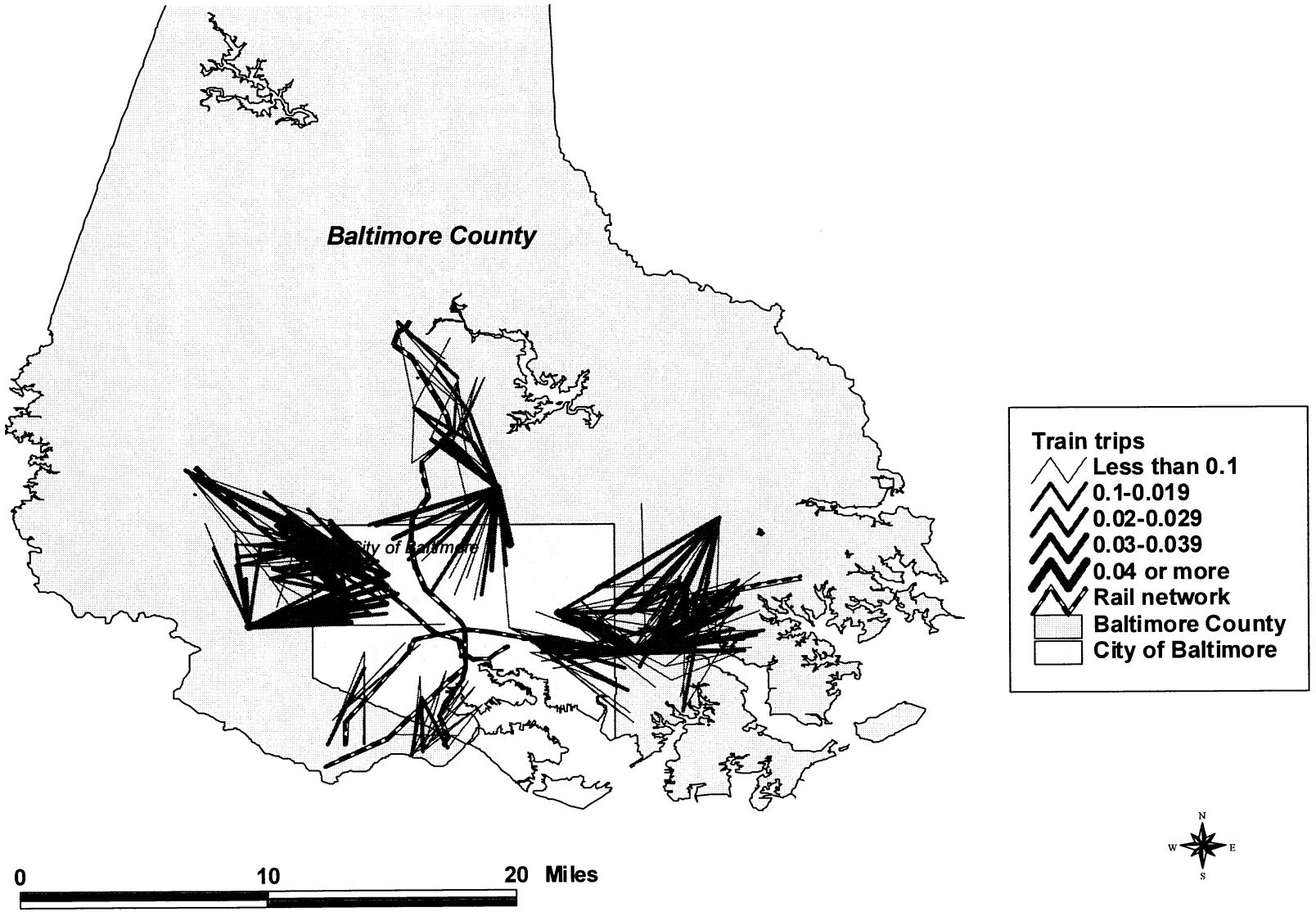
Usefulness of Mode Split Modeling

The mode split model is a logical extension of the travel demand framework. For transportation planning, it is an important step in the process. But, it also is important for crime analysis. First, it addresses the complexity of travel by separating the trips from specific origins to specific destinations into distinct modes. In this sense, it adds more realism to our understanding of criminal travel behavior. The Journey to Crime literature, which has been used by crime analysts and criminal justice researchers to "understand" criminal travel behavior, is simplistic in this respect. It assumes a single mode, though that is rarely articulated by the researchers. By pointing out typical travel distances by offenders circumvents the critical question of how they made the trip. This was, perhaps, not as critical 50-60 years ago when most crimes were committed within a smaller community and it could be assumed that most offenders walked to the crime location. But in post- World War era, automobile travel has become increasingly dominate. This model assumes that the vast majority of crime trips are taken by automobile. While there is currently no data to prove that assertion, it follows from the transportation patterns that have become widespread in the U.S. and elsewhere.

There is a second reason why an analysis of crime travel mode can be important. *If* the limitations of travel mode information could be improved through better and more careful data collection by police and other law enforcement agencies, this type of analysis could be very useful for policing. For one thing, it could allow more focused police deployment. For neighborhoods with a predominance of walking crime trips, then a police foot patrol could be most effective. Conversely, for neighborhoods with a predominance of driving crime trips, then patrol cars are probably the most effective. Police intuitively understand these characteristics, but the crime mode split model makes this more explicit.

Figure 15.14:

Mode Split: Intra-Urban Train Trips



For another thing, a mode split analysis of crime can better help crime prevention efforts. As the Baltimore data suggest, many of the local (intra-zonal) crime trips are committed around housing projects and in very low income communities. Most likely, this is a by product of poverty, lack of local employment opportunities, deteriorated housing, and even poor surveillance. Since teenagers are more likely to *not* own vehicles, it might be expected that the majority of these local crime trips are committed by very young offenders. This can be useful in crime prevention. Again, the "Weed and Seed" and after-school programs are generally targeted to youth from very low income neighborhoods. What is shown by the mode split analysis is probably the crime patterns associated with these neighborhoods. Even though it is intuitively understood, the mode split analysis quantifies these relationships in an explicit manner.

In short, a mode split analysis of crime trips is an important tool for crime analysts and criminal justice researchers. If correctly calibrated, it can help focus police enforcement and crime prevention efforts more specifically and can improve the theory of criminal travel behavior.

Hopefully, police departments will start to improve the quality of data in capturing likely travel modes while taking incident reports. Even though most police departments have an item similar to "Method of departure", there has not been a lot of emphasis on this information and most crime data sets are deficient on this information. However, with improved data will come more accurate accessibility functions and, hopefully, even real utility functions where actual costs are measured. The expectation is that this will happen and we should work towards accelerating the process.

Limitations to the Mode Split Methodology

There are also limitations to the method, particularly the aggregate approach. The aggregate approach does not consider individuals, only properties associated with zones (e.g., average travel time between two zones). As mentioned earlier, the accessibility function used (or the underlying utility theory) is much simpler for zones than for individuals. Consequently, the analysis is cruder at an aggregate level than at an individual level. Policy scenarios are much more limited with aggregate mode split than with individual-level models. For example, if an analyst wanted to explore what was the likely effect of increased public surveillance on walking behavior by pickpockets, it is more difficult to do with aggregate data than with individual data. For example, it could be hypothesized that actual pickpockets are more sensitive to increased public surveillance than, say, car thieves, but this can't be tested at the aggregate level. Instead, some general characteristics are assigned to all individuals (e.g., the number of security personnel in a zone).

Second, the zonal model for mode split (as with trip distribution) cannot explain intra-zonal travel. The accessibility function is applied to inter-zonal trips; for intra-zonal trips, it is inaccurate and generally defaults to simple choices (e.g., walking, biking or driving). For example, bus or train mode can rarely be applied at an intra-zonal level because there are usually too few network segments that traverse a zone and the segments

rarely stop within the zone. While this deficiency also applies to the trip distribution model, the dependence on a network for transit modes, particularly, lead to underestimation of transit use for very short trips.

Third, the zonal mode split model cannot explain individual differences. This goes back to the first point that a single utility function is being applied at the zonal level. Thus, the value of time to different individuals living in the same zone cannot be examined; instead, everyone is given the same value.

Fourth, the aggregate mode split model does not analyze time of day very well. The probabilities are assigned to all trips, rather than to trips taken at particular times of the day. To conduct that analysis, an analyst has to break down crimes by time of day and model the different periods separately. Aside from being awkward, the summed trips need to be balanced to ensure that they sum to the total number of trips.

Fifth, and finally, the mode split model, both aggregate and disaggregate, cannot explain *linked trips* (sometimes called *trip chaining*). An offender might leave home one day, go out to eat, visit a friend, commit a street robbery, go to a 'fence' to distribute the goods, buy drugs from a drug dealer, and then finally go home. The mode split model treats each of these as separate trips; in the case of crime mode split, there are three distinct crime trips - committing the robbery, selling the stolen goods to the 'fence', and buying the drugs from the drug dealer. The model doesn't understand that these are related events, but instead assigns separate mode probabilities to each trip. Thus, it is possible to produce absurd choices, such as driving to the crime scene, taking the bus to the drug dealer, and then biking home. In this respect, the disaggregate approach is equally flawed as the aggregate since both treat each trip as independent events. The solution to this lies in a 'third generation' of travel modeling in which individual trip makers are simulated over a day; *activity-based modeling*, as it is known, is still in a research stage (Goulias, 1996; Miller, 1996; Pas, 1996). But, it will eventually emerge as the dominant travel demand modeling algorithm.

Conclusions

Nevertheless, mode split modeling can be a very useful analysis step for crime analysis. It represents a new approach for crime analysis and one with many useful possibilities. It will require building more systematic databases in order to document travel modes. But, the possibilities that it offers up can be important for crime analysts and criminal justice researchers alike.

In the next chapter, the final step in the crime travel demand model will be discussed, network assignment.

Endnotes for Chapter 15

1. There is no reason this data could not be collected. Typically, many police departments collect information on 'Method of departure' from a crime scene. When a police report is taken, the victim is sometimes asked how the offender left the scene. In most cases, the information is not recorded on the police forms, or at least those that have been examined. This information is probably unreliable in any case since many offenders will take the bus or leave their car nearby while they walk/run to the crime scene. Still, if police departments were to put more effort into collecting this information and, perhaps, to validating it with arrested offenders, then it is possible to build up reliable data sets that can be used to model mode split. Until then, unfortunately, we have to rely on theory rather than evidence.

2. In a survey of the travel behavior of homeless persons, it was noted that most homeless walked very short distances over the day even though the value of their time was very low. For longer trips, they still tended to take the bus rather than walk. Survey on the travel behavior of very low income individuals. Urban Planning Program, University of California at Los Angeles, 1987 (with Martin Wachs).

3. In tests, I did find that the two models produced similar patterns. They were off in terms of the magnitude of the predicted trips, but the relative pattern was very similar.

4. Houston-Galveston Area Council. Personal communication. 2004.

Chapter 16
Network Assignment

In this chapter, the fourth, and last, component of the crime travel demand model will be described. Network assignment involves the assigning of predicted trips to particular routes. The predicted trips are those that are either predicted from the trip distribution stage or from the mode split stage. In the former case, all trips from each origin zone to each destination zone are assigned to a particular travel route, usually on the assumption that they all travel with the same mode of travel (usually walking, biking or driving). In the latter case, the predicted trips from each origin-destination zone pair by specific travel modes are assigned to a particular route which is mode specific. Thus, bus trips are assigned to bus routes; train trips are assigned to train routes; driving trips are assigned to a road network; walking trips are assigned to a more limited road network; and biking trips are assigned to a mixture of roads and bike paths. In other words, the assignment of travel modes is specific to a particular network.

Once the trips are assigned to routes, several statistics can be calculated. First, the predicted path from an origin zone to a destination zone can be displayed. This can very useful for police who could increase their patrol on high crime routes. Second, the entire trip load on road segments can be calculated. Since many crime trip routes pass over the same network segments (e.g., freeways, major arterial roads), the total number of predicted trips on a segment can be enumerated. The result is a map of the most heavily traversed segments in the network. Again, this can be very useful for police.

Thus, the network assignment completes the four stage modeling process of the crime travel demand framework. To summarize, in the first stage - trip generation, separate models of the number of crimes originating in each zone and the number of crimes ending in each zone are developed. In the second stage - trip distribution, the predicted number of crimes originating in each zone are allocated to each destination zone; the result is a prediction of the number of trips that occur between each origin-destination zone pair. In the third stage - mode split, each predicted origin-destination trip pair is separated (split) into distinct travel modes (e.g., walking, biking, driving, bus, train) with the result being a mode-specific origin-destination zone pair. Finally, the fourth stage - network assignment, assigns these trips to specific routes.

Theoretical Background

To understand the background, we need to look, first, at the nature of networks and second, at types of routing algorithms.

Networks

The most fundamental element of assignment is, of course, a network. The network can be a road network, a bus network (e.g., bus routes with stops), a train network (e.g., train lines with stations), or even a bicycle network (e.g., a mixture of roads and bicycle paths). Other kinds of networks can also be considered, for example telecommunication lines or even trade routes. We will concentrate on urban transportation networks, however.

The mathematical properties of networks are known as graph theory (Sedgewick, 2002). A network (or graph) is a set of nodes (or vertices) and a set of segments (or edges) that connect pairs of nodes. If there are V nodes (vertices), then there are V^2 pairs of nodes, including the distance from a node to itself. A graph with V nodes has, at most, V(V-1)/2 segments (edges); if multiple segments share nodes, then there will be even fewer.

Figure 16.1 illustrates a simple network. Travel occurs along the segments through the connecting nodes. A path is a sequence of nodes in which each successive node is connected to its predecessor in the path. Thus, in the figure, there cannot be direct travel between node A and node C, but must go through an intermediate node (e.g., through B or through a path from D to E to C).

Impedance of a Network

There are several properties of a network that are important for travel modeling. First, the length of a segment is proportional to its impedance (see chapters 14 and 15). The most simple kind of impedance is distance in which each unit length of the network corresponds to some unit of distance in the real world (e.g., one inch = 1 miles; one centimeter = 5 kilometers). This is analogous to the scale used in mapping systems. More complex types of impedance involve travel time, speed, or even generalized cost (a collection of several cost elements). Thus, to use the example in figure 16.1, node A is connected to nodes B and D. The path from A to B is 50 units long; similar lengths are found for the other segments in the example. This could represent distance (e.g., 50 miles), travel time (e.g., 50 minutes), or generalized cost (e.g., \$50).[1] To a graph, the units are irrelevant. As long as the user is explicit about these and consistent, path calculations will work properly.

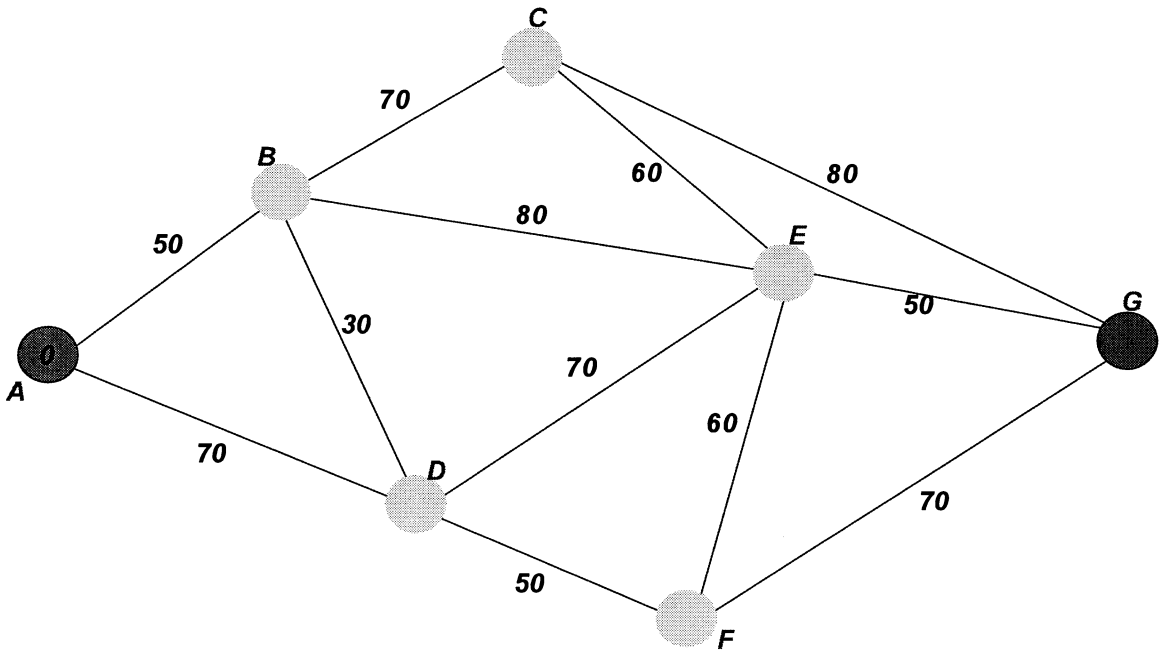
Bi-directional and Single Directional Networks

Bi-directional networks

Second, typical transportation networks are either bi-directional or single directional. In a bi-directional network, travel can occur in either direction. Again, using figure 16.1, if the network is bi-directional, then travel can occur

Figure 16.1:

Simple Network



from A to B or from B to A. A well known example of a bi-directional network is the TIGER system of the U.S. Census Bureau (2004). This is a representation of all major urban lines, including streets, railroad lines, census geography boundaries, jurisdictional boundaries, Congressional boundaries, and other features. It is used to map out census areas for the purpose of collecting the decennial Census. Virtually the entire United States is now mapped in the TIGER system. Depending on how carefully each jurisdiction updates the database for new roads and changes in existing roads, the TIGER system can be a very accurate spatial representation of the an urban road system. It is a widely available system and is often the first network that most police departments use when they create a crime mapping system. Figure 16.2 shows a TIGER network for Baltimore County and the City of Baltimore. There are 49,015 road segments in the TIGER map shown in the figure.

Problems with the TIGER system for travel modeling

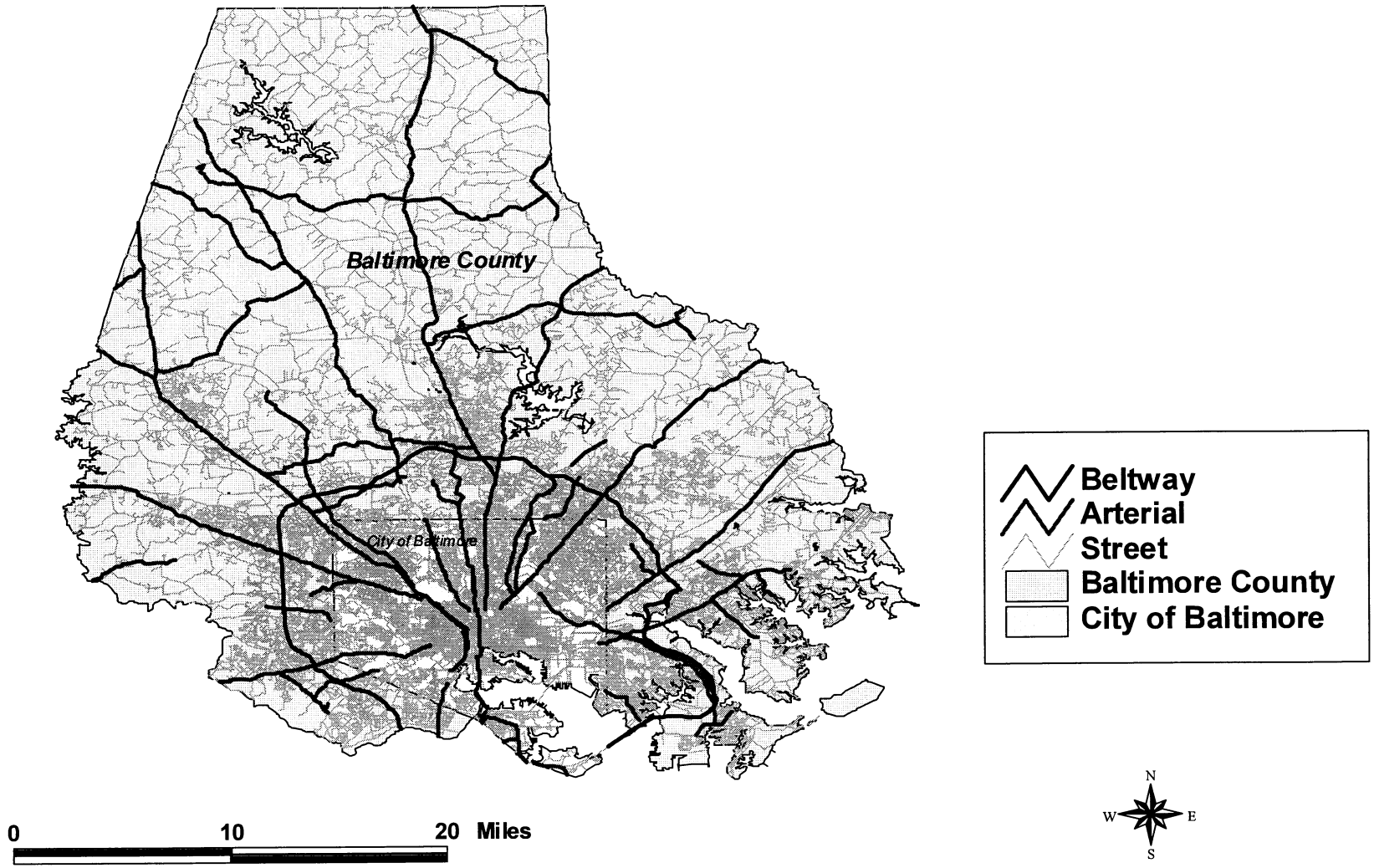
On the other hand, for travel modeling, there are substantial problems with bi-directional networks and with TIGER in particular. TIGER is typically less accurate with respect to rail lines and has virtually no information about bus routes, which are local in nature. Depending on how diligent the local government is in updating the database, the representation may not be as accurate as possible (though, in general, it's getting better over time).

A major problem is that connectivity is often not tested. Since the aim of the TIGER system is to represent a metropolitan area for the purpose of collecting the Census, connectivity is not guaranteed since it's irrelevant for that purpose. It's not clear that all roads are properly represented, a feature that could substantially alter a shortest path algorithm. For example, in figure 16.1, if the segment from A to B was not connected, then travel from A to C would have to take a circuitous path from A to D to E to C. Having an accurate and edited network is critical for modeling travel behavior. With a large number of segments in a TIGER system, it is often not clear where in a file connectivity is not properly linked.

Another major deficiency of the TIGER system is the lack of information about travel time or travel cost. Travel along a TIGER network is defined by distance, which does not change by time of day. It does not have cost information either, which makes it less flexible for examining alternative routes as a function of additional cost factors (e.g., an analysis of travel through an area with high surveillance versus travel through an area with low security presence even if travel through the first area is shorter in time than through the second area). The TIGER system does have information about functional class of road (interstate, state highway, collector road) and it's possible to assign a priori speeds to the different segments based on these classes (e.g., 35 miles per hour for Interstate highways, 25 miles per hour for

Figure 16.2:

TIGER Street Network
49,015 Road Segments



principal arterial roads). But, because the network is bi-directional, it's impossible to assign speeds for travel in opposite directions; in reality, there are usually differences in travel speeds in opposite directions (e.g., travel into the central business district in the morning might be at 15 miles per hour whereas travel in the opposite direction might be at 35 miles per hour).

Another major problem with TIGER and with a bi-directional network in general is in the representation of one-way streets. The TIGER system does not provide this information. Consequently, in using a TIGER file for modeling travel, a shortest path could easily travel up a one-way street in the wrong direction. To make the system work properly, there needs to be an additional field in the database that identifies a segment as one-way.

Single directional networks

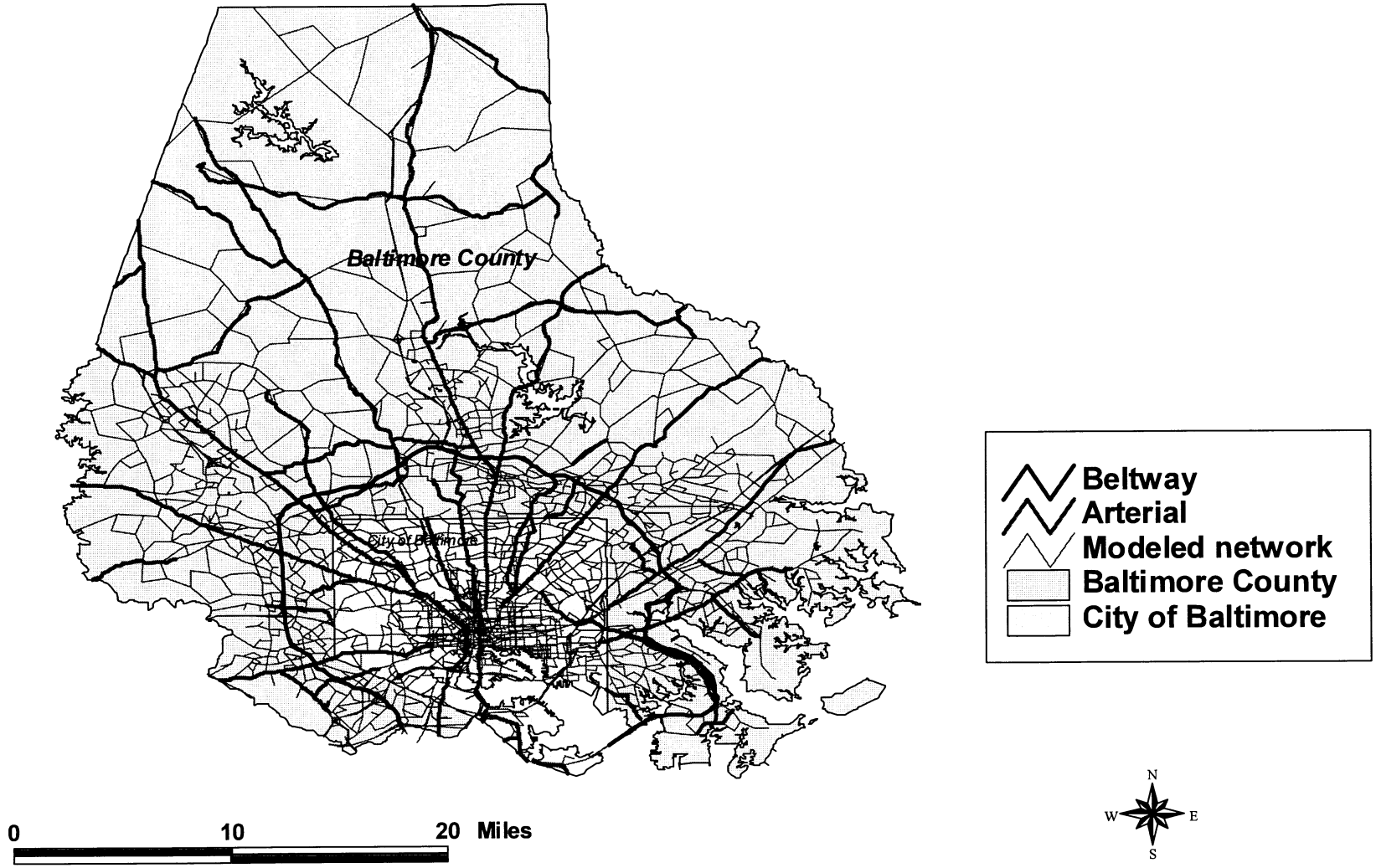
A single directional (or uni-directional) network, on the other hand, allows travel in only one direction. This has the advantage of keeping travel consistently defined. Two-way travel is represented by two segments, one in each direction (e.g., one for travel from A to B and one for travel from B to A). One-way streets can be characterized by only one of the paired directions. Most transportation modeling networks are single directional since an accurate representation of travel is critical. Travel times, speeds or costs can be assigned to the different directions of travel between two nodes and can be further assigned to different times of the day (e.g., 20 miles per hour in the morning peak period, 15 miles per hour in the afternoon peak period, 30 miles per hour in the off-peak daytime period, and 45 miles per hour at nighttime).

An example of a single directional network is that used for travel demand modeling by most Metropolitan Planning Organizations (MPO). These are used to model travel over an entire metropolitan area (regional travel) and are generally updated regularly; connectivity is continuously tested and errors are few in number. The travel modeling network is usually a 'skeleton' network, covering all the major roads - freeways, principal arterial roads, minor arterial roads, and some collector roads. They usually do not include much information about local or neighborhood streets since these are not very relevant for regional travel modeling. Figure 16.3 shows a modeling network used by the Baltimore Metropolitan Council for their travel demand model. There are only 11,045 road segments in the file, less than one fourth the size of the corresponding TIGER network. Considering that each segment in a single direction, effectively only about 5,000-6,000 actual roads are being represented in the file.

Most importantly, modeling networks usually include information about travel time or travel speed (which can be converted to travel time by dividing distance by speed) and are usually broken down into different time periods. Thus, it becomes possible to analyze travel at different times of the day to account for the major congestion effects that occur at the peak travel

Figure 16.3:

Modeled Street Network
11,045 Road Segments



periods, particularly the afternoon peak. Some modeling networks also include information on travel costs, which include parking, toll roads, and other costs that impact a trip. As mentioned in chapter 11, any analyst wishing to develop a crime travel demand model should contact the local MPO about obtaining a copy of the modeling network used.

> Hint: A single directional network can also be treated as bi-directional. In this case, all the trips on that roadway will generally be assigned to only one of the paired segments (for a two-way pair). For the network load output, particularly, this can be useful for showing the total number of trips on a road segment, independent of direction. Otherwise, if defined as a single directional network, the loads in each direction will be displayed separately.

Problems with modeling networks

Modeling networks also have their problems. The biggest one is that they do not include all roads, but only the more important regional ones. This can lead to unrealistic paths being modeled at a neighborhood level (e.g., entering or leaving a neighborhood from a centroid, rather than from a real street; taking circuitous routes to travel a short distance in space when, in fact, there are connecting local roads that actually exist but aren't included in the file). However, neighborhood roads can usually be added to the network to provide more detail at the neighborhood level and to correct modeling errors. It's a tedious process, but a police department could slowly update such a system over time and improve its accuracy. Care must be taken in doing this, however, to ensure that connectivity is correctly portrayed.

Another problem, which may or may not be critical, is that the representation of roads in a modeling network is spatially simplified. Road segments are straight lines, rather than having curvature. In the TIGER system, the basic record of a segment is a straight line connecting two nodes, but also includes up to 10 intermediate 'shape grammar' nodes that define curvature (integrated with spatially more accurate information from the U.S. Geological Survey). Thus, a modeling network looks a little 'unreal' at a neighborhood level since there are nothing but straight lines. But, as mentioned above, additional segments can be added to the file to improve local connectivity as well as familiarity.

Transportation Networks

The third property of a network for travel modeling is the type of network. Road networks were mentioned above. But there are also transit networks (e.g., bus routes, train routes) and even bicycle networks (e.g., bike paths).

If a trip distribution matrix of trips from origins to destinations is analyzed by travel mode, then it is critical to have a mode-specific network. Using TIGER or a simple modeling network will imply that all trips occur by the existing road system. For transit trips (bus and rail) particularly, but also for biking trips and possibly walking trips, features that are specific to the travel mode must be included. Bus routes will use the existing road system, but they don't use all roads, typically only the major arterial roads. Train systems rarely use the existing road system, but usually have dedicated tracks. There are exceptions. Some light rail systems do run on arterial roads. Other rail systems will run on an arterial road, but with a grade separation. Depending on how the MPO conceptualizes this, there may be separate lines for the rail or not.

Thus, it's very important to check and edit all networks that are used. For transit networks, in particular, the lines needs to be connected and thoroughly tested. Figure 16.4 repeats the Baltimore bus network map from chapter 15 (figure 15.12). Each of the lines on the map represent bus routes; there can be (and usually are) more than one bus route at any one line. Typically, these are drawn as separate line objects and are overlaid on each other. This particular network does not have information about bus stops. Consequently, a shortest path algorithm will choose the end nodes of segments to allow a trip to "enter" or "leave". Thus, it is possible that a bus trip would start at a location where there is not a bus stop. However, given that buses in Baltimore and elsewhere stop very frequently (every two or three blocks on average), the amount of error introduced is quite small.

With trains, however, it is absolutely critical that station locations be used to define the rail lines; people cannot enter or leave a train between stations. Figure 16.5 shows each of the four intra-urban rail lines with the station locations. Later in this chapter, there will be a discussion of a utility for creating rail lines from station locations. But a critical point is that each of the end points of the rail segments be associated rail stations. In the figure, each of the four rail lines is shown in separate color. For modeling in CrimeStat, however, the individual lines need to be merged into a single file in order for the shortest path routine to be able to move between rail lines (i.e., if there are separate line objects for each line, the routine will not know how to move from one line to another). Figure 16.6 shows the full rail line network.

Shortest Path Algorithms

Once a network has been created, edited and thoroughly tested for accurate connectivity, it can be used for a shortest path analysis. In a shortest path for a single trip (from an origin zone to a destination zone), the route with the lowest overall impedance is selected. As mentioned, impedance can be defined in terms of distance, travel time, speed, or generalized cost.

Figure 16.4:

Baltimore Bus Network
Existing Bus Routes: 2004

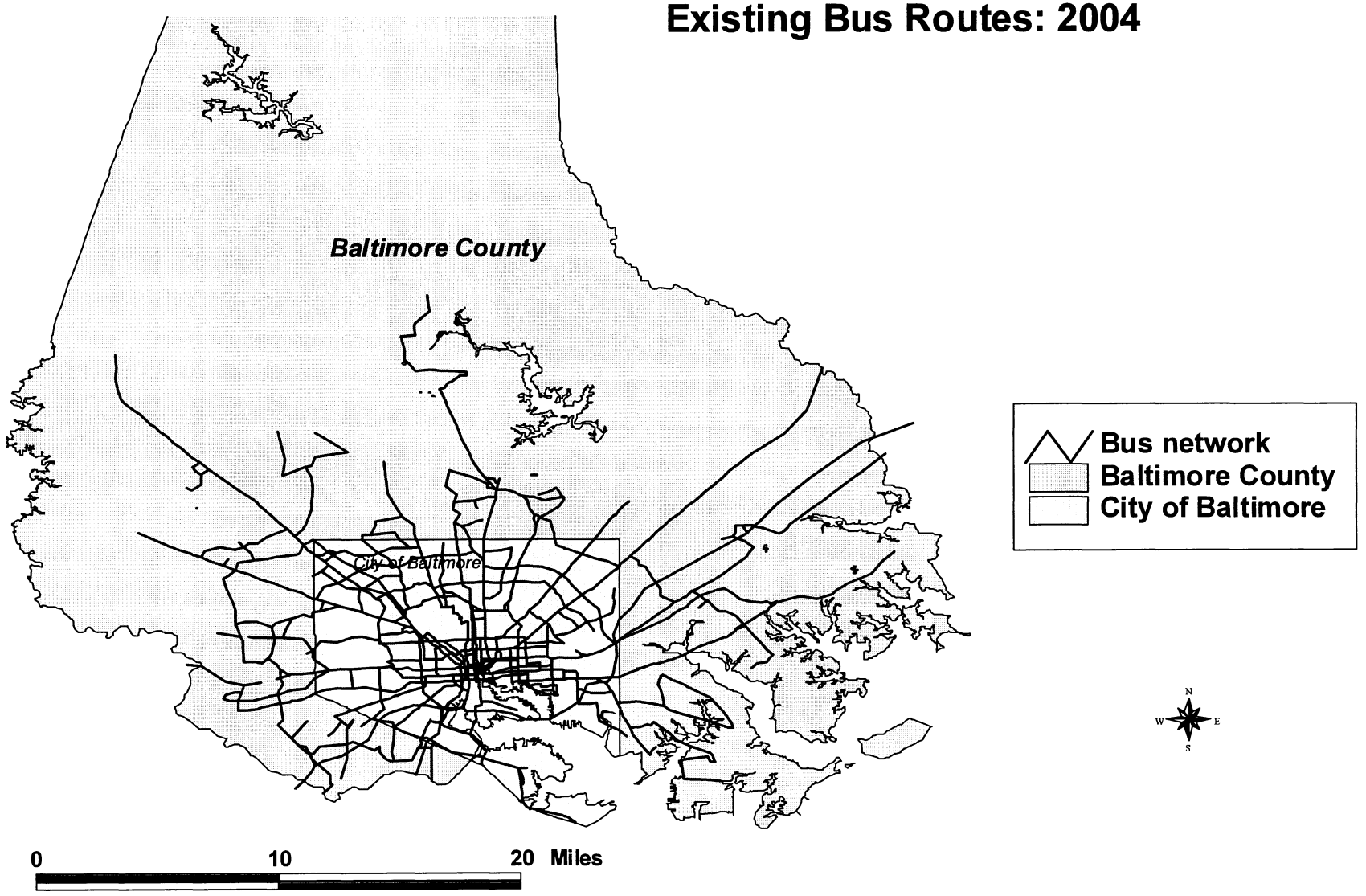


Figure 16.5:

Baltimore Intra-urban Rail Network: 2004
Intra-urban Lines and Rail Stations

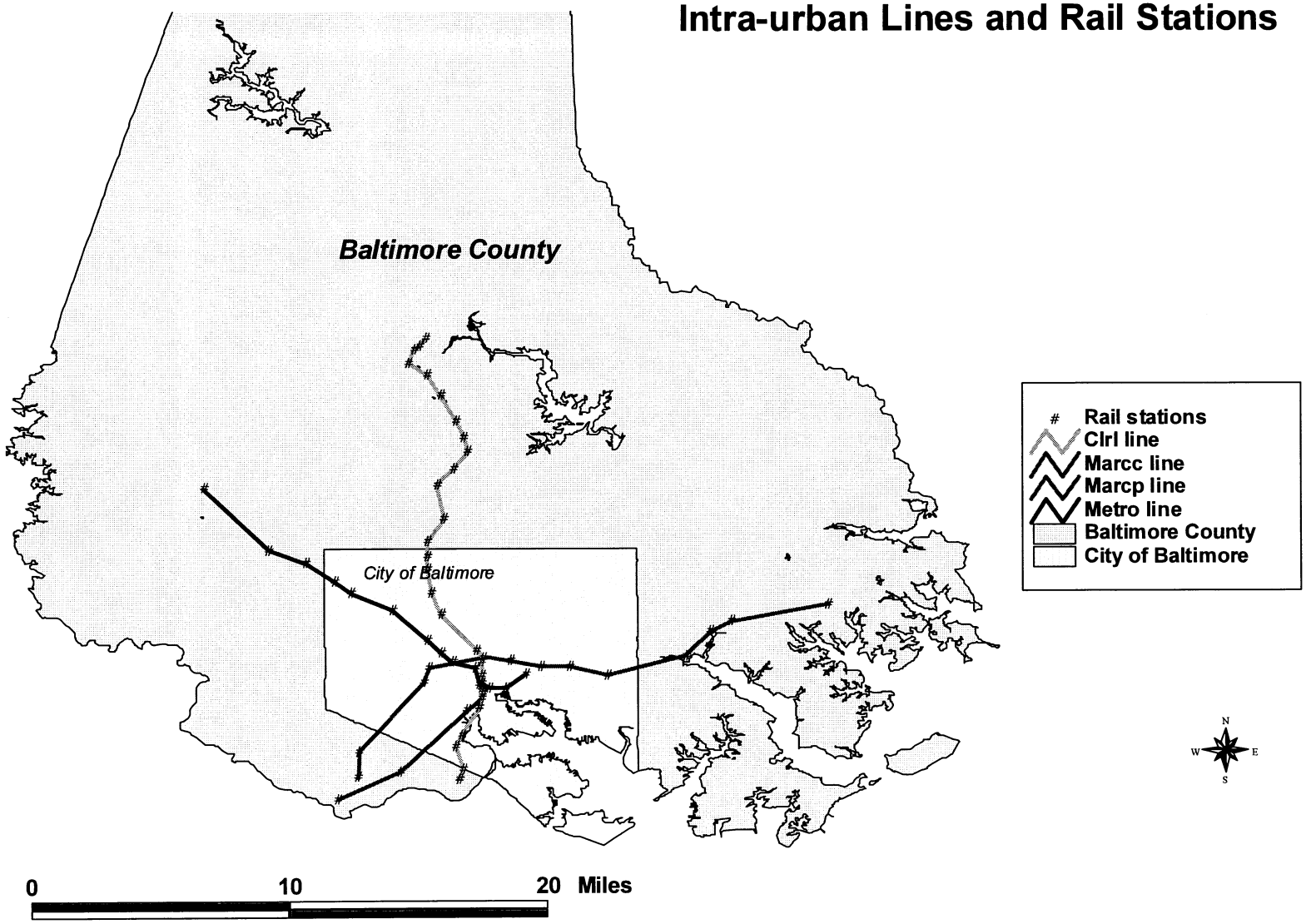
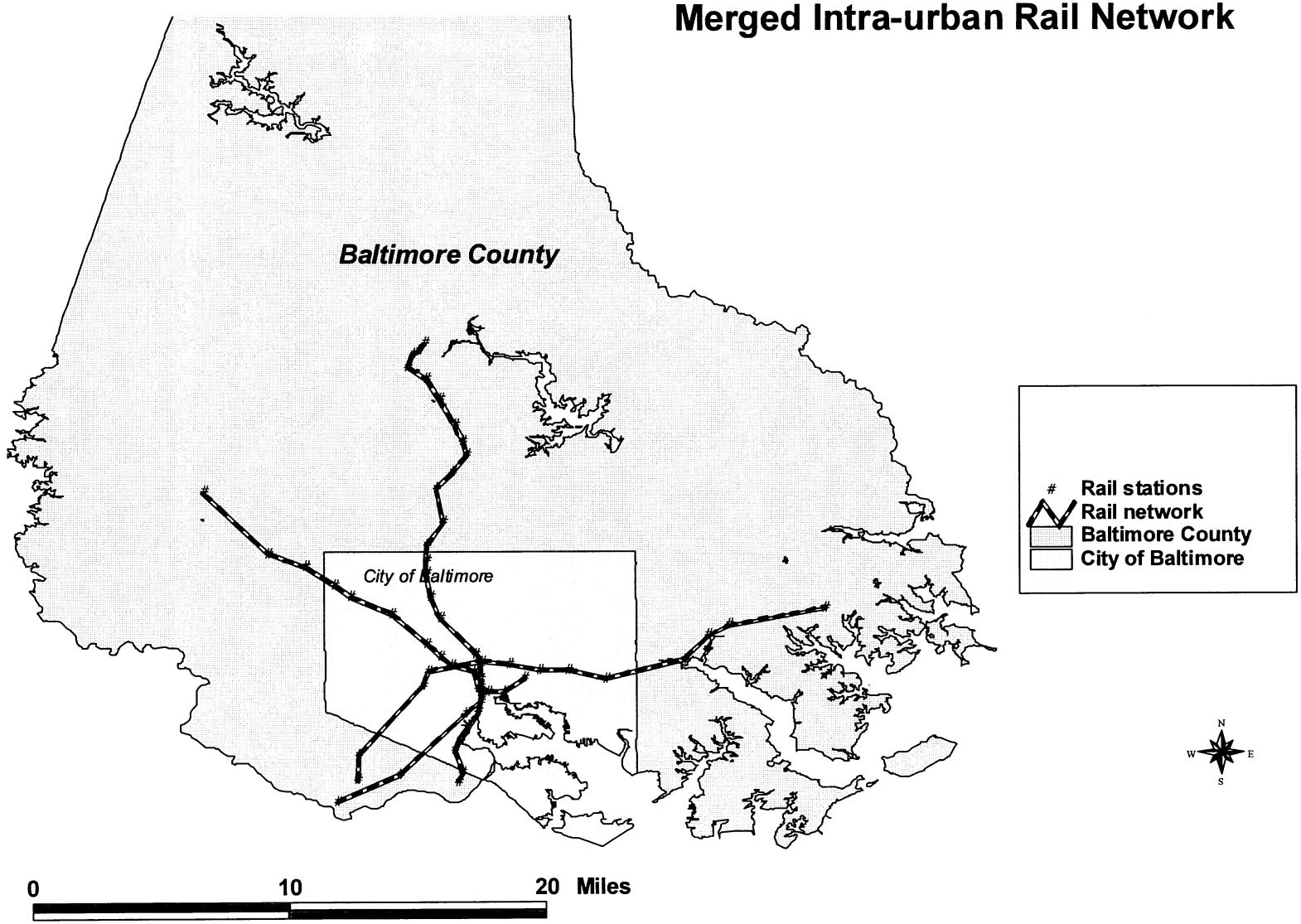


Figure 16.6:

Baltimore Rail Network
Merged Intra-urban Rail Network



There are a number of shortest path algorithms that have been developed (Sedgewick, 2002). They differ in terms of whether they are breadth-first (i.e., search all possibilities) or depth-first (i.e., go straight to the target) algorithms and whether they examine a one-to-many relationship (i.e., from a single origin node to many nodes) or a many-to-many relationship (All pairs; from each node to every other node).

The algorithm that is most commonly used for shortest path analysis of moderate-sized data sets (up to a million cases) is called A*, which is pronounced "A-star" (Nilsson, 1980; Stout, 2000; Rabin 2000a, 2000b; Sedgewick, 2002). It is a one-to-many algorithm but is an improvement over another commonly-used algorithm called Dijkstra (Dijkstra, 1959). Therefore, I'll start first by describing the Dijkstra algorithm before explaining the A* algorithm.

Dijkstra Algorithm

The Dijkstra algorithm is a one-to-many search strategy in which a shortest path from a single node to all other nodes is calculated. The routine is a breadth-first algorithm in that it searches all possible paths, but it builds the path one segment at a time. Starting from an origin location (node), it identifies the node that is nearest to it and which has not already been identified on the shortest path. After each node has been identified to be on the shortest path, it is removed from the search possibilities. The algorithm proceeds until the shortest path to all nodes has been determined. In terms of a matrix of origin nodes (on the vertical) and destination nodes (on the horizontal - see figure 14.1 in chapter 14), the search algorithm estimates the shortest path for any one row (i.e., from a particular origin to all destinations).

The algorithm can also be structured to find the shortest path between a particular origin node and a particular destination node. In this case, it will quit once the destination node has been identified on the shortest path. The algorithm can also be structured to find the shortest path from each origin node to each destination node. It does this one path at a time (e.g., it finds the shortest path from node A to all other nodes; then it finds the shortest path from node B to all other nodes; and so forth).

Let's use the network in figure 16.1 as an example. Figure 16.7 presents the network in terms of a particular origin node (A = Start) and a particular destination node (G = Finish). In the first step (not shown), the algorithm finds the node that is closest to A that has not already been put on the shortest path. In this case, it is to itself (i.e., A to A is the shortest path at this point). It thus removes A from the list of possible nodes and puts it in a shortest path node list. Next, the routine finds the node that is closest to A that has not already been put on the shortest path list. This will be node B, which is 50 units distance from A (figure 16.8). Thus, the shortest path now

Figure 16.7:

Example of Dijkstra Algorithm

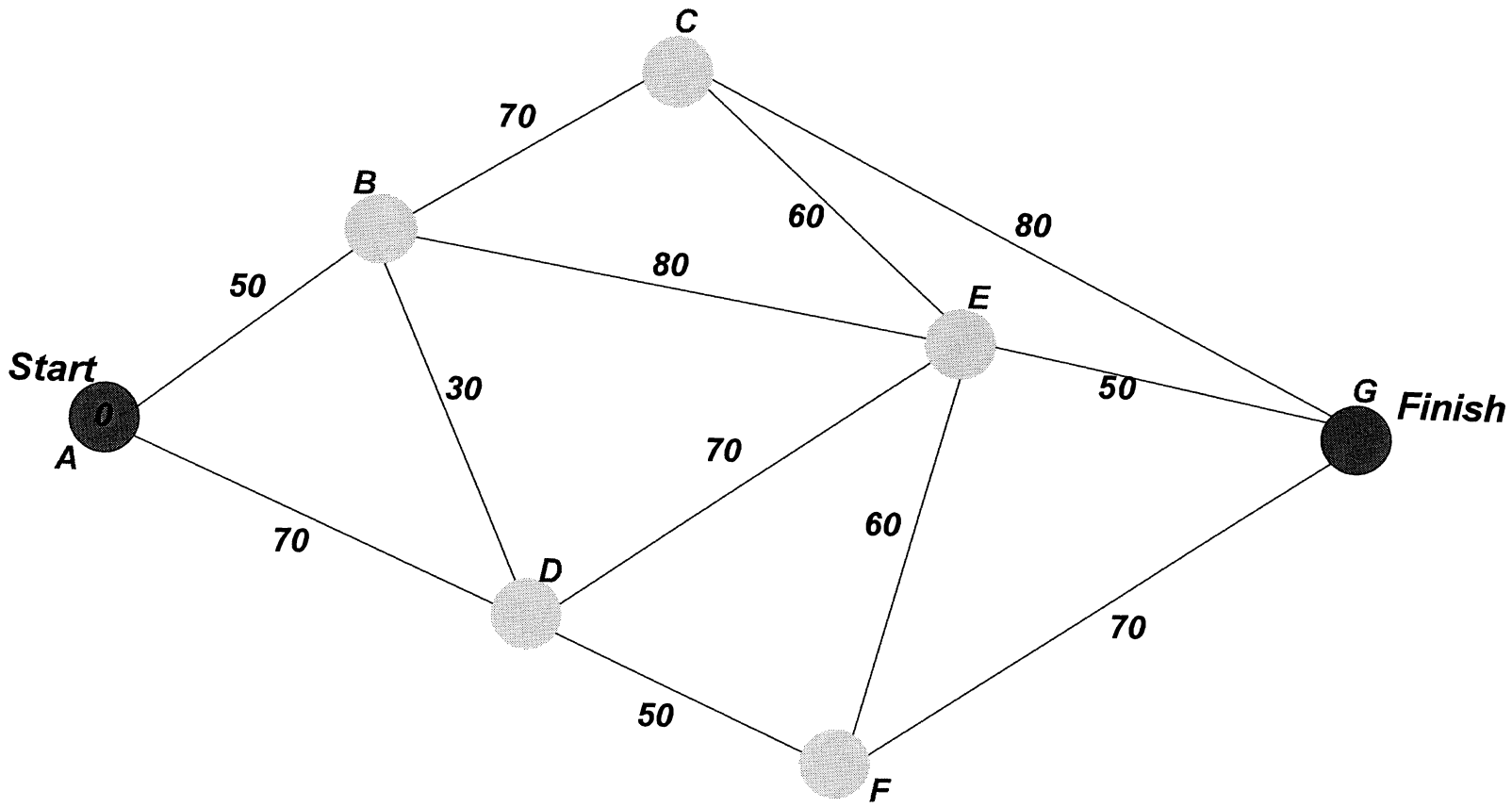
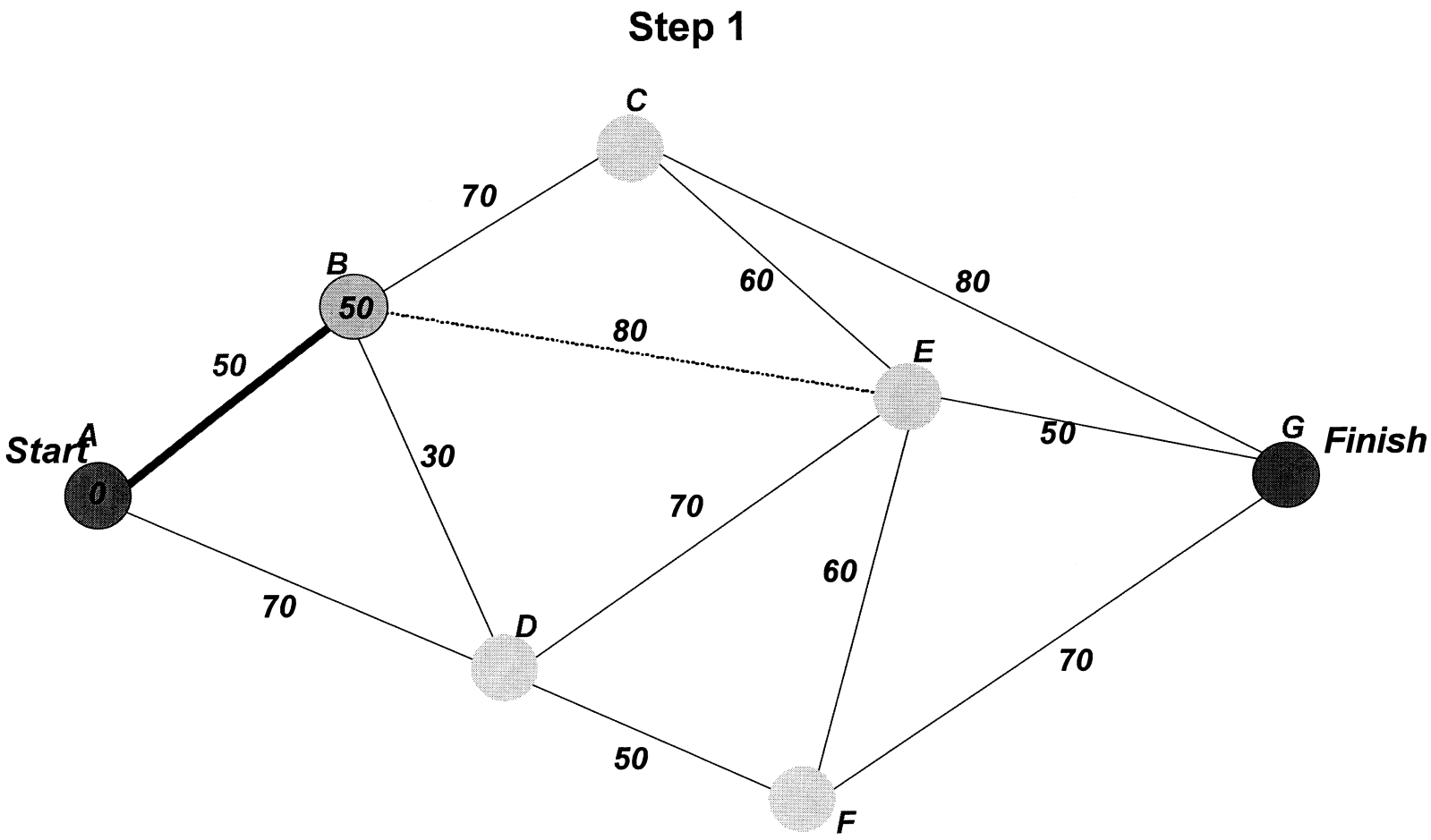


Figure 16.8:

Example of Dijkstra Algorithm



goes from A to B. Subsequently, node B is removed from the list of possible new nodes and is put on the shortest path list.

In step 2, the routine finds the node that is closest to one of the existing nodes on the shortest path list but which has not already been put on that list. This will be node D, which is 70 units from A (figure 16.9). That is, if A and B have already been put on the shortest path list, then only two nodes are connected to these - C and D. The distance from A to C is 120 (50 + 70) while the distance from A to D is 70. Thus, the routine selects node D next. Subsequently, node D is removed from the list of possible new nodes and is put on the shortest path list.

In step 3 (figure 16.10), the routine determines the node that is closest to A and which has not yet been put on the shortest path. There are two possibilities - C and F; both are 120 units distance from A. In the case of a tie, the routine 'flips a coin' and chooses one, in this case node F. Subsequently, node F is removed from the list of possible new nodes and is put on the shortest path list.

In step 4 (figure 16.11), the routine adds node C to the shortest path. Note that had the 'coin flip' in step 3 chosen node C instead of F, in this stage node F would have been selected; thus, the routine produces the same solution, just in a different order. Both nodes C and F are 120 units distance from node A. Node C is now removed the list of possible new nodes and is put on the shortest path list.

In step 5 (figure 16.12), the routine adds node E to the shortest path list because the distance to E through B is shorter than any other route that has not yet been determined (130 units from A). Notice that the path to E through C or D would have been longer than through B (180 and 140 units respectively).

Finally, in step 6 (figure 16.13), the routine goes to the finish, node G. The path through B and E is shorter than by any other path to G (180 total units). Thus, the Dijkstra algorithm has searched every node in the network and determined a shortest path from node A to each of them (figure 16.14). Even though we are only interested in the path from A to G, the algorithm solves all shortest paths from A to all nodes.

A* Algorithm

The biggest problem with the Dijkstra algorithm is that it searches the path to every single node. If the purpose were to find the shortest path from a single node to all other nodes, then this would produce the best solution. However, with an origin-destination matrix, we really want to know the distance between a pair of nodes (one origin and one destination). Consequently, the Dikjstra algorithm is very, very slow compared to what we

Figure 16.9:

Example of Dijkstra Algorithm

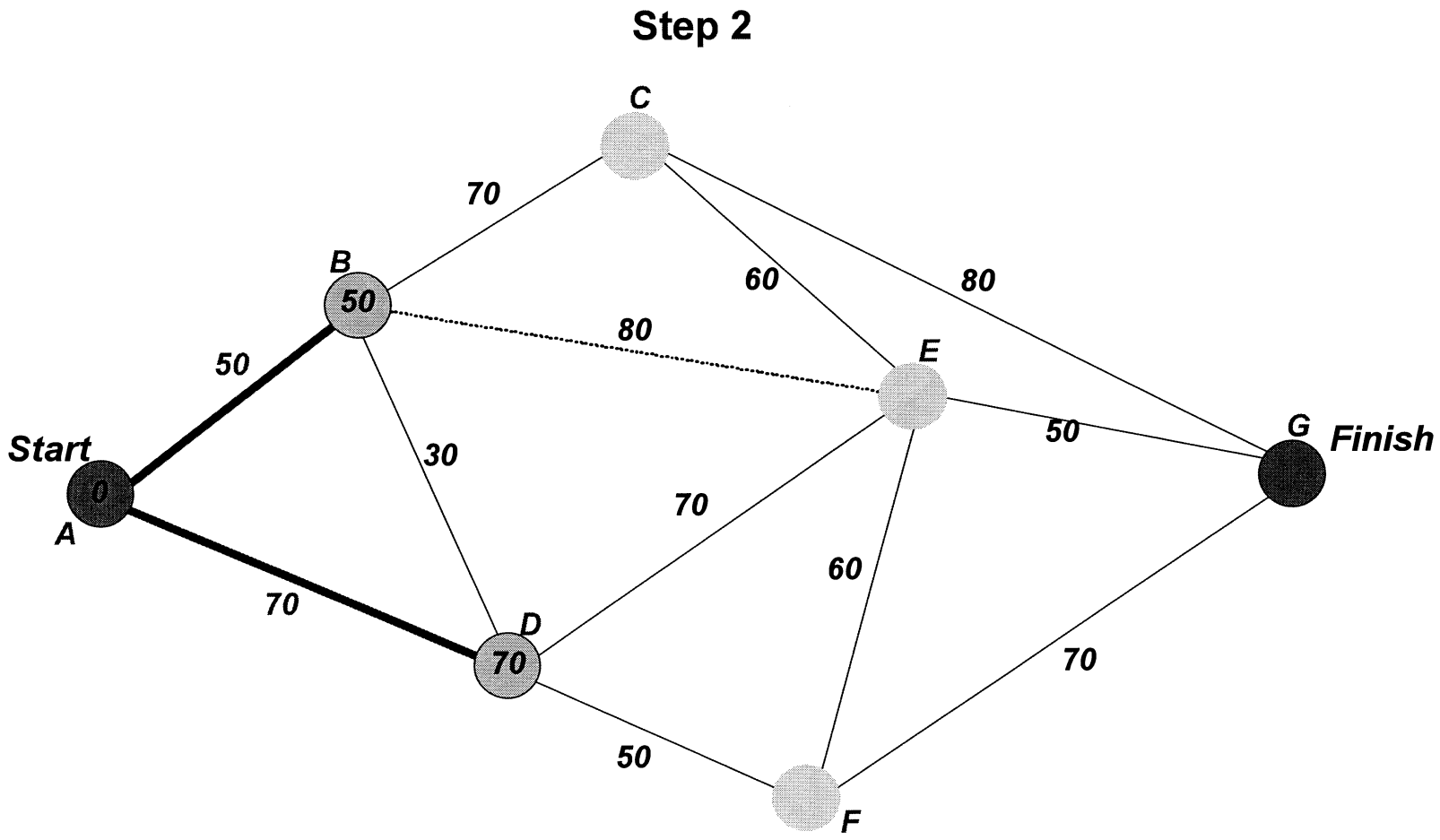


Figure 16.10:

Example of Dijkstra Algorithm

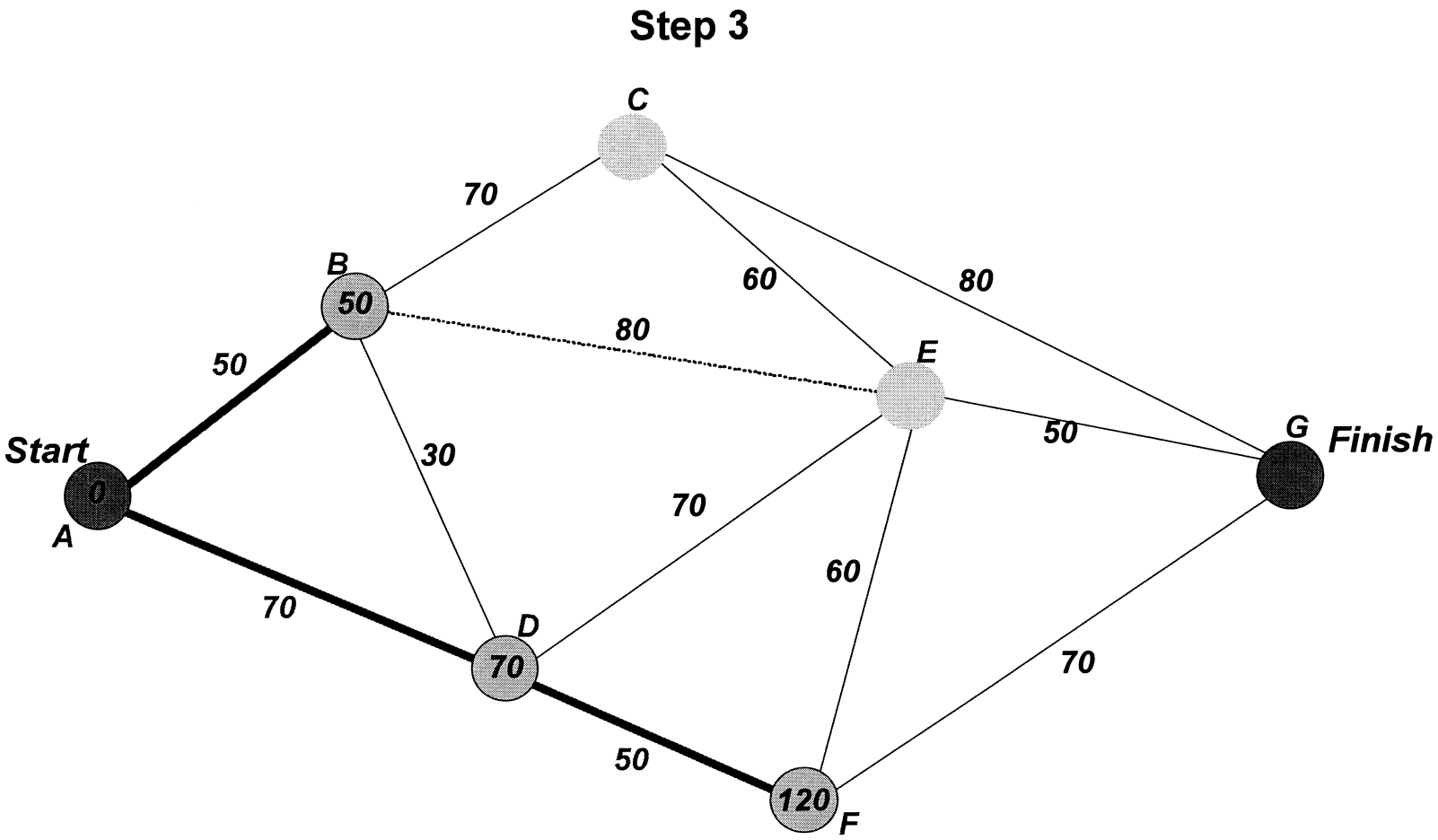


Figure 16.11:

Example of Dijkstra Algorithm

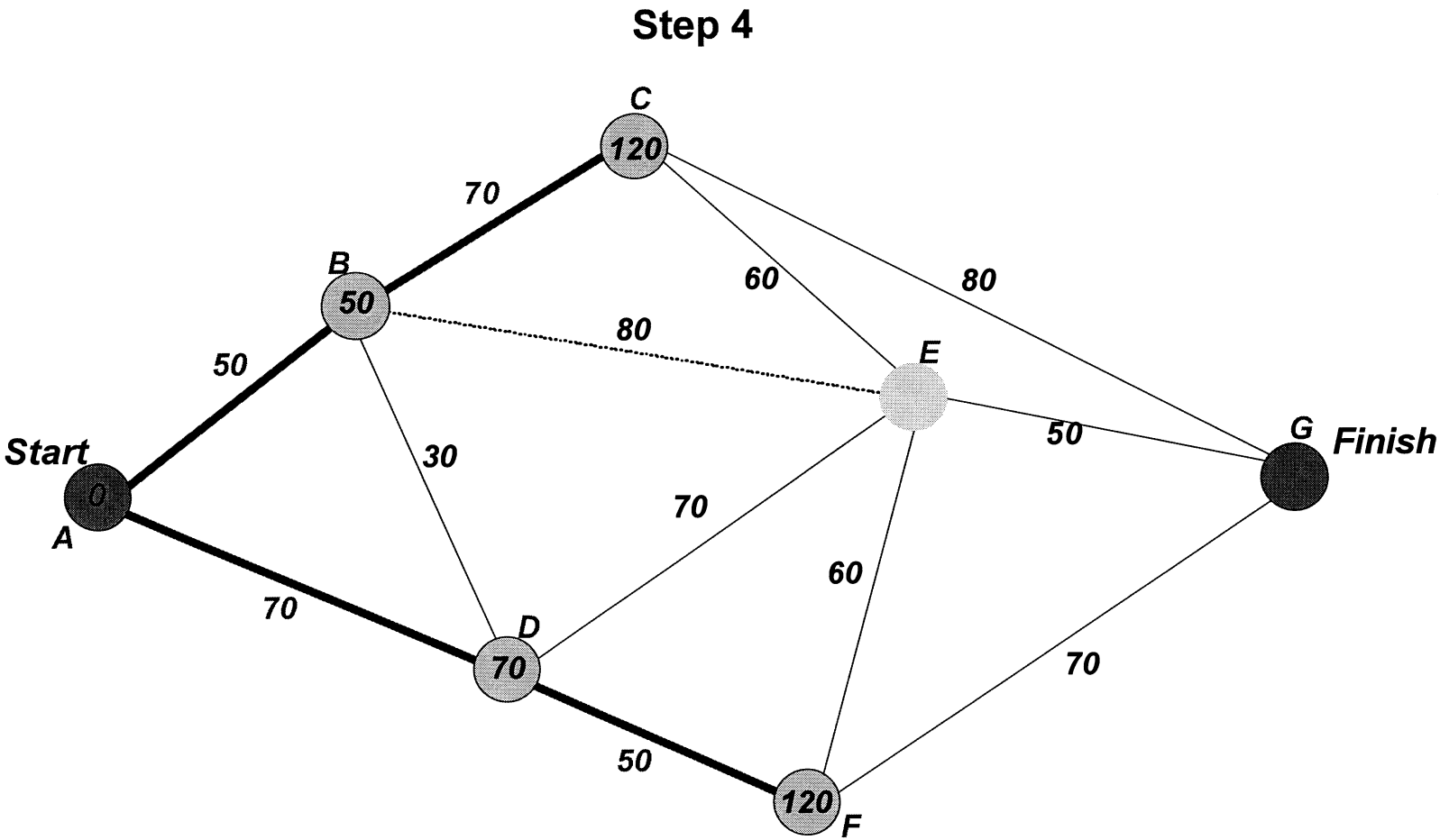


Figure 16.12:

Example of Dijkstra Algorithm

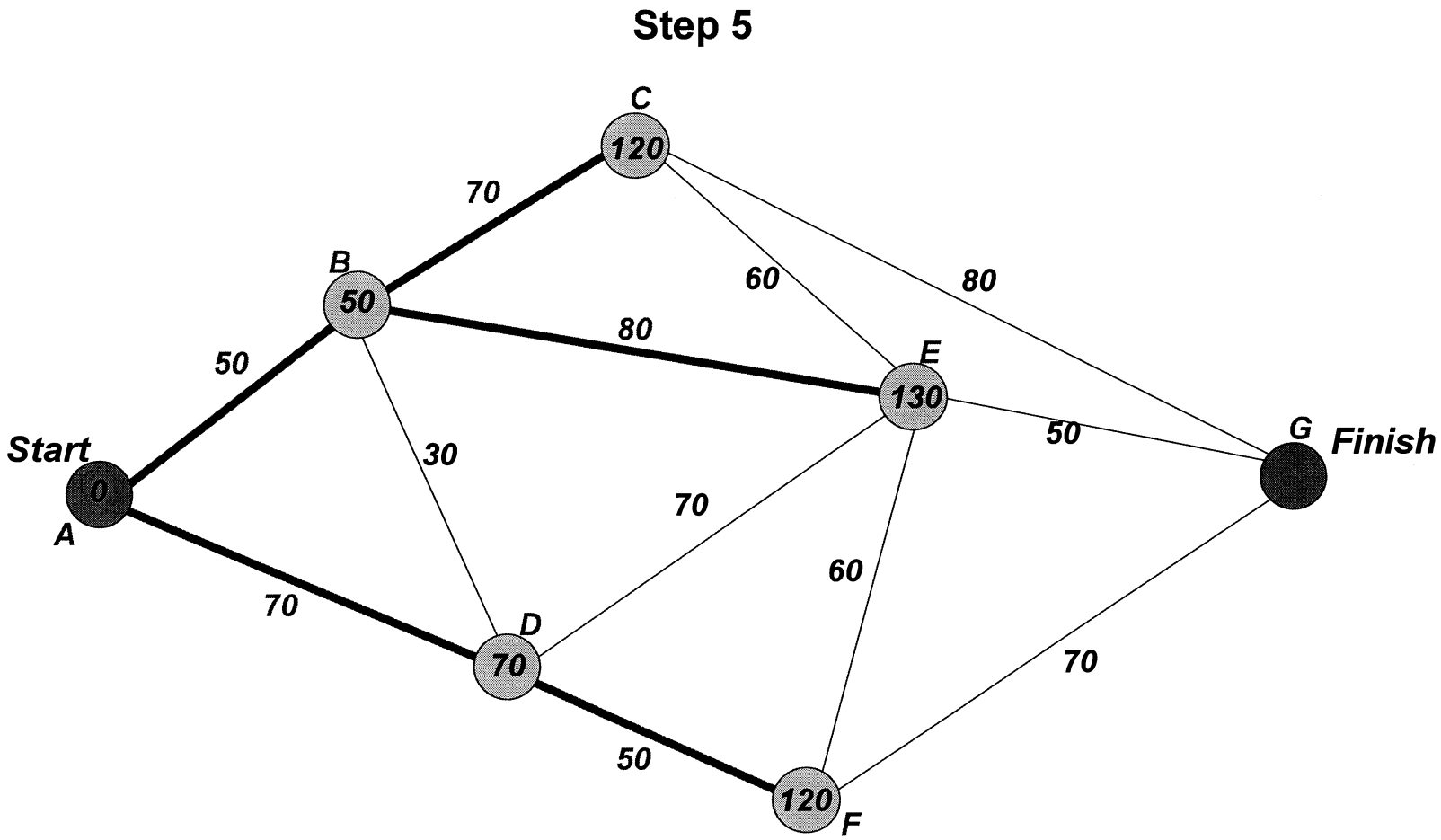


Figure 16.13:

Example of Dijkstra Algorithm

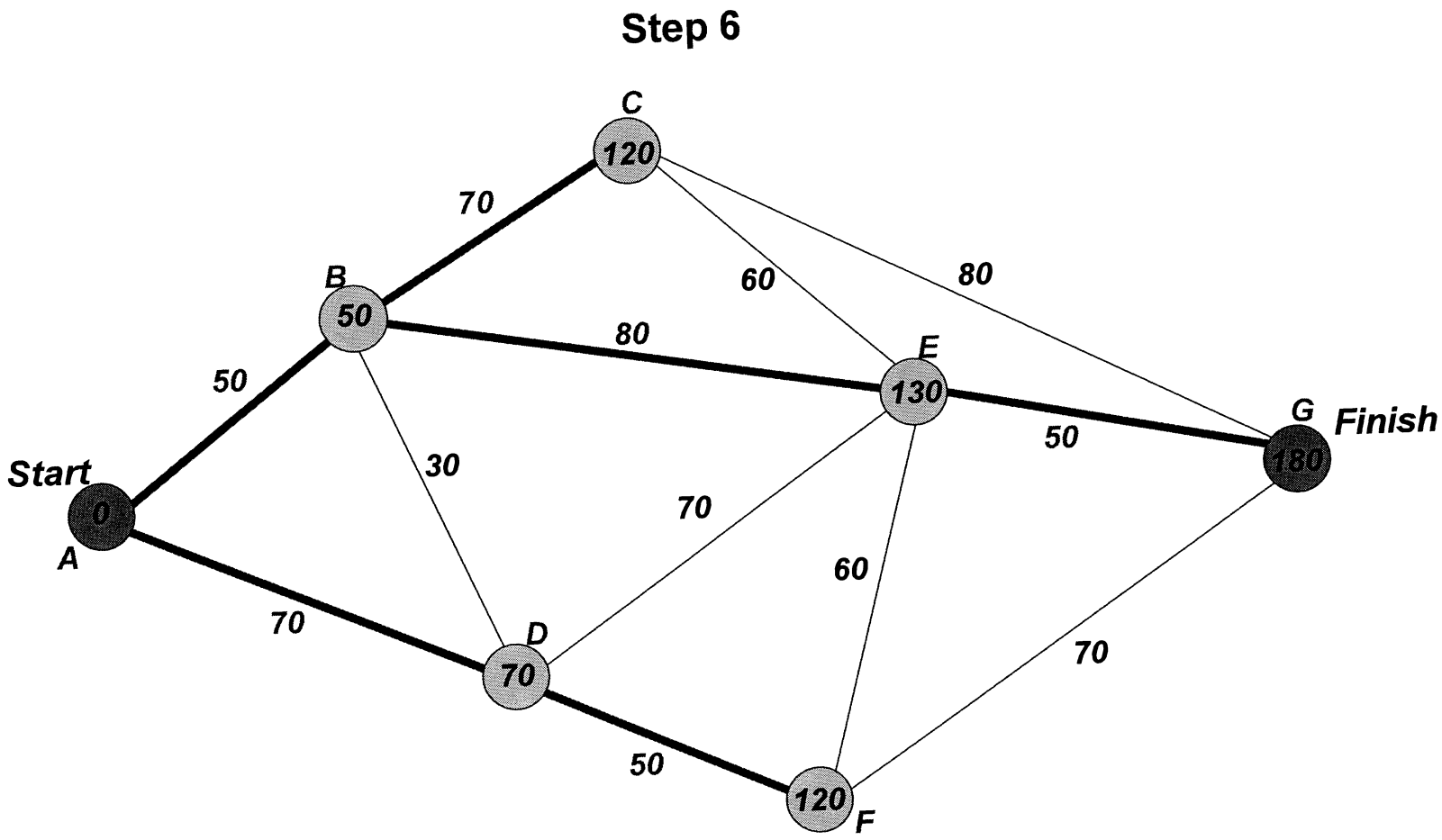
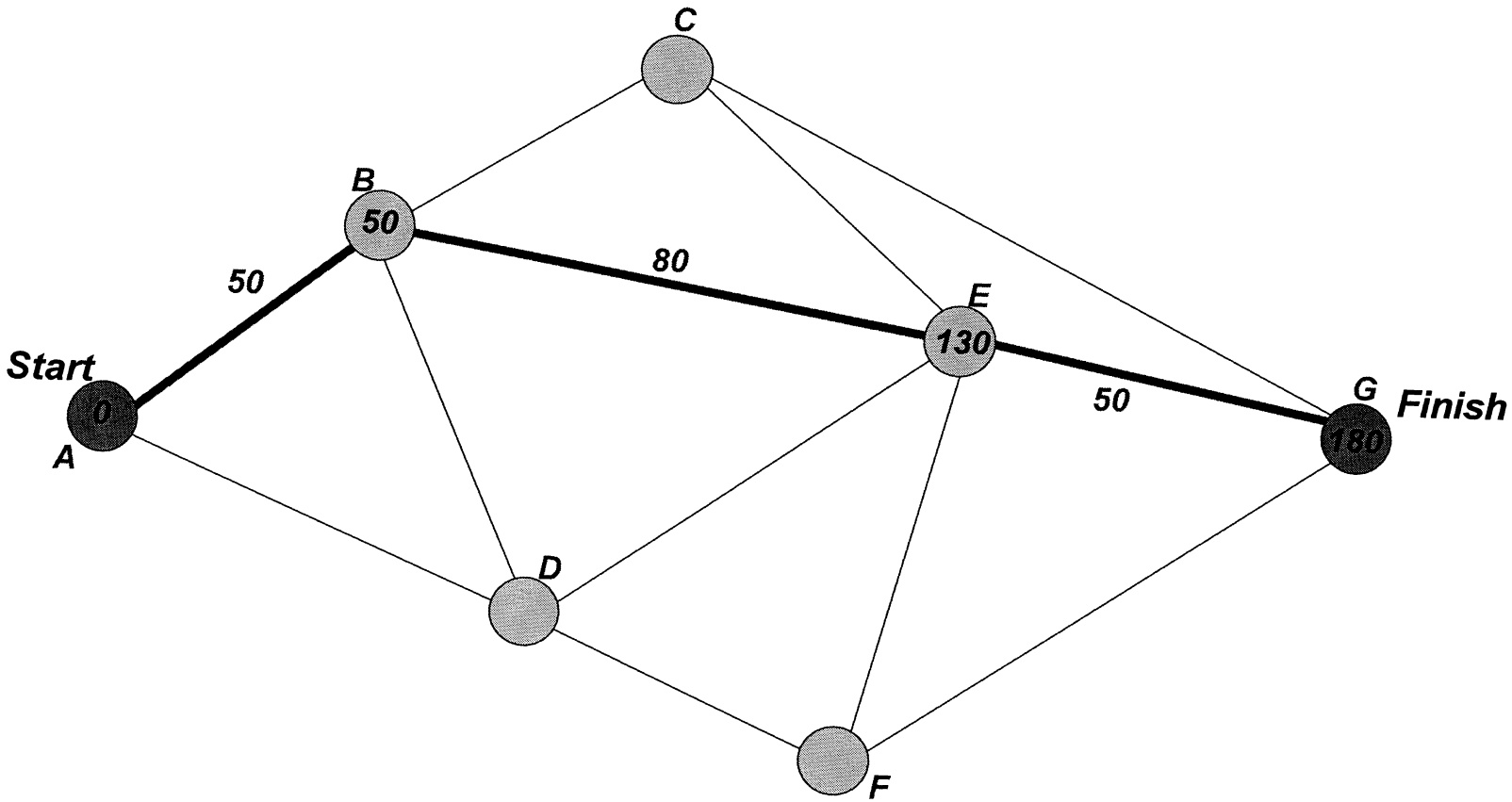


Figure 16.14:

Example of Dijkstra Algorithm

Shortest Path from Start to Finish



need. It would be a lot quicker if we could find the distance from each origin-destination pair one at a time, but quit the algorithm as soon as that distance has been determined.

This is where the A* algorithm comes in. A* was developed within the artificial intelligence research area as a means for developing a heuristic rule for solving a problem (Nilsson, 1980). In this case, the heuristic rule is the remaining distance from a solved node to the final destination. That is, at every step in the Dijkstra routine, an estimate is made of the remaining distance from each possible choice to the final destination. The node that is chosen for the shortest path is that which has the least total combined distance from the previously determined node to the final goal. Thus, for any step, if D_{i1} is the distance to a node, i, which has not already been put on the shortest path and D_{i2} is an estimate of the distance from that node to the final destination, the estimated total distance for that node is:

$$D_i = D_{i1} + D_{i2} \qquad (16.1)$$

Of all the nodes that could be chosen, the node, i, which has the shortest total distance is selected next for the shortest path. There are two caveats to this statement. First, the node, i, cannot have already been selected for the shortest path; this is just re-stating the rules by which we search for nodes which have not yet been put on the shortest path list. Second, the estimate of the remaining distance to the final destination must be less than or equal to the actual distance to the final destination. In other words, the estimated distance, D_{i2}, cannot be an overestimate (Nilsson, 1980). However, the closer the estimated distance is to the real distance, the more efficient will be the search.

How then do we determine a reasonable estimate for D_{i2}? The answer is a straight line from the possible node to the final destination since the shortest distance between two points is a straight line (or, on a sphere, a Great Circle distance since the shortest distance between two points is an arc). If we simply calculate the straight-line from the node that we are exploring to the final node, then the heuristic will work.

Let's look at the example again. Figure 16.15 displays the network again. Like the Dijkstra algorithm, the routine first finds a node closest to A, which is itself. Next, it finds a node that has the least total distance from A to the final destination, G (figure 16.16). There are two possibilities, go through B or go through D. The distance from A to B is 50 and the remaining distance from B to G is 130. Thus, the total distance through B would be 180. On the other hand, the distance from A to D is 70 and the remaining distance from D to G is 120. Thus, the total distance through D would be 190. Since 180 is smaller than 190, we choose node B.

need. It would be a lot quicker if we could find the distance from each origin-destination pair one at a time, but quit the algorithm as soon as that distance has been determined.

This is where the A* algorithm comes in. A* was developed within the artificial intelligence research area as a means for developing a heuristic rule for solving a problem (Nilsson, 1980). In this case, the heuristic rule is the remaining distance from a solved node to the final destination. That is, at every step in the Dijkstra routine, an estimate is made of the remaining distance from each possible choice to the final destination. The node that is chosen for the shortest path is that which has the least total combined distance from the previously determined node to the final goal. Thus, for any step, if D_{i1} is the distance to a node, i, which has not already been put on the shortest path and D_{i2} is an estimate of the distance from that node to the final destination, the estimated total distance for that node is:

$$D_i = D_{i1} + D_{i2} \qquad (16.1)$$

Of all the nodes that could be chosen, the node, i, which has the shortest total distance is selected next for the shortest path. There are two caveats to this statement. First, the node, i, cannot have already been selected for the shortest path; this is just re-stating the rules by which we search for nodes which have not yet been put on the shortest path list. Second, the estimate of the remaining distance to the final destination must be less than or equal to the actual distance to the final destination. In other words, the estimated distance, D_{i2}, cannot be an overestimate (Nilsson, 1980). However, the closer the estimated distance is to the real distance, the more efficient will be the search.

How then do we determine a reasonable estimate for D_{i2}? The answer is a straight line from the possible node to the final destination since the shortest distance between two points is a straight line (or, on a sphere, a Great Circle distance since the shortest distance between two points is an arc). If we simply calculate the straight-line from the node that we are exploring to the final node, then the heuristic will work.

Let's look at the example again. Figure 16.15 displays the network again. Like the Dijkstra algorithm, the routine first finds a node closest to A, which is itself. Next, it finds a node that has the least total distance from A to the final destination, G (figure 16.16). There are two possibilities, go through B or go through D. The distance from A to B is 50 and the remaining distance from B to G is 130. Thus, the total distance through B would be 180. On the other hand, the distance from A to D is 70 and the remaining distance from D to G is 120. Thus, the total distance through D would be 190. Since 180 is smaller than 190, we choose node B.

Figure 16.15:

A* Modifies the Dijkstra Algorithm

Adding an Estimate of the Remaining Distance to the Dijkstra distance

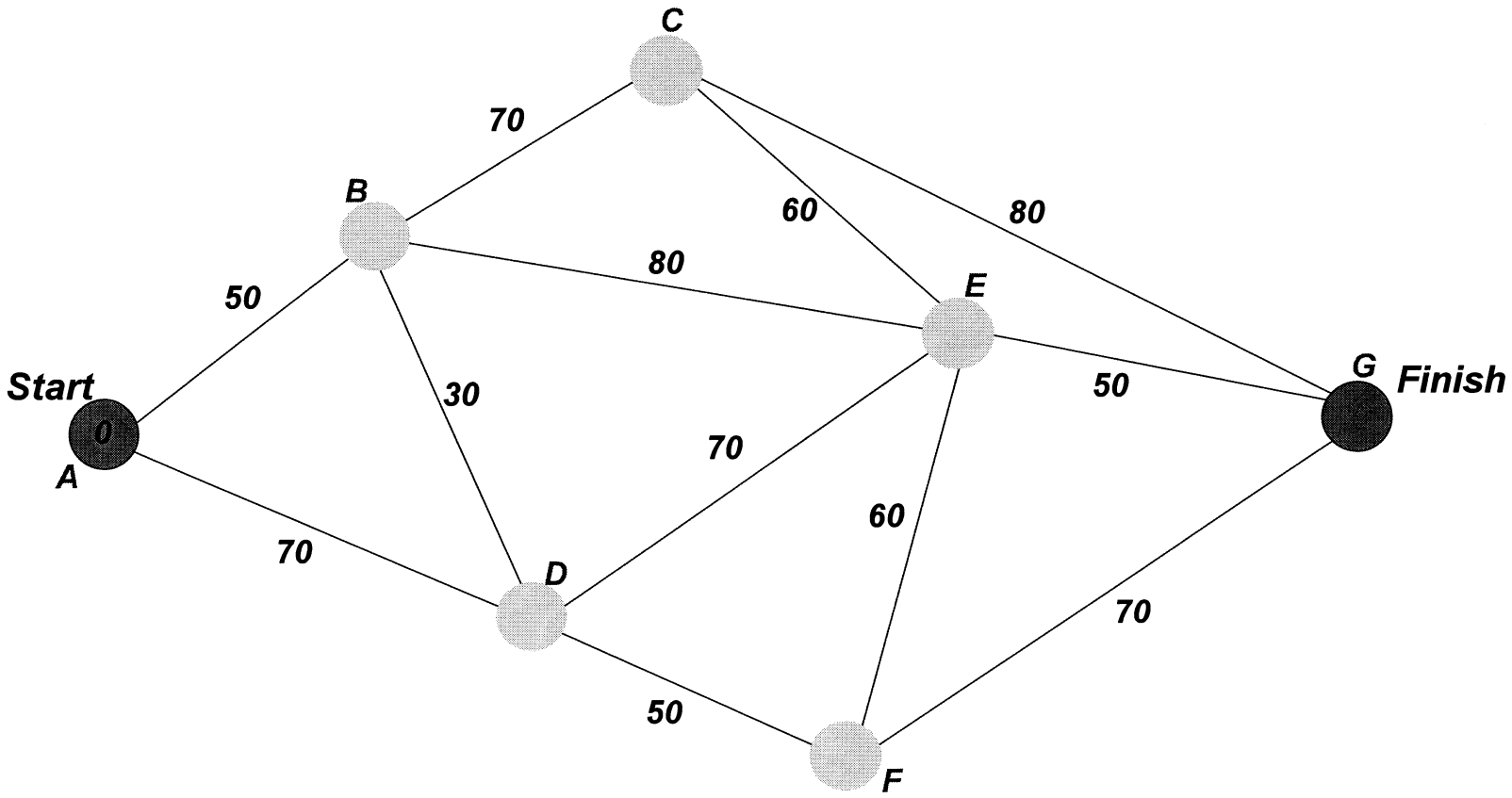
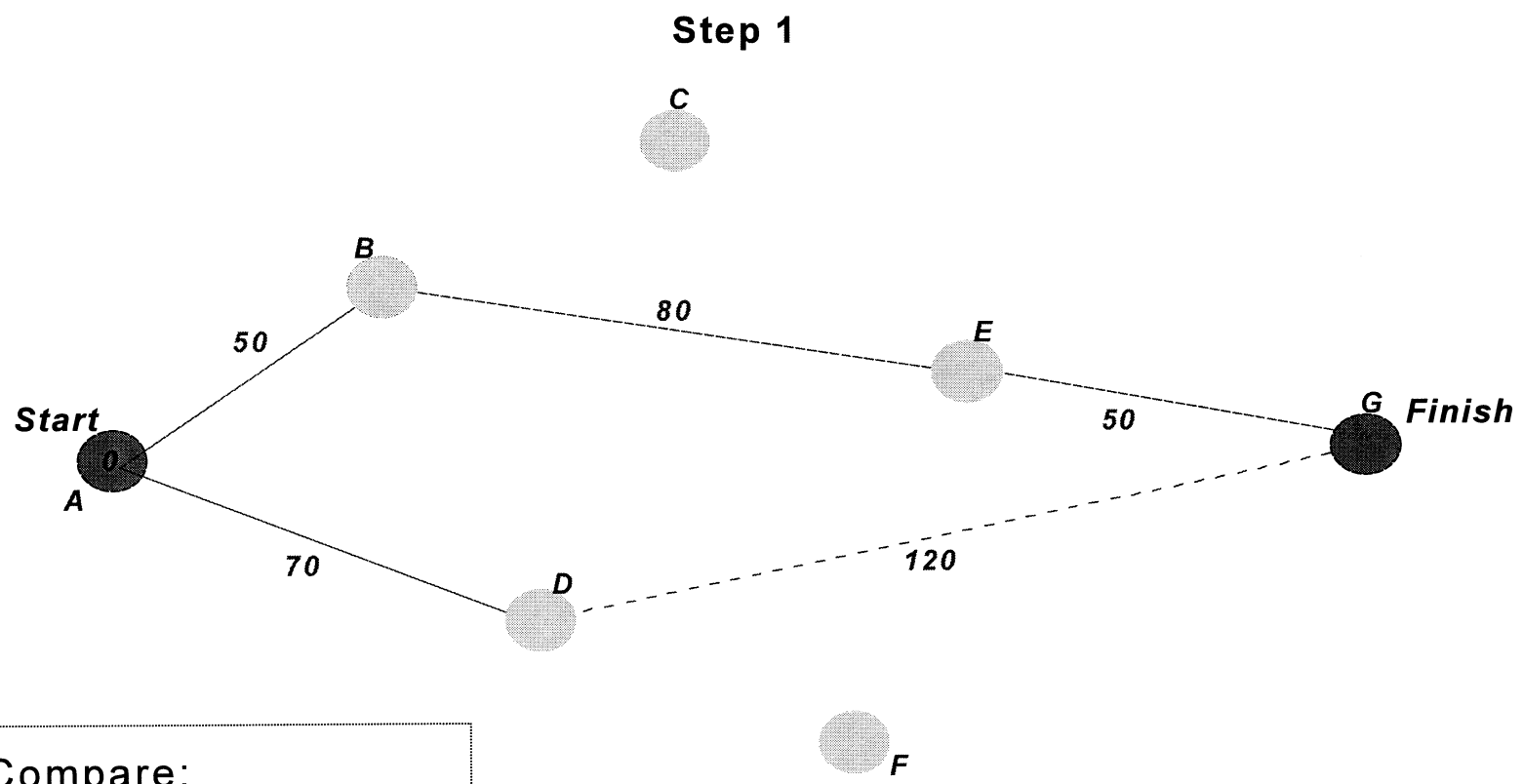


Figure 16.16:

A* Algorithm



In step 2 (figure 16.17), three possibilities are explored for reaching G from A - through B and E; through B and C; and through D. The total distance through B and E is 180 (50 + 80 + 50) while the total distance through B and C is 200 (50 + 70 + 80) and through D is 190 (70 + 120). Thus, the routine chooses through B and E.

In step 3 (figure 16.18), it is but a short path from E to the final destination G. The total distance through B and E to G is 180 while the total distance through B and C is 200 and through D is 190. Thus, the A* algorithm has determined a shortest path in three steps, rather than the 6 it took the Dijkstra algorithm (figure 16.19).

In general, if V is the number of nodes in the network, the Dijkstra algorithm requires V^2 searches whereas the A* algorithm requires only V searches (Sedgewick, 2002). As can be seen, this is much more efficient than having to search every single possible node, which is what Dijkstra requires.

Applying A* to multiple origins

As with the Dijkstra algorithm, A* can be applied to multiple origins. It does it one origin-destination combination at a time. If an origin-destination matrix is represented by the origins as rows and the destinations as columns, then the A* algorithm takes each origin-destination combination and finds the shortest path. Since it does not search all possible nodes (only those in which the total distance to the destination is shortest), it cannot determine in one step the distance from an origin to all possible destinations. However, it is so quick as an algorithm that it can be applied to each cell of the origin-destination matrix and still come out much faster than a Dijkstra search.

Weighting of Segments

As mentioned above, the units of the network can be any type of impedance - distance, travel time, or cost. These can be thought of as weights applied to a segment. The A* algorithm does not really care what are the units of the segments as long as they are consistent and proportional to cost. The algorithm will determine the path with the shortest total cost (or total weight).

Thus, this algorithm can be applied to a trip distribution or mode split matrix of origin-destination pairs. It will determine the shortest path from each origin zone to each destination zone and can do this in the measurement units that are selected for weighting.

The advantages for travel demand modeling are enormous. It means that if the weighting variable is travel time, then the algorithm will find the shortest time path through the network for each origin-destination pair. If the weighting variable is generalized cost, then the algorithm will find the

Figure 16.17:

A* Algorithm

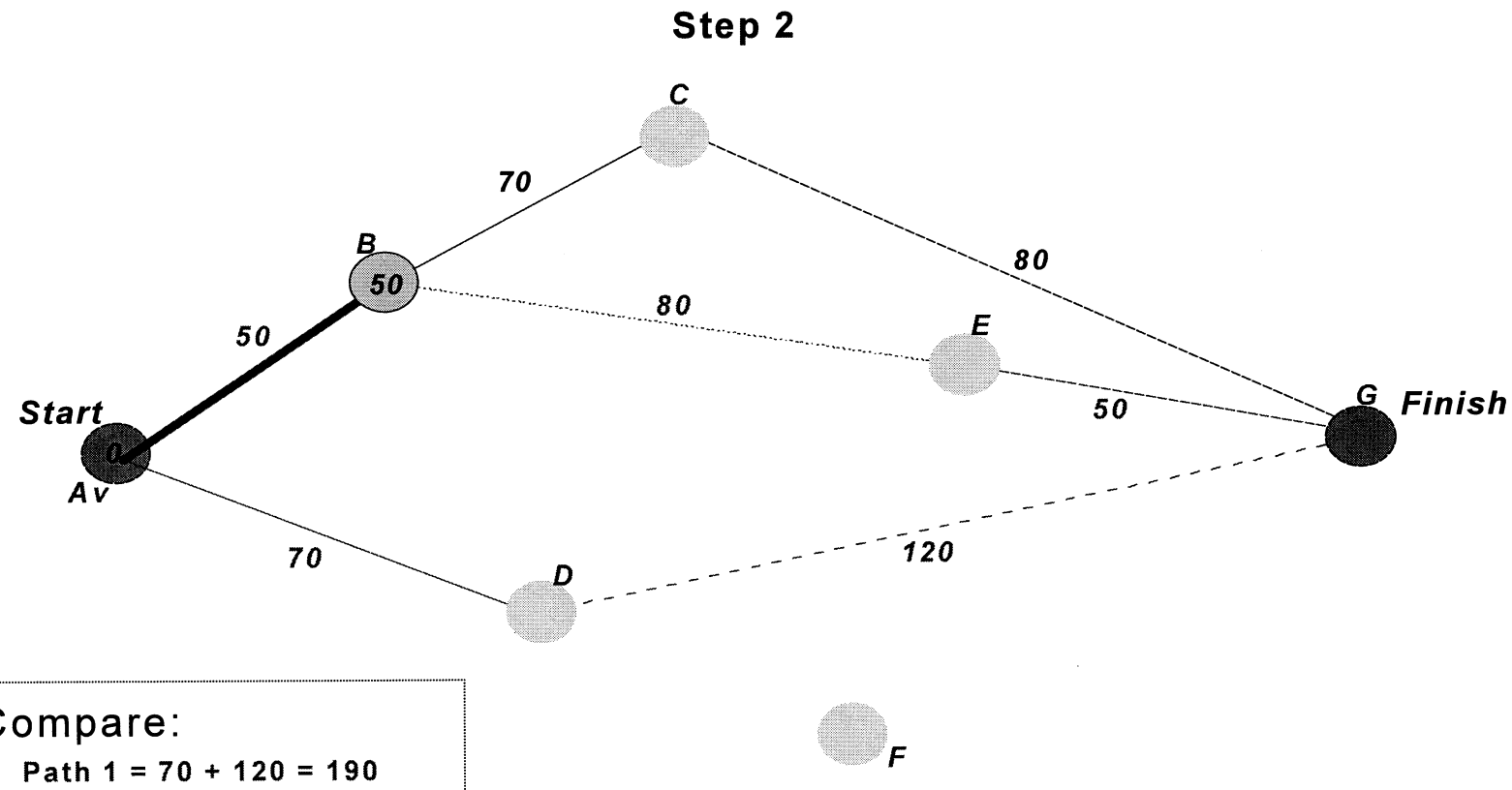
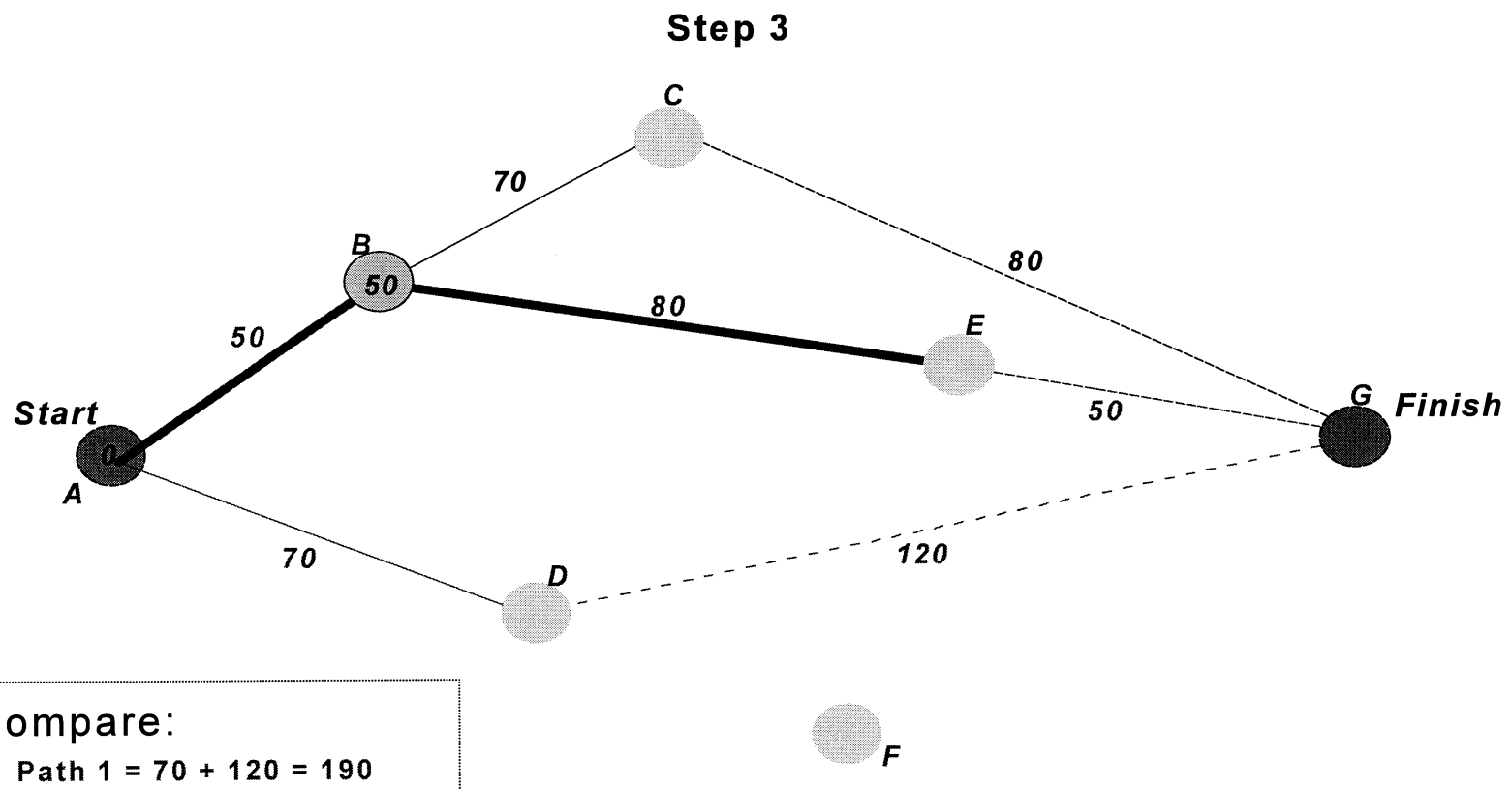


Figure 16.18:

A* Algorithm



Compare:

Path 1 = 70 + 120 = 190

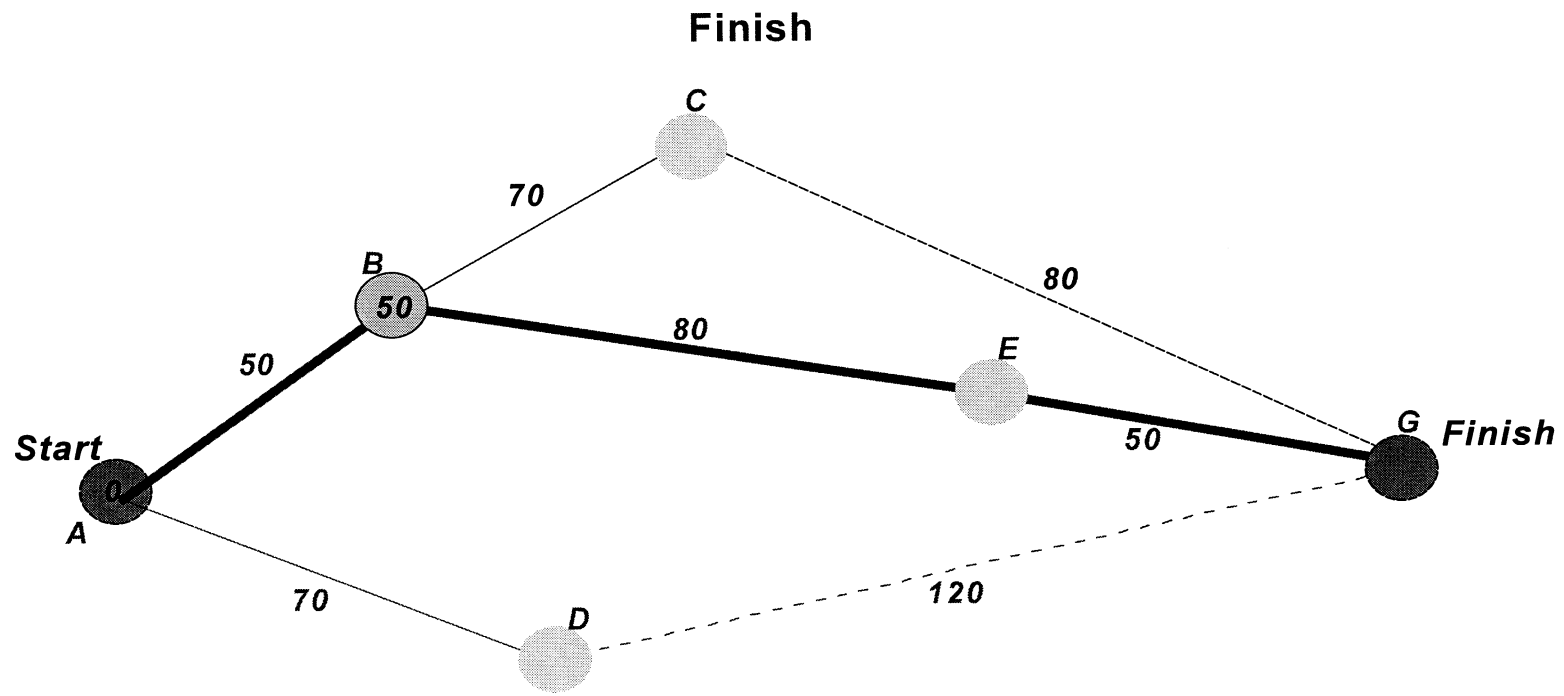
Path 2 = 50 + 80+ 50 = 180

Path 3 = 50 + 70 + 80 = 200

Choose path 2

Figure 16.19:

A* Algorithm



Compare:

A* solved in 3 steps

Dijkstra solved in 6 steps

shortest cost path through the network. Finally, if the weighting variable is speed, then this must be converted into an impedance weight by dividing the distance of the segment by the speed to yield travel time. In short, the A* algorithm is an amazing one and allows the building of a routing algorithm.[2]

Routing Algorithms

In applying a shortest path analysis to a network assignment, several assumptions have to be made. As mentioned earlier, network assignment involves assigning trips to particular routes. Given a network of segments (e.g., road segments, train segments), a *routing algorithm* allocates the predicted number of trips to one or more routes. In other words, the network assignment is done through a routing algorithm. What makes this complex is that there are a number of different routing algorithms, of which a shortest path is only one. Most of them are based on the assumption of travel cost relative to network capacity (Ortuzar and Willumsen, 2001).

The simplest type of routing algorithm is an *All or None* assignment. For each origin-destination pair (either for all trips or trips by specific travel mode), the algorithm calculates the shortest path through the network and assigns *all* trips to that path. This is the most rational model in that the cost of travel (whether measured by distance, travel time, or some cost measure) is minimized.

A second routing algorithm is a *stochastic path* in which each route has a certain probability of being selected. Multiple paths can be selected, but with a probability inversely proportional to their cost. The shortest path will be selected most often; the second shortest path next most often; the third shortest path third most often; and so forth. This type of algorithm attempts to capture the variability in travel behavior that can come from traveler's perceptions or incomplete information about the choice of path.

A third routing algorithm is a *congested assignment* in which there is feedback from the capacity of the network to the choice of route. In the classic case, as travel volumes increase on network segments, the capacity of the segment to absorb traffic is approached. The higher the ratio of the volume-to-capacity (V/C), the slower traffic becomes on the segment. In other words, the cost of travel increases. Eventually, if the volume keeps increasing, the speed slows so much as to eventually reduce the amount of traffic that can enter the segment (ITE, 2000). In theory, if there is so much traffic volume relative to the capacity, traffic comes to a complete halt (gridlock). However, in practice this doesn't happen as drivers take other routes. Consequently, with high V/C ratios, other routes become more desirable and some traffic spills over on to those segments. This type of model is frequently used in metropolitan travel demand models for transportation since congestion is a major factor in most urban areas.

There are advantages and disadvantages to each of these approaches. The "All or none" assignment is the closest to a rational choice model; the route with the lowest total cost is chosen. On the other hand, this algorithm will continue to assign trips to a route even if the road segment becomes extremely congested, which is not realistic. A stochastic model has the advantage of accounting for variability. If individual-level data could be obtained that measured individual choices and perceptions of routes, then it's possible that a realistic proportional split among routes could be detected. More often, however, such information is lacking and a variation on the mode split model is used to proportion the trips among the different possible routes (see Ortuzar and Willumsen, 2001, chapter 10 for more information).

The "Congested assignment" algorithm can be seen as a more realistic variation on the "All or none" in that the costs of travel change as the network capacity is reached. Most transportation models use that type of model because it is a more realistic representation.

Lack of information about crime trips

The problem with crime trips, however, is that the number of trips is liable to represent only a very small proportion of the total trips on any segment of a network. Thus, there is not liable to be any feedback from the capacity limits of segments to crime trips per se. Any feedback is liable to apply to all trips, of which the crime trips are a sub-set. It might be possible to link a crime trip route choice algorithm to a general congested assignment in order to approximate this situation, but the amount of information that would be necessary to be obtained and the complexity of the modeling algorithm would probably not produce much tangible benefits beyond what a simple model predicts.

Further, there could be feedback from surveillance and other policing practices that might increase the cost to an offender of traveling along a particular route. However, without any detailed information about perceived costs of particular routes, it is difficult to postulate any type of model for choosing alternatives. This would be a very valuable area of research in understanding the travel behavior of offenders.

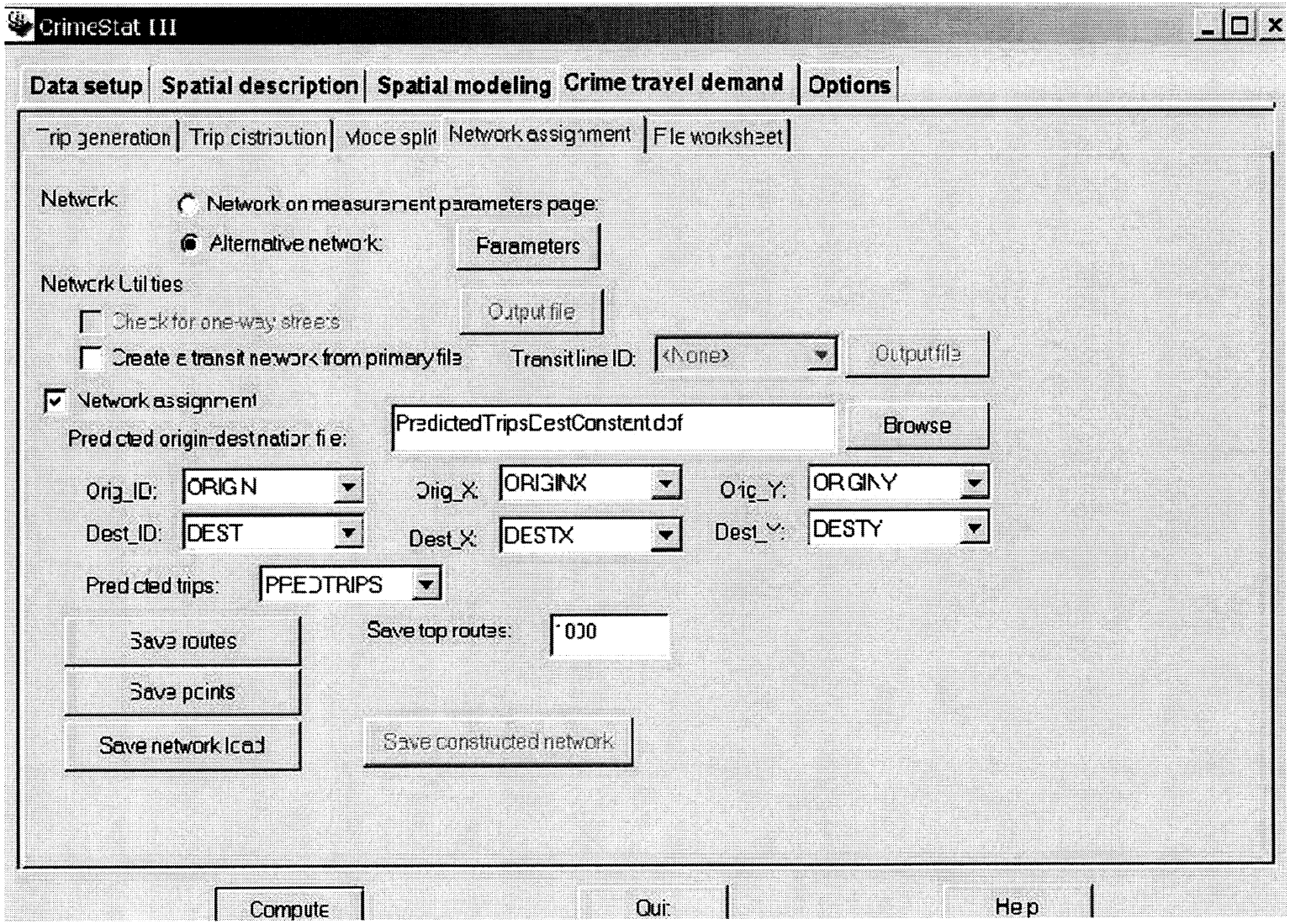
But, short of that information, an "All or none" assignment routine is probably the easiest to implement for allocating the predicted crime trips to routes.

The CrimeStat Network Assignment Routine

The CrimeStat network assignment routine implements an "All or none" assignment based on the A* shortest path algorithm. Figure 16.20 shows the setup page for network assignment. On the page, there is a network

Figure 16.20:

Network Assignment Module



assignment routine and there are some network utilities. Let's start with the network assignment routine.

Network Used

The first input that needs to be made is which network is to be used. The choices are the network specified on the Measurement parameters page (the default) or an alternative network.

Network on measurement parameters page

Check the 'Network on Measurement parameters page' box to use that network. All the parameters will have been defined for that setup (see Measurement parameters page).

Alternative network

If an alternative network is to be used, it must be defined. Check the 'Alternative network' box and click on the 'Parameters' button. Figure 16.21 shows the dialogue box for the alternative network.

> Note: if a network is also used on the Measurement Parameters page, then it must be defined there as well. CrimeStat will check whether that file exists; if it does not, the routine will stop and an error message will be issued. Therefore, if an alternative network is used, the user should probably change the distance measurement on the Measurement Parameters page to direct or indirect distance.

Type of network

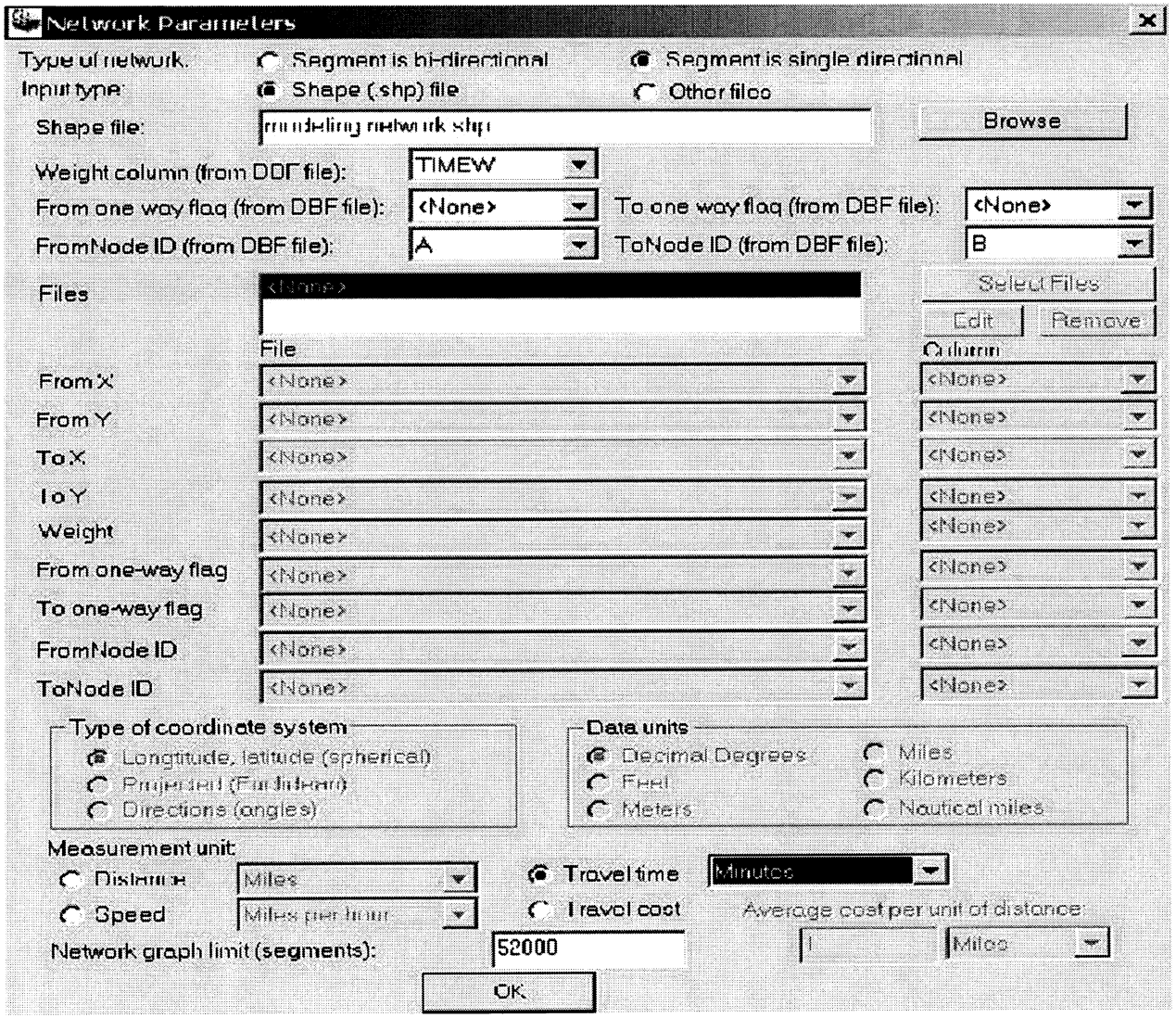
Network files can be bi-directional (e.g., a TIGER file) or single directional (e.g., a transportation modeling file). In a bi-directional file, travel can be in either direction. In a single directional file, travel is only in one direction. Specify the type of network to be used.

Input file

The network file can either be a shape file (line, polyline, or polylineZ file) or another file, either dBase IV 'dbf', Microsoft Access 'mdb', Ascii 'dat', or an ODBC-compliant file. The default is a shape file. If the file is a shape file, the routine will know the locations of the nodes. For a dBase IV or other file, the X and Y coordinate variables of the end nodes must be defined. These are called the "From" node and the "End" node. An optional weight variable is allowed for both a shape or dbf file. The routine identifies nodes and segments and finds the shortest path. If there are one-way streets in a bi-directional file, the flag fields for the "From" and "To" nodes should be defined.

Figure 16.21:

Alternative Network Dialogue



Weight field

By default, each segment in the network is not weighted. In this case, the routine calculates the shortest distance between two points using the distance of each segment. However, each segment can be weighted by travel time, speed or travel costs. If travel time is used for weighting the segment, the routine calculates the shortest time for any route between two points. If speed is used for weighting the segment, the routine converts this into travel time by dividing the distance by the speed. Finally, if travel cost is used for weighting the segment, the routine calculates the route with the smallest total travel cost. Specify the weighting field to be used and be sure to indicate the measurement units (distance, speed, travel time, or travel cost) at the bottom of the page. If there is no weighting field assigned, then the routine will calculate using distance.

From one-way flag and To one-way flag

One-way segments can be identified in a bi-directional file by a 'flag' field (it is not necessary in a single directional file). The 'flag' is a field for the end nodes of the segment with values of '0' and '1'. A '0' indicates that travel can pass through that node in either direction whereas a '1' indicates that travel can only pass from the other node of the same segment (i.e., travel cannot occur from another segment that is connected to the node). The default assumption is for travel to be allowed through each node (i.e., there is a '0' assumed for each node). There is a 'From one-way flag' field and a 'To one-way flag' field. For each one-way street, specify the flags for each end node. A '0' allows travel from any connecting segments whereas a '1' only allows travel from the other node of the same segment. Flag fields that are blank are assumed to allow travel to pass in either direction.

FromNode ID, ToNode ID

If the network is single directional, there are individual segments for each direction. Typically, two-way streets have two segments, one for each direction. On the other hand, one-way streets have only one segment. The FromNode ID and the ToNode ID identify from which end of the segment travel should occur. If no FromNode ID and ToNode ID is defined, the routine will chose the first segment of a pair that it finds, whether travel is in the right or wrong direction. To identify correctly travel direction, define the FromNode and ToNode ID fields.

Type of coordinate system

The type of coordinate system for the network file is the same as for the primary file.

Measurement unit

By default, the shortest path is in terms of distance. However, each segment can be weighted by travel time, travel speed, or travel cost.

1. For travel time, the units are minutes, hours, or unspecified cost units. For speed, the units are miles per hour and kilometers per hour. In the case of speed as a weighting variable, it is automatically converted into travel time by dividing the distance of the segment by the speed, keeping units constant.

2. For travel cost, the units are undefined and the routine identifies routes by those with the smallest total cost.

Network Utilities

There are two network utilities that can be used.

Check for one-way streets

First, there is a routine that will identify one-way streets if the network is single directional. In a single directional file, one-way streets do not have a reciprocal pair (i.e., a segment traveling in the opposite direction). This is indicated by a reciprocal pair of ID's for the "From" and "To" nodes. If checked, the routine identifies those segments that do not have reciprocal node ID's. The network is saved with a new field called "Oneway". One-way segments are assigned a value of '1' value and two-way segments are assigned a value of '0'. The output is saved as an ArcView '.shp', MapInfo '.mif' or Atlas*GIS '.bna' file.

Create a transit network from primary file

Second, there is a routine that will create a transit network from the primary file. This is useful for creating a transit network from a collection of bus stops (bus network) or rail stations (rail network). If checked, the routine will read the primary file and will draw lines from one point to another in the order in which the points appear in the primary file. Note, it is essential to order the points in the same order in which the network should be drawn (otherwise, an illogical network will be obtained). It is easy to do this in a spreadsheet program.

Transit Line ID

The routine can handle multiple lines, for example different rail lines or bus routes (e.g., Line A, Line B, Route 1, Route 2). In the primary file, the points must be grouped by lines, however, and must be classified by a Transit Line ID field. Within each group, the points must be arranged in order of

occurrence; the routine will draw lines from one point to another in that order. In the Transit Line ID field, indicate which variable is the classification variable. The output is saved as an ArcView '.shp', MapInfo '.mif' or Atlas*GIS '.bna' file.

Figure 16.5 above shows the effect of creating four separate rail lines from the station locations while figure 16.6 shows the merged four lines implemented with the Group ID.

Network Output

There are three types of output for the network assignment routine. First, the most frequent inter-zonal (i.e., trips between different zones) routes can be output as polylines. Second, the most frequent intra-zonal (i.e., trips within the same zone) routines can be output as points. Third, the entire network can be output in terms of the total number of trips that occur on each segment (network load).

Save routes

The shortest routes can be saved as separate polyline objects for use in a GIS. Specify the output file format (ArcView '.shp', MapInfo '.mif' or Atlas*GIS '.bna') and the file name.

Save top routes

Because the output file is very large (number of origin zones x number of destination zones), the user can select a zone-to-zone route with the most predicted trips. The default is the top 100 origin-destination combinations. Each output object is a line from the origin zone to the destination zone with a Route prefix. The prefix is placed before the output file name. The graphical output includes:

1. An ID number from 1 to K, where K is the number of links output (ID)
2. The feature prefix (ROUTE)
3. The origin zone (ORIGIN)
4. The destination zone (DEST)
5. The X coordinate for the origin zone (ORIGINX)
6. The Y coordinate for the origin zone (ORIGINY)
7. The X coordinate for the destination zone (DESTX)
8. The Y coordinate for the destination zone (DESTY)
9. The number of trips on that particular route (FREQ)
10. The distance between the origin zone and the destination zone (DIST).

Figure 16.22 shows the top 300 routes calculated with the modeling network. The assignment was weighted by travel time and the thickness and color of the line is proportional to the number of predicted trips.

To see how this differs from the trip distribution matrix, figure 16.23 zooms into a high volume route in eastern Baltimore County. The modeling streets are displayed as are the predicted links from the trip distribution for that area. As seen, the trip distribution simply produces straight-line links between origins and destinations. In this case, the crime trips come into to the centroid of the Traffic Analysis Zone (TAZ) in the middle of this hot spot of crimes (TAZ 610). The actual routes, on the other hand, follow the streets (in this case, the modeling network) and are more circuitous. Several of the streets are used much more heavily than others, according to the assignment.

An additional point should be noted, however. Since the modeling network was used rather than the TIGER network, the trips into and from the centroid of the TAZ do not follow any particular road; the algorithm simply draws a straight line from the centroid to the nearest road segment. In subsequent modeling, it might be worthwhile to digitize additional streets in this neighborhood since there are many crimes being attracted to it. A crime mapping analyst can easily add the additional features to improve resolution. The model would have to re-run, however, to get a more accurate display.

Save points

Intra-zonal trips (trips in which the origin and destination are the same zone) can be output as separate point objects as an ArcView '.shp', MapInfo '.mif' or Atlas*GIS '.bna' file. Again, the top K points are output (default=100). Each output object is a point representing an intra-zonal trip with a RoutePoints. The prefix is placed before the output file name.

The graphical output for each includes:

1. An ID number from 1 to K, where K is the number of links output (ID)
2. The feature prefix (ROUTEPoints)
3. The origin zone (ORIGIN)
4. The destination zone (DEST)
5. The X coordinate for the origin zone (ORIGINX)
6. The Y coordinate for the origin zone (ORIGINY)
7. The X coordinate for the destination zone (DESTX)
8. The Y coordinate for the destination zone (DESTY)
9. The number of trips on that particular route (FREQ)

These are not illustrated in this chapter because they are identical to the intra-zonal output of the trip distribution module (see chapter 14).

Figure 16.22:

Predicted Baltimore County Crime Trips: 1993-1997 Routes and Links for Zone-to-Zone Trips: All Crimes Weighted by Travel Time

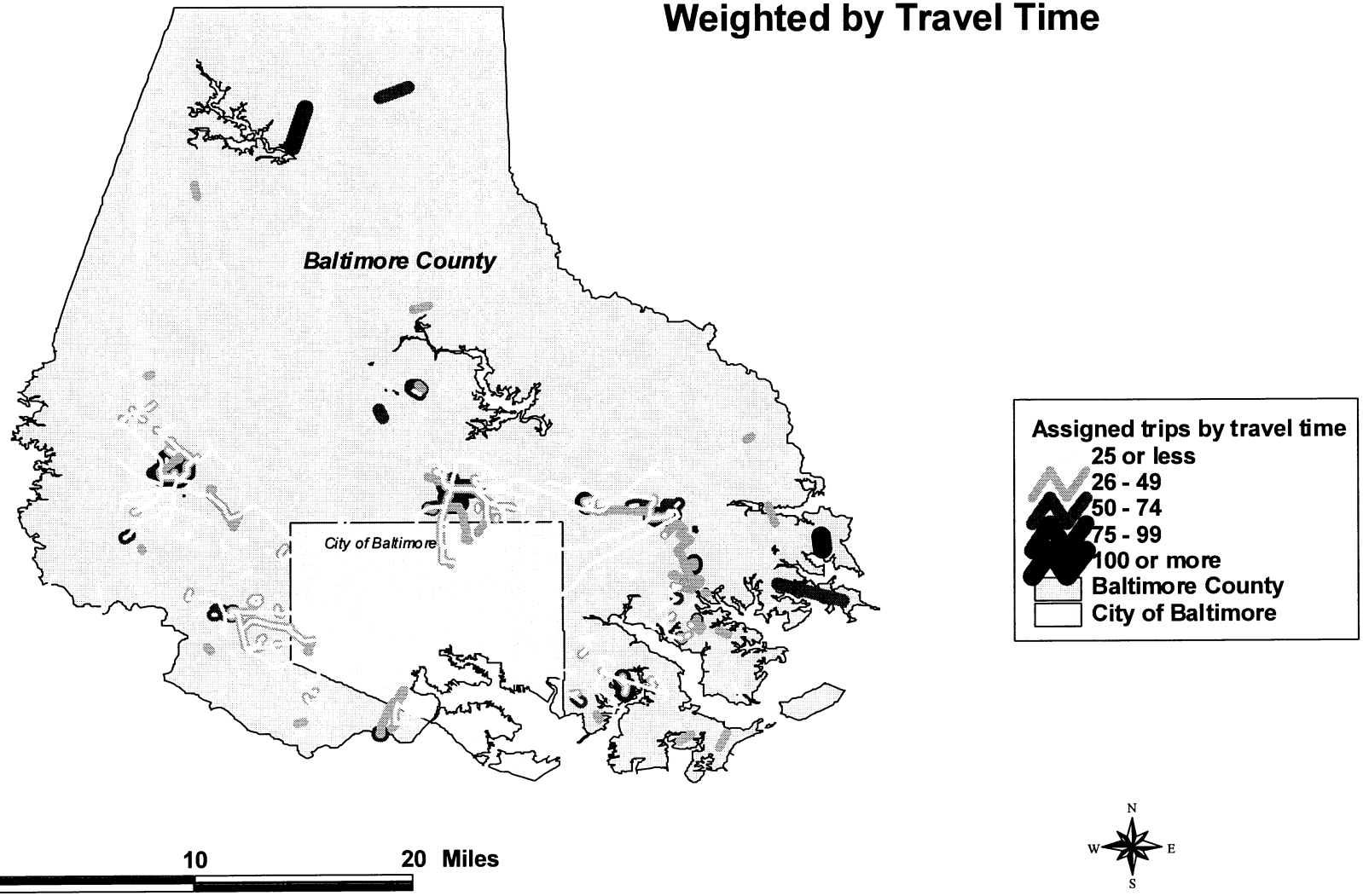
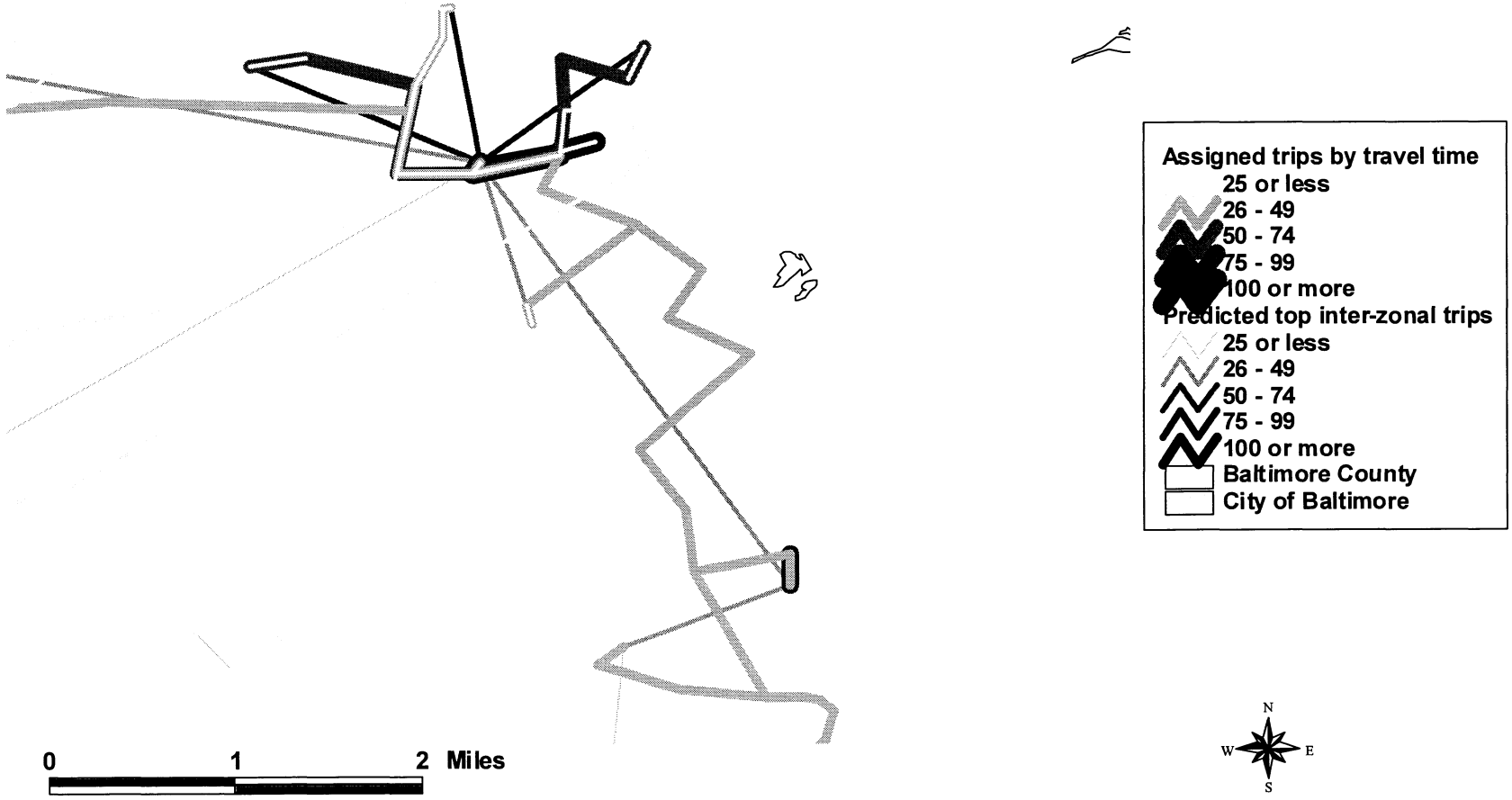


Figure 16.23:

Predicted Baltimore County Crime Trips: 1993-1997 Routes and Links for Zone-to-Zone Trips: All Crimes Weighted by Travel Time



Save network load

It is also possible to save the total network load as an ArcView '.shp', MapInfo '.mif' or Atlas*GIS '.bna' file. This is the total number of trips on each segment of the network. The routine takes every origin zone to destination zone combination and sums the number of trips that occur on each segment of the network. Click on the "Save output network" box and specify a file name for the output.

Figure 16.24 shows the entire crime trip volume on the network (network load). The assignment was weighted by travel time. Notice how there are many trips on the circular Baltimore Beltway (I-695). Because the road is a freeway, travel is generally much faster than on most arterial roads. Consequently, there are many crime trips being assigned to the freeway even though it is longer than many direct links.

To see how this differs from a shortest distance assignment, the routine was re-run using only distance as the weighting variable. Figure 16.25 displays the results. As seen, the routine does not use the Beltway very much, but instead uses the arterial roads more, particularly the diagonal arterial roads coming out of the City of Baltimore. Since the routine was determining the shortest path on the basis of distance only, it will inevitably find the most direct routes in terms of distance. In terms of travel time, however, many of those routes will be much slower because of traffic lights, cross-traffic, drivers pulling in and out of parking spaces, and so forth. Thus, the freeway is almost always quicker for travel than an arterial road except at peak rush hour conditions. This points out the importance of using travel time and, better yet, travel cost as an impedance variable. Distance is much too simple an indicator of it.

The network load routine can even be used for specific travel modes (and usually is for transportation travel demand modeling). Figure 16.26, for example, shows the network volumes (load) of bus crime trips, again weighted by travel time. According to the model, many of these trips originate in the City of Baltimore. But at the high crime locations, multiple bus routes tend to converge producing a high bus trip volume on the adjacent streets. Because of the very small number of bus crime trips predicted by the mode split model, the volumes are not high, even for the highest volume links. Also, notice how the Beltway is not used very much for bus trips, compared to the total network load in figure 16.24. The reason is that most bus routes do not use the freeway but stay on arterial roads (express buses would be an exception, but those tend to be used primarily for commuting).

Figure 16.27 shows the network volumes of train trips. Since there was no data on travel times along each train segment, the volumes are weighted only by distance. The number of trips, of course, are very few, as was noted in chapter 15. Also, notice how most of the crime trips taken by train occur on

Figure 16.24:

Crime Volume by Road Segment Weighted by Travel Time

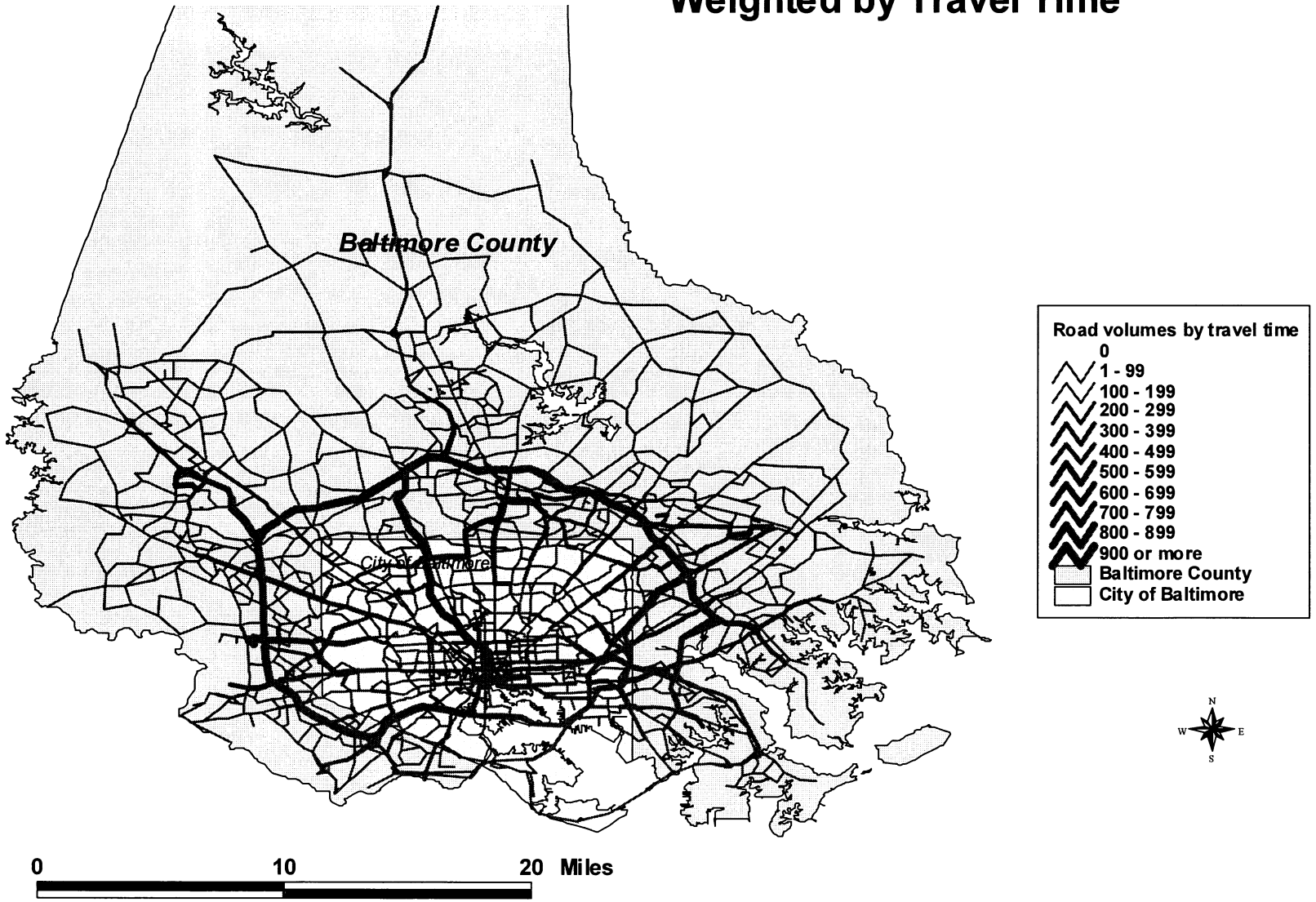


Figure 16.25:

Crime Volume by Road Segment Weighted by Distance

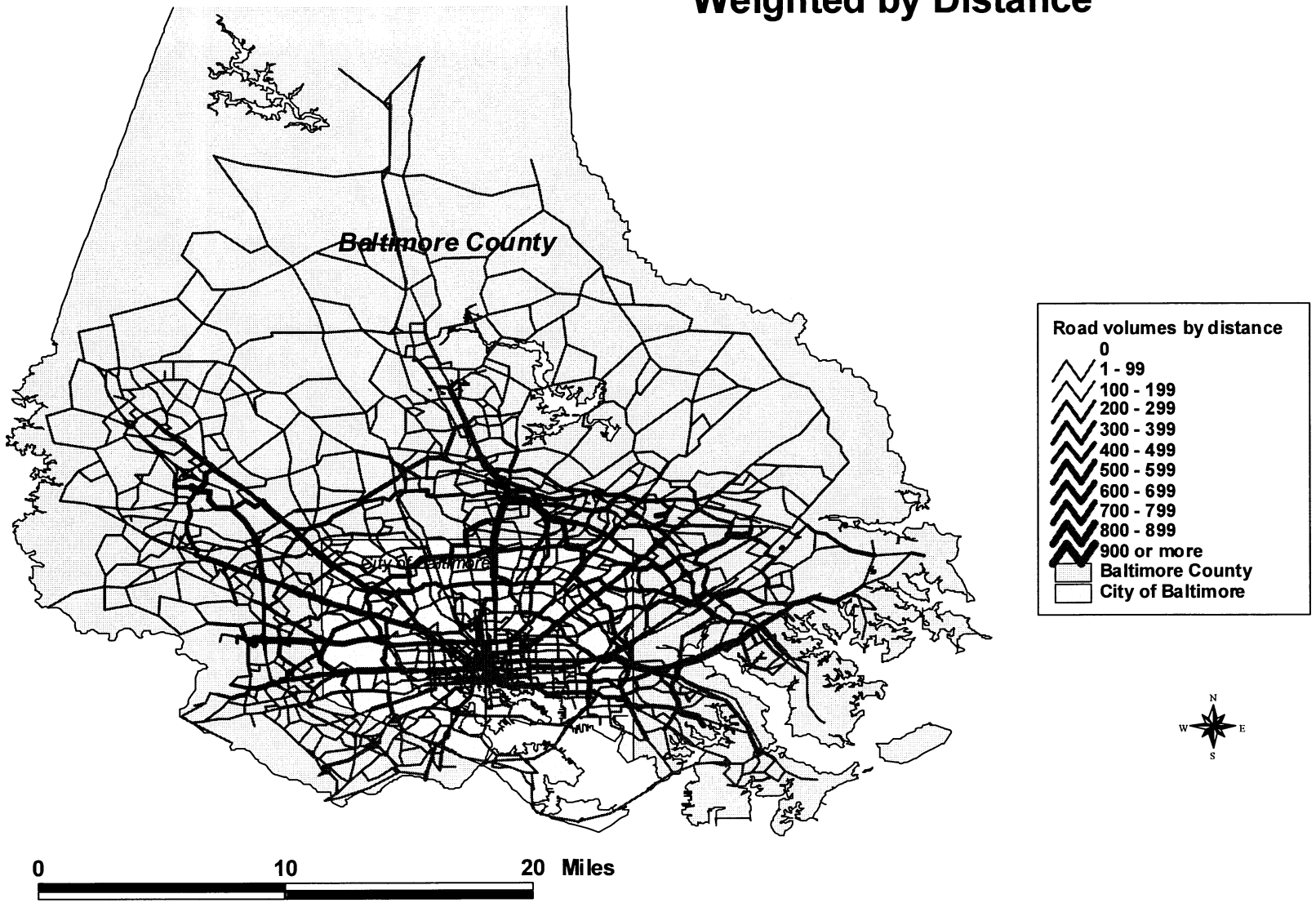


Figure 16.26:

Crime Volume by Bus Route Segment Weighted by Travel Time

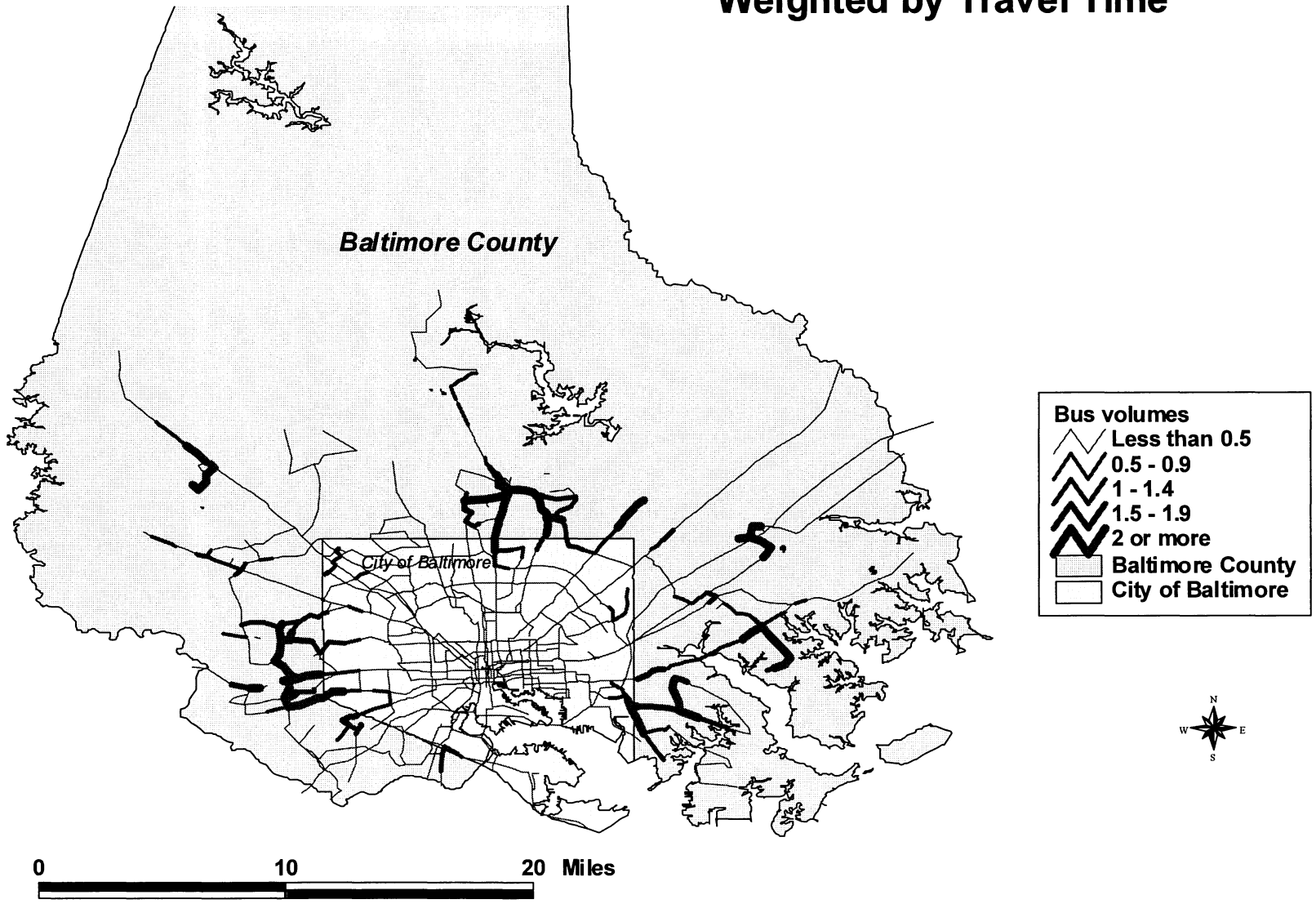
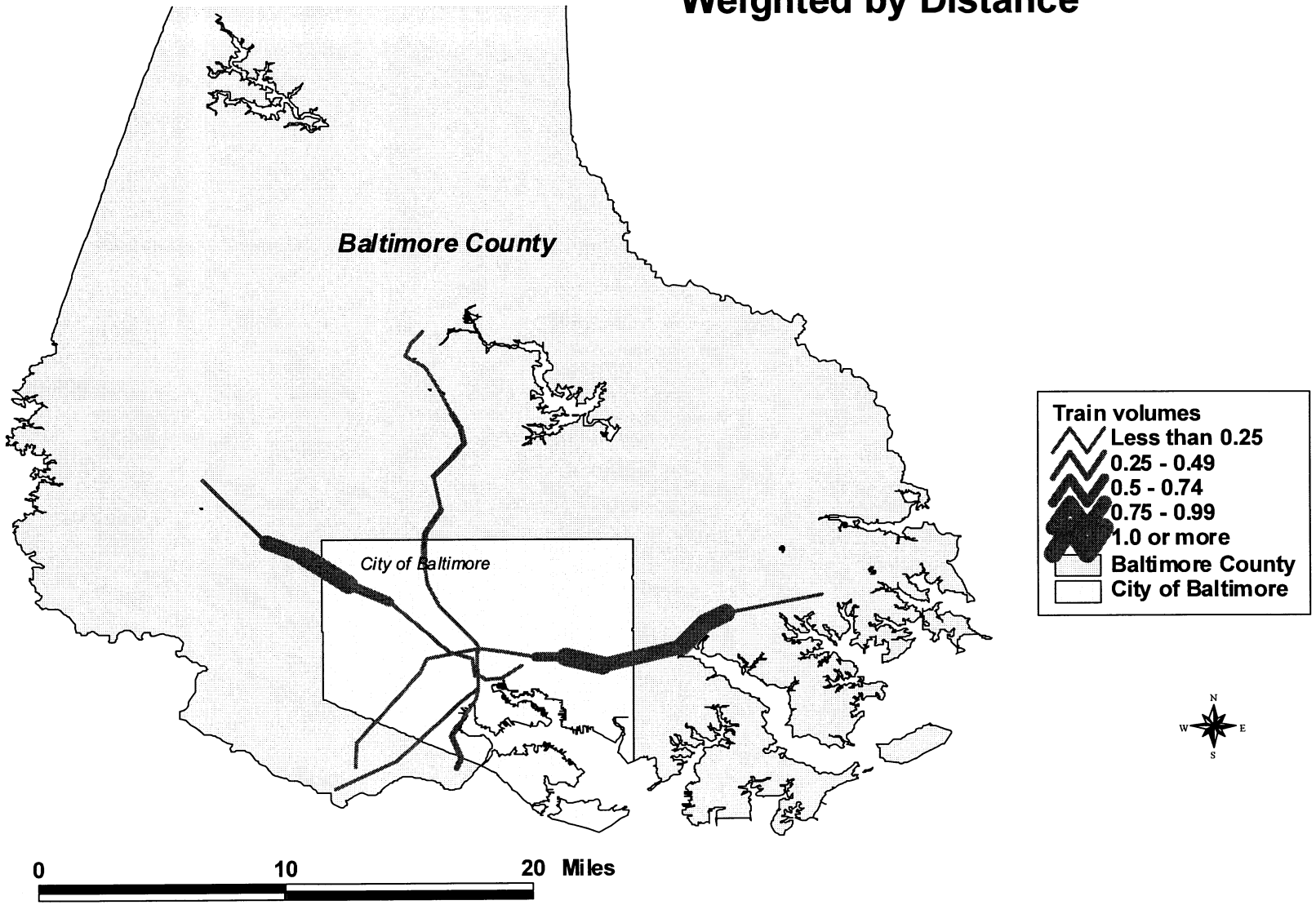


Figure 16.27:

Crime Volume by Rail Segment Weighted by Distance



two lines, the Metro line to the west and the MarcP line to the east. In both cases, the train trips start in the City of Baltimore and travel to Baltimore County. These, of course, are predictions of crime travel volumes on the rail network, not empirical verifications.

Crime Types

The network assignment routine can be applied to specific crime types. In general, it is a good idea to calibrate a general assignment for all crimes before analyzing specific crimes. The reason is that there are volume dimensions that assign most crime trips to the same segments. Still, some differences can be observed. Figure 16.28 shows the likely routes for vehicle thefts (in blue) and compares it to the likely routes for all crimes (in red). There are similarities and differences. There is overlap in the predicted routes in the southeast and southwest edges of the County with the City of Baltimore, and there is some overlap at the northwest border with the City of Baltimore. At the same time, though, some differences are visible, particularly at the western border with the City of Baltimore.

In other words, the network assignment model shows different routes for vehicle thefts than for crimes in general. This difference, of course, represents differences in the trip distribution matrix of the vehicle thefts compared to all crimes.[3]

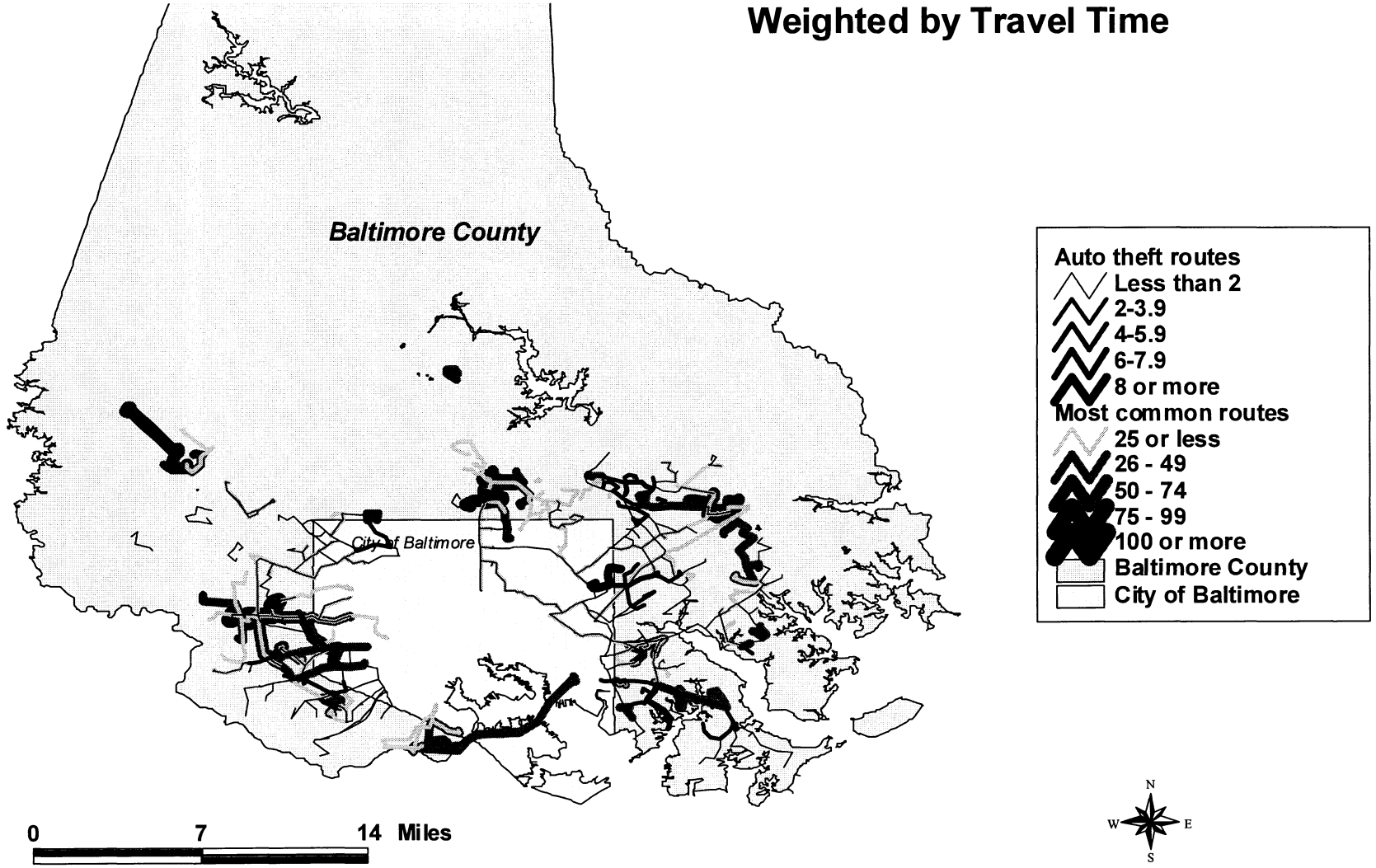
Uses of Network Assignment

A network assignment routine is the culmination of the crime travel demand modeling process. Essentially, it assigns predicted trips (whether for entire origin-destination trip pairs or for mode-specific trip pairs) to an actual network and usually on the basis of least cost. The algorithm used in the CrimeStat network assignment routine calculated the shortest path (in terms of distance, travel time, or cost) and assigned all the trips for each origin-destination pair to this route. The representation is more complex than a simple trip link (which is a straight line) since it uses information on the actual network used. The result is a prediction of routes that are taken to commit crimes and a prediction of the total crime trip volume on each network segment. This is clearly an advance on the geographic profiling/journey-to-crime approach, which has simply analyzed travel distance as an explanatory variable.

Network assignment also has many uses for police. First, it can point out where police need to focus their deployment. In this sense, the progression of the four modeling stages represents adding information to the knowledge of the crime events. Simply mapping the crime events tells a police department where the crimes are occurring. Analyzing the trip distribution tells the department from where the crimes might be originating.

Figure 16.28:

Predicted Routes by Crime Type: 1993-1997
All Crimes and Vehicle Thefts
Weighted by Travel Time



Splitting the distribution by travel model provides information about the likely travel mode used. Finally, assigning the predicted trips to actual routes gives information about how offenders may have traveled to the crime location. The model provides a lot more information than a simple description of a high crime area.

Second, knowing the likely routes of offenders can allow for increased surveillance' and target hardening. Not only can police patrol the likely routes in a more focused manner, but other surveillance tools can be used, too. For example, surveillance cameras that monitor traffic can be used for a variety of purposes. In the U.S., they have tended to be used for monitoring traffic signals for red-light running (IIHS, 2004). However, in Europe they are widely used for a variety of traffic monitoring purposes - speed enforcement, bus lane enforcement, entering the London congestion zone, as well as monitoring traffic signals. In London, for example, the entire monitoring process is automated. For a vehicle making a violation, the camera takes a picture and a software package identifies the license plate. The license number is then matched against a database of vehicles and a traffic citation is sent to the owner. There is no reason why this type of technology could not be structured to also look for stolen vehicles or vehicles belonging to individuals for which outstanding citations have been issued. In short, knowing on which roads high crime trips volumes are likely to occur can help police focus a range of surveillance tools on those locations.

Conclusions

In short, network assignment is a logical step in the modeling of crime trips and one that brings the trips down to actual routes that are used. It is a more realistic representation of travel behavior and one that can allow focused deployment by police.

In the next chapter, two case studies are examined. Dick Block and Dan Helms apply the crime travel demand theory to Chicago and Las Vegas respectively.

Endnotes for Chapter 16

1. Speed could be used, but it is inversely proportional to impedance (i.e., the higher the speed, the less the impedance). Most shortest path algorithms treat the weight as proportional. However, speed can be converted into travel time by dividing distance by speed. To use the example, if the length is 1 mile long and the speed is 50 miles per hour, then the travel time is 1/50 hours (or 1.2 minutes).

2. For larger databases greater than, say, 1 million records, however, A* is too slow. An algorithm that is appropriate for very large databases can be found in Shekhar and Chawla (2003).

3. The differences could be due to the mode split routine as well as the trip distribution matrix. However, in the case of vehicle thefts, the travel mode is not very relevant since the return trip is always by vehicle - the stolen vehicle.

Chapter 17
Case Studies in Crime Travel Demand Modeling

In this chapter, Richard Block and Daniel Helms present case studies in crime travel demand modeling for Chicago and Las Vegas respectively.

I. Travel Patterns of Chicago Robbery Offenders

Richard Block
Loyola University
Chicago

Some neighborhoods are dangerous others are safe. Crime clusters in specific areas. So too do criminals. Criminologists, police, and civilians have known this for nearly 150 years. However, relatively little research has been done on the travel patterns of offenders. Using a modification of standard transportation models, *CrimeStat III* allows police and researchers to describe and predict travel patterns based on four sequential models.

The object of research presented here is to test the usefulness and feasibility of CrimeStat's Crime Travel Demand model utilizing police reports of all robberies occurring in Chicago in 1997 and 1998 that had at least one known offender who lived in the city. In sum, the objectives of this study of robbery in Chicago were:

1. To test the *CrimeStat III* crime travel demand model in a mature central city.

2. To describe the travel patterns of robbery offenders based upon offenders home and location of incident.

3. To predict the travel patterns of robbers in 1998 based upon characteristics of the offender's resident neighborhood and the incident neighborhood and a gravity model of the relationship between the two..

4. To predict the travel patterns of robbers in 1998 based upon the patterns of 1997.

5. To assess the quality of the predictions and their value to the police.

Two Models: Econometric and Opportunistic

As outlined in chapter 13, a travel demand model is a four-step sequential model. The first stage is trip generation, whereby the number of crimes originating in a neighborhood and the number of crimes ending in a neighborhood are modeled. The second stage is trip distribution which is a model of the number of trips that go from each origin zone to each destination zone. The third stage is mode split, which models the number of trips for each zone pair (origin zone and destination zone) that travels by a particular

travel. The fourth, and final stage, is network assignment which models the likely routes taken by offenders in traveling between particular zone pairs.

This mapping of links assumes that travel decisions are based upon minimizing costs to get to a valued destination–as sort of geographic rationality. When I go to work, I weigh costs and benefits. I choose the route that will get me there quickest with the fewest problems. Early theories of criminology assumed that criminal activity was no different than other behavior. It was determined rationally. By extension, travel routes and crime locations are also determined rationally.

Trips of offenders are similar to any repeated activity. Most of our activities occur near where we live or work or on the path in between. This is our knowledge space. Trips within it maximize our efficiency and minimize costs. Daily purchases occur close to home with a rapid fall off with distance. But major purchases are an exception. They may occur far away. This distance decay can be generalized to travel cost decay. The more expensive in time, money, and distance, the less likely a trip will occur. Applied to robbery, most incidents occur close to home, but a bank robber might incur greater costs to find a good target. Most previous research has found that predatory criminals avoid incidents too close to home for fear that they will be recognized. Combined with distance decay, this creates a buffer zone of few criminal incidents (Rossmo, 2000).

Environmental criminology assumes that most activity occurs in a knowledge space that includes nodes of residence work and play and the routes between these (Brantingham and Brantingham, 1984,1990) However, the components of travel for criminals may not be the same as other people. For example, for someone with a full time job, getting to work as quickly as possible is important; time is money. For a jobless criminal, time may be less important.

Routine activities theory assumes that both targets and offenders choose their activities based on a weighing of costs and benefits. Offenders seek out targets in locations where they are likely to congregate (e.g. Bars at closing time, rapid transit stations). A crime occurs when an offender and a target converge in the absence of a capable guardian (Felson, 2002). The routine activities of offender may mostly be hanging out rather than rationally seeking targets. What is the basis of convergence? Chance or the decisions of offenders? Any potential robbers decision is effected by both chance and cost. Time and distance are both measures of cost. However, within a short distance of home time and distance costs are near to zero.

An alternative hypothesis is that robbers do not weigh costs and benefits of travel. Rather, the may see an opportunity for crime and take it. Because much of their day to day activity is near home, many incidents occur near the robbers's home. Travel patterns are irrelevant for these crimes. The number of robberies decline with distance from the offender's home because fewer of the robber's daily activities occur far from home. On the other hand, more professional robbers may seek out specific areas or locations where lucrative targets are found and may be willing to travel great distances.

In Chicago, an opportunistic robber's knowledge of good targets may be limited to the isolated area around his residence. In addition, trips within the area cost almost nothing, although other costs, such as risk of capture may be relatively high. The differences between Chicago and Baltimore County or between Chicago and its suburbs has to do as much with knowledge of the distribution of opportunities as with the cost of travel. Chicago's neighborhoods are so isolated that some offenders may have little knowledge of opportunities outside their resident area. The crime travel demand model holds that in the aggregate offenders appear to weigh costs and benefits. However, the data analyzed here says nothing about individual decisions. Decisions may be made with other factors not captured by shortest distance or time.

In one of the few studies of non-arrested robbers Wright and Decker (1997) found that most St Louis robbers are opportunistic and rob close to home. Rationality and careful cost calculation have little to do with their decisions. These are people who have a need for quick money. If they saw an opportunity near home, they would take it. Opportunities were most likely to occur as the potential offender and victim go about their daily routine activities. Many of them are close to home. Therefore, crime occurs close to home.

The closer to an offender's home that an incident occurs the more likely the incident results from a chance meeting. The further away that it occurs the more likely that it is planned. Part of the planning is transportation costs. Usually this is calculated in terms of income. It is difficult to do this for offenders. The best we can do is estimate travel time.

Crime Travel Demand Models in Chicago

The Offender Travel Model is a new application of the Travel Demand Model. The travel demand model has been in development since the 1950's. It is used in every metropolitan area in the United States. *CrimeStat*'s crime travel demand model was outlined in Chapter 13.

As applied to robbery in Chicago, description is as important as prediction. While the CPD has long collected information of the location of the incident and residence of the offender, these were not linked in any systematic way. In meetings with the department, credible descriptive maps, proved to be the most convincing reason to use the new *CrimeStat* travel demand module. Before a new technique is tested, its potential crediblity must be demonstrated. Therefore, the last phase, in the Chicago Travel Demand Model emphasized both the predicted travel demand model and the observed travel of offenders.

Analysis of Chicago's Crime Travel Demand proceeds in three stages. The first step (trip generation) is a prediction of variables associated with the number of crimes originating in each zone and the number of crimes ending in each zone.

The second step is the prediction of links between zones based on zonal characteristics of incident locations and offender residences and a measure of the

attraction between the two zones. These predictive models are compared to the observed links and trips and the previous year's trips used as a prediction.

The mode split step was not run because of the lack of data. Unfortunately, the Chicago police data does not permit an analysis by different modes of transportation (see chapter 15). Data on whether the offender drove, walked, or rode rapid transit to the incident are not collected.

The final step is the description of probable travel routes from the offender's home zone to the incident zone based on shortest time or distance along a transportation network. The links modeled in the second step can be converted to a probable route between home and incident zones over a road network or a summary network load which aggregates travel of all offenders along a transportation network.

Data for the Study

Incident and Arrest Files

The analysis presented here merged information from many sources. This research is based on incident and arrest records from the CPD. Excluding O'Hare Airport, the city of Chicago is divided into 946 traffic analysis zones. Incidents are assigned to these zones for both residence location (the origin) and the crime location (the destination). These include all Chicago robberies in 1997 and 1998 that had at least one known offender who lived in Chicago. These were geo-coded by the address of the incident and all known offenders. Offenders who traveled longer distances are probably under-represented (Block, 2004). About 20% of all reported robberies are included. In 1997, there were 25,000 robberies reported to the police. Of these robberies, 4,636 resulted in the arrest of at least one Chicago resident. Including robberies with multiple offenders, there were 6,643 crime trips.

Traffic Analysis Zones

These incidents and offenders are counted in 946 Traffic Analysis Zones (TAZ). O'Hare Airport is excluded. Chicago's traffic analysis zones are mostly based on a uniform grid of 1/2 mile squares. These are not based on census tracts or other city divisions. However, some census data is available for these zones along with information on employment. About 100 of them have no census population and therefore are unlikely to include the residence of an offender. Land use, employment, population, and robbery incident and offender residence counts are available for all zones. Land use goes beyond the standard census measures to include characteristics from many data sources that might be related to crime. Among these are code violations, vacant parcels, fires, liquor licenses, pawn shops, entertainment venues, distance from the central business district

and other potentially criminogenic characteristics.[1] These traffic analysis zones are the units of analysis. Trips are defined from the center of a zone.

Chicago's Road Network

The base of Chicago's road network is a grid with 1/8 mile between blocks, a feeder street every half mile, and a main street every mile. Layered on top of this grid is a series of diagonal streets that tend to be major shopping streets and a relatively small number of expressways that converge at the edge of the central city. A semi-expressway, Lake Shore Drive, runs along the lakefront for 25 miles. Chicago has a well developed rapid transit system that, unfortunately, could not be included in the current analysis.

Two street networks were available for analysis:

1. **Modified TIGER Line File**: A mostly complete map of all streets and rail lines. Following police practice, the modified TIGER file allows for geo-coding in non-addressed areas, such as parks, by extending the base grid. All public streets are included, but one-way streets are not taken into account and the shortest distance may be on a route that no one would travel. Some areas of the city are not well mapped.

2. **Modeling network:** This includes Expressways, principal arterials and collector roads. Each road segment is uni-directional; that is, it expresses travel in only one direction. Thus, for a two-way road, there will be two records for every segment, one in each direction. This has the advantage that one-way streets can examined since there will not be an opposite direction pair. On the other hand, a modeling network is less complete since minor streets are ignored. This type of map is useful for capturing trips that occur over a mile or more, but is not very useful for the many trips of less than 1/2 mile that occur in Chicago. It does take into account one-way streets. Using distance, the network will over-emphasize surface diagonal streets and will under-emphasize expressways.

One of the advantages of the modeling network is that street segments can be weighted by speed or travel time, rather than just distance. There are eight distinct time periods with the travel time on each segment by period being indicated. Each street segment can be weighted by its travel time in minutes during a specific time period (e.g.; 7-9 AM) to allow a more realistic description of travel behavior. Further, travel in opposite directions can be treated differently since travel times can be different for each direction. During rush hour, travel in one direction may be much quicker than travel in the other direction. Weighting by travel time will allow larger arterial roads and expressways to be

---

[1] In contrast to many cities, Chicago has a large population living in the central business district and lacks a ring of impoverished communities surrounding downtown.

chosen more since travel speeds will generally be faster on the larger capacity roads. This network tends to be most realistic for longer trips but, again, is not useful for very short 'local' trips since the local, neighborhood road network is not included. A greater percentage of the travel is on expressways.

Trip Generation

Using the arrest data, events were aggregated to the TAZ's by both the origins and the destinations. As expected, the distribution of crimes by origin zone and by destination zone were highly skewed. For example, 419 zones had no robberies originate in them while one zone had 27 origins and another had 24 origins.

A similar condition held for the number of crimes by destination. For example, 409 zones had no robberies occur within them while one zone had 24 crimes occur and two had 23 crimes occur.

Separate models of these incidents were developed at the zone level. The regression analysis tools in CrimeStat are excellent, but choosing regression predictors requires both skill and theory. Many explanatory variables were tested. The independent variables chosen for analysis were based on those previously found to be important predictors of violent crime in Chicago. Significant variables were:

1. POP2000 The most important was the 2000 population because the dependent variable was a predicted count of origins or destination. Other variables that were included were:

2. ETHNICPER The percentage of the dominant racial or ethnic group within the TAZ. Recent research (Sampson & Raudenbush, 2001) has found that racial isolation and poverty predicted high community levels of violence.

4. POVPERCENT The percent of the households below the poverty level. Sampson and Raudenbush (2001) found this to be a dominant variables explaining community disorder.

5. VENUE The number of entertainment venues (clubs, theaters, bowling allies) in a TAZ. This is information gathered from the MetroMix and the Reader in 2002. It was negatively related to the residence of the offender and was probably more a measure of perceived neighborhood safety than availability of targets.

6. PAWNSHOP The number of pawnshops is included in several regressions. A pawnshop is both a focus for potential targets and a good place to get cash.

7. VACANT: Count of vacant buildings in the TAZ. Perhaps this is an indicator of general neighborhood dilapidation (Broken Windows).

The variables that were not significantly related to origins or destinations included many that are typically related to travel demand including employment and distance from the central business district. In addition, variables that are often associated with robbery, such as counts of drug arrests, convenience stores, liquor licences, banks and currency exchanges were unrelated to origins or destinations after poverty and population were accounted for. Few TAZ characteristics that might attract an offender to commit a crime were significantly related to the number of robbery incidents in a TAZ. In general the results of the regressions and the resulting travel demand matrix supported the depiction of robbery in Chicago occuring in or near the offender's relatively isolated home neighborhood.

Poisson regressions for origin and destination zone counts for overnight trips were similar in 1997 and 1998. Tables 17.1 and 17.2 present the final Poisson regression models for the resident zone of robbers in 1998 and the location zones for robberies that occurred overnight. In all regression models, population had a positive relationship to the number of crimes, both origins and destinations. Similarly, the poverty variable and the ethnic homogeneity variable were positively related to the number of crimes, both origins and destinations.

Table 17.1

Final Overnight 1998 Robbery Origin Model

Data file:	Chicago TAZ with Time.dbf
Type of model:	Origin
DepVar:	**Robbery Origins 8PM-5:59AM**
N:	946
Df:	940
Type of regression model:	Poisson with over-dispersion correction
Log Likelihood:	-2011.35
Likelihood ratio(LR):	2962.73
P-value of LR:	0.0001
AIC:	4034.71
SC:	4063.82
Dispersion multiplier:	1.00
R-square:	0.443
Deviance r-square:	0.445

Predictor	DF	Coefficient	Stand Error	Pseudo-Tolerance	z-value	p-value
CONSTANT	1	**-2.072610**	0.170828	.	-12.132746	0.001
POP2000	1	**0.000235**	0.000011	0.876420	22.156415	0.001
ETHNICPER	1	**0.015786**	0.001746	0.909463	9.042151	0.001
POVPERCENT	1	**0.037134**	0.002144	0.872974	17.321707	0.001
VACANT	1	**0.016970**	0.002528	0.835809	6.712064	0.001
VENUE	1	**-0.115182**	0.033458	0.933336	-3.442566	0.001

The pseudo-R-square values are reasonably good and an analysis of the residual errors do not reveal any major outliers. Given the large number of zones (n=946) the regressions predict variations in the count of origins and destination fairly well.

Table 17.2
Final Overnight 1998 Robbery Destination Model

Data file:	Chicago TAZ with Time.dbf
Type of model:	Destination
DepVar:	**Robbery Destinations 8PM-5:59AM**
N:	946
Df:	941
Type of regression model:	Poisson with over-dispersion correction
Log Likelihood:	-2041.56
Likelihood ratio(LR):	2661.30
P-value of LR:	0.0001
AIC:	4093.11
SC:	4117.37
Dispersion multiplier:	1.00
R-square:	0.380
Deviance r-square:	0.474

Predictor	DF	Coefficient	Stand Error	Pseudo-Tolerance	z-value	p-value
CONSTANT	1	**-1.946591**	0.032370	.	-60.135432	0.001
POP2000	1	**0.000218**	0.000008	0.898680	26.418877	0.001
ETHNICPER	1	**0.015913**	0.000874	0.944910	18.201093	0.001
PAWNSHOP	1	**0.335678**	0.029184	0.954563	11.501940	0.001
POVPERCENT	1	**0.035707**	0.001888	0.989400	18.913079	0.001

Trip Distribution

After the two predicted models were developed, the trip distribution stage was modeled, in other words the number of trips that go from each origin zone to each destination zone (the trip distribution). The inputs were the predicted origins and predicted destinations for robberies in 1998 from tables 17.1 and 17.2.

The test of CrimeStat's crime travel demand module, began with analysis of 1997. Preparatory analysis indicated that 29% of robbery trips occurred in the offender's home zone. While the number of intra-zonal trips can be mapped and predicted, travel within a zone cannot be described.

Using observed crime trips, the number of trips from each zone to every other zone was calculated. Figure 17.1 depicts the volume of observed inter- and intra-zonal trip links in 1997. The zone shadings indicate the number of intra-zonal trips. The width of the links indicates the frequency of trip links. Impoverished areas of the west and south side dominate this analysis. Most inter-zonal links are quite short. Many begin in zones that

also have many intra-zonal trips. In Las Vegas and Baltimore County many links are associated with specific sites such as shopping malls or entertainment areas. Within the City of Chicago, the links lack a clear focal zone for incidents. However, few robbery trips are made to the central business district.

A trip distribution analysis includes both inter- and intra-zonal trips in a single analysis. The analysis is not of travel from home to destination, but from a home zone to a destination zone. For transportation planners inter-zonal trips are more important than intra-zonal trips because these predict changing transportation needs. The volume of within zone travel can be predicted but not specific routes However, many Chicago robberies (29% in 1997, 26% in 1998) are intra-zonal.

Therefore, two techniques are tested to account for the many intra-zonal trips. In the first analysis, both inter- and intra-zonal overnight robberies trips are included in the same analysis. In the second analysis, to see whether different variables were predicting incidents close to the offender's home address from those further away, inter- and intra-zonal trips were analyzed separately. Ultimately, I concluded that there was little to be gained by separating the two types of trips.

Trip Distribution

The gravity model that underlies CrimeStat's trip distribution model assumes that travel between or within zones is dependent upon the offender pool, opportunities, and costs. Conceptually, this can be written as:

$$T(ij) = \alpha(\text{Offender Pool})\ \beta(\text{Opportunities})/\text{Cost}^{\lambda} \qquad (17.1)$$

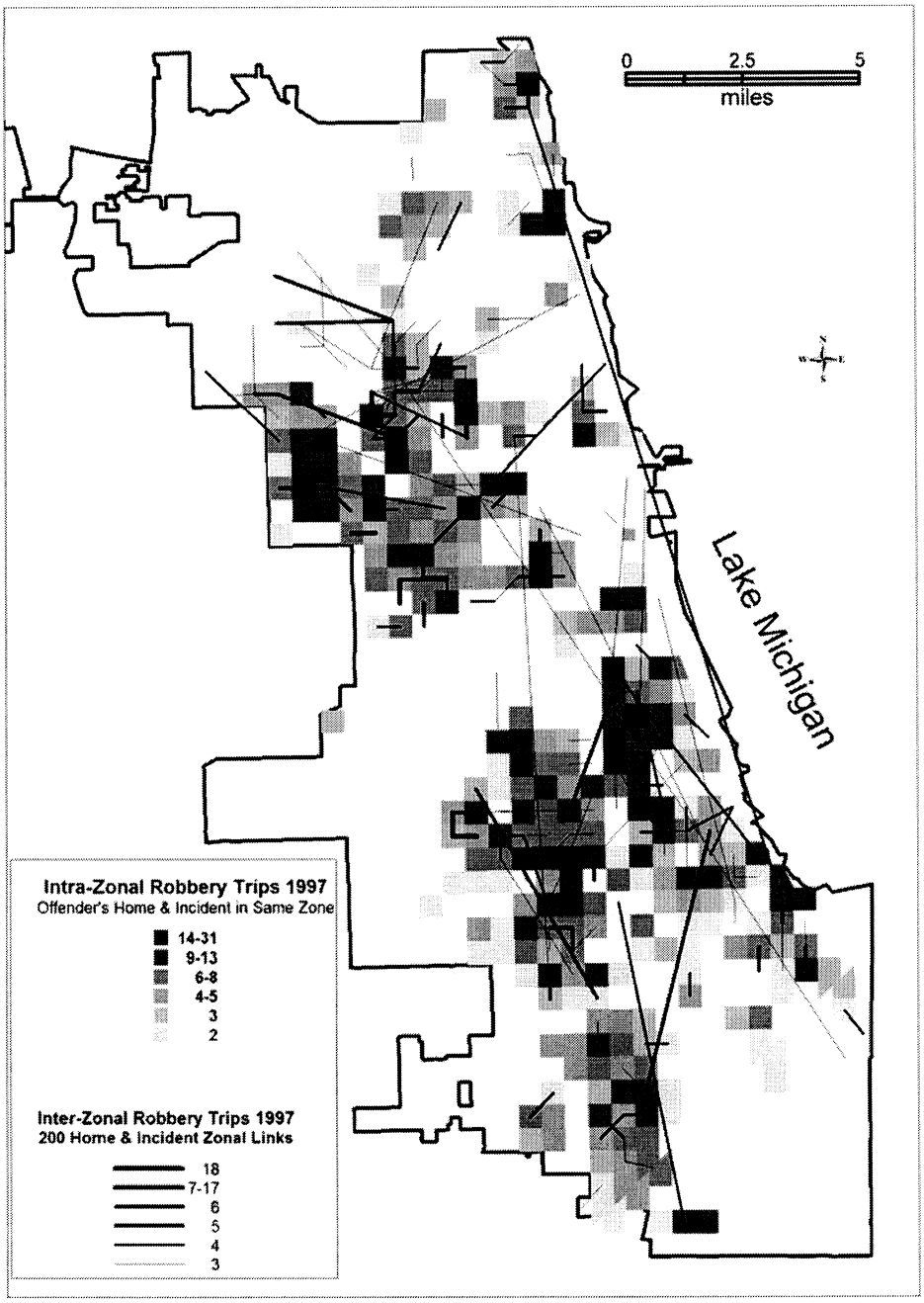
where α and β are coefficients and λ is an exponent. The impedance (or 'cost') component is modeled with a mathematical function. After experimentation, I found that the best impedance function was a lognormal distribution with a mean of 2 miles and a standard deviation of 5. The resulting model fit the actual trip length distribution quite well. The coincidence ratio was about the same for both the 1997 and 1998 comparisons (figure 17.2).

To graphically indicate the trips, straight lines are used to indicate links between zones and widths to indicate volume (figure 17.3). An inspection of figure 17.3, shows that many specific links were not well predicted. In general, the prediction underestimated very short trips but overestimated middle distance trips (2-4 miles).

Predicting 1998 Trips From 1997 Trips

From a police perspective, even the distribution of crime trips can be of value for tactical purposes and for planning interventions. However, the description of 1998 night time robberies travel demand was retrospective-done long after 1998. Can this distribution be successfully predicted? In time series analysis, the best prediction of one period is generally the period that immediately preceded it. In spatial analysis, this is also likely to

Figure 17.1:



Robbery 1997: Intra-Zonal & Inter-Zonal Links

Source: Chicago Police Department Cartography: Richard Block, Loyola University Chicago

Figure 17.2:

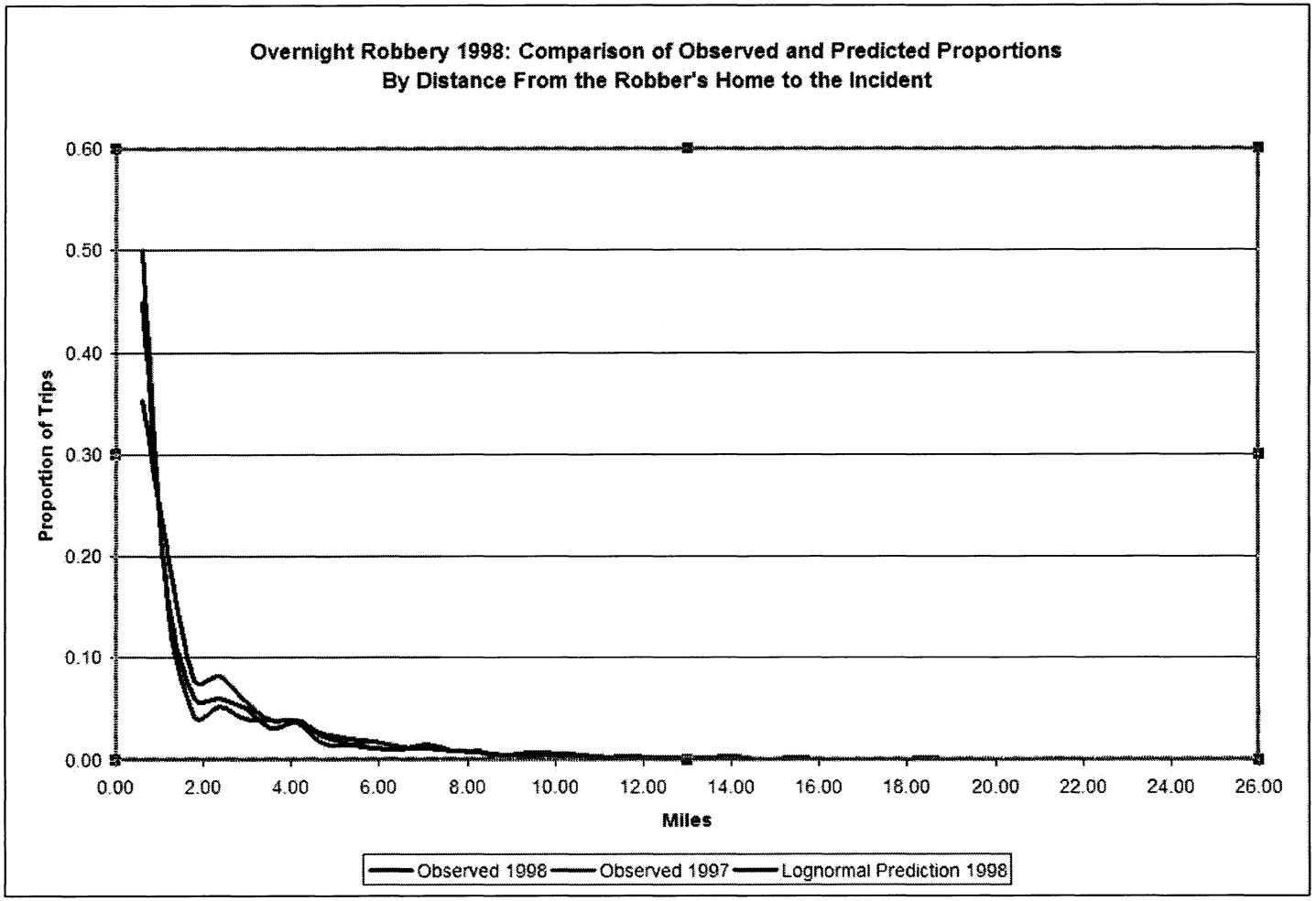
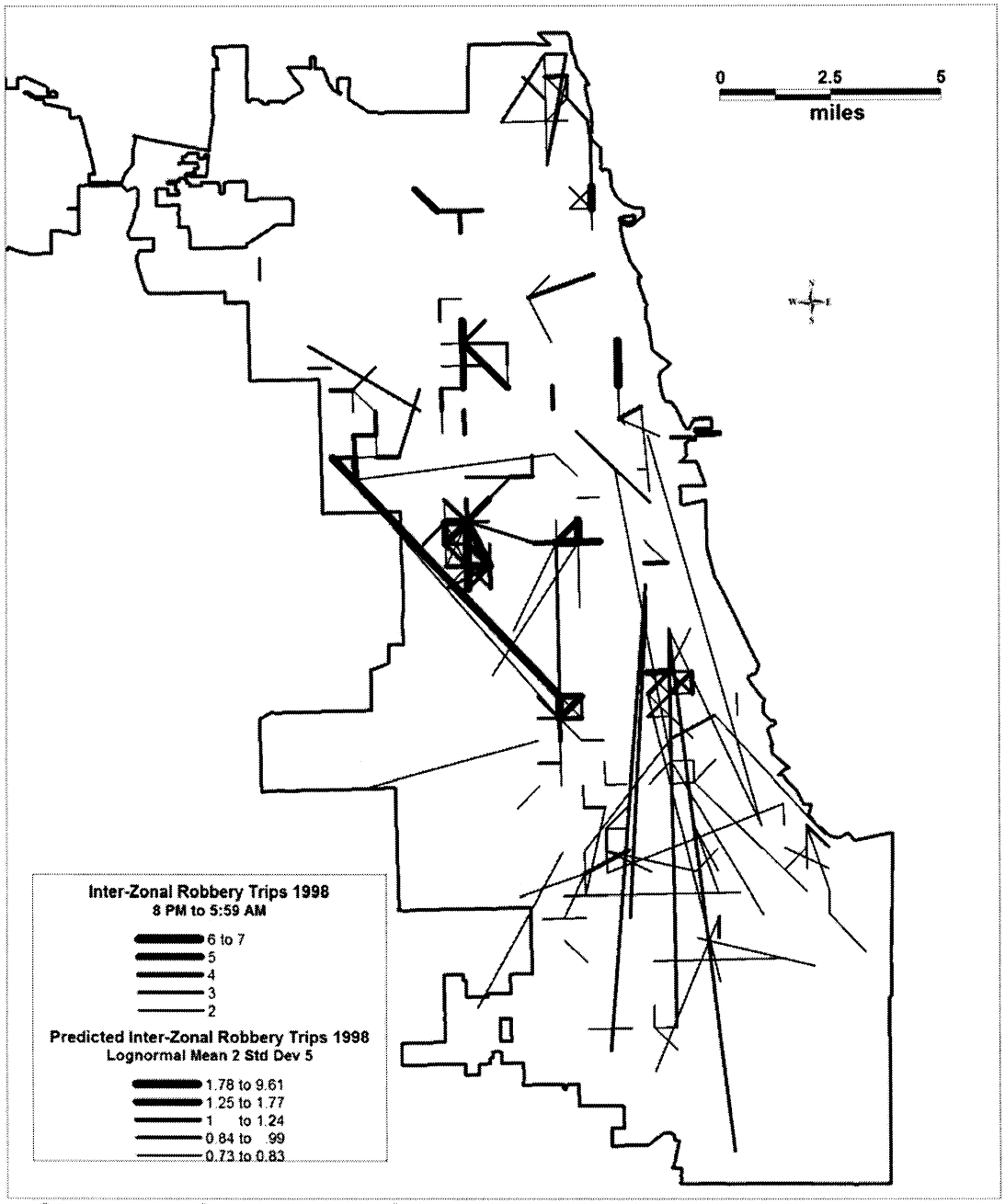


Figure 17.3



Observed & Predicted Overnight Robbery Links 1998

Source: Chicago Police Department Cartography: Richard Block, Loyola University Chicago

be true, especially in a mature city. However, while neighborhood characteristics change slowly in Chicago, they do change. During the late 1990's many public housing projects were emptied and some were torn down. While few neighborhoods deteriorated, many gentrified. Any of these might cause a change in the distribution of robbery trips.

The 1997 observed robbery travel matrix was used to predict observed travel in 1998. *CrimeStat III,* in conjunction with a GIS and a statistical package, provides several comparison tools. Comparing 1997 and 1998, the fit is quite good. Including street segments that had no trips in either year, 55% of the trip links in 1998 were predicted by the trip links in 1997 (r2=.741). The coincidence ratio of .86 for 1998 and the distance distribution in figure 17.2 above indicate a high degree of similarity. However, a comparison of the top 300 trip links illustrates that, while zones with many intra-zonal incidents are fairly well predicted, inter-zonal trips are not well predicted. Mapping these makes clear that 1997 inter-zonal links cannot accurately predict specific 1998 links (figure 17.4). However, specific links may be less important from a police perspective than knowledge of the frequency of offender travel on specific streets.

Predicting Overnight Robbery Trips

After selecting only those 1998 robberies that occurred from 8 PM to 5:59 AM, a zone to zone matrix was constructed. Unlike the analysis above, this matrix included both intra-zonal (31.5% of the total) and inter-zonal trips. As shown in Figure 17.5, zones with many intra-zonal overnight trips also had many inter-zonal trips. Intra-zonal links were widely dispersed throughout the city with an area of concentration on the west side, but there was no clear pattern.

Mode Split

Because of the lack of information about travel mode, the mode split model was not run. It is hoped that, with better information, this type of model could be run in the future.

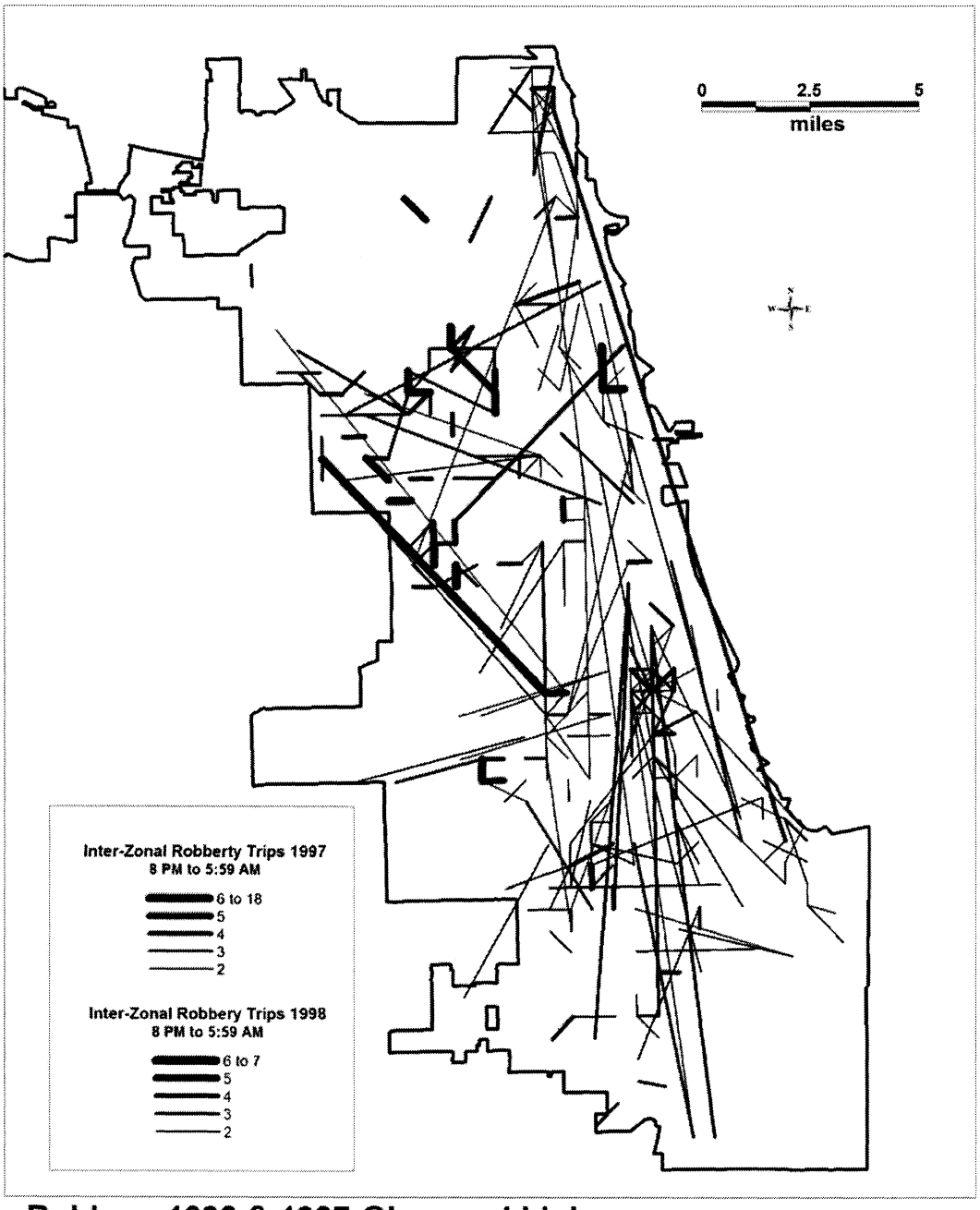
Network Assignment

The third, and final step, in the analysis was to examine the likely routes taken as well as the total demand placed on the road network. Network assignment is an especially useful tool for police work because it suggest possible locations for intervention. Because it is based on the actual street network, it is more concrete than a depiction of links. Therefore, I tested several ways to depict network assignment for 1997 robbery travel before proceeding to the 1998 analysis.

The network assignment routine in *CrimeStat III* outputs two results:

1. The shortest routes on a street network. For each zone-to-zone pair, the shortest path is calculated.

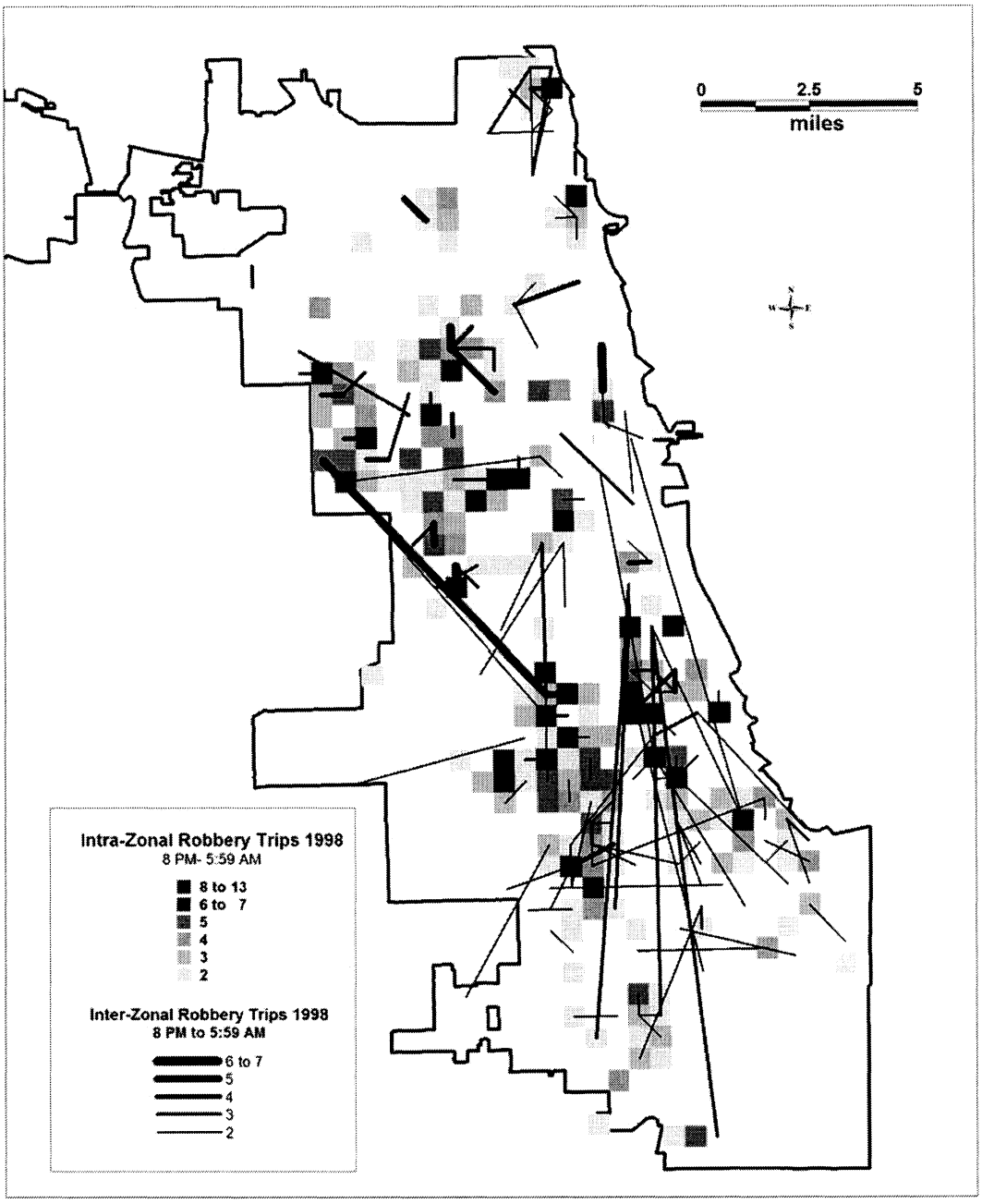
Figure 17.4:



Robbery 1998 & 1997:Observed Links

Source: Chicago Police Department Cartography: Richard Block, Loyola University Chicago

Figure 17.5:



Robbery 1998: Overnight Intra-Zonal & Inter-Zonal Links

Source: Chicago Police Department Cartography: Richard Block, Loyola University Chicago

2. The Network load. Network load counts the number of trips over each street segment regardless of origin or destination and sums these.

Both the shortest routes and the total network load can be based on time or cost rather than distance.

First, all inter-zonal robberies in 1997 were mapped along Chicago's street network by shortest distance. The 4000 trips were counted along each of Chicago's 51,000 street segments and mapped as a network load. As the width and color changes from blue to red in Figure 17.6, the number of trips that passed over a segment increases. However, this map is difficult to interpret and lacks credibility. Much of the load is along small side-streets. Diagonal streets are emphasized and expressways are ignored because they usually are not the shortest route in terms of distance. Also, travel in the wrong direction on a one way street is possible since only distance was used to calculate the shortest path. The CPD did not believe this to be a useful map.

The same inter-zonal links were mapped again along using the Chicago modeling network, but weighting segments only by distance (figure 17.7). While this resulted in a greatly simplified map, it still lacked some credibility. Expressways are rarely the shortest distance, therefore, their use is under emphasized. The algorithm results in an over emphasis on diagonal main streets. Some connected segments looked like a stair case following along Chicago's grid of main and secondary streets from one high incident neighborhood to another on the west and southwest sides.

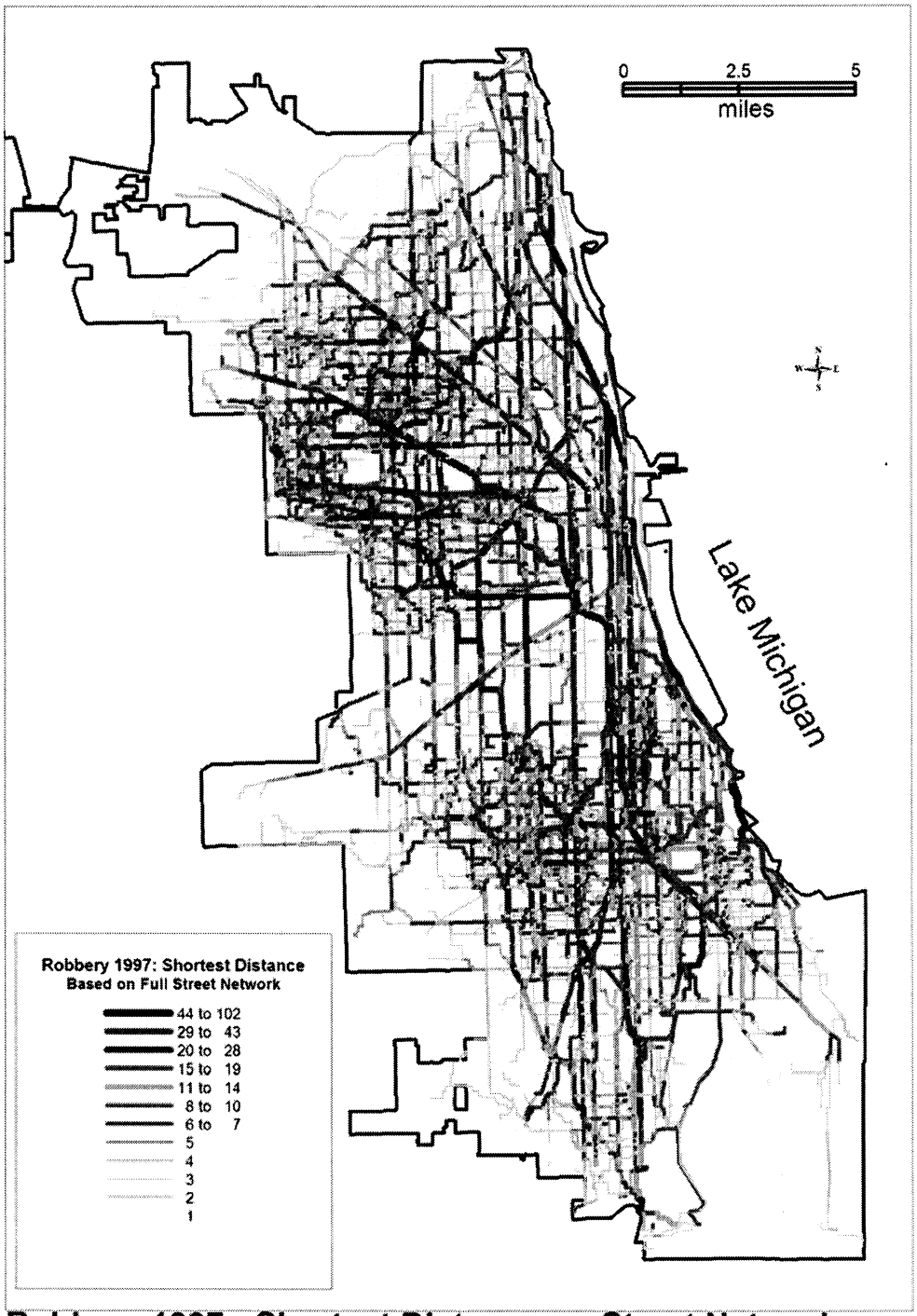
In other words, distance did not seem to be a good representation of travel routes. Given that police records include time of incident and travel time along Chicago's road network is available, and that *CrimeStat* allows for analysis by travel time, I re-conceptualized travel cost as shortest time rather than distance.

Shortest Time or Shortest Distance?

What does distance measure? Traveling ten miles during Chicago's evening rush is quite different than at midnight. However, the two blocks from my house to the nearest convenience store is unaffected by the time of day and little effected by the mode of transportation. While distance appears to be a straight forward measure, it is not. At close distance, it specifies knowledge space or the location of routine activities. Further from home, it is related to a lack of knowledge, but is also an inaccurate measure of the cost of travel. Better measures than distance are often available. All U.S. major metropolitan areas map travel time by time of day on major streets, feeder streets, and expressways using modeling networks (see chapter 16). These maps along with police data on time of incident can be combined to realistically describe shortest travel time rather than shortest distance.

The Chicago Area Transportation Survey (CATS) divides the day into eight time periods based on travel demand. Whether a crime trip was intra- or inter-zonal was unaffected by time of day (χ^2=7.07 sig=.421 in 1998). Not surprisingly, the robber's daily

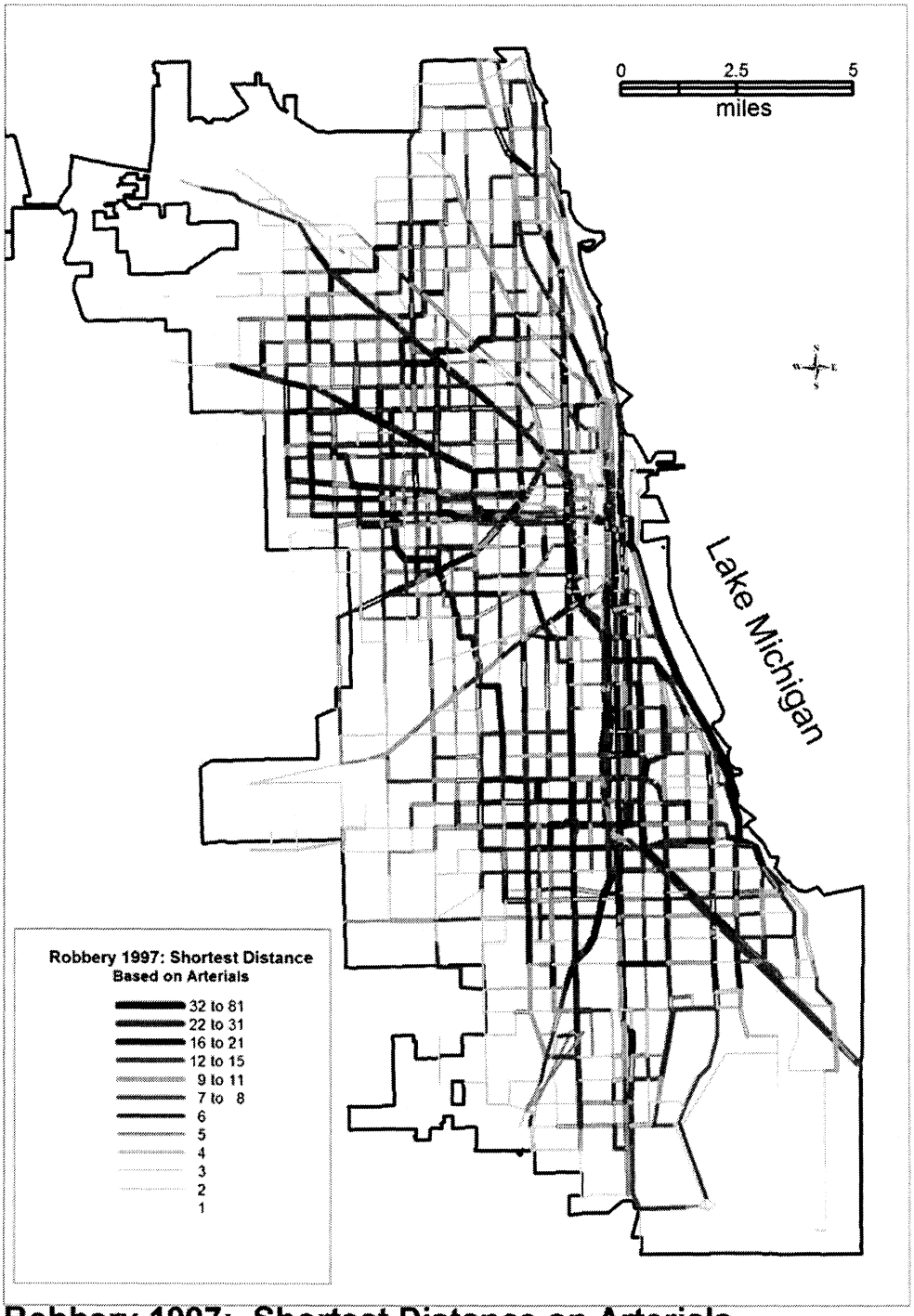
Figure 17.6:



Robbery 1997: Shortest Distance on Street Network

Source Chicago Police Department Cartography: Richard Block, Loyola University Chicago

Figure 17.7:



Robbery 1997: Shortest Distance on Arterials

Source Chicago Police Department Cartography: Richard Block, Loyola University Chicago

travel cycle is different than the general population. In 1998, robbers show little demand for travel in the morning rush hour period (6 AM to 10 AM). Of the remaining trips, about half (46% in 1998) occurred from 8 PM to 5:59 AM. These overnight trips are the subject of the analysis presented here.

Overnight Robbery Trips

Overnight network load was mapped on Chicago's arterial roads according to both shortest distance (figure 17.8 left) and shortest time (figure 17.8 right). As before, the two maps are very different. Expressways are rarely included in the shortest distance between zones. Much of the travel is on diagonal surface streets. However, if time is taken into account, many of the trips are on expressways and on Lake Shore Drive. This is probably a more realistic description of longer distance trips.

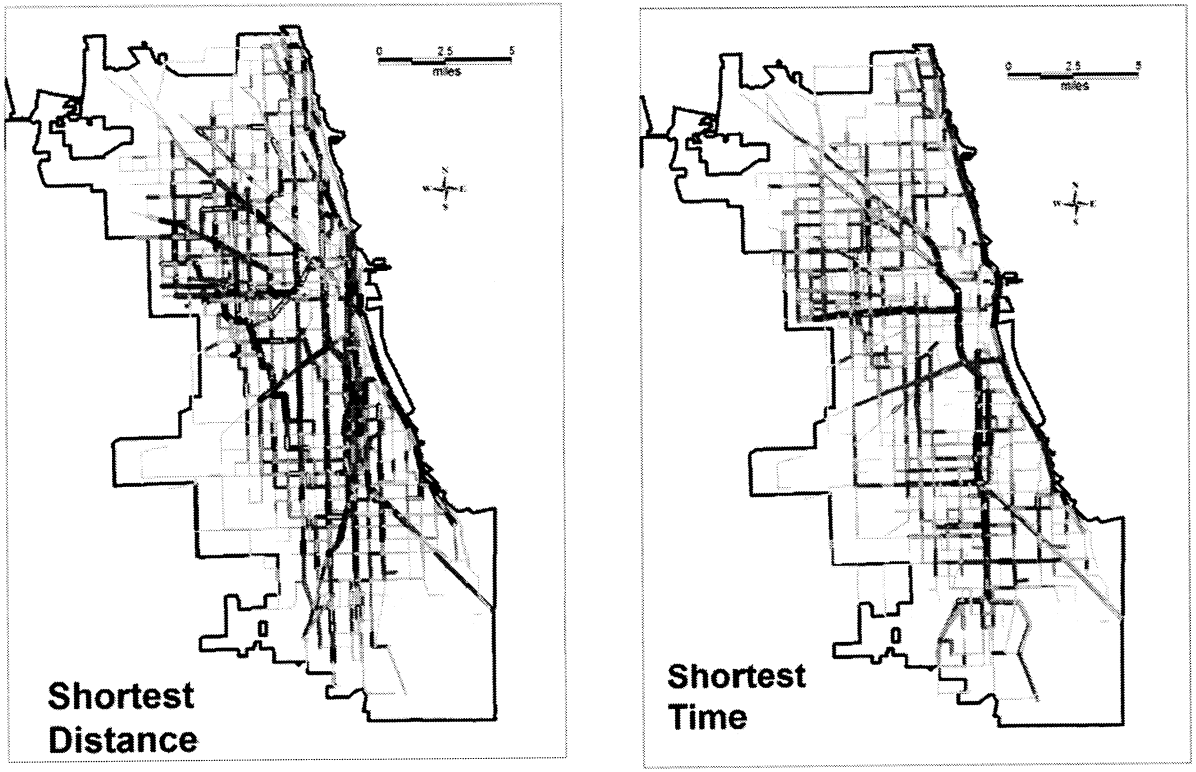
In moving from a complete street network to a simplified network using distance as an impedance to a time-based network, the description moves from an unrealistic and probably un-interpretable map to one that probably corresponds to the routes taken by offenders. Does this add to police knowledge? Of the 10,763 mapped segments in the network, 65.1% had no predicted trips assigned to them. Two percent of the road segments, those with 15 or more trips, contributed 20.2% of the 16,162 robber's movements across road segments. These were typically arterial roads or expressways. Nevertheless, by identifying these streets as those most likely to carry crime trips, these 'hot street' segments could become a focus for police patrol or for intervention to prevent crime.

Feasibility & Advantages

The police already collect information on the location and time of incidents and the home address of arrested offenders. Can this information be utilized to describe and predict the travel patterns of Chicago robbers? First, *CrimeStat*'s trip distribution module was used to describe zonal patterns of travel for all known 1997 Chicago robbery offenders. Around 30% of Chicago robberies are committed near to the offender's home. For these a zonal model cannot predict travel patterns. For other robberies, a time-weighted travel pattern resulted in a more credible description than one based on distance. However, even this description resulted in an over emphasis on travel along surface grid streets and diagonal streets rather than expressways.

The key to analyzing the robber's travel pattern is to reconsider the meaning of distance. Close to home or work, distance represents a knowledge space and an opportunity space, a place the offender knows in which he or she spends a lot time. This is an area where the benefits of knowledge may outweigh the costs of possible capture or it may simply be where the offender hangs out.. Further away, shortest distance is a poor representation of travel cost. In major metropolitan areas, a better representation is shortest travel time. Combining travel time of day with time of incident, results in a more realistic travel pattern.

Figure 17.8



Robbery 1998, Offender Travel Network for Incidents Occuring from 8 PM to 5:59 AM. Shortest Travel Time & Distance Compared

These intra- and inter-zonal links are, themselves, a new way to look at the relationship between offender and incident. However, they need some representation before they are useful to the police for tactical analysis or crime prevention. In my discussion with the Chicago Police Department, a network load map seemed to be most useful. Network load summarizes the number of crime trips that passed over each segment in a road network.

Limiting analysis to robberies occurring overnight (8PM to 5:59 AM), 1997 travel patterns were a good predictor of travel distances, intra-zonal robberies, and network load in 1998. However, 1997 travel patterns only weakly predicted specific links between traffic analysis zones. For 1998 incidents, a trip distribution model (using Poisson regression of the zonal count of robbers' homes and incident locations, and a impedance function) modeled the overnight travel links between home and incident. Substituting a lognormal impedance function - that better matched the observed overnight robbery pattern, resulted in predictions that were nearly as good as the 1997 observed travel patterns. A combination of these predictions with analysis of travel patterns over several years might eventually result in an excellent zonal prediction of crime travel patterns.

Crime travel demand analysis is complex and time consuming and requires a relatively powerful PC with a large memory capacity. Is it worth it? Yes. Information on crime trips is automatically gathered by the police, but it is not fully utilized. However, unlike transportation planners, police are generally concerned with the short term and with acute rather than chronic problems. They work on an existing street network rather than planning for the future. Crime travel demand models may better serve the police as short term descriptions rather than long term predictions and can probably be used to describe the effect of specific police interventions such as road blocks or drug interdictions. The crime travel demand model along with a GIS can identify hot street segments–those segments that are most likely to be on the travel routes of offenders and most useful for intervention to prevent crime.

For researchers, on the other hand, a crime travel demand model is a good way to ask long-term, structural questions. If the travel patterns remain relatively constant over time, then these relationships can be modeled using a limited number of variables. The result is a way to compare different metropolitan areas as well as a way to look at the same metropolitan area over different time periods. It's a framework for analysis that is broader than just a journey-to-crime type of description.

Limitations

There are also limitations to the model:

1. Only crimes with at least one known offender are analyzed. To the extent that offender travel patterns in unsolved crimes are different than those with known offenders, travel patterns will be misrepresented.

2. The model works best if records are gathered in such a way that the address of an offender home can be linked to the address of an incident.

2. The travel demand model assumes that the offender's home address is accurate. Offenders may not have a stable address or may give a false address.

3. The travel demand model assumes that offenders travel directly from home neighborhood to incident neighborhood; many probably do not.

4. The crime travel demand model is an aggregate model, not a individual one. It predicts travel from the center of one zone to the center of another. It cannot predict specific trips or the behavior of specific offenders and cannot predict travel within a zone.

5. The model must be crime and city specific. Chicago robbers were much more likely to attack close to home than those in Baltimore County or Las Vegas. Because these homes were distributed throughout the city, the travel patterns of Chicago robbers were much less focused on single target zones than in the other test sites.

6. The study of Chicago was limited to incidents that occurred in the city of Chicago. It does not model travel patterns of incidents occurring outside the city and can say nothing about them.

7. The data available from the Chicago Police Department did not allow for a test of travel mode used. It cannot be assumed that criminal trips use the same modes of transportation as non-criminal trips.

Conclusions: Chicago

Chicago is a city of isolated neighborhoods. Even nearby neighborhoods may be *terra incognita*. Crime travel follows the pattern of neighborhoods. In Chicago, many robberies occur very close to the home address of the offender. The crime travel demand model cannot analyze these crime trips because each zone is represented by a single point. In some impoverished neighborhoods, robbery is very common. An offender can opportunistically attack on any block. Even when offenders travel they tend to stay nearby their home neighborhood. The isolation of robbery in the a few neighborhoods results in a downtown that is relatively free of incidents and crime trips are relatively short.

Chicago is a mature city. Neighborhoods change slowly. Large scale changes in housing, poverty, or attractors do occur---the destruction of public housing, widespread gentrification and the replacement of rail yards with upscale housing. With these changes come new opportunities for crime and changing crime travel patterns. These may be predicted with the new crime travel demand module.

II. Application of Travel Demand Behavior Model on Crime Data from Las Vegas, Nevada

Dan Helms
GIS & Crime Analysis Specialist
Crime Mapping & Analysis Program
National Law Enforcement & Corrections Technology Center
Rocky Mountain Region
Denver, CO

Introduction

Strategic crime forecasting has for many years relied on a limited and simplistic suite of methods to predict approximately where future events may occur in broad strokes. Extrapolation of percentile change is probably the most commonly used means of forecasting future crime frequencies, based on the notion fundamental to all predictions, that the future will resemble the past. Unfortunately, this method is completely unable to cope with changes in the demographics, population, and social makeup of a jurisdiction.

For a number of years, innovative crime analysts and criminologists have looked to other disciplines outside the study of criminal behavior for methods of predicting how the future will unfold. Economics, epidemiology, meteorology, and biology have all offered significant contributions, as their more sophisticated and creative methods for foretelling future frequencies have been adapted to criminology with varying degrees of success.

Transportation modeling is the most recent external science to suggest potential means of predicting criminal behavior. The success of travel-demand modeling in the civilian world of transportation behavior has presented us with another possible technique which could be adapted to forecasting crime.Travel-demand modeling offers an algorithm for estimating not only how much activity will occur in a given region, but also how offenders will travel across the jurisdiction to commit their crimes. This model has been implemented in the *CrimeStat* software application for use against crime data.

In this study, we will review the application of this model against data from the metropolitan Las Vegas area over a period of three years.

The Las Vegas Metropolitan Area

The Las Vegas metropolitan area is comprised of Clark County, Nevada, and several independent municipalities within it. The Las Vegas Metropolitan Police Department (LVMPD) serves Clark County (in the capacity of a Sheriff's Office) as well as the City of Las Vegas (in the capacity of a municipal police department). Although the vast majority of the land area, population, and businesses within this area are policed by the LVMPD, there are three other significant jurisdictions: The City of North Las Vegas, the City of Henderson, and the City of Boulder City, each having their own police department.

In addition to these important sibling agencies, several other law enforcement agencies have overlapping jurisdiction within areas principally policed by the LVMPD: The Paiute Tribal Police, the Southern Pacific Railway Police, the Nevada Highway Patrol, US Air Force Security Police, US Air Force Office of Special Investigations, Federal Bureau of Investigation, Veteran's Administration Police, and others. Although these agencies perform valuable police functions, the LVMPD unquestionable deals with the vast majority of crime in the vast majority of locations, making it an attractive candidate for offender travel research.

In many ways, Las Vegas resembles an island. Surrounded by barren desert, with very few roads entering or leaving the city, it is an urban oasis in a sparsely populated desert wilderness, consisting of largely impassable terrain. This geographic position and isolation make Las Vegas highly interesting from the perspective of a transportation (or crime trip movement) modeler.

Another unique feature of the Las Vegas area is the highly transient nature of the population, which falls into three discrete categories:

1. First, the Resident Population consists of some one million persons, approximately 880,000 of which live in the jurisdiction of the LVMPD (the remainder being served primarily by Henderson and North Las Vegas). These permanent residents are the mainstay of the community and the source for demographic data used by the census bureau and planning agencies.

2. Second, we must consider the Visitor Population, consisting of some 35,000,000 - 40,000,000 persons per year. On any given day, between 100,000 and 500,000 visitors will be staying in the Las Vegas area - a critical factor in transportation, demography, and crime! These tourists sometimes act as crime importers (e.g., criminal street gangs from neighboring Californian cities often visit Las Vegas for weekend mayhem, or more professional criminal purposes); in most instances, however, they serve as a pool of prey for local criminals.

3. Third, and finally, there is a substantial Homeless Population in Las Vegas, drawn by the seasonally warm climate and the ease with which this city can be reached as a destination. Although not famous for a "friendly" attitude toward the homeless, these persons are protected by law enforcement in Las Vegas and are well served by many charitable social institutions and services. Because Las Vegas is also an easy place to sin, homeless individuals with drug, alcohol, and gambling addictions often gravitate here; the possibility of "winning big" and instantly reversing a life of misfortune also weighs in the consideration of many homeless who choose to make their base in Las Vegas. Although not a major source of annoyance as criminals, nor overly victimized by criminals, these persons do constitute a significant (although never well-measured) fraction of the local population, and

therefore of local crime statistics. However, due to the inability to accurately measure a "home" location for these persons when they do commit crimes, few of these have been represented in this study.

This study will focus on the criminal movement behavior of the resident population of the greater Las Vegas metropolitan area.

Source Data Provenance and Organization

Data concerning the Las Vegas metropolitan area was provided by the Las Vegas Metropolitan Police Department's Investigative Division. Often, researchers underestimate the severe difficulties and chronic shortcomings of law enforcement data. Thanks to a first-rate RMS, and a voluminous tactical database repository, the Las Vegas Metropolitan Police Department's data presented relatively few problems; however, geocoding accuracy issues, missing data fields from *modus operandi* tables, and erroneous arrestee home locations resulted in some difficulties. These had to be overcome before any analysis or testing of new methods was possible.

Crime report data for the LVMPD is maintained in an SQL-Server 7.0 database constructed by the Printrak (now owned by Motorola) company, makers of the Law RMS (LRMS) police records management system used by Las Vegas, among others. This repository currently houses many hundreds of thousands of crime reports, field interviews, and other critical police data in a well-organized, relational database.

Crime reports are filled out by either sworn officers (when taken in the field) or by station personnel (when reported in person at an LVMPD substation or city hall). These paper reports include ample MO detail and descriptive information in compartmentalized, "force-choice" fields, as well as substantial expository narratives. "Forced-choice" fields are also typically supplemented by "Other" options which can then be individually explained, to deal with very unusual crime behaviors, descriptions, or details.

At the end of each shift, officers submit their reports to their sergeant for review; after a quick check to ensure the most basic levels of data quality and integrity, the reports are then placed in a mailbox for pickup, which occurs several times each day and night. Reports are transferred by intradepartmental couriers to city hall, where they are collected by the Records Section. Professional data entry specialists then meticulously type each report into the LRMS database.

The data entry process includes several validation and error-trapping elements. These usually greatly enhance the completeness and accuracy of each report, but are sometimes bypassed by busy clerks. Perhaps the most significant validity check which can be bypassed is the address verification system, which performs a brute-force match against a "geofile" of known, valid locations. When a matching address is entered into the system, geographic coordinates and other useful data is automatically propagated into the file. Because many crimes do not occur at valid, documented physical street addresses (crimes

in remote or desert areas, or in new construction zones, or on buses or in taxi cabs, for example), however, data entry clerks have grown accustomed to overriding the address verification module. This is also sometimes done in the interests of speed and expediency, even when a valid, matchable address is provided in the crime report. When this happens, the resulting address must be cleaned using a data cleaning application prior to successfully matching in a geocoding operation. Once entered into the LRMS database, crime report information may be extracted through a variety of standard methods.

The LVMPD routinely downloads crime reports on a daily basis into an ATAC analytical database where crime analysts and investigators can examine and study the data without creating any drag on the primary server. The ATAC database is streamlined for analysis, and is much easier to query and analyze than the LRMS repository itself. The ATAC databases are Microsoft Jet-compliant relational database very similar to the MS Access 2000 database.

Data used for the Next-Generation Offender Crime Travel Model project were derived from records stored in several ATAC analytical databases created and maintained by the LVMPD Crime Analysis Section. These databases are archived by calendar year and by crime category. The archive dates for calendar year are assigned based on the year of occurrence. Crime categories are: Auto Crimes (including motor vehicle thefts, burglaries from motor vehicles, and criminal damage to automobiles); Burglaries (including all burglary statutes); Larcenies (including all Larceny/Theft statutes); and Personal (including all sexual offenses, assaults and aggravated assaults, robberies and home invasions, kidnappings, and homicides).

These databases contain MO, Persons, and Vehicles tables, related by event number. The MO table contains all information pertinent to the location, timing, category, and methods of each crime event; the Persons table all information on personal identification, description, and histories, not only for suspect and arrestees, but also victims, witnesses, reporting parties, etc.; the Vehicle table all information concerning any vehicles which may be involved in the offense, including descriptive and identification information, whether the vehicle relates to the criminals, victims, or has some other relationship to the crime.

For purposes of this project, the LVMPD authorized access and transmission of the contents of the complete ATAC database inventory for the Crime Analysis Section. Of the fifty-odd databases provided, the Personal Crimes databases for the years 1996 - 2002 were initially selected.

Data Screening

Three broad categories were selected from the complete data inventory provided:

1. Confrontational
2. Burglary, and
3. Vehicular crimes.

These intentionally disparate data were selected in the interests of increasing the latitude of the study. It was hypothesized that travel behavior would vary between these categories of events. Confrontational crimes included sexual assaults, robberies, kidnappings, and murders. These crimes were included in a single group as part of this initial appraisal of the effectiveness of travel-demand modeling on criminal behavior, even though it is obvious that the behaviors exhibited by offenders across these crime types are likely to vary. These crimes were grouped in spite of these likely differences because similarities in targeting behavior across these crimes might make them amenable to collective analysis; a hypothesis which can be tested using the techniques built into the travel-demand module.

Burglaries used in this analysis included both residential and commercial burglaries, but not burglaries from motor vehicles. Only crimes in which a building or property was illegally entered for the purpose of theft were included in this study, thereby eliminating the prolific larceny category.

Vehicular crimes included both auto thefts and burglaries from motor vehicles. "Carjackings" were not specifically included, but some auto thefts in which the modus operandi followed the confrontational "carjacking" pattern may have been included when specifically statutory designations were missing to differentiate these from more typical auto thefts.

Some operational definitions of these crimes are in order.

1. Sexual assaults used in this analysis included forcible rapes with victims of either sex, as well as any other physical, sexual abuse of another person of either sex - such as digital or objective penetration, fondling, etc. - and also open and gross lewdness (e.g., "flashing"). Statutory sexual seduction ("statutory rape") was excluded.

2. Robberies used in this analysis included all robbery-related statutes in the Nevada Revised Statutes (2002), including home invasions.

3. Kidnappings were included in confrontational crimes, but the application of kidnapping as a statutory offense by law enforcement in Las Vegas (and elsewhere) may be counter-intuitive to some readers. Kidnapping is often attached as an additional offense to other crimes, such as robberies or sexual assaults, in any case in which the victim is forcibly moved from one location to another. This practice is used primarily as an adjunct to prosecution, because kidnapping (unlike either robbery or sexual assault) is a federal crime, and in some cases may be easier to prove in court.

4. Homicides used in this analysis included all murder statutes, as well as all manslaughter statutes. No justified homicides were included.

Once the target crime categories have been defined, separate databases for each of the three categories were compiled. Although data for several years was made available, all but three years of data were excluded from the study. Data prior to 1997 was often relatively poorly maintained and prepared, and sometimes contained serious omissions which made it unreliable. Data for the year 2002 was incomplete when this study was commissioned. Although crime data for the years 1997 and 1998 was functionally reliable, socio-economic and transportation data for these years was not readily obtainable at the time this study commenced; since these data were necessary for implementation of this model, these years, too, were excluded from analysis. Therefore, only the years 1999, 2000, and 2001 were included in this study.

Because this study focuses on spatial relationships between crime event locations and criminal home locations, only solved crimes could be used. Crimes were included as "Solved" when an arrest was made - unfortunately, difficulties in obtaining data from the justice system and the long delays inevitable in the prosecutorial process made it impossible to identify crimes in which a conviction had been obtained; an arrest was the closest approximation to a reliable solution possible for this research.

Of those "solved" crimes in which an arrest was made, only those in which the offender's home address and the precise location of the crime itself were both known could be used. Even when crimes were closed by arrest, and adequate data was available to geographically plot and analyze the case, some have still been excluded. Instances in which the offender and victim both live at the scene of the crime have been excluded from these analyses, since no travel was involved; however, instances in which either party lived at the scene of the crime but the other did not have been retained. The reasoning behind this decision is that the decision to commit a crime at a given place does include the decision to commit a crime in one's own home. Therefore, the spatial travel (none) component of this decision should still be reflected in the model if we hope to eventually derive a valid statistical representation of offender travel behavior.

Also, crime in which the offender lived outside the study area (Clark County, Nevada) have been excluded in most cases - but not all. In some cases, "tourist" offenders may have been included when their temporary "base of operations" (i.e., local lodgings) have been recorded. In these instances, the hotel, motel, resort, or private dwelling they lived in has been used as a "home" location for purposes of originating a crime trip.

The number of cases usable for each category of crime varied significantly from year to year (table 17.3).

Table 17.3
Confrontational Crimes Available for Analysis

Year	Total Offenses	Usable Offenses
1999	5272	1080
2000	7560	1643
2001	3588	991

The large increase in number of offenses from 1999 to 2000 is difficult to explain; the following substantial drop (52%!) is even more troubling. A similar, but inverted, discrepancy emerges in the frequency of burglaries reported during those years (Table 17.4).

Table 17.4
Burglary Crimes Available for Analysis

Year	Total Offenses	Usable Offenses
1999	17234	2520
2000	12899	2040
2001	16403	2733

A final enigma, most significant of all, is obvious when we look at the frequency of auto crimes over the same three-year period (table 17.5).

Table 17.5
Vehicular Crimes Available for Analysis

Year	Total Offenses	Usable Offenses
1999	6871	646
2000	15025	1219
2001	8349	894

These disparities are hard to account for.

On the whole, 1999 had a middling number of auto thefts and confrontations, but a shockingly high number of burglaries; in 2000, on the other hand, the confrontations and auto thefts radically increased (the auto crimes by more than double!), but burglaries dropped notably. Finally, in 2001, confrontational crimes drop to the lowest levels (a staggering decrease), as do auto crimes, while burglaries leap up to nearly 1999 levels!

How can we explain these strange fluctuations? Given the large percentages involved, it's tempting to imagine some change in counting or reporting procedures in 2000; however, a scrutiny of the policies and procedures for the LVMPD does not seem to bear this out. Previous years (1996 - 1999) do not evince a similar wide degree of variation. The reason or reasons for these crime reporting "mood swings" remains unknown. Is there reason, therefore, to distrust these data?

For purposes of this study, the answer appears to be, "No." That is, the data used for these analyses should, even allowing for as yet-unexplained vagaries in reporting, comprise a representative sample of the reported crime activity in Las Vegas over these years.

Since forecasting the frequency of crime is a relatively minor component of the travel-demand model, these numeric sine-waves shouldn't cause us too much concern.

Instead, since the focus of this model is the effective explanation and representation of the distribution of crime trip generators and crime trip destinations (and, as a function thereof, of the crime trip paths between them), the frequencies themselves should matter little.

Reference Data

The Traffic Analysis Zone (TAZ) file for Las Vegas was selected as the optimum polygonal reference theme for this study (figure 17.9). This file was provided by the Metropolitan Planning Office for Las Vegas, the Regional Transportation Commission through the courtesy of David Granata, Senior GIS Analyst, himself an expert in transportation modeling through the use of geographic information systems (GIS). The data provided included historical data for 1999, 2000, and 2001, enabling more accurate modeling of the importance of various factors longitudinally across time. The TAZ dataset was provided in ESRI shapefile format, which is intrinsically legible to the CrimeStat application on which the model is to be built.

The TAZ shapefile includes information on housing, employment, income, population, road mileage, and a variety of subset data specific to particular types of employment (e.g., "Strip" jobs, Nellis Air Force Base employment, entertainment-related jobs, vacant properties, number of pawn shops, etc.).

An additional reference theme is needed to apply the final step in the travel-demand model, the network assignment method. The Major Street Centerline file (LVMAJSCL.shp) in ESRI shapefile format was selected (figure 17.10). Although only including arterial streets, freeways, and major thoroughfares, this transportation network layer is all that is needed to describe the vast majority of trips (of any sort) in Las Vegas. The addition of bus route information may prove a useful supplementary network to future analyses using this model.

Assignment of Crime Trips

Data from each year, by category, is assigned to a simple tabular database consisting of an identifying variable (Event Number as primary key), Origination coordinates (coordinates of the offender's home address, or local base of operations in the case of external offenders), and Destination coordinates (coordinates of the crime scene). These data were then combined into an *MS Access 97*® database for analysis using CrimeStat. Figures 17.11 and 17.12 shows the assigned origins and destionations.

Each origin-destination pair is termed a "Crime Trip." Following the reasoning of transportation modelers, it is understood that offenders to not leave their homes, travel directly to a crime scene to commit an attack, then return home. Instead, each "sortie" is likely to consist of several stages.

For example, a sexually predatory offender may get up in the morning, leave home, drive to work (stopping for coffee along the way), then go out to lunch before returning to the office, then on his way home depart from his usual route to drive through a residential

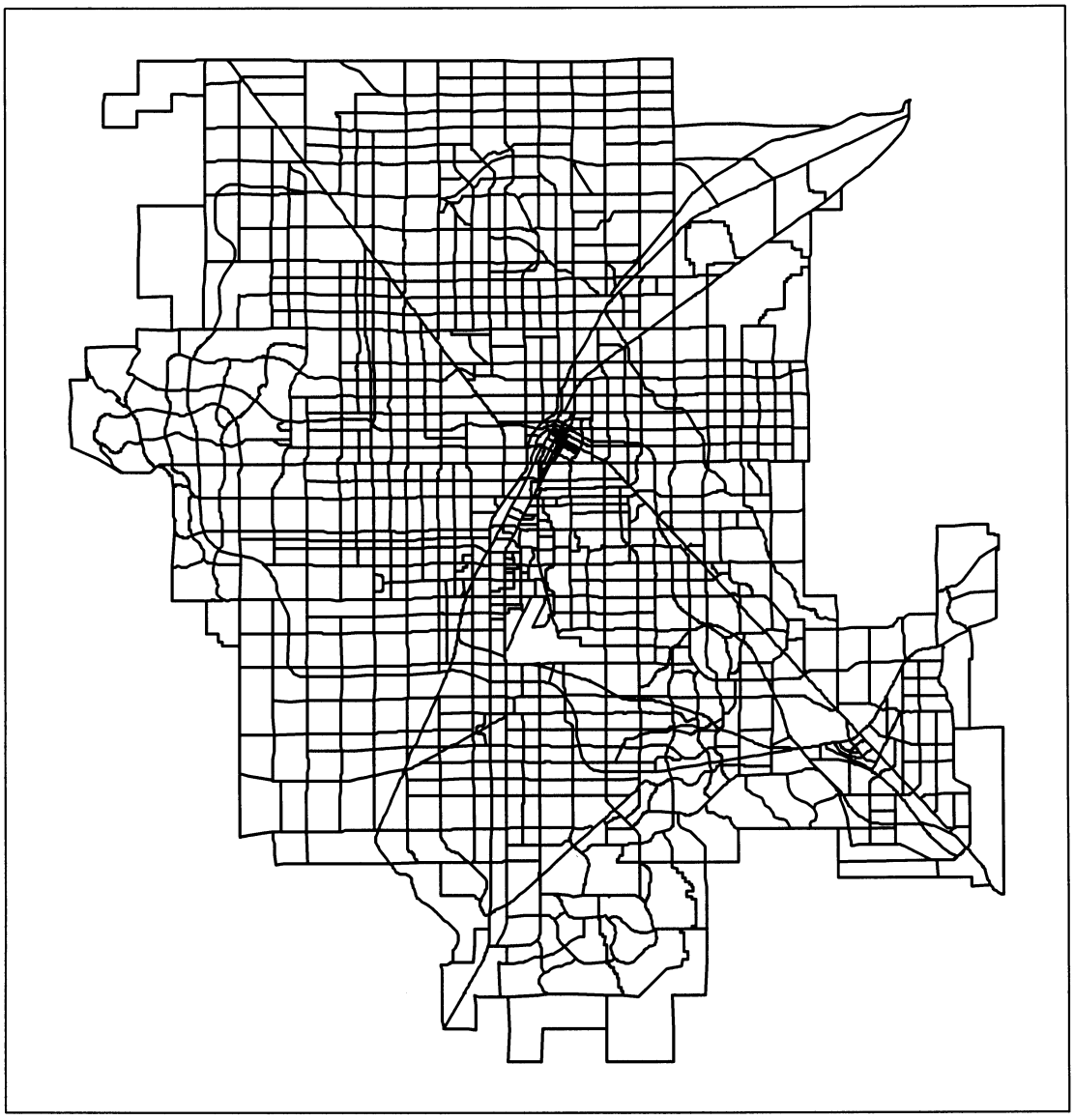


Figure-17.9: Traffic Analysis Zones in metropolitan Las Vegas

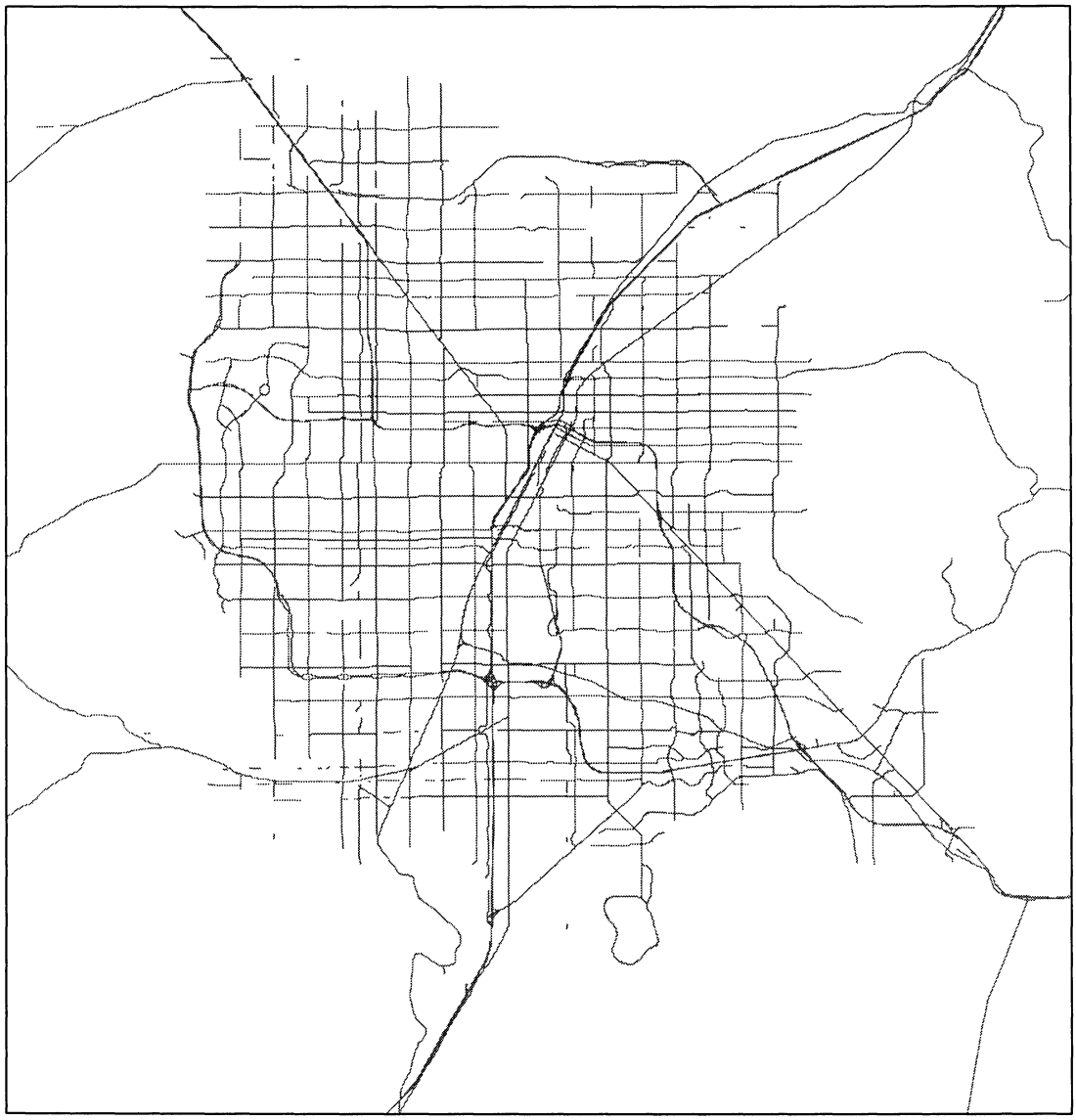


Figure-17.10: Las Vegas major street centerline network

Figure 17.11: Crime Trip Origins (All Confrontational Crimes, 1999 – 2001)

Figure 17.12: Crime Trip Destinations (All Confrontational Crimes, 1999 – 2001)

neighborhood, looking for targets for potential victims. If a promising target is observed, he may then commit an attack, then drive back toward his home area, stopping off for gas or at a drive-thru restaurant on the way, before parking at his house. Although this round-trip from home to home consists of multiple destinations, some of which are repeated throughout the day, the whole journey is considered to be a single "Crime Trip".

In some cases, a single offender was responsible for many crimes. When this happens, the single origin is paired with multiple destinations, resulting in separate Crime Trips. In other cases, one crime may be perpetrated by multiple offenders. When this happens, each offender's origin is paired with the single destination, again resulting in separate Crime Trips.

While it is possible to distinctly model each Crime Trip based on precise spatial locations, it is generally accepted to aggregate both origins and destinations to the centroid of each Traffic Analysis Zone. This enables the spatial assignment of TAZ variables such as income and population to the aggregate frequencies of both origins and destinations.

This assignment is performed in CrimeStat by centroid allocation - the nearest TAZ centroid is used to assign the TAZ data to each origin and destination. This method is faster and simpler than "point-in-polygon" spatial aggregation and assignment, but should result in comparatively few mistaken assignments due to unusual TAZ polygon shape or distribution. Since Crime Trip data is aggregated to the zonal level, therefore, the resulting analyses and forecasts are only applicable to this level and cannot meaningfully disaggregated to a more refined resolution.

The accepted travel-demand model framework contains a built-in "error factor" for external trips - that is, crime trips originating within the study area but having destinations falling outside the area, or, conversely, originating outside the study area but having internal destinations. These "external trips" were culled from the crime database during the data screening process; therefore, "External Zone" data is inapplicable to the trip generation stage of the analysis.

Trip Generation

Each origin/destination pair having been aggregated to the TAZ polygon layer, it is now possible to evaluate the relationship between socio-economic variables available in the TAZ database with the frequency of crime origins and destinations. This is accomplished through regression modeling, and may prove one of the most useful single features in the new modeling capabilities of the *CrimeStat* application.

There are two main regression options available in the software at present: Ordinary Least Squares (OLS) and Poisson. The Poisson estimation also includes a separate option which allows backward elimination of variables. This option, Poisson Regression with Backward Elimination, was the most effective of the techniques evaluated, resulting in consistently better visual fits to the data and lower residuals. This very useful

step examines each variable element suggested by the analyst for its predictive value as a coefficient in estimating the frequency of either origins or destinations by TAZ.

In every case, three variables within the TAZ database for Las Vegas proved consistently useful as predictive measures:

1. Income,
2. Population, and
3. Total Employment.

The measurable successfulness of these variables to account for the predictable distribution of both origins and destinations was somewhat counter-intuitive; it was suspected prior to the application of this model that other variables (in particular the number of pawn shops, the number of Strip employment opportunities, and the number of Nellis AFB employment opportunities) would be critical predictors of crime. In fact, however, all of these variables demonstrated strong multicolinearity with the three primary variables listed above. When these other, extraneous factors were excluded from the regression process, the effectiveness of the model's predictive capabilities was substantially improved.

A suggested and accepted travel-demand modeling techniques widely implemented by transportation planners is the adoption of "special generator" variables to explain unusual or unique factors implicit in some areas. It was expected that Nellis AFB, the Las Vegas Strip itself, and some other seemingly significant factors would likely fill the role of "special generator;" however, results indicated that none of these were as effective in a predictive or explanatory role as Income, Population, and Total Employment.

Latitudinal forecasting of crime trip origins and destinations performed fairly well; comparison of expected versus observed trip numbers did not match particularly well, but the relative distribution by TAZ was a very close match (figures 17.13-17.16).

Longitudinal forecasting of crime trip frequency by data from one year to the next year performed very poorly; this is probably an artifact of the still-unexplained drastic variation in frequency between the three years considered in this study. Results from other years, or other jurisdictions, may exemplify very different findings.

Side-by-side comparison of observed and predicted crime trip origins reveals some persuasive similarities, but significant discrepancies, also (figures 17.17 and 17.18). In general, relative proportions are very accurately described, but smaller-producing zones are somewhat underestimated (the model seems to perform better on zones with higher productions).

Side-by-side comparison of observed versus predicted crime trip destinations suggests that, proportionally, the model again performs very well, particularly on zones with higher production scores. Zones with very weak crime trip destination productions (of one or two crimes) are not as accurately depicted.

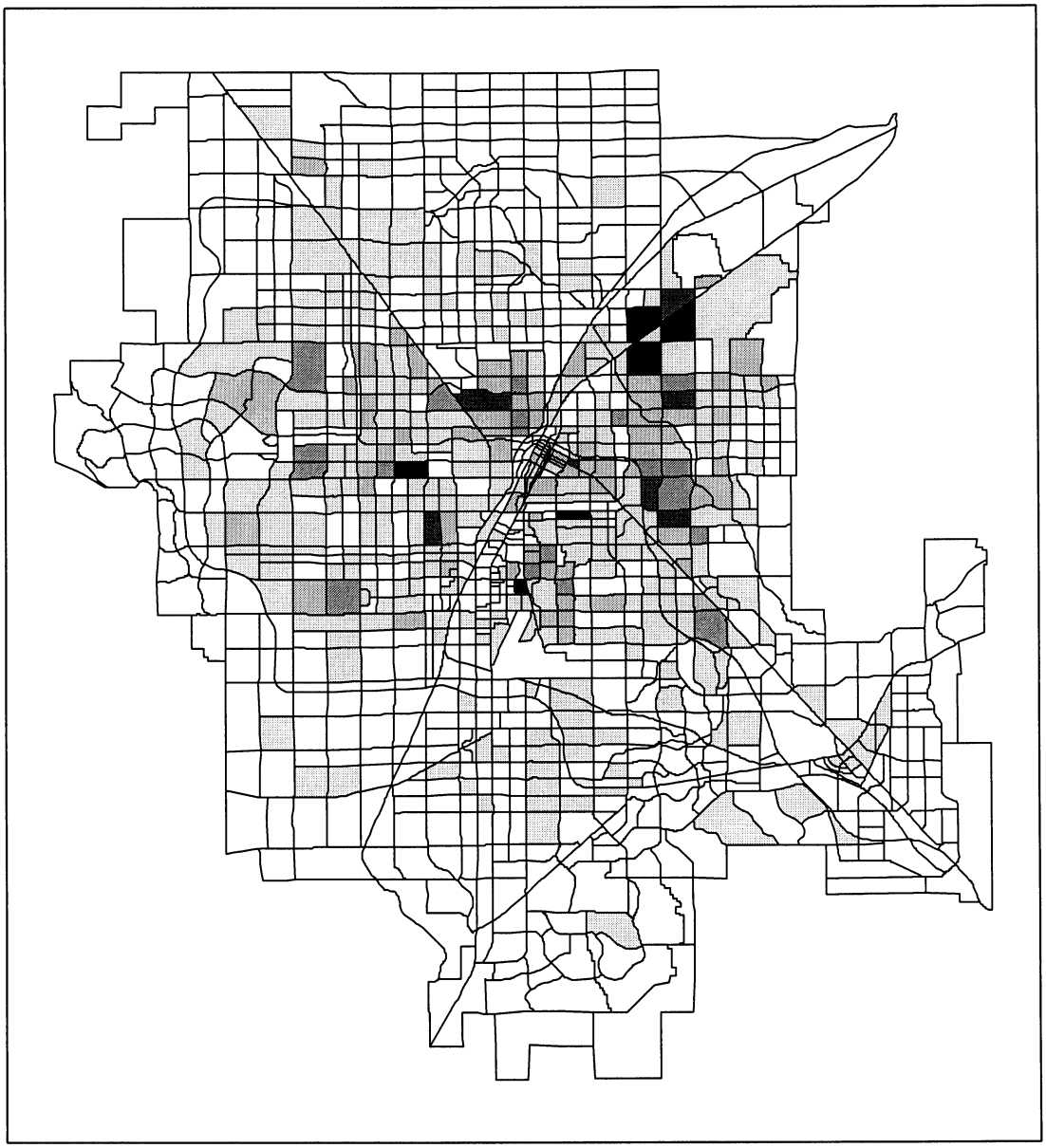


Figure 17.13: Relative equal-interval frequency distribution of actual crime trip origins

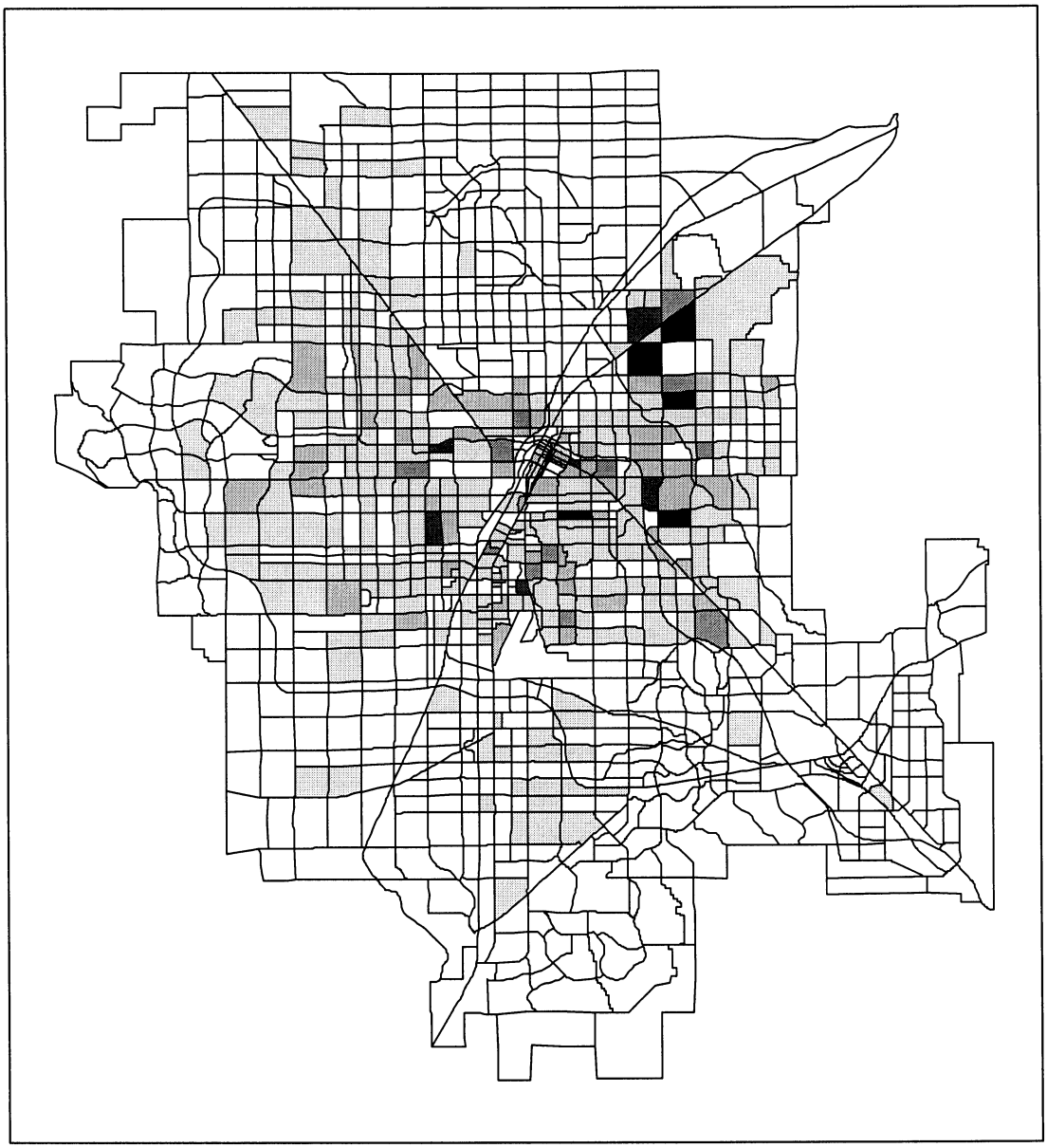


Figure 17.14: Relative equal-interval frequency distribution of actual crime trip destinations

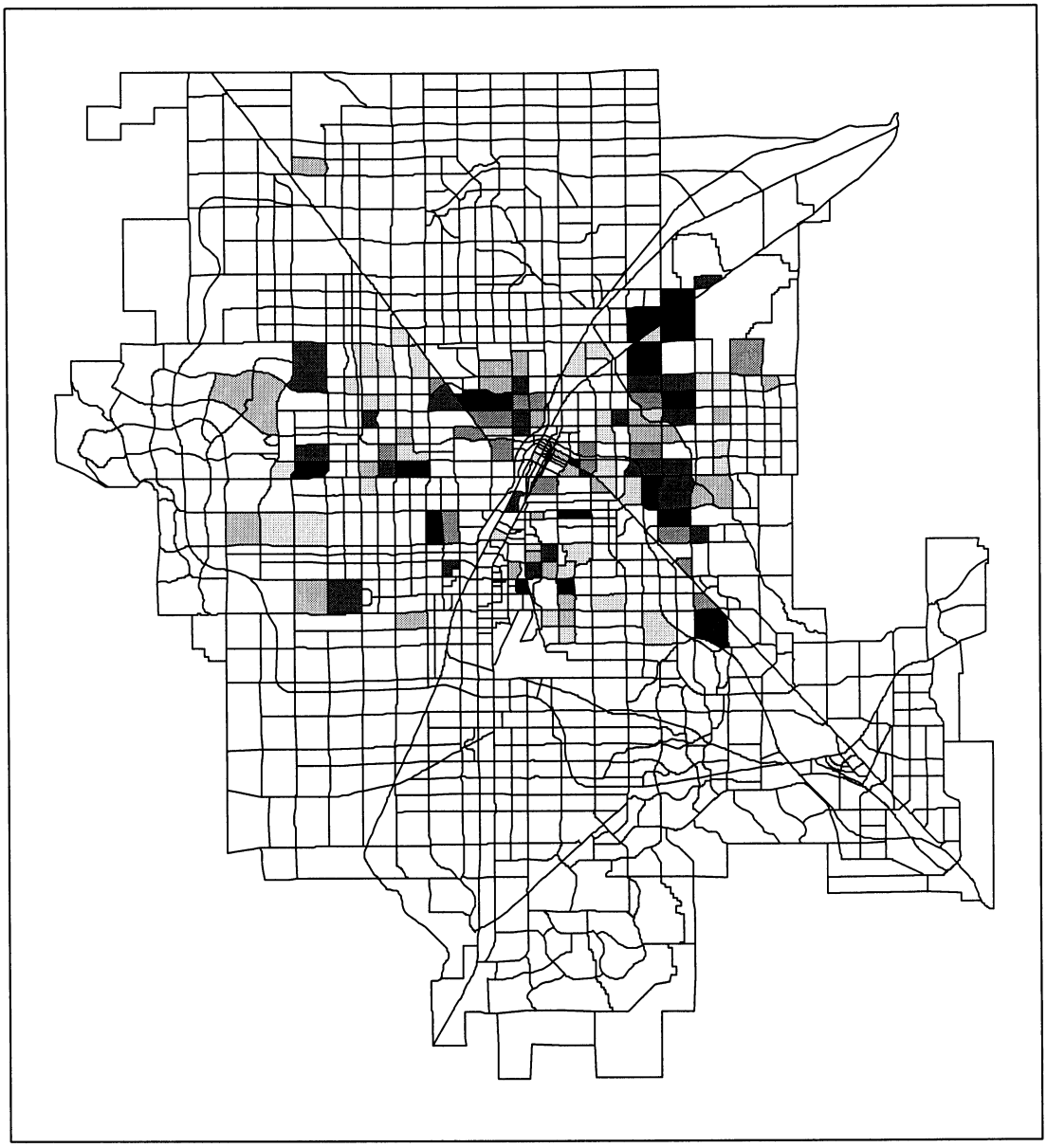


Figure 17.15: Relative equal-interval frequency distribution of predicted crime trip origins

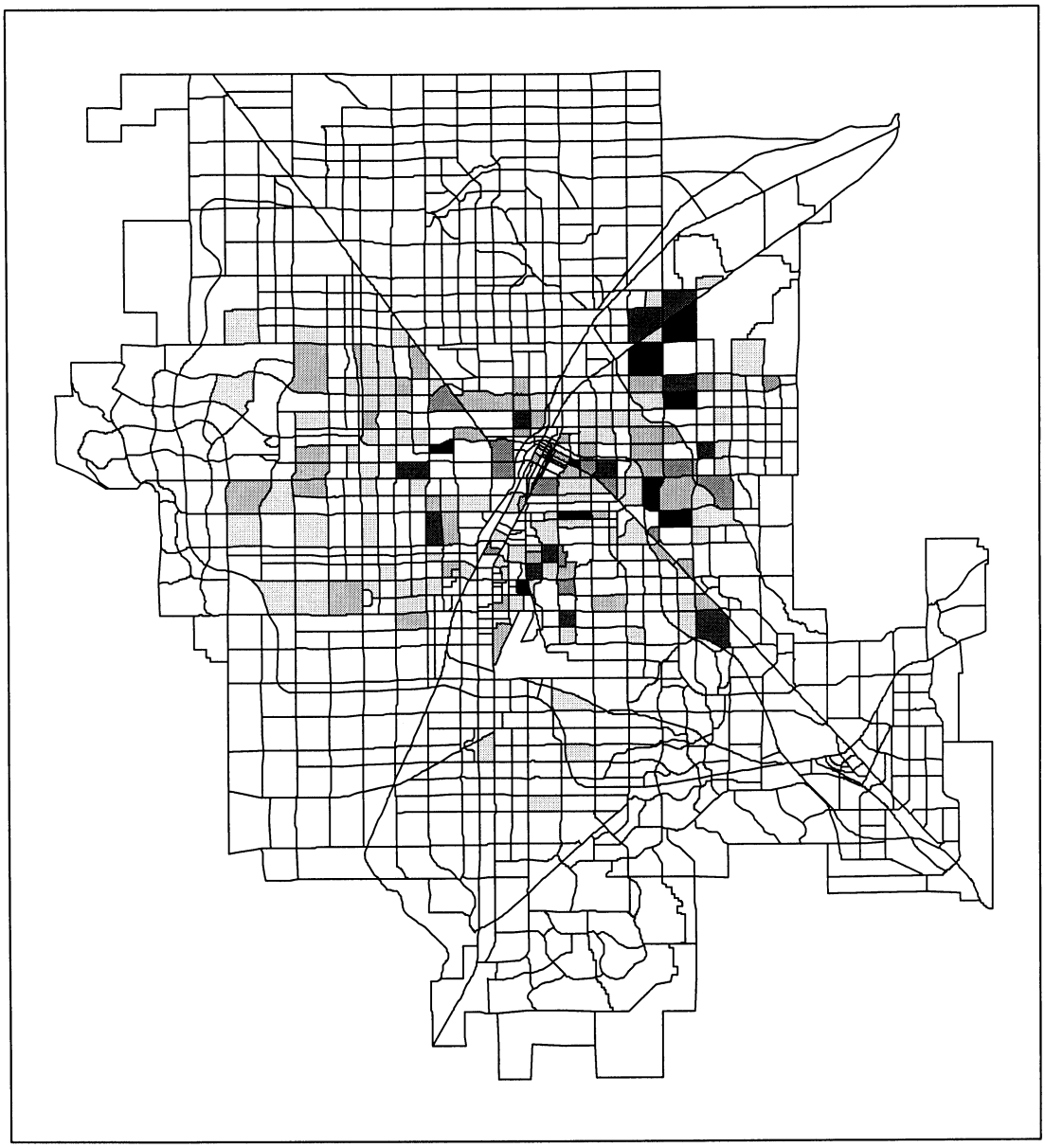
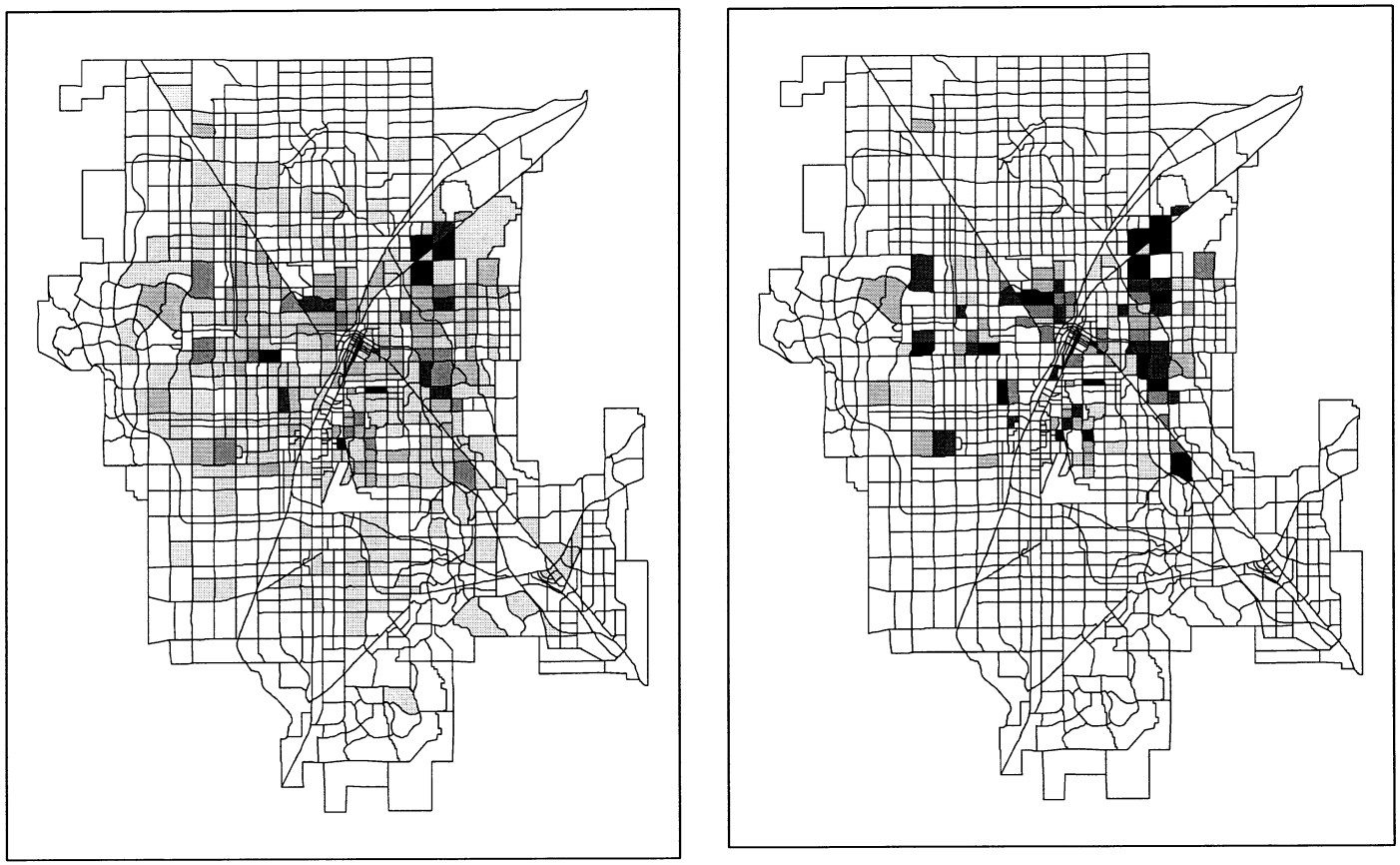


Figure 17.16: Relative equal-interval frequency distribution of predicted crime trip destinations



Figures 17.17: Comparison of observed (left) and predicted (right) origins

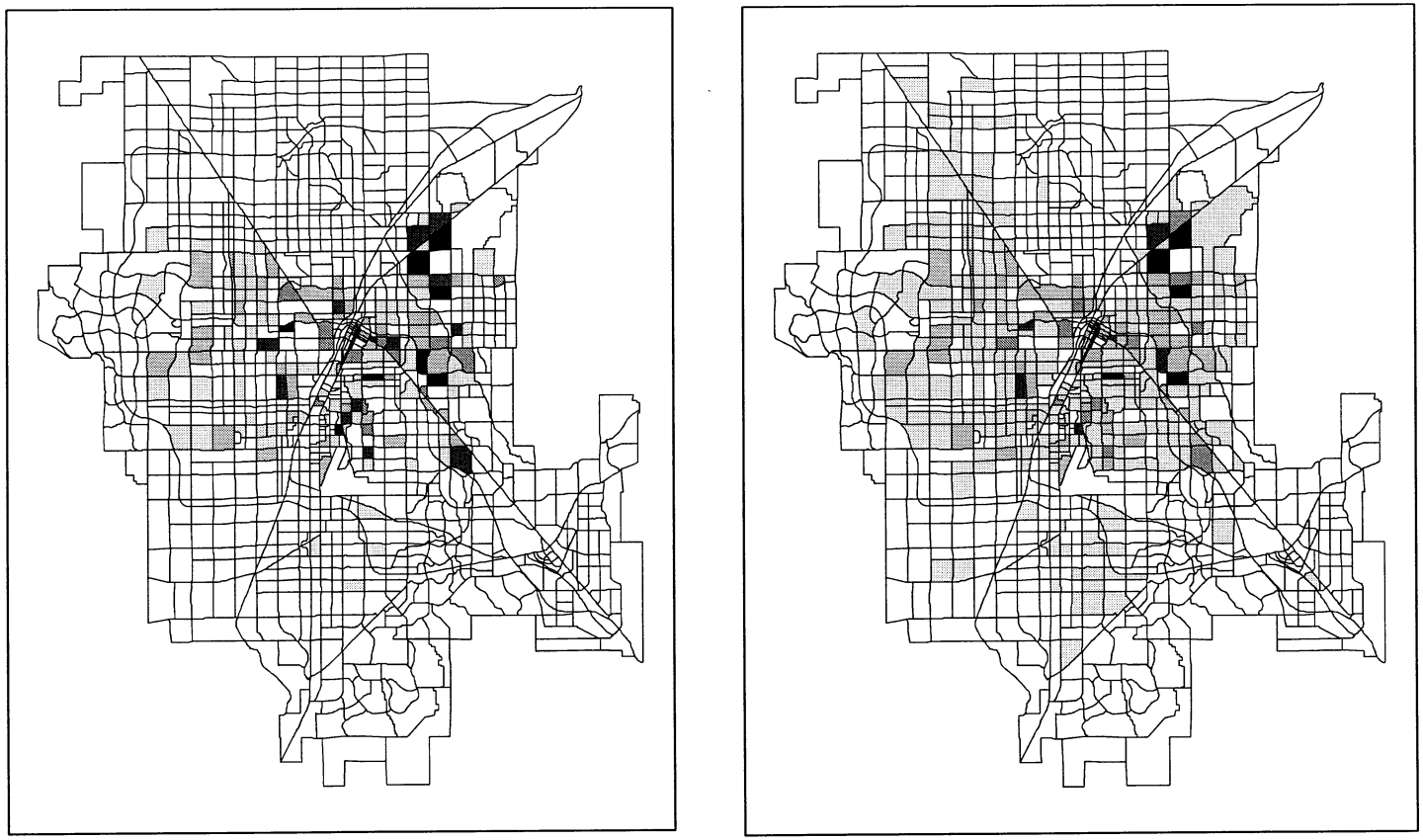


Figure 17.18: Comparison of observed (left) and predicted (right) destinations

Trip Distribution

Assignment of trip links between TAZ polygons performed very well (figure 17.19). Originally, some concern was felt that the assignment of crime events to TAZ centroids (rather than using the actual crime scene and home address coordinates) might result in significant distortion; however, this does not appear to have occurred. Compare the raw (actual) crime trip lines with the centroid-corrected trip lines to see how neatly they match (figure 17.20). The resulting distance decay and impedance functions perform perfectly well. There are almost no discrepancies visible to the naked eye.

Various impedance function calculations were attempted in the course of this study. Eventually, an adaptive (100-bin) normal interpolation with 100 minimum samples was selected as the best fit. However, a negative exponential impedance function also fit well, similar to the Baltimore County and Chicago models.

Intra-zonal crime trips - those having both origin and destination within the same TAZ - cannot be displayed as lines, since they have no length. Instead, they can be represented by points (figure 17.21). Inter-zonal crime trips, on the other hand, are better displayed by lines (figure 17.22).

Intra-zonal crime trips accounted for 42% of all crime trips overall, but only 12% of robberies, indicating a much longer "hunting range" for robbers; this may be in keeping with the hypothesis that the tourist corridors draw robbery crime trips as destinations which originate in other neighborhoods. More than 50% of sexual assaults were intra-zonal, indicating a shorter-than-usual hunting range for sexual attackers, who seem to prefer striking in their home neighborhoods.

Mode Split

Unfortunately, the mode split portion of the travel-demand model is the weakest element for the Las Vegas data.

Transportation modes across metropolitan Las Vegas are varied. Typical of a western city, the overwhelming majority of residents rely on private automobiles for transportation, as do many tourist visitors. However, this mainstay is supplemented by a robust bus system, as well as alternate personal transportation for short trips (i.e., walking, bicycling, or scooters). The picture of automobile transportation is somewhat muddled by the higher than usual dependency on taxi-cabs and limousines for transportation by out-of-state visitors.

Data provided by the LVMPD included a field called "Method of Departure" which was intended to contain information about how the offender departed the scene of the crime, which in turn would have been an effective way of calculating probable mode split for crime trips sampled. Unfortunately, this data field was blank in the overwhelming

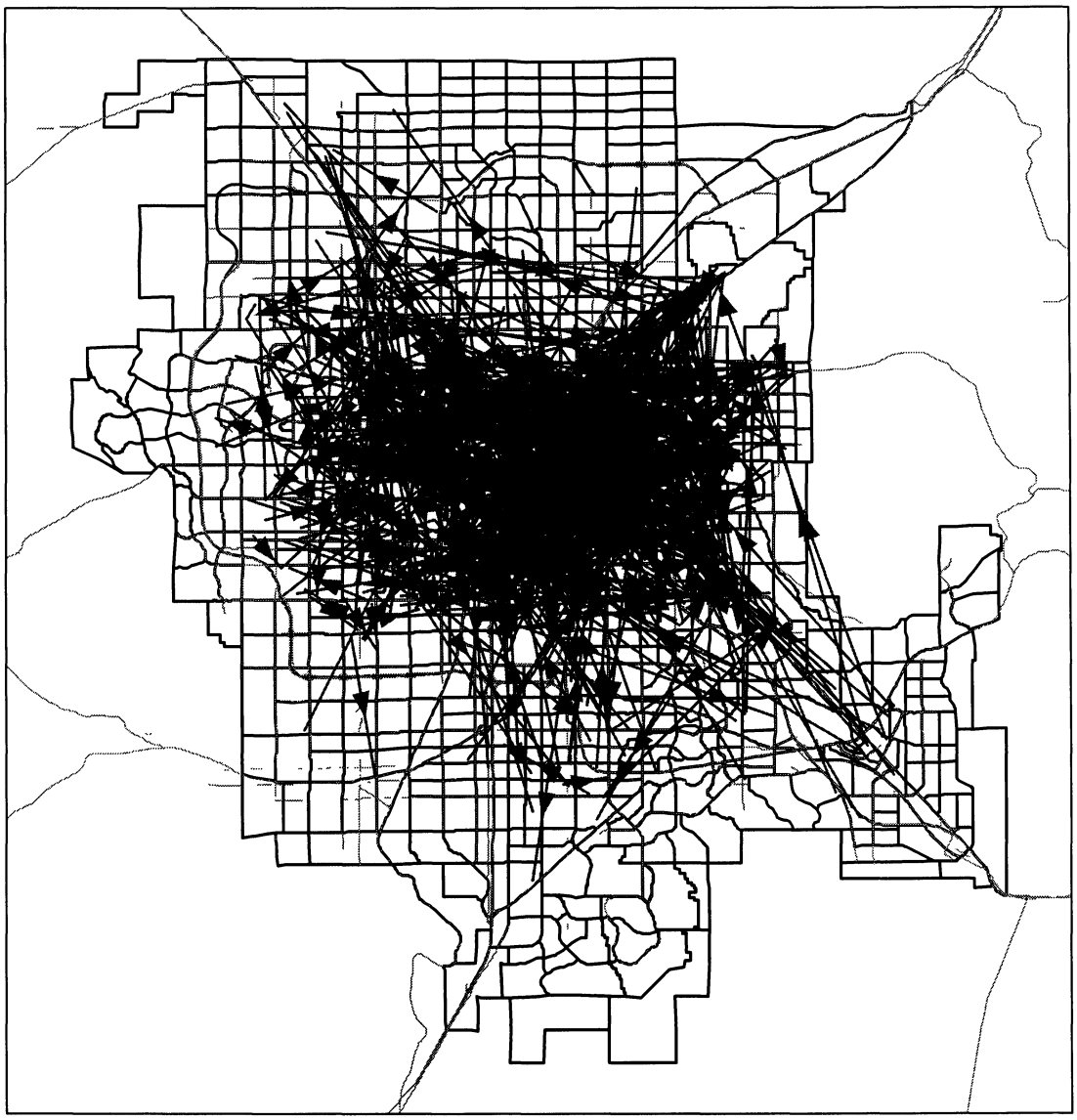


Figure 17.19: Raw Crime Trip Links

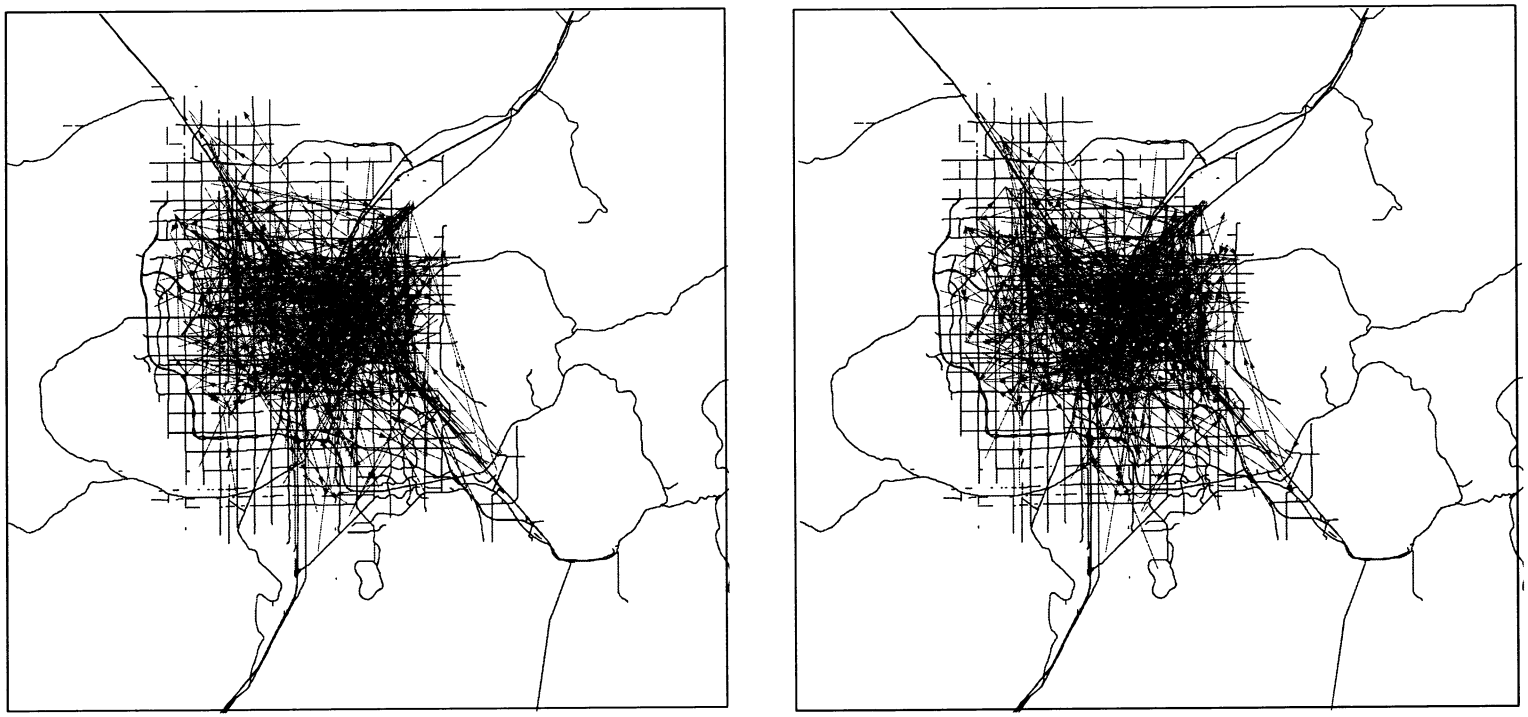


Figure 17.20: Actual (left) and Predicted (right) TAZ-centric crime trip links

Figure 17.21: Predicted Intra-zonal crime trips

Figure 17.22: Top 100 predicted inter-zonal crime trips

majority of cases (approximately 4% contained entries, and only 75% of these - 3% overall - contained apparently valid data).

Therefore, any empirical estimation of mode split for these data requires inference from other data. For example, auto theft crimes may safely be assumed to use a car to provide transportation for at least some portion of the crime trip. In other cases, the plain-text narrative includes vehicle descriptions or statements about how the offender moved that were not distilled into the correct field. Unfortunately, the large volume of cases makes recovering information from these free narratives impractical for the small number of cases in which mode split information can beneficially be derived.

Due to this lack of reliable data, only two mode split options were included in this analysis: Walking and Driving. Default impedance functions proved very acceptable for both modes: Inverse Exponential for walking trips and Lognormal for driving.

Network Assignment

The complete street centerline (SCL) file for the metropolitan Las Vegas area was available in a routable format (topologically rectified ESRI Shapefile); however, this file proved prohibitively large and unwieldy for the A* shortest-path/least-cost algorithm implemented in *CrimeStat*. Instead of the complete SCL data layer, a layer consisting only of arterial streets and freeways was used instead. This major roads file proved adequate to neatly explaining the probable transportation path choices made by the top 100 and top 300 inter-zonal crime trips (figures 17.23 and 17.24).

In general, the visual goodness-of-fit for predicted crime trips improved as the category of crime was narrowed. Predictions from one year to the next remained weak, probably as a result of the as-yet-unexplained radical variance in crime frequencies across all studies categories; however, within discrete crime categories predictive capabilities were sometimes visually impressive.

Modeling Auto Theft Site to Recovery Site

In the case of auto thefts, an attempt was made to isolate the movement from vehicle theft site to vehicle recovery site, rather than use the theft site and offender home location as the destination and origin, respectively, of the crime trip. It was hoped that this variation of the travel-demand model for crime trip analysis might prove more useful for this type of data than home-based crime trips, partly because more accurate location information was available for recovery sites than for home locations, as well as because it was hypothesized that the theft/recovery "trip" segment might prove more representative than the home/theft trip.

Results for auto thefts appeared weak, with predicted crime trips much longer than the observed (figure 17.25). While the observed trips focused tightly on the central core areas and densely-populated residential zones, the predicted trips seemed to skirt the edges of the metropolitan area. This is possibly due to an implied over-emphasis on

Figure 17.23: Top 100 Inter-zonal Crime Trips as allocated to Major Streets Network

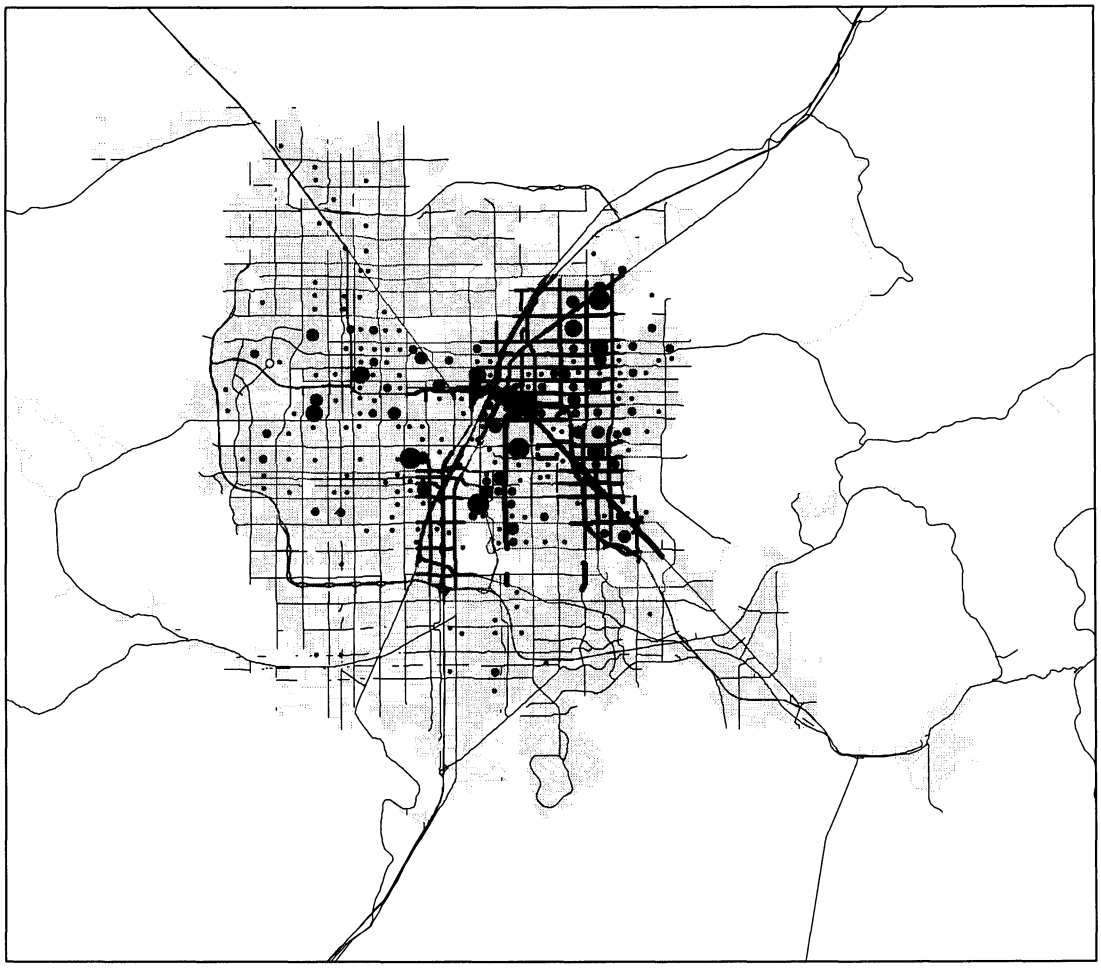


Figure 17.24: Top 300 Crime Trips (Intra- and Inter-zonal) as allocated to Major Streets Network

Figure 17.25: Observed (left) and Predicted (right) Top 100 Auto Theft Crime Trips as Allocated to Major Street Network

freeway travel which may be correctible with better network-allocation parameters. The median distance for observed crime trips was 2.3 miles.

Residential Burglaries

Differentiation of residential from commercial or auto burglaries was accomplished by three filtering criteria: Statute, Premise, and Zoning. Some specific Nevada Revised Statutes have been reserved for residential burglaries; burglaries in which these statues were cited were therefore accepted as residential in nature. Categorical Premise type data was provided in the MO data for each crime; when this data explicitly noted a residential site, these cases were also accepted as residential.

Some burglaries didn't specifically include a residential statue or explicitly residential premise code; but were spatially located in areas of the jurisdiction reserved for residential rather than commercial, industrial, or other zoning purposes. These cases were therefore also accepted as residential in character.

Results for analysis of residential burglaries was more promising than for auto thefts, or for burglaries overall (figure 17.26). While, again, observed crime trips focused on the most densely-populated residential neighborhoods, and predicted crime trips were much longer and spread more far afield, this spread was much smaller than that seen in auto thefts and more closely conformed to the observed distribution. The median distance for residential burglary crime trips was 1.1 miles.

Sexual Assaults

The spatial distribution of sexual assault crime trips in many ways seemed to invert the problems seen in the predicted crime trips for auto thefts and residential burglaries. In the previous examples, an observed tendency toward centrality seemed to be confused with a predicted tendency toward dispersion toward outlying areas. In this case, however, a very nebulous, outlying distribution of observed crime trips (centering in three faint clusters around the perimeter of the central metropolitan region) was observed. The predicted crime trip distribution mistakenly emphasized central areas, and seemed to completely fail to predict the southeastern-most "cluster" of crime trips (figure 17.27).

The large median crime trip length for sexual assaults - 3.2 miles - may help explain the relatively poor predictiveness of these results. Different impedance functions will probably help improve the reliability of this model against these types of crimes.

Robberies

Robbery crime trips in Las Vegas appear to closely parallel the major gaming and transportation corridors running north to south through the center of the metropolitan area (figure 17.28). The visual fit of predicted against observed crime trips was most impressive against these cases. Although the predicted crime trip distribution appears more compact and centralized than the observed, the directionality and polarity of the two

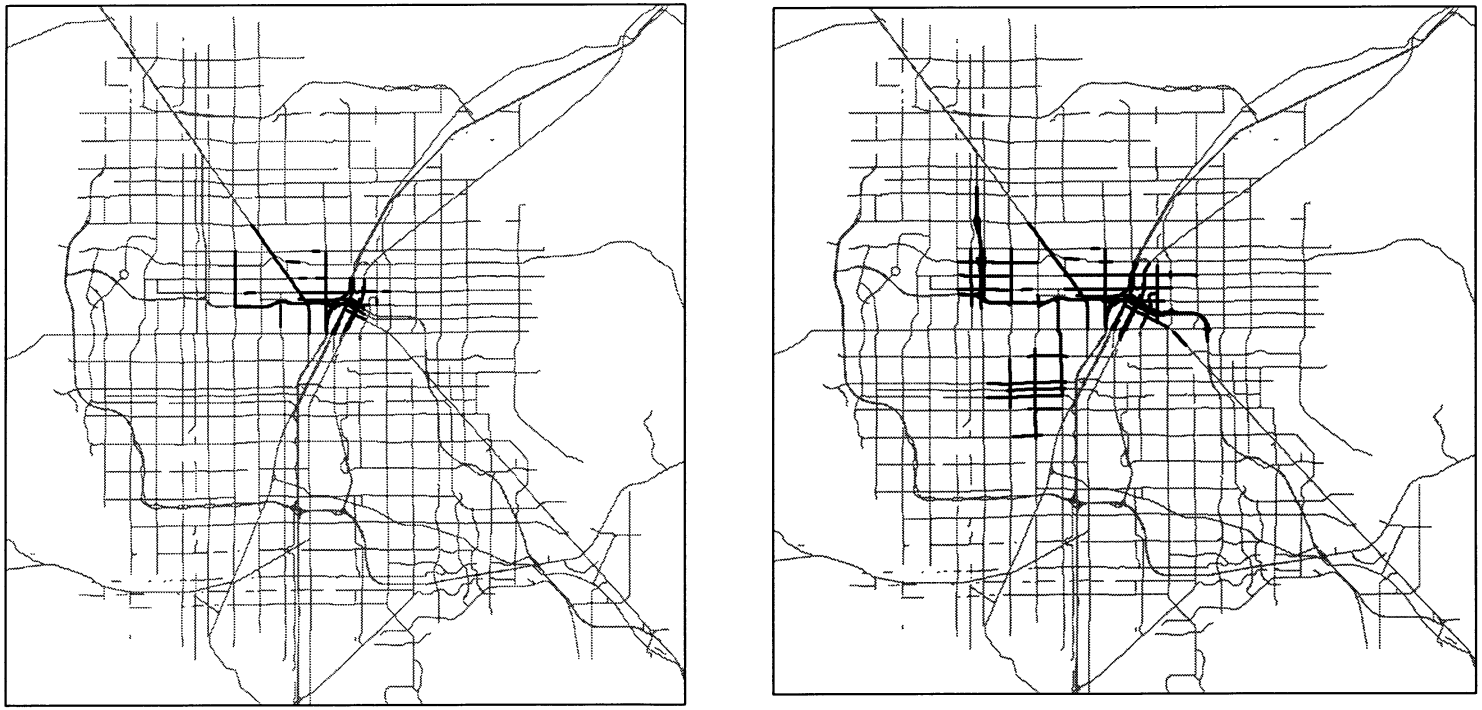


Figure 17.26: Observed (left) and Predicted (right) Top 100 Residential Burglary Crime Trips as Allocated to Major Street Network

Figure 17.27: Observed (left) and Predicted (right) Top 100 Sexual Assault Crime Trips as Allocated to Major Street Network

Figure 17.28: Observed (left) and Predicted (right) Top 100 Robbery Crime Trips as Allocated to Major Street Network

parallel nicely, and make a striking visual match. The median crime trip distance for robberies was 2.3 miles.

Conclusions: Las Vegas

Overall, the model appeared to perform well in some ways, but weaker in others. One of the most troubling problems facing the evaluation of the network assignment stage of the model is the lack of any good final metric other than visual approximation for determining the value of the resulting prediction. Some measurement of congruence is needed to make the determination of usefulness reliable and valid.

The first stage of the model - crime trip generation, is arguably the most useful to law enforcement. This elegantly simple model can readily be adapted to different types of data, and with the forthcoming inclusion of additional regression methods (specifically the negative binomial distribution model) to supplement the existing ordinary least squares (OLS) and Poisson variants, this feature is likely to remain useful for the foreseeable future.

The second stage of the model - crime trip distribution, is also potentially highly useful. The analysis not merely of where offenders live, or where crimes are committed, but of the travel and transportation decisions linking the two locations, may have significant repercussions for crime analysts. This type of analysis will be particularly useful for strategic and administrative analysts when recommending manpower allocation, beat boundaries and precinct/district configuration schemes, and assessing the impact of major developments such as transportation corridors, shopping malls, or sports complexes on the distribution of crime.

The mode split stage of the model was difficult to apply meaningfully to the Las Vegas data in this study, because of deficiencies in the data itself. Either transportation choice values were not recorded, or were recorded in irretrievable formats, making an empirical evaluation of offenders' transportation choice proclivities impractical. Failing the availability of empirical data, falling back on overall trends in public transportation choice are all that is possible for the analyst. Since it is possible that crime trips may be qualitatively different than other types of trips on which these statistical models have been based, further research is required to assess whether or not these standards will be applicable to criminal behavior.

The final stage of the model - network assignment, functioned mechanically as expected, but did result in some potentially weak results (such as overemphasis on the speed of freeways apparent in some results) which may be overcome with better mode split and network choice parameters.

One aspect of the model that caused for initial concern, the aggregation of crimes to the Traffic Analysis Zone polygon level, proved to have no significant impact on the resulting analysis. The TAZ structure seems admirably suited to analysis of this sort of movement - as indeed one might expect from its provenance.

The most successful predictive variables for estimating crime trip production, whether of origins or destinations, were infallibly Total Population, Total Employment, and Income. Inclusion of additional variables distorted rather than improved the predictive value of the model, most of the time with measurable multicolinearity which was not always apparent a priori.

With the mechanical aspects of the model - as implemented in the latest version of *CrimeStat*, complete and functioning correctly, it remains to be learned how to better calibrate and implement the model to make it an effective tool for law enforcement analysis and planning.

References Used in *CrimeStat* Manual

Aldenderfer, M. and R. Blashfield (1984). *Cluster Analysis*. Sage: Beverly Hills, CA.

Amir, Menachim (1971). *Patterns in Forcible Rape*. The University of Chicago Press: Chicago, 87-95.

Andersson, T. (1897). *Den Inre Omflyttningen*. Norrland: Mälmo.

Anselin, Luc (1995). "Local indicators of spatial association - LISA". *Geographical Analysis*. 27, No. 2 (April), 93-115.

Anselin, Luc. (1992). *SpaceStat: A Program for the Statistical Analysis of Spatial Data*. Santa Barbara, CA: National Center for Geographic Information and Analysis, University of California.

Anselin, Luc and Moss Madden (1990). *New Directions in Regional Analysis*. Belhaven Press: New York.

Aplin, Graeme (1983). *Order-Neighbour Analysis*. Concepts and Techniques in Modern Geography No. 36. Institute of British Geographers, Norwich, England: Geo Books.

Bachi, R. (1957). *Statistical Analysis of Geographical Series*. Central Bureau of Statistics, Kaplan School, Hebrew University: Jerusalem.

Bailey, Trevor C. and Anthony C. Gatrell (1995). *Interactive Spatial Data Analysis*. Longman Scientific & Technical: Burnt Mill, Essex, England.

Barber, C.; Dobkin, D.; and Huhdanpaa, H. (1997). "The Quickhull algorithm for convex hulls." *ACM Trans. Mathematical Software,* 22, 469-483.

Ball, G. H. and D. J. Hall (1970). "A clustering technique for summarizing multivariate data". *Behavioral Science*, 12, 153-155.

Barnard, G. A. (1963). "Comment on 'The Spectral Analysis of Point Processes' by M. S. Bartlett", *Journal of the Royal Statistical Society*, Series B, 25, 294.

Beale, E. M. L. (1969). *Cluster Analysis*. Scientific Control Systems: London.

Beimborn, Edward A. (1995). "A transportation modeling primer". In *Inside the Blackbox, Making Transportation Models Work for Livable Communities*. Http://www.uwm.edu/Dept/CUTS/primer.thm.

Ben-Akiva, Moshe and Steven Lerman (1985). *Discrete Choice Analysis: Theory and Application to Travel Demand*. MIT Press: Cambridge.

Bernasco, Wim and Floor Luykx (2002). "Using random utility models to explain location choice of offenders". Sixth Annual International Crime Mapping Research Conference, December, Denver CO.

Besag, Julian and James Newell (1991). "The detection of clusters in rare diseases". Journal of the Royal Statistic Society A, 154, Part I, 143-155.

Betlyon, Brian and Michael Culp (2001). *Overview of Travel Demand Forecasting.* Presentation. Federal Highway Adminstration, U.S. Department of Transportation: Washington, DC. http://tmip.fhwa.dot.gov/conf_courses/presentations/fmt_traveldemand/traveldemand_files/v3_document.htm.

Bezdek, J. C. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms.* Plenum Press: New York.

Block, Carolyn R. (1994). "STAC hot spot areas: a statistical tool for law enforcement decisions". In *Proceedings of the Workshop on Crime Analysis Through Computer Mapping.* Criminal Justice Information Authority: Chicago, IL.

Block, Richard and Carolyn Rebecca Block (1999) "Risky places: a comparison of the environs of rapid transit stations in Chicago and the Bronx" in John Mollenkopf (ed), *Analyzing Crime Patterns: Frontiers of Practice*, Sage Publishing: Beverly Hills, CA.

Block, Richard and Carolyn Rebecca Block (1995). "Space, place and cime: hot spot areas and hot places of liquor-related Crime" in John E. Eck and David Weisburd (eds.), *Crime and Place.* Crime Prevention Studies, Volume 4. Criminal Justice Press: Monsey, NY. 147-185.

Block, Carolyn R. and Lynn A. Green (1994). *The GeoArchive Handbook: A Guide for Developing a Geographic Database an Information Foundation for Community Policing.* Illinois Criminal Justice Information Authority: Chicago, IL.

Blumin, D. (1973). *Victims: A Study of Crime in a Boston Housing Project.* City of Boston, Mayor's Safe Street Act, Advisory Committee: Boston.

Boggs, S. L. (1965). "Urban crime patterns", *American Sociological Review*, 30, 899-908.

Borland.Com (1998). *dBase IV 2.0.* Inprise Corporation: Scotts Valley, CA.

Bossard, Earl G. (1993). "RETAIL: Retail trade spatial interaction". In Richard E. Klosterman, Richard K. Brail and Earl G. Bossard, *Spreadsheet Models for Urban and Regional Analysis.* Center for Urban Policy Research, Rutgers University: New Brunswick, NJ, 419-448.

Bowers, K. and A. Hirschfield (1999). "Exploring links between crime and disadvantage in North-West England: An analysis using Geographic Information Systems". *International Journal of Geogrpahical Information Science*, 13, 159-184.

Bowman, Adrian W. and Adelchi Azzalini (1997). *Applied Smoothing Techniques for Data Analysis: The Kernel Approach with S-Plus Illustrations*. Oxford Science Publications, Oxford University Press: Oxford, England.

Brantingham, P. and P. Brantingham (1999). "A Theoretical Model of Crime Hot Spot Generation", *Studies on Crime and Crime Prevention*, 8 (1), 7-26.

Brantingham, Paul and Patricia J. Brantingham (1984). *Patterns in Crime*. Macmillan Publishing: New York.

Brantingham, Patricia L. and Paul J. Brantingham (1981). "Notes on the geometry of crime". In Paul J. Brantingham and Patricia L. Brantingham, *Environmental Criminology*. Waveland Press, Inc.: Prospect Heights, IL, 27-54.

Bright, M. L. and D. S. Thomas (1941). "Interstate migration and intervening opportunities", *American Sociological Review*, 6, 773-783.

BTS (2003). "U.S. Vehicle-miles (millions)". Bureau of Transportation Statistics, U.S. Department of Transportation: Washington, DC. http://www.bts.dot.gov/publications/national_transportation_statistics/2003/html/table_01_32.html.

BTS (2002). *National Household Travel Survey: Daily Travel Quick Facts*. Bureau of Transportation Statistics, U.S. Department of Transportation: Washington, DC. http://www.bts.gov/publications/national_household_travel_survey/quick_sheets/daily_travel.html.

Burgess, Ernest W. (1925). "The growth of the city: an introduction to a research project". In R. E. Park, E. W. Burgess and R. D. Mackensie (ed), *The City*. University of Chicago Press: Chicago, 47-62.

Bursik, R. J., Jr. and H. G. Grasmick (1993). "Economic deprivation and neighborhood crime rates, 1960-1980". *Law and Society Review*, 27, 263-268.

Burt, James E. and Gerald M. Barber (1996). *Elementary Statistics for Geographers* (second edition). The Guilford Press: New York.

Cameron, A. Colin and Pravin K. Trivedi (1998). *Regression Analysis of Count Data*. Cambridge University Press: Cambridge, U.K.

Çan, Ayşe and Issac Megbolugbe (1996). "The geography of underserved mortgage markets". Paper presented at the American Real Estate and Urban Economics Association meeting. May.

Canter, David (2003). *Dragnet: A Geographical Prioritisation Package*. Center for Investigative Psychology, Department of Psychology, The University of Liverpool: Liverpool, UK. http://www.i-psy.com/publications/publications_dragnet.php.

Canter, D. (1994). *Criminal Shadows: Inside the Mind of the Serial Killer*. Harper Collins Publishers: London.

Canter, D. (1999). "Modelling the home location of serial offenders". Paper presented at the 3rd Annual International Crime Mapping Research Conference, Orlando, December.

Canter, D.V, Coffey, T., Huntley, M., & Missen, C. (2000). Predicting serial killers' home base using a decision support system. Journal of Quantitative Criminology, 16, 457-478.

Canter, D. and S. Tagg (1975). "Distance estimation in cities", *Environment and Behaviour*, 7, 59-80.

Canter, D. and P. Larkin (1993). "The environmental range of serial rapists", *Journal of Environmental Psychology*, 13, 63-69.

Canter, D. and A. Gregory (1994). "Identifying the residential location of rapists", *Journal of the Forensic Science Society*, 34 (3), 169-175.

Capone, D. L. and W. W. Nichols Jr. (1975). "Crime and distance: an analysis of offender behaviour in space", *Proceedings, Association of American Geographers*, 45-49.

Carnegie-Mellon University (1975). *Security of Patrons on Urban Public Transportation Systems*. Transportation Research Institute, Carnegie-Mellon University: Pittsburgh, PA.

Chaitin, Gregory (1990). *Information, Randomness and Incompleteness* (second edition). World Scientific: Singapore.

Chand, D and S. Kapur (1970). "An algorithm for convex polytopes". *J. ACM*, 17, 78-86.

Chen, A. & Renshaw, E. (1994) The general correlated random walk. *Journal of Applied Probability*, 31, 869-884.

Chen, A. & Renshaw, E. (1992). "The Gillis-Domb-Fisher correlated random walk." *Journal of Applied Probability*, 29, 792-813.

Chiricos, T. (1987). ""Rates of Crime and Unemployment" *Social Problems*, 34, 187-211

Chrisman, Nicholas (1997) *Exploring Geographic Information Systems*. John Wiley and Sons, Inc.: New York.

Citro, Constance F. and Robert T. Michael (eds) (1995). *Measuring Poverty : A New Approach*. Panel on Poverty and Family Assistance, Committee on National Statistics, Commission on Behavioral and Social Sciences and Education, National Research Council: Washington, DC.http://www.census.gov/hhes/www/img/povmeas/ack.pdf.

Clark, P. J. and F. C. Evans (1954). "Distance to nearest neighbor as a measure of spatial relationships in populations". *Ecology*, 35, 445-453.

Clayton, David and John Kaldor (1987). "Empirical Bayes estimates of age-standardized relative risks for use in disease mapping". *Biometrics*, 43, 671-681.

Cleveland, William S., Eric Grosse, and William M. Shyu (1993). "Local regression models". In John M. Chambers and Trevor J. Hastie, *Statistical Models in S*. Chapman & Hall: London.

Cliff, Andrew D. and Peter Haggett (1988). *Atlas of Disease Distributions*. Blackwell Reference: Oxford.

Cliff, A. and J. Ord (1973). *Spatial Autocorrelation*. Pion: London.

Cohen, L.E. and Felson, M. (1979) "Social change and crime rate trends: a routine activity approach", *American Sociological Review*, 44: 588-608.

Cohen, Lawrence E. 1981 "Modeling crime trends: a criminal opportunity perspective", *Journal of Research in Crime and Delinquency,* 18:138-163.

Cole, A. J. and D. Wishart (1970). "An improved algorithm for the Jardine-Sibson method of generating overlapping clusters". *Comparative Journal*, 13, 156-163.

Committee on Map Projections (1986). *Which Map is Best*, American Congress on Surveying and Mapping, Falls Church, VA., 1986.

Cornish, Derek and Ronald Clarke (1986). *The Reasoning Criminal*. Springer-Verlag: New York.

Cressie, Noel (1991). *Statistics for Spatial Data*. New York: J. Wiley & Sons, Inc.

Cromley, Robert G. (1992). *Digital Cartography*. Prentice Hall: Englewood Cliffs, NJ.

Culp, Michael (2002). *Land Use and Travel Demand Forecasting*. Powerpoint presentation. Federal Highway Adminstration, U.S. Department of Transportation: Washington, DC. http://tmip.fhwa.dot.gov/conf_courses/presentations.

Curtis, L. A. (1974). *Criminal Violence.* Lexington Books: Lexington, MA.

D'andrade,R. (1978). "U-Statistic Hierarchical Clustering" *Psychometrika*, 4,58-67.

de Berg, M.; van Kreveld, M.; Overmans, M.; and Schwarzkopf, O. (2000). "Convex hulls: mixing things." In *Computational Geometry: Algorithms and Applications*, 2nd rev. ed. Springer-Verlag: Berlin, 235-250,

Demographia (1999). *U.S. Central Cities and Suburban Crime Rates Ranked: 1999.* Wendell Cox Consultancy: Belleville, IL. http://www.demographia.com/db-crime99r.htm.

Demographia (1998). *U. S. Metropolitan Areas: 1998 Central City and Suburban Population.* Wendell Cox Consultancy: Belleville, IL. http://www.demographia.com/db-usmsacc98.htm.

Diggle, P. J. (2003). *Statistical Analysis of Spatial Point Patterns.* Arnold: London.

Dijkstra, E. W. (1959). "A note on two problems in connection with graphs", *Numerische Mathematik*, 1, 269-271.

Domencich, Thomas and Daniel McFadden (1975). *Urban Travel Demand: A Behavioral Analysis.* North Holland Publishing Company: Amsterdam & Oxford (republished in 1996). Also found at http://emlab.berkeley.edu/users/mcfadden/travel.html.

Draper, Norman and Harry Smith (1981). *Applied Regression Analysis, Second Edition.* John Wiley & Sons: New York.

Durkheim, Emile (1895). *The Rules of Sociological Method.* Free Press: New York. 1964.

Dwass, M (1957). "Modified randomization tests for nonparametric hypotheses". *Annals of Mathematical Statistics*, 28, 181-187.

Ebdon, David (1988). *Statistics in Geography* (second edition with corrections). Blackwell: Oxford.

Ehrlich, I. (1975). "On the relation between education and crime". In F. T. Juster (ed), *Education, Youth and Human Behavior.* McGraw-Hill: New York, 313-337.

Eldridge, J. D. and J. P. Jones (1991). "Warped space: a geography of distance decay", *Professional Geographer*, 43 (4), 500-511.

Engelen, Rodney E. (1986). "Transportation planning". In Frank S. So, *The Practice of State and Regional Planning.* American Planning Association: Chicago, ch. 17, 431-453.

ESRI (1998a). *ArcView GIS 3.1.* Environmental Systems Research Institute: Redland, CA.

ESRI (1998b). *ArcInfo 7.2.1*. Environmental Systems Research Institute: Redland, CA.

ESRI (1998c). *Atlas*GIS 4.0*. Environmental Systems Research Institute: Redland, CA.

ESRI (1997). *ArcView Spatial Analyst*. Environmental Systems Research Institute: Redland, CA.

Everett, Brian (1974). *Cluster Analysis*. Heinemann Educational books, Ltd: London.

Farewell, Daniel (1999). "Specifying the bandwidth function for the kernel density estimator". http://www.iph.cam.ac.uk/bugs/documentation/coda03/node44.html.

Felson, Marcus (2002) *Crime & Everyday Life* (3rd Ed). Sage: Thousand Oaks, CA.

FHWA (2001a). "Chapter 2: A critical review of the trip-based, four-step procedure of urban passenger demand forecasting". In *Activity-based Modeling System for Travel Demand Forecasting*.The Travel Model Improvement Program, Federal Highway Administration, U.S. Department of Transportation: Washington, DC. http://tmip.fhwa.dot.gov/clearinghouse/docs/amos/ch2.stm.

FHWA (2001b). "Appendix 2: Motor vehicle use and the Clean Air Act: Boosting Efficiency by reducing travel". *Transportation Conformity and Demand Management: Vital Strategies for Clean Air Attainment*. The Travel Model Improvement Program, Federal Highway Administration, U.S. Department of Transportation: Washington, DC. http://tmip.fhwa.dot.gov/clearinghouse/docs/airquality/vsca/app2.stm.

FHWA (1997). *Model Validation and Reasonableness Checking Manual*. Prepared by Barton-Aschman Associates, Inc and Cambridge Systematics, Inc for the Travel Model Improvement Program, Federal Highway Administration, U.S. Department of Transportation: Washington, DC. http://tmip.fhwa.dot.gov/clearinghouse/docs/mvrcm/.

FHWA (1996). "Latest VMT growth estimates", *Highway Information Update*, 1(1), Federal Highway Administration, U.S. Department of Transportation: Washington, DC., http://www.fhwa.dot.gov//ohim/vol1no1.html.

Field, Brian and Bryan MacGregor (1987). *Forecasting Techniques for Urban and Regional Planning*. UCL Press, Ltd: London.

Fisher, W. (1958). "On grouping for maximum homogeneity." *Journal of the American Statistical Association*. **53**, 789-798.

Foot, D. (1981). *Operational Urban Models*. Methuen: London.

Fotheringham, A. Stewart, Chris Brunsdon, and Martin Charlton (2002). *Geographically Weighted Regression: The Analysis of Spatially Varying Relationships*. John Wiley & Sons: New York.

Fotheringham, A. Stewart and M. E. O'Kelly (1989). *Spatial Interaction Models: Formulations and Applications*. Kluwer Academic Publishers: Boston.

Fotheringham, A. Stewart and M. E. O'Kelly (1989). *Spatial Interaction Models: Formulations and Applications*. Kluwer Academic Publishers: Boston.

Fowles, R. & M. Merva. (1996). "Wage Inequality and Criminal Activity", *Criminology*, 34, 163-82.

Friedman, H. P. and J. Rubin (1967). "On some invariant criteria for grouping data", *Journal of the American Statistical Association*, **62**, 1159-1178.

Furfey, P. H. (1927). "A note on Lefever's 'Standard deviational ellipse'". *American Journal of Sociology*. XXIII, 94-98.

Gaile, Gary L. and James E. Burt (1980). *Directional Statistics*. Concepts and Techniques in Modern Geography No. 25. Institute of British Geographers, Norwich, England: Geo Books.

Geary, R. (1954). "The contiguity ratio and statistical mapping". *The Incorporated Statistician*, 5, 115-145.

Gersho, A. and R. Gray (1992). *Vector Quantization and Signal Compression*. Kluwer Academic Publishers: Dordrecht, Netherlands.

Getis, Arthur (1991). "Spatial interaction and spatial auto-correlation: a cross-product approach". *Environment and Planning A*, 23, 1269-1277.

Getis, Arthur and J. Keith Ord (1996). "Local spatial statistics: an overview". In Paul Longley and Michael Batty (eds), *Spatial Analysis: Modelling in a GIS Environment*. GeoInformation International: Cambridge, England, 261-277.

Getis, Arthur and Barry Boots (1978). *Models of Spatial Processes: An Approach to the Study of Point, Line and Area Patterns*. London: Cambridge University Press.

Golden Software. 1994. *Surfer® for Windows (Ver. 6)*. Golden Software, Inc.: Golden, CO.

Goulias, Konstadinos G. (1996). "Activity-based travel forecasting: What are some issues?". Proceedings of Activity-based Travel Forecasting conference. Federal Highway Administration, U. S. Department of Transportation: Washington, DC. http://tmip.fhwa.dot.gov/clearinghouse/docs/abtf/.

Gowers, J. C. (1967). "A comparison of some methods of cluster analysis". *Biometrics*, 23, 623-628.

Graham, R (1972). "An efficient algorithm for determining the convex hull of a finite planar point set". *Info. Proc. Letters*, 1, 132-133.

Greenhood, David (1964). *Mapping*. The University of Chicago Press: Chicago.

Griffith, Daniel A. (1987). *Spatial Autocorrelation: A Primer*. Resource Publications in Geography, The Association of American Geographers: Washington, DC.

Groff, Elizabeth R. (2002). "Modeling the spatial dynamics of homicide". Paper presented at Mapping and Analysis for Public Safety annual conference. Denver, CO. December. http://www.ojp.usdoj.gov/nij/maps/Conferences/02conf/Groff.ppt

Grubesic, Tony H. and Alan T. Murray (2001). "Detecting hot spots using cluster analysis and GIS". Paper presented at Annual Conference of the Crime Mapping Research Center, Dallas, TX. http://www.ojp.usdoj.gov/cmrc.

Hagan, J. & R. Peterson (1994). *Inequality and Crime*. Stanford University Press: Palo Alto, CA.

Hägerstrand, T. (1957). "Migration and area: survey of a sample of Swedish migration fields and hypothetical considerations on their genesis". *Lund Studies in Geography, Series B, Human Geography*, 4, 3-19.

Haggett, Peter and Edward Arnold (1965). *Locational Analysis in Human Geography* (1st edition). Edward Arnold: London.

Haggett, Peter, Andrew D. Cliff, and Allan Frey (1977). *Locational Analysis in Human Geography* (2nd edition). Edward Arnold: London.

Hall, D. B. (2000). "Zero-inflated Poisson and binomial regression with random effects: a case study". *Biometrics*, 56, 1030-1039.

Hammond, Robert, and Patrick McCullagh (1978). *Quantitative Techniques in Geography: An Introduction*. Second Edition. Clarendon Press: Oxford, England.

Härdle, Wolfgang (1991). *Smoothing Techniques with Implementation in S*. Springer-Verlag: New York.

Harries, Keith (1999). *Mapping Crime: Principle and Practice*. NCJ 178919, National Institute of Justice, U. S. Department of Justice: Washington, DC., http://www.ncjrs.org/html/nij/mapping/pdf.html.

Harries, Keith (1980). *Crime and the Environment*. Charles C. Thomas Press: Springfield.

Harries, Keith and Phil Canter (1998). "The use of GPS in geocoding crime incidents". Personal Communication.

Hartigan, J. A. (1975). *Clustering Algorithms*. John Wiley & Sons, Inc.: New York.

Henderson, Robin (1981). *The Structural Root Systems of Sitka Spruce and Related Stochastic Processes*. PhD Thesis, University of Edinburgh: Edinburgh.

Henderson, Robin, Eric Renshaw, and David Ford (1983). "A note on the recurrence of a correlated random walk". Journal of Applied Probability, 20, 696-699.

Henderson, Robin, Eric Renshaw, and David Ford (1984). "A correlated random walk model for two-dimensional diffusion". Journal of Applied Probability, 21, 233-246.

Henderson, R., E. D. Ford, E. Renshaw, and J. D. Deans (1983). "Morphology of the structural root system of Sitka Spruce 1. Analysis and Quantitative Description". *Forestry*, 56 (2), 121-135.

Hensher, David A. and Kenneth J. Button (2002). *Handbook of Transport Modeling*. Elsevier Science: Cambridge, UK.

Horowitz, Joel L., Frank S. Koppelman, and Steven R. Lerman (1986). *A Self-instructing Course in Disaggregate Mode Choice Modeling*. Federal Transit Administration, U.S. Department of Transportation: Washington, DC. http://ntl.bts.gov/DOCS/381SIC.html.

Huff, D. L. (1963). "A probabilistic analysis of shopping center trade areas". *Land Economics*, 39, 81-90.

Hultquist, J., L. Brown and J. Holmes (1971). "Centro: a program for centrographic measures". Discussion paper no. 21, Department of Geography, Ohio State University: Columbus, OH.

Huxhold, William E. (1991). *An Introduction to Geographic Information Systems*. Oxford University Press: Oxford, New York, 147-184.

IIHS (2004). *Red Light Running*. Insurance Institute for Highway Safety: Arlington, VA. http://www.hwysafety.org/safety%5Ffacts/rlc.htm.

Insightful Corporation (2001). *S-PLUS 6.0 Professional for Windows*. Insightful Corporation: Seattle, WA.

Isard, Walter (1979). *Location and Space-Economy: A General Theory Relating to Industrial Location, Market Areas, Land Use, Trade, and Urban Structure* (originally published 1956). Program in Urban and Regional Studies, Cornell University: Ithaca, NY.

Isard, Walter (1960). *Methods in Regional Analysis*. John Wiley and Sons: New York.

Isbel, E. C. (1944). "Internal migration in Sweden and intervening opportunities", *American Sociological Review*, 9, 627-639.

ITE (2003). *Trip Generation* (7th edition). Institute of Transportation Engineers: Washington, DC.

ITE (2000). *Highway Capacity Manual.* Institute of Transportation Engineers: Washington, DC.

Jardine, N. and R. Sibson (1968). "The construction of hierarchic and non-hierarchic classifications". *Comparative Journal*, 11, 117-184.

Jefferis, Eric (1998). "A multi-method exploration of crime hot spots". Crime Mapping Research Center, National Institute of Justice: Washington, DC.

Johnson, M.A. (1978). "Attribute importance in multiattribute transportation decisions", *Transportation Research Record*, 673, 15-21.

Johnson, S.C. (1967), "Hierarchical Clustering Schemes" *Psychometrika*, 2,241-254

Jones, K. S. and D. M. Jackson (1967). "Current approaches to classification and clump finding at the Cambridge Language Research Unit". *Comparative Journal*, 10, 29-37.

Kafadar, K. (1996). "Smoothing geographical data, particularly rates of disease". *Statistics in Medicine* 15(23), 2539-2560.

Kallay, M. (1984). "The complexity of incremental convex hull algorithms in Rd", *Info. Proc. Letters* 19, 197.

Kaluzny, Stephen P., Silvia C. Vega, Tamre P. Cardoso, and Alice A. Shelly (1998). *S+ Spatial Stats: User Manual for Windows and Unix.* Springer: New York.

Kanji, Gopal K. (1993). *100 Statistical Tests.* Sage Publications: Thousand Oaks, CA.

Kelsall, J. E. and P. J. Diggle (1995a). "Kernel estimation of relative risk", *Bernoulli*, 1, 3-16.

Kelsall, J. E. and P. J. Diggle (1995b). "Non-parametric estimation of spatial variation in relative risk". *Statistical Medicine*, 14, 2335-2342.

Kent, Josh, Michael Leitner, and Andrew Curtis (2004). "Evaluating the usefulness of functional distance measures when calibrating Journey-To-Crime distance decay functions". *Computers, Environment and Urban Systems.* In press.

Kim, Karl E. and Michael Parke (1996). The use of GPS and GIS in traffic safety. Report to Motor Vehicle Safety Office, State of Hawaii Department of Transportation: Honolulu.

Kind, Stuart S. (1987). "Navigational ideas and the Yorkshire Ripper investigation". *Journal of Navigation*, 40 (3), 385-393.

King, B. F. (1967). "Step wise clustering procedures". *Journal of the American Statistical Association*. 62, 86-101.

Kohfeld, C. W. and J. Sprague (1988). "Urban unemployment drives crime". *Urban Affairs Quarterly*, 24, 215-241.

Knox, E. G. (1988). "Detection of clusters". In P. Elliott (ed), *Methodology of Enquiries into Disease Clustering*, London School of Hygiene and Tropical Medicine: London.

Knox, E. G. (1964). "The detection of space-time interactions". *Applied Statistics*, 13, 25-29.

Knox, E. G. (1963). "Detection of low intensity epidemicity: application in cleft lip and palate". *British Journal of Preventive and Social Medicine*, 18, 17-24.

Krueckeberg, D. A. and A. L. Silvers (1974). *Urban Planning Analysis: Methods and Models*. John Wiley and Sons: New York.

Kuhn, H.W. and R. E. Kuenne (1962). "An efficient algorithm for the numerical solution of the generalized Weber problem in spatial economics", *Journal of Regional Science* 4, 21-33.

Kulldorff, Martin (1997). "A spatial scan statistic", *Communications in Statistics - Theory and Methods*, 26, 1481-1496.

Kulldorff, M. and G. Williams (1997). *SaTScan v 1.0: Software for the Space and Space-Time Scan Statistics*, Bethesda, MD: National Cancer Institute.

Kulldorff, Martin and Nevelle Nagarwalla (1995). "Spatial disease clusters: Detection and inference", *Statistics in Medicine*, 14, 799-810.

Lam, Nina Siu-ngan and Lee De Cola (1993). *Fractals in Geography*. The Blackburn Press: Caldwell, NJ.

Lander, B. (1954). *Toward an Understanding of Juvenile Delinquency*. Columbia University Press: New York.

Langbein, Laura Irwin and Allan J. Lichtman (1978). *Ecological Inference.* Sage University Paper series on Quantitative Applications in the Social Sciences, series no. 07-010. Beverly Hills and London: Sage Publications.

Langworthy, Robert H. and Eric Jefferis (1998). "The utility of standard deviational ellipses for project evaluation". Discussion paper, National Institute of Justice: Washington, DC.

LaVigne, Nancy and Julie Wartell (1998). *Crime Mapping Case Studies: Success in the Field (volume 1).* Police Executive Research Forum and National Institute of Justice, U. S. Department of Justice: Washington, DC.

LaVigne, Nancy and Julie Wartell (2000). *Crime Mapping Case Studies: Success in the Field (volume 2).* Police Executive Research Forum and National Institute of Justice, U. S. Department of Justice: Washington, DC., http://www.mn-8.com/Merchant2/merchant.mvc?Screen=PROD&Product_Code=841&Category_Code=CAR.

LeBeau, James L. (1997). *Demonstrating the Analytical Utility of GIS for Police Operations: A final report*, NCJ 187104, National Institute of Justice, U. S. Department of Justice: Washington, DC., http://www.ncjrs.org/pdffiles1/nij/187104.pdf.

LeBeau, James L. (1992). "Four case studies illustrating the spatial-temporal analysis of serial rapists". *Police Studies*, 15(3), 124-145.

LeBeau, James L. (1987a). "The journey to rape: geographic distance and the rapist's method of approaching the victim", *Journal of Police Science and Administration*, 15 (2), 129-136.

LeBeau, James L. (1987b). "The methods and measures of centrography and the spatial dynamics of rape", *Journal of Quantitative Criminology*, 3 (2), 125-141.

Lefever, D. (1926). "Measuring geographic concentration by means of the standard deviational ellipse". *American Journal of Sociology*, 32(1): 88-94.

Levine, Ned (2002).*CrimeStat: A Spatial Statistics Program for the Analysis of Crime Incident Locations* (version 2.0). Ned Levine & Associates, Houston, TX; National Institute of Justice, Washington, DC.

Levine, Ned (2000). *CrimeStat: A Spatial Statistics Program for the Analysis of Crime Incident Locations* (version 1.1). Ned Levine & Associates, Annandale, VA.; National Institute of Justice, Washington, DC.

Levine, Ned (1999) *CrimeStat: A Spatial Statistics Program for the Analysis of Crime Incident Locations* (version 1.0). Ned Levine & Associates, Annandale, VA.; National Institute of Justice, Washington, DC.

Levine, Ned (1999). "The effects of local growth management on regional housing production and population redistribution in California", *Urban Studies*, 36(12), 2047-2068.

Levine, Ned (1996). "Spatial statistics and GIS: software tools to quantify spatial patterns". *Journal of the American Planning Association*. 62 (3), 381-392.

Levine, Ned and Karl E. Kim (1999). "The spatial location of motor vehicle accidents: A methodology for geocoding intersections". *Computers, Environment, and Urban Systems*. 22 (6), 557-576.

Levine, Levine, Karl E. Kim, and Lawrence H. Nitz (1995a). "Spatial analysis of Honolulu motor vehicle crashes: I. Spatial patterns". *Accident Analysis & Prevention*, 27(5), 663-674.

Levine, Ned, Karl E. Kim, and Lawrence H. Nitz (1995b). "Spatial analysis of Honolulu motor vehicle crashes: II. Generators of crashes". *Accident Analysis & Prevention*, 27(5), 675-685.

Levine, Ned and Martin Wachs (1986a). "Bus Crime in Los Angeles: I - Measuring The Incidence". *Transportation Research*. 20 (4), 273-284.

Levine, Ned and Martin Wachs (1986b). "Bus Crime in Los Angeles: II - Victims and Public Impact". *Transportation Research*. 20 (4), 285-293.

Levine, Ned, Martin Wachs and Elham Shirazi (1986). "Crime at Bus Stops: A Study of Environmental Factors". *Journal of Architectural and Planning Research*. 3 (4), 339-361.

Lind, Andrew W. (1930). "Some ecological patterns of community disorganization in Honolulu". *American Journal of Sociology*, 36 (2). 206-220.

Los Angeles Times (1998). *Eye on the Sky*. Business section, July 20.

Lottier, S. (1938). "Distribution of criminal offences in metropolitan regions", *Journal of Criminal Law, Criminology, and Police Science*, 29, 37-50.

Lundrigan, S., & Canter, D., (2001) A multivariate analysis of serial murderers' disposal site location choice in Journal of Environmental Psychology, 21, 423-432.

MacQueen, J. (1967). "Some methods for classification and analysis of multivariate observations". *5th Berkeley Symposium on Mathematics, Statistics and Probability*. Vol 1, 281-298.

McBratney, A. B. and J. J. deBruijter (1992). "A continuum approach to soil classification by modified fuzzy k-means with extragrades", *Journal of Soil Science*, 43, 159-175.

McDonnell, Porter W. Jr. (1979). *Introduction to Map Projections*. New York: Marcel Dekker, Inc.

McGuckin, Nancy A. And Nanda Srinivasan (2003). *Journey to Work in the United States and its Major Metropolitan Areas*. FHWA-EP-03-058, Office of Planning, Federal Highway Administration: Washington, DC.

McQuitty, L. L. (1960). "Hierarchical syndrome analysis". *Educational and Psychological Measurement*, 20, 293-304.

Maling, D. H. (1973). *Coordinate Systems and Map Projections* (1973). George Philip and Sons, London.

Malkiel, Burton G. (1999). *A Random Walk Down Wall Street* (revised edition). W. W. Norton & Company: New York.

Maltz, Michael D., Andrew C. Gordon, and Warren Friedman (1989). *Mapping Crime in Its Community Setting: A Study of Event Geography Analysis.*

Mantel, Nathan (1967). "The detection of disease clustering and a generalized regression approach". *Cancer Research*, 27, 209-220.

Mantel, N. and J. C. Bailar (1970). "A class of permutational and multinomial test arising in epidemiological research", *Biometrics*, 26, 687-700.

MapInfo (1998). *MapInfo Professional 5.0.1.* MapInfo Corporation: Troy, NY.

Marcon, Eric and Florence Puech (2003). "Evaluating the geographic concentration of industries using distance-based methods". *Journal of Economic Geography*, 3, 409-428.

Mardia, K.V. (1972). *Statistics of Directional Data.* Academic Press: New York.

Massey, F. J., Jr (1951). "The distribution of the maximum deviation between two sample cumulative step functions". *Annuals of Mathematical Statistics*, 22, 125-128.

Mather, A. S. (1986). *Land Use.* John Wiley & Sons: New York.

McCullagh, P. And J. A. Nelder (1989). *Generalized Linear Models* (2nd edition). Chapman & Hall/CRC: Boca Raton, FL.

McFadden, Daniel L. (2002). "The path to discrete-choice models". *Access*, No. 20, Spring. 20-25. http://www.uctc.net/access/access.asp#20.

Messner, S. (1986). "Economic inequality and levels of urban homicide", *Criminology*, 23, 297-317.

Messner, Steven and Kenneth Tardiff (1986). "The social ecology of urban homicide: an application of the 'Routine Activities approach'". *Criminology*, 22, 241-267.

Microsoft (2001). *Excel™.* Microsoft Corporation: Redmond, WA.

Microsoft (1999). *Welcome to the ODBC Section of the Microsoft Universal Data Access Web Site.* Microsoft: Redmond, WA. http://www.microsoft.com/data/obdc.

Microsoft (2002). *Windows XP.* Microsoft: Redmond, WA.

Microsoft (1998a). *Windows NT Workstation 4.0.* Microsoft: Redmond, WA.

Microsoft (1998b). *Windows NT Server 4.0.* Microsoft: Redmond, WA.

Microsoft (1998c). *Windows 98*. Microsoft: Redmond, WA.

Microsoft (1995). *Windows 95*. Microsoft: Redmond, WA.

Miaou, Shaw-Pin, Joon Jin Song, and Bani K. Mallick (2003). "Roadway traffic crash mapping: a space-time modeling approach", *Journal of Transportation and Statistics*, 6 (1), 33-57.

Miaou, Shaw-Pin (1996). *Measuring the Goodness-of-Fit of Accident Prediction Models*. FHWA-RD-96-040. Federal Highway Administration, U.S. Department of Transportation: Washington, DC.

Miller, Eric J. (1996). "Microsimulation and activity-based forecasting". Proceedings of Activity-based Travel Forecasting conference. Federal Highway Administration, U. S. Department of Transportation: Washington, DC. http://tmip.fhwa.dot.gov/clearinghouse/docs/abtf/.

Moran, P. A. P. The interpretation of statistical maps. *Journal of the Royal Statistical Society B*, 10, 1948; 243-251.

Moran, P. A. P. Notes on continuous stochastic phenomena. *Biometrika*, 37, 1950; 17-23.

Murray, A.T. and T.H. Grubesic. 2002. "Identifying Non-hierarchical Clusters." *International Journal of Industrial Engineering*, 9, 86-95.

NCHRP (1998). *Integration of Land Use Planning with Multimodal Transportation Planning*. Project 8-32(3). Prepared by Parsons Brinkerhoff Quade and Douglas, Inc. for the National Cooperative Highway Research Program, Transportation Research Board, National Research Council: Washington DC. October.

NCHRP (1995). *Travel Estimation Techniques for Urban Planning*. Project 8-29(2). National Cooperative Highway Research Program, Transportation Research Board: Washington, DC. http://www4.trb.org/trb/crp.nsf/0/647c1cb3a6b6bfe285256748005619fa?OpenDocument

Needham, R. M. (1967). "Automatic classification in linguistics". *The Statistician*, 17, 45-54.

Neft, David Samuel (1962). *Statistical Analysis for Areal Distributions*. Ph.D. dissertation, Columbia University: New York.

Newell, Allen, J.C. Shaw, and H. A. Simon (1957). "Empirical Explorations of the Logic Theory Machine", Proceedings of the Western Joint Computer Conference, pp. 218-239.

Newman, O. (1972). *Defensible Space: Crime Prevention Through Urban Design*. Macmillan: New York.

Nilsson, Nils J. (1980). *Principles of Artificial Intelligence.* Morgan Kaufmann Publishers, Inc.: Los Altos, CA.

NIST (2004). "Gallery of distributions". *Engineering Statistics Handbook*. National Institute of Standards and Technology: Washington, DC. http://www.itl.nist.gov/div898/handbook/eda/section3/eda366.htm.

Normandeau, Andre (1967). *Trends and Patterns in Robbery: Philadelphia, PA, 1960-66.* Ph.D. Dissertation, University of Pennsylvania: Philadelphia.

Oppenheim, Norbert (1980). *Applied Models in Urban and Regional Analysis*. Prentice-Hall, Inc.: Englewood Cliffs, NJ.

Openshaw, S. A. W. Craft, M. Charlton, and J. M. Birch (1988). "Invetigation of leukemia clusters by use of a geographical analysis machine", *Lancet*, 1, 272-273.

Openshaw, S. A., M. Charlton, C. Wymer, and A. W. Craft (1987). "A mark 1 analysis machine for the automated analysis of point data sets", *International Journal of Geographical Information Systems*, 1, 335-358.

Ortuzar, Juan de Dios and Luis G. Willumsen (2001). *Modeling Transport* (3rd edition). J. Wiley & Sons: New York.

Ottawa (1997). *Transportation Master Plan*. Regional Municipality of Ottawa-Carleton. http://www.ottawa.ca/city_services/planningzoning/17_2_en.shtml.

Papachristos, Andrew (2003). "The social structure of gang homicides in Chicago". Annual conference of the American Society of Criminologists, Denver.

Park, R. & E. Burgess. (1924). *Introduction to the Science of Sociology*. Chicago University Press: Chicago.

Parzen, E. (1962). "On the estimation of a probability density and mode". *Annuals of Mathematical Statistics*, 33, 1065-1076.

Pas, Eric I. (1996). "Recent advances in activity-based travel demand modeling". Proceedings of Activity-based Travel Forecasting conference. Federal Highway Administration, U. S. Department of Transportation: Washington, DC. http://tmip.fhwa.dot.gov/clearinghouse/docs/abtf/.

Patterson Philip (1998). *Factors that Affect VMT Growth*. U.S. Department of Energy: Washington. DC. http://www.ott.doe.gov/pdfs/vmtwhite.pdf.

Phillip, P.D. (1980) "Characteristics and typology of the journey to crime." In D.E. Georges-Abeyie and K.D. Harries (eds), *Crime: A Spatial Perspective*, Columbia Univ. Press: New York, 156-166.

Porojan, A. (2000). "Trade flows and spatial effects: the Gravity Model revisited". Conference on Managing Economic Transition in Eastern Europe: Emerging Research Issues. The Manchester Metropolitan University: Manchester, England, January. http://www.business.mmu.ac.uk/research/met/papers/aporojan.pdf .

Portland (1999). *Model Procedures Handbook*. City of Portland Office of Transportation METRO: Portland., Appendix I. http://www.portlandtransportation.org/planning/ModeChoice.htm.

Porter, Christopher, John Suhrbier, and William L. Schwartz (1999). "Forecasting bicycle and pedestrian travel: State of the practice and research needs". *Transportation Research Record*, 1674, 94-101.

Preparata, Franco and S.J. Hong (1977). "Convex hulls of finite sets of points in two and three dimensions", *Comm. ACM,* 20, 87-93.

Pyle, G. F. (1974). *The Spatial Dynamics of Crime.* Department of Geography Research Paper No. 159, University of Chicago: Chicago.

Pyle, Gerald F., Edward W. Hanten, Patricia Garstang Williams, Allen L. Pearson, II, J. Gary Doyle and Kwame Kwofie (1974). *The Spatial Dynamics of Crime.* Department of Geography, University of Chicago: Chicago.

Rabin, Steve (2000a). "A* aesthetic optimizations". In DeLoura, Mark. *Game Programming Gems*. Charles River Media, Inc.: Rockland, MA., 264-271.

Rabin, Steve (2000b). "A* speed optimizations". In DeLoura, Mark. *Game Programming Gems*. Charles River Media, Inc.: Rockland, MA., 272-287.

Rand, Alicia (1986). "Mobility triangles" in Robert M. Figlio, Simon Hakim and George Rengert (ed), *Metropolitan Crime Patterns*. Criminal Justice Press: Monsey, NY, 117-126.

Ravenstein, E. G. (1885). "The laws of migration". *Journal of the Royal Statistical Society.* 48.

RDC, Inc. (1995). "Activity-based modeling system for travel demand forecasting". Prepared for the Metropolitan Washington Council of Governments and the Federal Highway Administration, U.S. Department of Transportation: Washington, DC. http://tmip.fhwa.dot.gov/clearinghouse/docs/amos.

Recker, Will (2000). "A bridge between travel demand modeling and activity-based travel analysis". *Center for Activity Systems Analysis.* Paper UCI-ITS-AS-WP-00-11. http://repositories.cdlib.org/itsirvine/casa/UCI-ITS-AS-WP-00-11/.

Reilly, W. J. (1929). "Methods for the study of retail relationships". *University of Texas Bulletin*, 2944.

Rengert, G., A. R. Piquero, and P. R. Jones (1999). "Distance decay re-examined", *Criminology*, 37 (2), 427-445.

Rengert, George F. "Comparing cognitive hot spots to crime hot spots". In Carolyn Rebecca Block, Margaret Dabdoub and Suzanne Fregly, Crime Analysis Through Computer Mapping. Police Executive Research Forum: Washington, DC. 1995. 33-47.

Rengert, George F (1981). "Burglary in Philadelphia: a critique of the opportunity structure model". In Paul J. Brantingham and Patricia L. Brantingham, *Environmental Criminology*. Waveland Press, Inc.: Prospect Heights, IL, 189-202.

Rengert, George F. (1975). "Some effects of being female on criminal spatial behavior". *The Pennsylvania Geographer*, 13 (2), 10-18.

Renshaw, Eric (1985). "Computer simulation of sitka spruce: spatial branching models for canopy growth and root structure". Journal of Mathematics Applied in Medicine and Biology, 2, 183-200.

Repetto, T. A. (1974). *Residential Crime*. Ballinger: Cambridge, MA.

Rhodes, William M. and Catherine Conly (1981). "Crime and mobility: an empirical study". In Brantingham, Paul J. and Patricia L. Brantingham, *Environmental Criminology*. Waveland Press, Inc.: Prospect Heights, IL, 167-188.

Rich, Thomas (2001). "Crime mapping and analysis by community organizations in Hartford, Connecticut", *Research in Brief*. National Institute of Justice, U. S. Department of Justice: Washington, DC., http://www.ncjrs.org/pdffiles1/nij/185333.pdf.

Ripley, Brian D (1981). *Spatial Statistics*. John Wiley & Sons: New York.

Ripley, Brian D. (1976). "The second-order analysis of stationary point processes". *Journal of Applied Probability* 13: 255-66.

Robinson, A. H., R. D. Sale, J. L. Morrison and P. C. Muehrcke (1984). *Elements of Cartography* (5th edition). J. Wiley and Sons: New York.

Roemer, F. And K. Sinha (1974). "Personal security in buses and its effects on ridership in Milwaukee", *Transportation Research Record*, 487, 13-25.

Rosenblatt, M. (1956). "Remarks on some non-parametric estimates of a density function". *Annuals of Mathematical Statistics*, 27, 832-837.

Rossmo, D. Kim (2000). *Geographic Profiling*. CRC Press: Boca Raton Fl.

Rossmo, D. Kim (1993a). "Multivariate spatial profiles as a tool in crime investigation". In Carolyn Rebecca Block and Margaret Dabdoub (eds), *Workshop on Crime Analysis Through Computer Mapping: Proceedings*. Illinois Criminal Justice Information Authority and Loyola University Sociology Department: Chicago. (Library of Congress HV7936.C88 W67 1993).

Rossmo, D. Kim (1993b). "Target patterns of serial murderers: a methodological model". *American Journal of Criminal Justice*, 17, 1-21.

Rossmo, D. Kim (1995). "Overview: multivariate spatial profiles as a tool in crime investigation". In Carolyn Rebecca Block, Margaret Dabdoub and Suzanne Fregly, *Crime Analysis Through Computer Mapping*. Police Executive Research Forum:Washington, DC. 65-97.

Rossmo, D. Kim (1997). "Geographic profiling". In Janet L. Jackson and Debra A. Bekerian, *Offender Profiling: Theory, Research and Practice*. John Wiley and Sons: Chicester, 159-175.

Rushton, Gerard (1979). *Optimal Location of Facilities*. COMPress: Wentworth, NH.
SPSS, Inc. (1999). *SPSS 9.0 for Windows*. SPSS, Inc.: Chicago.

SAS Institute Inc. (1998). *Statistical Analysis System, Version 7*. Cary, NC.

Shachter, Jason (2001). "Geographical mobility: March 1999 to March 2000". *Current Population Reports*, P20-538, March. U.S. Census Bureau: Hyattsville, MD.

Schnell, J. B., A. J. Smith, K. R. Dimsdale, and L. J. Thrasher (1973). *Vandalism and Passenger Security: A Study of Crime and Vandalism on Urban Mass Transit Systems in the United States and Canada*. Prepared by the American Transit Association for the Urban Mass Transportation Administration (now Federal Transit Administration), U. S. Department of Transportation. National Technical Information Service: Springfield, VA. PB 236-854.

Schwartz, W.L., C. D. Porter, G.C. Payne, J.H. Suhrbier, P.C. Moe, and W.L. Wilkinson III (1999). *Guidebook on Methods to Estimate Non-Motorized Travel: Overview of Methods*. Turner-Fairbanks Highway Research Center, Federal Highway Administration: McLean, VA. July. http://www.tfhrc.gov/safety/pedbike/vol1/title.htm.

Scott, D. W. (1992). *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley and Sons: New York.

Sedgewick, Robert (2002). *Algorithms in C++: Part 5 Graph Algorithms* (3rd edition). Addison-Wesley: Boston.

Shaw, Clifford R. (1929). *Delinquency Areas*. University of Chicago Press: Chicago.

Shaw, Clifford R. & Henry D. McKay (1942). *Juvenile Delinquency in Urban Areas.* Chicago: University of Chicago Press, 1942.

Shaw, Clifford and Henry McKay (1972). *Juvenile Delinquency and Urban Areas* (revised edition). University of Chicago Press: Chicago.

Shekhar, Shashi and Sanjay Chawla (2003). *Spatial Databases: A Tour.* Prentice-Hall: Upper Saddle River, NJ.

Sherman, Lawrence W. and David Weisburd (1995). "General deterrent effects of police patrol in crime "hot spots": a randomized controlled trial". *Justice Quarterly.* 12, 625-648.

Shifton, Yoram, Moshe Ben-Akiva, Kimon Proussaloglu, Gerard de Jong, Yasasavi Popuri, Krishnan Kasturirangan, and Shlomo Bekhor (2003). "Activity-based modeling as a tool for better understanding travel behaviour". *Conference Proceedings.* 10th International Conference on Travel Behaviour Research, Lucerne, Switzerland. August. http://www.ivt.baug.ethz.ch/allgemein/pdf/shiftan.pdf.

Shoup, Donald (2002). "Roughly right vs. precisely wrong". *Access*, No. 20, Spring. 20-25. http://www.uctc.net/access/access.asp#20.

Siegel, Sidney (1956). *Nonparametric Statistics for the Behavioral Sciences.* McGraw-Hill: New York.

Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis.* Chapman & Hall: London.

Skiena, S. S. (1997). "Convex hull." §8.6.2 in *The Algorithm Design Manual.* Springer-Verlag: New York, 351-354.

Smirnov, N. V. (1948). "Table for estimating the goodness of fit of empirical distributions". *Annuals of Mathematical Statistics*, 19, 279-281.

Smith, T. S. (1976). "Inverse distance variations for the flow of crime in urban areas". *Social Forces*, 25(4), 804-815.

Smith, William, Sharon Glave and Davison, Elizabeth (2000). "Furthering the integration of routine activity and social disorganization theories: Small units of analysis and the study of street robbery as a diffusion process". *Criminology*, 38, 489-523.

Sneath, P. H. A. (1957). "The application of computers to taxonomy". *Journal of General Microbiology*, 17, 201-226.

Snyder, John P. (1987). *Map Projections - A Working Manual.* U.S. Geological Survey Professional Paper 1395. U. S. Government Printing Office: Washington, DC.

Snyder, John P. and Philip M. Voxland (1989). *An Album of Map Projections*. U.S. Geological Survey Professional Paper 1453. U. S. Government Printing Office: Washington, DC.

Sokal, R. R. and P. H. A. Sneath (1963). *Principles of Numerical Taxonomy*. W. H. Freeman and Co.: San Francisco.

Sokal, R. R. and C. D. Michener (1958). "A statistical method for evaluating systematic relationships". *University of Kansas Science Bulletin*, 38, 1409-1438.

Spitzer, Frank (1976). *Principles of Random Walk* (second edition). Springer: New York.

Stack, S. (1984). "Income inequality and property crime", *Criminology*, 22, 229-257.

Stephenson, L. (1980). "Centrographic analysis of crime". In D. George-Abeyie and K. Harries (eds), *Crime, A Spatial Perspective*, Columbia University Press: New York.

Stewart, J. Q. (1950). "The development of social physics". *American Journal of Physics*, 18, 239-53.

Stoe, Debra, Carol R. Watkins, Jeffrey Kerr, Linda Rost, and Theodosia Craig (2003). *Using Geographic Information Systems to Map Crime Victim Services: A Guide for State Victims of Crime Act Adminstrators and Victim Service Providers*. National Institute of Justice, U. S. Department of Justice: Washington, DC., http://www.ojp.usdoj.gov/ovc/publications/infores/geoinfosys2003/welcome.html.

Stopher, Peter R. and Arnim H. Meyburg (1975). *Urban Transportation Modeling and Planning*. Lexington, MA: Lexington Books.

Stouffer, S. A. (1940). "Intervening opportunities: a theory relating mobility and distance". *American Sociological Review*, 5, 845-67.

Stout, Bryan (2000). "The basics of A* for path planning". In DeLoura, Mark. *Game Programming Gems*. Charles River Media, Inc.: Rockland, MA., 254-263.

Systat, Inc. (2000). *Systat 10: Statistics I*. SPSS, Inc.: Chicago.

Tabachnick, B. G. and L. S. Fidell (1996). *Using Multivariate Statistics* (3rd ed). Harper Collins: New York.

Talbot T. O., Kulldorff M., Forand S. P., Haley V. B. (2000). "Evaluation of spatial filters to create smoothed maps of health data". *Statistics in Medicine*, 19, 2399-2408.

Taylor, Peter James (1970). *Interaction and Distance: An Investigation into Distance Decay Functions and a Study of Migration at a Microscale.* PhD thesis, University of Liverpool: Liverpool.

Tea3.org (2004). *Surface Transportation Policy Project*. http://www.transact.org/

Thompson, H. R. (1956). "Distribution of distance to nth neighbour in a population of randomly distributed individuals". *Ecology*, 37, 391-394.

Thorndike, R. L. (1953). "Who belongs in a family?". *Psychometrika*, 18, 267-276.

Thrasher, F. M. (1927). *The Gang*, University of Chicago Press: Chicago.

Thünen, J. H. Von (1826). *Der Isolierte Staat in Beziehung auf Landwirtshaft und Nationalökonomie* (The Isolated State in Relation to Agriculture). Hamburg.

Turnbull, B. W., E. J. Iwano, W. S. Burnett, H. L. Howe, and L. C. Clark (1990). "Monitoring for clusters of disease: application to leukemia incidence in upstate New York", *American Journal of Epidemiology*, 132, S136-S143.

Turner, S. (1969). "Delinquency and distance". In M. E. Wolfgang and T. Sellin (eds), *Delinquency: Selected Studies*. John Wiley and Sons: New York.

Turner, Shawn, Gordon Shunk, and Aaron Hottenstein (1998). *Development of a Methodology to Estimate Bicycle and Pedestrian Travel Demand*. Report 1723-S, Texas Transportation Institute: College Station. http://tti.tamu.edu/product/catalog/reports/1723-s.pdf.

U.S. Census Bureau (2004a). *TIGER/Line 2004*. Bureau of the Census, U. S. Department of Commerce: Washington, DC.

U.S. Census Bureau (2004b). *Journey to Work and Place of Work*. Bureau of the Census, U.S. Department of Commerce: Washington, DC. http://www.census.gov/population/www/socdemo/journey.html

U.S. Census Bureau (2002). "Vehicles available and household income in 1999: 2000". Table QT-H11, Census 2000 Summary File 3 (SF 3) - Sample Data, Bureau of the Census, U. S. Department of Commerce: Washington, DC. http://factfinder.census.gov/servlet/QTTable?_bm=y&-geo_id=D&-qr_name=DEC_2000_SFAIAN_QTH11&-ds_name=D&-_lang=en.

U.S. Census Bureau (2000). "All across the USA: Population distribution, 1999", In *Population Profile of the United States: 1999*. Bureau of the Census, U. S. Department of Commerce: Washington, DC., chapter 2.

USDOJ (2000). *Regional Crime Analysis Geographic Information System (RCAGIS).* Criminal Division, U.S. Department of Justice: Washington, DC. http://www.usdoj.gov/criminal/gis/rcagishome.htm.

USDOT (2003). *Title XXIII, Part 450.* Code of Federal Regulations. Code of Federal Regulations, Title 23, Part 450, Volume 1. 23CFR450. Washington, DC.

van Koppen, Peter J, Jasper J van der Kemp and Christianne J. de Poot (2002) "Geografische daderprofilering" (Geographic offender profiling) in van Koppen et al, *Het Recht Van Binnen: Psychologie Van Het Recht* (The Law from Inside: Psychology of the Law), Deventer, Netherlands Kluwer.

van Koppen, Peter, Christanne J. de Poot, and M. Vere van Koppen (2000). "Cirkels van delicten: over pleegplaatsen van misdrijven en de woonplaats van de daders" (Circles of crime: incident location and the residences of the offenders), *De Psycholoog: Psychologie en Recht* (The Psychologist, Psychology and Law), Oktober, 435-442.

van Koppen, Peter J. and Robert W. J. Jansen (1998). "The Road to robbery: travel patterns in commercial robberies". *British Journal of Criminology*, 38 (2), 230-246.

van Koppen, Peter J. and Jan W. de Keijser (1997). "Desisting distance decay: on the aggregation of individual crime trips". *Criminology*, 35 (3), 505-516.

Venables, W.N. and B.D. Ripley (1997). *Modern Applied Statistics with S-Plus (second edition).* Springer-Verlag: New York.

Wachs, Martin, Brian Taylor, Ned Levine and Paul Ong, "The Changing Commute: A Case Study of the Jobs/Housing Relationship Over Time". *Urban Studies*. 1993. 30 10, 1711-1729.

Ward, J. H. (1963). "Hierarchical grouping to optimize an objective function". *Journal of the American Statistical Association*. 58, 236-244.

Warren, J., Reboussin, R., Hazelwood, R., Cummings, A., Gibbs, N. & Trumbetta, S. (1998). "The distance correlates of serial rape". *Journal of Quantitative Criminology*, 14, 35-58.

Wartell, Julie and Tom McEwen (2001). *Privacy in the Information Age: A Guide for Sharing Crime Maps and Spatial Data.* National Institute of Justice, U. S. Department of Justice: Washington, DC., http://www.ncjrs.org/pdffiles1/nij/188739.pdf.

WASHCOG (1974). *Citizen Safety and Bus Transit.* Metropolitan Washington Council of Governments. National Technical Information Service, Springfield, VA. PB 237-740/AS.

Weber, A. (1909). *Über den Standort der Industrien* (Theory of Location of Industries).

Weisburd, David and Tom McEwen (1998). *Crime Mapping Crime Prevention*. Criminal Justice Press: Monsey, NY.

Weisburd, David and Lorraine Green (1995). "Policing drug hot spots: the Jersey City drug market analysis experiment". *Justice Quarterly*. 12 (4), 711-735.

White, R, Clyde (1932). "The relationship of felonies to environmental factors in Indianapolis". *Social Forces*, 10 (4), 488-509.

Whittle, P. (1958). "On the smoothing of probability density functions". *Journal of the Royal Statistical Society, Series B*, 55, 549-557.

Wilson, A. G. (1970). *Entropy in Urban and Regional Planning*. Leonard Hill Books: Buckinghamshire.

Wilson, J.Q. & G. Kelling. (1982) "Broken Windows: The Police and Neighborhood Safety." *Atlantic Monthly*, March. 29-38.

White, R. C. (1932). "The relation of felonies to environmental factors in Indianapolis". *Social Forces*, 10(4), 498-509.

Wright, Richard T. and Scott H. Decker (1997). *Armed Robbers in Action: Stickups and Street Culture*. Northeastern University Press, Boston.

Xie, X. L. and G. Beni (1991). "A validity measure for fuzzy clustering". IEEE Trans. Pattern Analysis Machine Intell., 13, 841-847.

Zipf, G. K. (1949). *Human Behaviour and the Principle of Least Effort*. Cambridge.

Zhao, Fang, Lee-Fang Chow, Min-Tang Li, Albert Gan, and David L. Shen (2001). *Refinement of FSUTMS Trip Distribution Methodology*. Lehman Center for Transportation Research, Florida International University: Miami, FL. http://www.eng.fiu.edu/LCTR/re-project-link/techmemo3.pdf.

Appendix A

Dynamic Data Exchange (DDE) Support

CrimeStat supports Dynamic Data Exchange (DDE). This allows the program to be linked to another program, which can call up *CrimeStat* as a routine. The following are the programming codes used to support DDE commands.

CrimeStat's DDE Topics That Support the DDE "poke" Command

Topic	Item	Data format
Primary File	File	RemoveAll
		AddTextFile| <name>| <columns>| <separator>| <header rows>
		AddDbfFile| <name>
		AddShpFile| <name>
	X	<file name>| <column name>
	Y	<file name>| <column name>
	Weight	<file name>| <column name>
	Intensity	<file name>| <column name>
	Direction	<file name>| <column name>
	Coordinate	<coordinate>| <unit> Valid coordinates: Longtitude, latitude Projected Direction Valid units: Decimal degrees Feet Meters
Secondary File	File	See Primary File
	X	See Primary File
	Y	See Primary File
	Weight	See Primary File
	Intensity	See Primary File
	Direction	See Primary File
Reference File	Source	<source> Valid sources: From File Generated
	File	See Primary File

	X	See Primary File
	Y	See Primary File
	True Grid	<number of columns> a value of zero (0) will uncheck the check box.
	Bound	<lower-left x>| <lower-left y>| <upper-right x>| <upper-right y>
	Cell specification	<source>| <value> Valid sources: By cell-spacing By number of columns
Measurement Parameters	Measurement Type	<type> Valid types: Direct Indirect
	Area	<area>| <area unit> Valid units: See Dialog
	Length	<length>| <length unit> Valid units: See Dialog

***CrimeStat*'s DDE Topics That Support the DDE "request" Command**

Topic	Item	Return value
System	SysItems	Name of the supported items of the "system" topic.
	ReturnMessage	Detailed information (if any) of the last message.
	Status	Server status, either "Ready" or "Busy".
	Formats	Supported data formats.
	Help	Detailed help on CrimeStat's DDE support.
	TopicItemList	Name of the supported items of the current topic.

***CrimeStat*'s DDE Topics That Support the DDE "execute" Command**

Topic	Command	Description
System	Quit	Close CrimeStat.
Primary File	Select	Select the Primary file tab.
Secondary File	Select	Select the Secondary file tab.
Reference File	Select	Select the Reference file tab.

Measurement Parameters	Select	Select the Measurement parameters tab.
Spatial Distribution	Select	Select the Spatial distribution tab.
Distance Analysis	Select	Select the Distance analysis tab.
'Hot spot' Analysis I	Select	Select the 'Hot spot' analysis I tab.
'Hot spot' Analysis II	Select	Select the 'Hot spot' analysis II tab.
Interpolation	Select	Select the Interpolation tab.

Example: Controlling *CrimeStat* from within Visual basic

```
Public Function OpenCrimeStat(topic As String) As Variant
   On Error Resume Next
   Dim channel, I
   Dim file As String
   file = "CrimeStat.exe"
   channel = DDEInitiate("CrimeStat", topic)
   If Err Then
      Err = 0
      I = Shell(file, 1)
      If Err Then
         Return
      End If
      channel = DDEInitiate("CrimeStat", topic)
   End If
   OpenCrimeStat = channel
End Function

Public Sub TestCrimeStatDde(foo As String)
   On Error Resume Next
   Dim file As String
   Dim channel

   file = "SampleData.dbf"
   channel = OpenCrimeStat("Primary File")
   DDEPoke channel, "Coordinate", "Projected| Feet"
   DDEPoke channel, "File", "RemoveAll"
   DDEPoke channel, "File", "AddDbfFile| " & file
   DDEPoke channel, "X", file & "| LON"
   DDEPoke channel, "Y", file & "| LAT"
   DDEPoke channel, "Coordinate", "Longtitude, latitude| Decimal degrees"
   DDETerminate channel

   file = "Grid.dbf"
   channel = OpenCrimeStat("Reference File")
```

```
    DDEPoke channel, "Source", "From File"
    DDEPoke channel, "True Grid", "0"
    DDEPoke channel, "File", "RemoveAll"
    DDEPoke channel, "File", "AddDbfFile| " & file
    DDEPoke channel, "X", file & "| LON"
    DDEPoke channel, "Y", file & "| LAT"
    DDEPoke channel, "True Grid", "108"
    DDEPoke channel, "Source", "Generated"
    DDEPoke channel, "Bound", "-78.5| 22.4| -75.3| 24.2"
    DDEPoke channel, "Cell Specification", "By cell-spacing| 0.5"
    DDEPoke channel, "Cell Specification", "By number of columns| 20"
    DDETerminate channel

    channel = OpenCrimeStat("Measurement Parameters")
    DDEPoke channel, "Measurement Type", "Direct"
    DDEPoke channel, "Measurement Type", "Indirect"
    DDEPoke channel, "Area", "734.12| Square meters"
    DDEPoke channel, "Length", "1734.12| meters"
    DDETerminate channel

    channel = OpenCrimeStat("Interpolation")
    DDEExecute channel, "select"
    DDETerminate channel
End Sub

Private Sub CrimeStatQuit_Click()
    On Error Resume Next
    Dim channel
    channel = OpenCrimeStat("System")
    DDEExecute channel, "quit"
    DDETerminate channel
End Sub
Private Sub TestCrimeStat_Click()
    TestCrimeStatDde "bar"
End Sub
```

Appendix B

Some Notes on the Statistical Comparison of Two Samples

The following presents methods for testing the spatial differences between two distributions. At this point, *CrimeStat* does not include routines for testing the differences between two or more samples. The following is provided for the reader's information. Chapter 4 discussed the calculation of these statistics as a single distribution.

Differences in the Mean Center of Two Samples

For differences between two samples in the mean center, it is necessary to test both differences in the X coordinate and differences in the Y coordinates. Since *CrimeStat* outputs both the mean X, mean Y, standard deviation of X, and standard deviation of Y, a simple t-test can be set up. The null hypothesis is that the mean centers are equal

$$H_0: \quad \mu_{XA} = \mu_{XB}$$
$$\mu_{YA} = \mu_{YB}$$

and the alternative hypothesis is that the mean centers are not equal

$$H_1: \quad \mu_{XA} \neq \mu_{XB}$$
$$\mu_{YA} \neq \mu_{YB}$$

Because the true standard deviations of sample A, σ_{XA} and σ_{YA}, and sample B, σ_{XB} and σ_{YB}, are not known, the sample standard deviations are taken, S_{XA}, S_{YA}, S_{XB} and S_{YB}. However, since there are two different variables being tested (mean of X and mean of Y for groups 1 and 2), the alternative hypothesis has two fundamentally different interpretations:

Comparison I: That EITHER $\mu_{XA} \neq \mu_{XB}$ OR $\mu_{YA} \neq \mu_{YB}$ is true

Comparison II: That BOTH $\mu_{XA} \neq \mu_{XB}$ AND $\mu_{YA} \neq \mu_{YB}$ are true

In the first case, the mean centers will be considered not being equal if either the mean of X *or* the mean of Y are significantly different. In the second case, both the mean of *and* the mean of Y must be significantly different for the mean centers to be considered not equal. The first case is clearly easier to fulfill than the second.

Significance levels

By tradition, significance tests for comparisons between two means are made at the $\alpha \leq .05$ or $\alpha \leq .01$ levels, though there is nothing absolute about those levels. The significance levels are selected to minimize *Type 1 Errors*, inadvertently declaring a difference in the means when, in reality, there is not a difference. Thus, a test establishes that the

likelihood of falsely rejecting the null hypothesis be less than one-in-twenty (less strict) or one-in-one hundred (more strict).

However, with multiple comparisons, the chances increase for finding 'significance' due to the multiple tests. For example, with two tests - a difference in the means of the X coordinate and a difference in the means of the Y coordinate, the likelihood of rejecting the first null hypothesis ($\mu_{XA} \neq \mu_{XB}$) is one-in-twenty and the likelihood of rejecting the second null hypothesis ($\mu_{YA} \neq \mu_{YB}$) is also one-in-twenty, then the likelihood of rejecting either one null hypothesis or the other is actually one-in-ten.

To handle this situation, comparison I - the 'either/or' condition, a Bonferoni test is appropriate (Anselin, 1995; Systat, 1996). Because the likelihood of achieving a given significance level increases with multiple tests, a 'penalty' must be assigned in finding either the differences in means for the X coordinate or differences in means for the Y coordinates significant. The Bonferoni criteria divides the critical probability level by the number of tests. Thus, if the $\alpha \leq .05$ level is taken for rejecting the null hypothesis, the critical probability for each mean must be .025 (.05/2); that is, differences in either the mean of X or mean of Y between two groups must yield a significance level less than .025.

For comparison II - the 'both/and' condition, on the other hand, the test is more stringent since the differences between the means of X and the means of Y must both be significant. Following the logic of the Bonferoni criteria, the critical probability level is multiplied by the number of tests. Thus, if the $\alpha = .05$ level is taken for rejecting the null hypothesis, then *both* tests must be significant at the $\alpha \leq .10$ level (i.e., .05*2).[1]

Tests

The statistics used are for the t-test of the difference between means (Kanji, 1993).

a. First, test for equality of variances by taking the ratio of the variances (squared sample standard deviations) of both the X and Y coordinates:

$$F_X = \frac{S_{XA}^2}{S_{XB}^2} \qquad (B.1)$$

$$F_Y = \frac{S_{YA}^2}{S_{YB}^2} \qquad (B.2)$$

with $(N_A - 1)$ and $(N_B - 1)$ degrees of freedom for groups *A* and *B* respectively. This test is usually done with the larger of the variances in the numerator. Since there are two variances being compared (for X and Y, respectively), the logic should follow either *I* or *II* above (i.e., if either are to be true, then the critical α will be actually $\alpha/2$ for each; if both must be true, then the critical α

will be actually 2*α for each).

b. Second, if the variances are considered equal, then a t-test for two group means with unknown, but equal, variances can be used (Kanji, 1993; 28). Let

$$S_{XAB} = \text{SQRT}\left[\frac{\sum_{i=1}^{N(A)} (X_{Ai} - \bar{X}_A)^2 + \sum_{i=1}^{N(B)} (X_{Bi} - \bar{X}_B)^2}{(N_A + N_B - 2)}\right] \quad \text{(B.3)}$$

$$S_{YAB} = \text{SQRT}\left[\frac{\sum_{i=1}^{N(A)} (Y_{Ai} - \bar{Y}_A)^2 + \sum_{i=1}^{N(B)} (Y_{Bi} - \bar{Y}_B)^2}{(N_A + N_B - 2)}\right] \quad \text{(B.4)}$$

where the summations are for i=1 to N within each group separately. Then the test becomes

$$t_X = \frac{(\bar{X}_A - \bar{X}_B) - (\mu_{XA} - \mu_{XB})}{S_{XAB} * \text{SQRT}\left[\frac{1}{N_A} + \frac{1}{N_B}\right]} \quad \text{(B.5)}$$

$$t_Y = \frac{(\bar{Y}_A - \bar{Y}_B) - (\mu_{YA} - \mu_{YB})}{S_{YAB} * \text{SQRT}\left[\frac{1}{N_A} + \frac{1}{N_B}\right]} \quad \text{(B.6)}$$

with $(N_A + N_B - 2)$ degrees of freedom for each test.

c. Third, if the variances are not equal, then a t-test for two group means with unknown and unequal variances should be used (Kanji, 1993; 29).

$$t_X = \frac{(\bar{X}_A - \bar{X}_B) - (\mu_{XA} - \mu_{XB})}{\text{SQRT}\left\{\left[\frac{S_{XA}^2}{N_A} + \frac{S_{XB}^2}{N_B}\right]\right\}} \quad \text{(B.7)}$$

$$t_Y = \frac{(\bar{Y}_A - \bar{Y}_B) - (\mu_{YA} - \mu_{YB})}{\text{SQRT}\left\{\left[\frac{S_{YA}^2}{N_A} + \frac{S_{YB}^2}{N_B}\right]\right\}} \quad \text{(B.8)}$$

with degrees of freedom

$$v = \left\{\frac{\left[\frac{S_A^2}{N_A} + \frac{S_B^2}{N_B}\right]}{\left[\frac{S_A^4}{N_A^2(N_A + 1)} + \frac{S_B^4}{N_B^2(N_B + 1)}\right]}\right\} - 2 \quad \text{(B.9)}$$

for both the X and Y test. Even though this latter formula is cumbersome, in practice, if the sample size of each group is greater than 100, then the t-values for infinity can be taken as a reasonable approximation and the above degrees of freedom need not be tested (t=1.645 for α=.05; t=1.960 for α=.01).

d. The significance levels are those selected above. For comparison I - that either differences in the means of X or differences in the means of Y are significant, the critical probability level is $\alpha/2$ (e.g., .05/2 = .025; .01/2 = .005). For comparison II - that both differences in the means of X and differences in the means of Y are significant, the critical probability level is $\alpha*2$ (e.g., .05*2 = .10; .01*2 - .02).

e. Reject the null hypothesis if:

Comparison I: Either tested t-value (t_x or t_y) is greater than the Critical t for $\alpha/2$

Comparison II: Both tested t-values (t_x and t_y) are greater than the critical t for $\alpha*2$

Example 1: Burglaries and Robberies in Baltimore County

To illustrate, compare the distribution of burglaries in Baltimore County with those of robberies, both for 1996. Figure B.1 shows the mean center of all robberies (blue square) and all residential burglaries (red triangle). As can be seen, the mean centers are located within Baltimore City, a property of the unusual shape of the county (which surrounds the city on three sides). Thus, these mean centers cannot be considered an unbiased estimate of the metropolitan area, but unbiased estimates for the County only. When the relative positions of the two mean centers are compared (figure 4.12 in chapter 4), the center of robberies is south and west of the center for burglaries. Is this difference significant or not?

To test this, the standard deviations of the two distributions are first compared and the F-test of the larger to the smaller variance is used (equations B.1 and B.2). *CrimeStat* provides the standard deviation of both the X and Y coordinates; the variance is the square of the standard deviation. In this case, the variance for burglaries is slightly larger than for robberies for both the X and Y coordinates.

$$F_X = \frac{S_{XA}^2}{S_{XB}^2} = \frac{0.0154}{0.0145} = 1.058$$

$$F_Y = \frac{S_{YA}^2}{S_{YB}^2} = \frac{0.0058}{0.0029} = 2.007$$

Because both samples are fairly large (1180 robberies and 6051 burglaries), the degrees of freedom are also very large. The F-tables are a little indeterminate with large samples, but the variance ratio approaches 1.00 as the sample reaches infinity. An approximate critical F-ratio can be obtained by the next largest pair of values in the table (1.22 for $p \leq .05$ and 1.32 for $p \leq .01$). Using this criteria, differences in the variances for the X coordinate are probably not significant while that for the Y coordinates definitely are significant. Consequently, the test for a difference in means with unequal variances is used (equations B.7, B.8 and B.9).

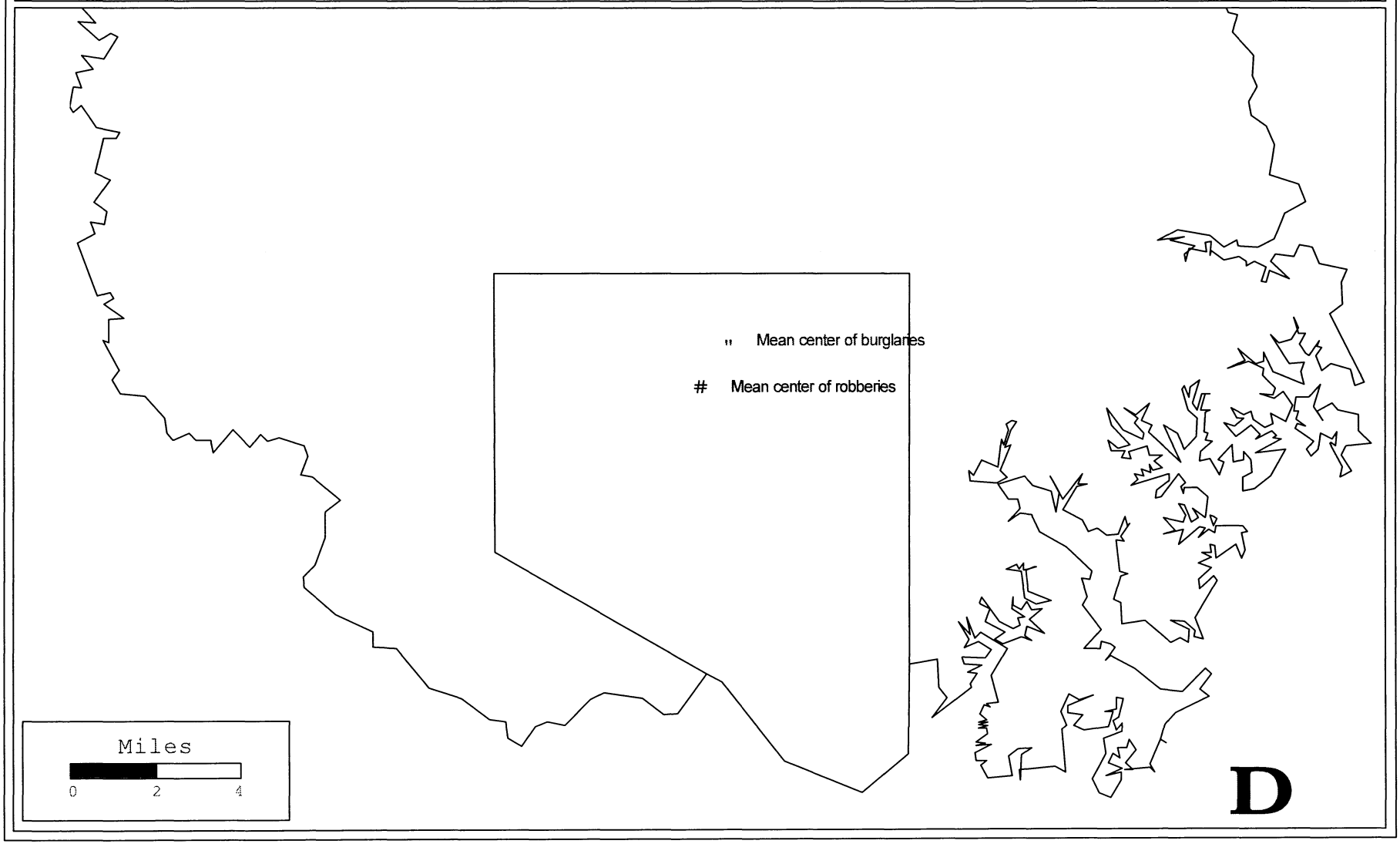
$$t_X = \frac{(\bar{X}_A - \bar{X}_B) - (\mu_{XA} - \mu_{XB})}{\text{SQRT}\left\{\left[\frac{S_{XA}^2}{N_A} + \frac{S_{XB}^2}{N_B}\right]\right\}} = \frac{-76.608482 - (-76.620838)}{\text{SQRT}\left\{\left[\frac{0.0154}{6051} + \frac{0.0145}{1180}\right]\right\}}$$

$$= \frac{0.0124}{0.0039} = 3.21 \ (p \leq .005)$$

Figure B.1:

1996 Burglaries and Robberies in Baltimore County

Comparison of Mean Centers



$$t_Y = \frac{(\bar{Y}_A - \bar{Y}_B) - (\mu_{YA} - \mu_{YB})}{\text{SQRT}\{[\frac{S_{YA}^2}{N_A} + \frac{S_{YB}^2}{N_B}]\}} = \frac{39.348368 - 39.334816}{\text{SQRT}\{[\frac{0.0058}{6051} + \frac{0.0029}{1180}]\}}$$

$$= \frac{0.0136}{0.0018} = 7.36 \quad (p \leq .005)$$

Therefore, whether we use the 'either/or' test (critical $\alpha \leq .025$) or the 'both/and' test (critical $\alpha \leq .1$), we find that the difference in the mean centers is highly significant. Burglaries have a different center of gravity than robberies in Baltimore County.

Differences in the Standard Distance Deviation of Two Samples

Since the standard distance deviation, S_{XY} (equation 4.6 in chapter 4) is a standard deviation, differences in the standard distances of two groups can be compared with an equality of variance test (Kanji, 1993, 37).

$$F = \frac{S_{XYA}^2}{S_{XYB}^2} \tag{B.10}$$

with $(N_A - 1)$ and $(N_B - 1)$ degrees of freedom for groups A and B, respectively. This test is usually done with the larger of the variances in the numerator. Since there is only one variance being compared, the critical α are as listed in the tables.

From *CrimeStat*, we find that the standard distance deviation of burglaries is 8.44 miles while that for robberies is 7.42 miles. In chapter 4, figure 4.12 displayed these two standard distance deviations. As can be seen, the dispersion of incidents, as defined by the standard distance deviation, is greater for burglaries than for robberies. The F-test of the difference is calculated by

$$F = \frac{S_{XYA}^2}{S_{XYB}^2} = \frac{8.44^2}{7.42^2} = 1.29$$

with 6050 and 1180 degrees of freedom respectively. Again, the F-tables are slightly indeterminate with respect to large samples, but the next largest F beyond infinity is 1.25 for $p \leq .05$ and 1.38 for $p \leq .01$. Thus, it appears that burglaries have a significantly greater dispersion than robberies, at least at the $p \leq .05$ level.

Differences in the Standard Deviational Ellipse of Two Samples

In a standard deviational ellipse, there are actually six variables being compared:

Mean of X
Mean of Y
Angle of rotation
Standard deviation along the transformed X axis
Standard deviation along the transformed Y axis
Area of the ellipse

Differences in the mean centers

Comparisons between the two mean centers can be tested with the above statistics.

Differences in the angle of rotation

Unfortunately, to our knowledge, there is not a formal test for the difference in the angle of rotation. Until this test is developed, we have to rely on subjective judgements.

Differences in the standard deviations along the transformed axes

The differences in the standard deviations along the transformed axes (X and Y) can be tested with an equality of variance test (Kanji, 1993, 37).

$$F_{Sx} = \frac{S_{X1}^2}{S_{X2}^2} \qquad \text{(B.11)}$$

$$F_{SY} = \frac{S_{Y1}^2}{S_{Y2}^2} \qquad \text{(B.12)}$$

with $(N_A - 1)$ and $(N_B - 1)$ degrees of freedom for groups A and B respectively. This test is usually done with the larger of the variances in the numerator. The example above for comparing the mean centers of Baltimore County burglaries and robberies illustrated the use of this test.

Differences in the areas of the two ellipses

Since an area is a variance, the differences in the areas of the two ellipses can be compared with an equality of variance test (Kanji, 1993, 37).

$$F = \frac{Area_A}{Area_B} \quad (B.13)$$

with $(N_A - 1)$ and $(N_B - 1)$ degrees of freedom for groups 1 and 2 respectively. This test is done with the larger of the variances in the numerator.

Significance levels

The testing of each of these parameters for the difference between two ellipses is even more complicated than the difference between two mean centers since there are up to six parameters which must be tested (differences in mean X, mean Y, angle of rotation, standard deviation along transformed X axis, standard deviation along transformed Y axis, and area of ellipse). However, as with differences in mean center of two groups, there are two different interpretations of differences.

Comparison I: That the two ellipses differ on ANY of the parameters

Comparison II: That the two ellipses differ on ALL parameters.

In the first case, the critical probability level, α, must be divided by the number of parameters being tested, α/p. In theory, this could involve up to six tests, though in practice some of these may not be tested (e.g., the angle of rotation). For example, if five of the parameters are being estimated, then the critical probability level at $\alpha \leq .05$ is actually $\alpha \leq .01$ (.05/5).

In the second case, the critical probability level, α, is multiplied by the number of parameters being tested, $\alpha * p$, since *all* tests must be significant for the two ellipses to be considered as different. For example, if five of the parameters are being estimated, then the critical probability level, say, at $\alpha \leq .05$ is actually $\alpha \leq .25$ (.05*5).

Differences in Mean Direction Between Two Groups

Statistical tests of different angular distributions can be made with the directional mean and variance statistics. To test the difference in the angle of rotation between two groups, a Watson-Williams test can be used (Kanji, 1993; 153-54). The steps in the test are as follows:

1. All angles, θ_i, are converted into radians

$$Radian_i = Angle_i * \pi/180 \quad (B.14)$$

2. For each sample separately, *A* and *B*, the following measures are calculated

$$C_j = \sum_{A=1}^{N_1} \cos \theta_j \qquad S_j = \sum_{A=1}^{N_1} \sin \theta_j \tag{B.15}$$

$$C_k = \sum_{B=1}^{N_2} \cos \theta_k \qquad S_k = \sum_{B=1}^{N_2} \sin \theta_k \tag{B.16}$$

where θ_j and θ_k are the individual angles for the respective groups, *A* and *B*.

3. Calculate the resultant lengths of each group

$$R_A = \text{SQRT}[\, C_A^2 + S_A^2 \,] \tag{B.17}$$

$$R_B = \text{SQRT}[\, C_B^2 + S_B^2 \,] \tag{B.18}$$

4. Resultant lengths for the combined sample are calculated as well as the length of the resultant vector.

$$C = C_A + C_B \tag{B.19}$$

$$S = S_A + S_B \tag{B.20}$$

$$R = \text{SQRT}[\, C^2 + S^2 \,] \tag{B.21}$$

$$N = N_A + N_B \tag{B.22}$$

$$R^* = \frac{(R_A + R_B)}{N} \tag{B.23}$$

5. An F-test of the two angular means is calculated with

$$F = g\,(N - 2)\,\frac{R_A + R_B - R}{N - (R_A + R_B)} \tag{B.24}$$

where

$$g = 1 - \frac{3}{8k} \tag{B.25}$$

with k being identified from a maximum likelihood Von Mises distribution by referencing R^* with 1 and N-2 degrees of freedom (Mardia, 1972; Gaile and Burt, 1980). Some of the reference k's are given in table B.1 (from Mardia, 1972; Kanji, 1993, table 38).

Table B.1

Maximum Likelihood Estimates for Given R^* in the Von Mises Case
(from Mardia, 1972; Kanji, 1993, table 38)

R^*	k
0.00	0.00000
0.05	0.10013
0.10	0.20101
0.15	0.30344
0.20	0.40828
0.25	0.51649
0.30	0.62922
0.35	0.74783
0.40	0.87408
0.45	1.01022
0.50	1.15932
0.55	1.32570
0.60	1.51574
0.65	1.73945
0.70	2.01363
0.75	2.36930
0.80	2.87129
0.85	3.68041
0.90	5.3047
0.95	10.2716
1.00	infinity

Table B.2

Comparison of Two Groups for Angular Measurements
Angle of Deviation From Due North

Group A		Group B	
Incident	**Measured Angle**	**Incident**	**Measured Angle**
1	160	1	196
2	184	2	212
3	240	3	297
4	100	4	280
5	95	5	235
6	120	6	353
		7	190
		8	340

6. Reject the null hypothesis of no angular difference if the calculated F is greater than the critical value $F_{1, N-2}$.

Example 2: Angular comparisons between two groups

A fourth example is that of sets of angular measurements from two different groups, A and B. Table B.2 provides the data for the two sets. The angular mean for Group A is 144.83^0 with a directional variance of 0.35 while the angular mean for Group B is 258.95^0 with a directional variance of 0.47. The higher directional variance for Group B suggests that there is more angular variability than for Group A.

Using the Watson-Wheeler test, we compare these two distributions.

1. All angles are converted into radians (equation B.14).

2. The cosines and sines of each angle are taken and are summed within groups (equations B.15 and B.16).

   $C_A = -3.1981$ $S_A = 2.2533$
   $C_B = -.8078$ $S_B = -4.1381$

3. The resultants are calculated (equations B.17 and B.18).

   $R_A = 3.9121$
   $R_B = 4.2162$

4. Combined sample characteristics are defined (equations B.19 through B.23).

   $C = -4.0059$
   $S = -1.8848$
   $R = 4.4271$
   $N = 14$
   $R^* = 0.5806$

5. Once the parameter, k, is obtained (approximated from table 4.1 or obtained from Mardia, 1972 or Kanji, 1993), g is calculated, and an F-test is constructed (equations B.24 and B.25).

   $k = 1.44$
   $g = 0.7396$
   $F = 5.59$

6. The critical F for 1 and 12 degrees of freedom is 4.75 ($p \leq .05$) and 9.33 ($p \leq .01$). The test is significant at the $p \leq .05$ level and we reject the null hypothesis of no angular differences between the two groups. Group A has a different angular distribution than Group B.

Endnotes for Appendix B

1. There are limits to the Bonferoni logic. For example, if there were 10 tests, having a threshold significance level of .005 (.05 / 10) for the 'either/or' conditions and a threshold significance level of .50 (.05 * 10) for the 'both/and' would lead to an excessively difficult test in the first case and a much too easy test in the second. Thus, the Bonferoni logic should be applied to only a few tests (e.g., 5 or fewer).

Appendix C

Ordinary Least Squares and Poisson Regression Models

by
Luc Anselin
University of Illinois
Champaign-Urbana, IL

This note provides a brief description of the statistical background, estimators and model characteristics for a regression specification, estimated by means of both Ordinary Least Squares (OLS) and Poisson regression.

Ordinary Least Squares Regression

With an assumption of normality for the regression error term, OLS also corresponds to Maximum Likelihood (ML) estimation. The note contains the statistical model and all expressions that are needed to carry out estimation and essential model diagnostics. Both concise matrix notation as well as more extensive full summation notation are employed, to provide a direct link to "loop" structures in the software code, except when full summation is too unwieldy (e.g., for matrix inverse). Some references are provided for general methodological descriptions.

Statistical Issues

The classical multivariate linear regression model stipulates a linear relationship between a *dependent* variable (also called a response variable) and a set of *explanatory* variables (also called independent variables, or covariates). The relationship is stochastic, in the sense that the model is not exact, but subject to random variation, as expressed in an *error* term (also called disturbance term).

Formally, for each observation i, the value of the dependent variable, y_i is related to a sum of K explanatory variables, x_{ih}, with h=1,...,K, each multiplied with a regression *coefficient*, β_h, and the random error term, ε_i:

$$y_i = \sum_{h=1}^{K} x_{ih} \beta_h + \varepsilon_i \qquad \text{(C-1)}$$

Typically, the first explanatory variable is set equal to one, and referred to as the *constant term*. Its coefficient is referred to as the *intercept*, the other coefficients are *slopes*. Using a constant term amounts to extracting a mean effect and is equivalent to using all variables as deviations from their mean. In practice, it is highly recommended to *always* include a constant term.

In matrix notation, which summarizes all observations, i=1,...,N, into a single compact expression, an N *by 1* vector of values for the dependent variable, y is related to an N *by* K matrix of values for the explanatory variables, X, a K *by 1* vector of regression coefficients, β, and an N *by 1* vector of random error terms, ε:

$$y = X\beta + \varepsilon \qquad \text{(C-2)}$$

This model stipulates that on average, when values are observed for the explanatory variables, X, the value for the dependent variable equals $X\beta$, or:

$$E(y \mid X) = X\beta \tag{C-3}$$

where E[|] is the conditional expectation operator. This is referred to as a specification for the conditional mean, conditional because X must be observed. It is a theoretical model, built on many assumptions. In practice, one does not know the coefficient vector, β, nor is the error term observed.

Estimation boils down to finding a "good" value for the β, with known statistical properties. The statistical properties depend on what is assumed in terms of the characteristics of the distribution of the unknown (and never observed) error term. To obtain a Maximum Likelihood estimator, the complete distribution must be specified, typically as a normal distribution, with mean zero and variance, σ^2. The mean is set to zero to avoid systematic under- or over-prediction. The variance is an unknown characteristic of the distribution that must be estimated together with the coefficients, β. The estimate for β will be referred to as b (with b_h as the estimate for the individual coefficient, β_h).

The *estimator* is the procedure followed to obtain an estimate, such as OLS, for b_{OLS}, or ML, for b_{ML}. The *residual* of the regression is the difference between the observed value and the *predicted value*, typically referred to as e. For each observation,

$$e_i = y_i - \sum_{h=1}^{K} x_{ih}\,\beta_h \tag{C-4}$$

or, in matrix notation, with $\hat{y}=Xb$ as short hand for the vector of predicted values,

$$e=y-\hat{y} \tag{C-5}$$

Note that the residual is *not* the same as the error term, but only serves as an estimate for the error. What is of interest is not so much the individual residuals, but the properties of the (unknown) error distribution. Within the constraints of the model assumptions, some of the characteristics of the error distribution can be estimated from the residuals, such as the error variance, σ^2, whose estimate is referred to as s^2.

Because the model has a random component, the observed y are random as well, and any "statistic" computed using these observed data will be random too. Therefore, the estimates b will have a distribution, intimately linked to the assumed distribution for the error term. When the error is taken to be normally distributed, the regression coefficient will also follow a normal distribution. Statistical inference (significance tests) can be carried out once the characteristics (parameters) of that distribution have been obtained (they are never known, but must be estimated from the data as well). An important result is that OLS is *unbiased*. In other words, the mean of the distribution of the estimate b is β, the true, but unknown, coefficient, such that "on average," the estimation is on target. Also, the variance of the distribution of b is directly related to the variance of the error term (and the values for the X). It can be computed by replacing σ^2 by its estimate, s^2.

An extensive discussion of the linear regression model can be found in most texts on linear modeling, multivariate statistics, or econometrics, for example, Rao (1973), Greene (2000), or Wooldridge (2002).

Ordinary Least Squares Estimator

In its most basic form, OLS is simply a fitting mechanism, based on minimizing the sum of squared residuals or residual sum of squares (RSS). Formally, b_{OLS} is the vector of parameter values that minimizes

$$RSS = \sum_{i=1}^{N} e_i^2 = \sum_{i=1}^{N} (y_i - \sum_{h=1}^{K} x_{ih} b_h)^2 \qquad \text{(C-6)}$$

or, in matrix notation,

$$RSS = e'e = (y-Xb)'(y-Xb) \qquad \text{(C-7)}$$

The solution to this minimization problem is given by the so-called *normal equations*, a system of K equations of the form:

$$\sum_{i=1}^{N} (y_i - \sum_{h=1}^{K} x_{ih} b_h) x_{ih} = 0 \qquad \text{(C-8)}$$

for $h=1$ *to* K, or, in matrix notation,

$$X'(y-Xb) = 0 \qquad \text{(C-9)}$$
$$X'Xb = X'y \qquad \text{(C-10)}$$

The solution to this system of equations yields the familiar matrix expression for b_{OLS}:

$$b_{OLS} = (X' X)^{-1} X'y \qquad \text{(C-11)}$$

An estimate for the error variance follows as

$$s^2_{OLS} = \sum_{i=1}^{N} (y_i - \sum_{h=1}^{K} x_{ih} b_{OLS,h})^2 / (N - K) \qquad \text{(C-12)}$$

or, in matrix notation,

$$s^2_{OLS} = e'e / (N-K) \qquad \text{(C-13)}$$

It can be shown that when the X are *exogenous*[1] only the assumption that $E[\varepsilon]=0$ is needed

[1] In practice, this means that each explanatory variable must be uncorrelated with the error term. The easiest way to ensure this is to assume that the X are fixed. But even when they are not, this property holds, as long as the randomness in X and ε are not related. In other words, knowing something about the value of an explanatory variable should *not* provide any information about the error term. Formally, this means that X and ε must be orthogonal, or $E[X'\varepsilon]=0$. Failure of this assumption will lead to so-called simultaneous equation bias.

to show that the OLS estimator is *unbiased*. With the additional assumption of a fixed error variance s^2, OLS is also most *efficient*, in the sense of having the smallest variance among all other linear and unbiased estimators. This is referred to as the BLUE (Best Linear Unbiased Estimator) property of OLS. Note, that in order to obtain these properties, no additional assumptions need to be made about the distribution of the error term. However, to carry out statistical inference, such as significance tests, this is insufficient, and further characteristics of the error distribution need to be specified (such as assuming a normal distribution), or asymptotic assumptions need to be invoked in the form of laws of large numbers (typically yielding a normal distribution).

Maximum Likelihood Estimator

When the error terms are assumed to be independently distributed as normal random variables, OLS turns out to be equivalent to ML.

Maximum Likelihood estimation proceeds as follows. First, consider the density for a single error term:

$$\varepsilon \sim N(0, \sigma^2), \text{ or} \tag{C-14}$$

$$f([\varepsilon]_i|s^2) = \frac{1}{\sqrt{2\Pi\sigma^2}} e^{-(1/2)(\varepsilon_i^2/\sigma^2)} \tag{C-15}$$

A subtle, but important, point is that the error itself is not observed, but only the "data" (y and X) are. We move from a model for the error, expressed in unobservables, to a model that contains observables and the regression parameter by means of a standard "transformation of random variables" procedure. Since y_i is a linear function of ε it will also be normally distributed. Its density is obtained as the product of the density of ε and the "Jacobian" of the transformation, using $\varepsilon_i = y_i - x_i\beta$ (with x_i as the i-th row in the X matrix). As it turns out, the Jacobian is one, so that

$$f([y_i|\beta]_i|s^2) = \frac{1}{\sqrt{2\Pi\sigma^2}} e^{-(1/2)((y_i - x_i\beta)^2/\sigma^2)} \tag{C-16}$$

The likelihood function is the joint density of all the observations, given a value for the parameters β and σ^2. Since independence is assumed, this is simply the product of the individual densities from equation C-16. The log-likelihood is then the log of this product, or the sum of the logs of the individual densities. The contribution to the log likelihood of each observation follows from equation C-16:

$$\begin{aligned} &\text{Log } f(y_i|\beta, \sigma^2) = \\ &L_i = -(1/2)\log(2\pi) - (1/2)\log(\sigma^2) - (1/2)[(y_i - x_i\beta)^2/\sigma^2] \end{aligned} \tag{C-17}$$

The full log-likelihood follows as:

$$L = \sum_{i=1}^{N} L_i = -(N/2)\log(2\pi) - (N/2)\log(\sigma^2) - (1/2\sigma^2) \sum_{i=1}^{N} (y_i - x_i\beta)^2 \tag{C-18}$$

or, in matrix notation,

$$L = -(N/2)\log(2\pi) - (N/2)\log(\sigma^2) - (1/2\sigma^2)(y - X\beta)'(y - X\beta) \qquad \text{(C-19)}$$

A Maximum Likelihood estimator for the parameters in the model finds the values for β and σ^2 that yield the highest value for equation C-19. It turns out that minimizing the residual sum of squares (or, least squares), the last term in equations C-18 and C-19, is equivalent to maximizing the log-likelihood. More formally, the solution to the maximization problem is found from the first-order conditions (setting the first partial derivatives of the log-likelihood to zero), which yield the OLS estimator for *b* and

$$s^2_{ML} = \sum_{i=1}^{N} e_i^2/N \qquad \text{(C-20)}$$

or, in matrix notation,

$$s^2_{ML} = e'e/N \qquad \text{(C-21)}$$

Inference

With estimates for the parameters in hand, the missing piece is a measure for the precision of these estimates, which can then be used in significance tests, such as t-tests and F-tests. The estimated variance-covariance matrix for the regression coefficients is

$$\text{Var}(b) = s^2(X'X)^{-1} \qquad \text{(C-22)}$$

where s^2 is either s_{OLS}^2 or s_{ML}^2. The diagonal elements of this matrix are the variance terms, and their square root the standard error. Note that the estimated variance using s_{ML}^2 will always be smaller than that based on the use of s_{OLS}^2. This may be spurious, since the ML estimates are based on asymptotic considerations (with a "conceptual" sample size approaching infinity), whereas the OLS estimates use a "degrees of freedom" (*N-K*) correction. In large samples, the distinction between OLS and ML disappears (for very large *N*, *N* and *N-K* will be very close).

Typically, interest focuses on whether a particular population coefficient (the unknown b_h) is different from zero, or, in other words, whether the matching variable contributes to the regression. Formally, this is a test on the null hypothesis that $b_h = 0$. This leads to a t test statistic as the ratio of the estimate over its standard error (the square root of the *h*,*h* element in the variance-covariance matrix), or

$$t = b_h / \sqrt{s^2(X'X)^{-1}_{hh}} \qquad \text{(C-23)}$$

This test statistic follows a Student t distribution with *N-K* degrees of freedom. If, according to this reference distribution, the probability that a value equal to or larger than the t-value (for a one-sided test) occurs is very small, the null hypothesis will be rejected and the

coefficient deemed "significant."[2]

Note that when s_{ML}^2 is used as the estimate for s^2, the t-test is referred to as an "asymptotic" t-test. In practice, this is a standard normal variate. Hence, instead of comparing the t test statistic to a Student t distribution, its probability should be evaluated from the standard normal density.

A second important null hypothesis pertains to all the coefficients taken together (other than the intercept). This is a test on the significance of the regression as a whole, or a test on the null hypothesis that, jointly, $b_h = 0$, for h=2,...,K (note that there are K-1 hypotheses). The F test statistic for this test is constructed by comparing the residual sum of squares (RSS) in the regression to that obtained without a model. The latter is referred to as the "constrained" (i.e., with all the β except the constant term set to zero) residual sum of squares (RSS_C). It is computed as the sum of squares of the y_i in deviation from the mean, or

$$RSS_C = \sum_{i=1}^{N} (y_i - \bar{y})^2 \qquad \text{(C-24)}$$

where $\bar{y} = \sum_{i=1}^{N} y_i/N$. The F statistic then follows as:

$$F = [(RSS_C - RSS) / (K - 1)] / [RSS / (N - K)] \qquad \text{(C-25)}$$

It is distributed as an F-variate with K-1,N-K degrees of freedom.

Model Fit

The most common measure of fit of the regression is the R^2, which is closely related to the F-test. The R^2 departs from a decomposition of the total sum of squares, or the RSS_C from equation C-C-24, into the "explained" sum of squares (the sum of squares of predicted values, in deviations from the mean), and the residual sum of squares, RSS. The R^2 is a measure of how much of this decomposition is due to the "model." It is easily computed as:[3]

$$R^2 = 1 - RSS/RSS_C \qquad \text{(C-26)}$$

In general, the model with the highest R^2 is considered best. However, this may be misleading since it is always possible to increase the R^2 by adding another explanatory variable, irrespective of whether this variable contributes "significantly." The adjusted R^2 (R_a^2) provides a better guide that compensates for "over-fitting" the data by correcting for the number of variables included in the model. It is computed by rescaling the numerator and denominator in equation C-26, as

$$R_a^2 = 1 - [RSS / (N - K)] / [RSS_C / (N - 1)] \qquad \text{(C-27)}$$

---

2 Any notion of significance is always with respect to a given p-value, or Type I error. The Type I error is the chance of making a wrong decision, i.e., of rejecting the null hypothesis when in fact it is true.

3 When the regression specification does not contain a constant term, the value obtained for the R^2 using equation (C-C-26) will be incorrect. This is because the constant term forces the residuals to have mean zero. Without a constant term, the RSS must be computed in the same waysameway as in equation (C-24), by subtracting the average residual $\hat{e} = \Sigma e_i/N$.

For very large data sets, this rescaling will have negligible effect and the R^2 and R_a^2 will be virtually the same.

When OLS is viewed as a ML estimator, an alternative measure of fit is the value of the maximized log-likelihood. This is obtained by substituting the estimates b_{ML} and s_{ML}^2 and into expression C-18 or C-19. With $e = y - Xb_{ML}$ as the residual vector and $s_{ML}^2 = e'e/N$, the log-likelihood can be written in a simpler form:

$$L = -(N/2)\log(2\pi) - (N/2)\log(e'e/N) - (1/2[e'e/N])(e'e) \quad \text{(C-28)}$$

$$= -(N/2)\log(2\pi) - (N/2) - (N/2)\log(e'e/N) \quad \text{(C-29)}$$

Note that the only term that changes with the model fit is the last one, the logarithm of the average residual sum of squares. Therefore, the constant part is not always reported. To retain comparability with other models (e.g., spatial regression models), it is important to be consistent in this reporting. The model with the *highest* maximized log-likelihood is considered to be best, even though the likelihood, as such, is technically not a measure of fit.

Similar to the R_a^2, there exist several corrections of the maximized log-likelihood to take into account potential over-fitting. The better-known measures are the Akaike Information Criterion (AIC) and the Schwartz Criterion (SC), familiar in the literature on Bayesian statistics. They are easily constructed from the maximized log-likelihood. They are, respectively:

$$\text{AIC} = -2L+2K, \quad \text{(C-30)}$$
$$\text{SC} = -2L+K\log(N) \quad \text{(C-31)}$$

The model with the *lowest* information criterion value is considered to be best.

Poisson Regression

Next, the Poisson regression model is examined.

Likelihood Function

In the Poisson regression model, the dependent variable for observation i (with i=1,...,N), y_i is modeled as a Poisson random variate with a mean λ_i that is specified as a function of a K by 1 (column) vector of explanatory variables x_i, and a matching vector of parameters β. The probability of observing y_i is expressed as:

$$\text{Prob}(y_i) = \frac{e^{-\lambda_i}\lambda_i^{y_i}}{y_i!} \quad \text{(C-32)}$$

The conditional mean of y_i, given observations on x_i is specified as an exponential function of x:

$$E[y_i|x_i] = l_i = e^{x_i'b}, \quad \text{(C-33)}$$

where x_i' is a row vector. Equivalently, this is sometimes referred to as a *loglinear* model, since

$$\ln l_i = x_i'b. \tag{C-34}$$

Note that the mean in C-33 is nonlinear, which means that the effect of a change in x_i will depend not only on β (as in the classical linear regression), but also on the value of x_i. Also, in the Poisson model, the mean equals the variance (equidispersion) so that there is no need to separately estimate the latter.

There is a fundamental difference between a classical linear regression model and the specification for the conditional mean in the Poisson regression model, in that the latter does not contain a random error term (in its "pure" form). Consequently, unlike the approach taken for the linear regression, the log-likelihood is not derived from the joint density of the random errors, but from the distribution for dependent variable itself, using C-32. Also, there is no need to estimate a residual variance, as in the classical regression model.

Assuming independence among the count variables (e.g., *excluding* spatial correlation), the log-likelihood for the Poisson regression model follows as:

$$L = \sum_{i=1}^{N} y_i x_i'b - e^{x_i'b} - \ln y_i! \tag{C-35}$$

Note that the third term is a constant and does not change with the parameter values. Some programs may not include this term in what is reported as the log-likelihood. Also, it is not needed in a Likelihood Ratio test, since it will cancel out.

The first order conditions, $\partial L/\partial \beta = 0$, yield a system of K equations (one for each β) of the form:

$$\sum_{i=1}^{N} (y_i - e^{x_i'b}) x_i = 0 \tag{C-36}$$

Note how this takes the usual form of an orthogonality condition between the "residuals" $(y_i - e^{x_i'b})$ and the explanatory variables, x_i. This also has the side effect that when x contains a constant term, the sum of the predicted values, $e^{x_i'b}$ equals the sum of the observed counts.[4] The system C-36 is nonlinear in β and does not have an analytical solution. It is typically solved using the Newton-Raphson method (see section).

Once the estimates of β are obtained, they can be substituted into the log-likelihood (equation C-36) to compute the value of the maximum log-likelihood. This can then be inserted in the AIC and BIC information criteria in the usual way.

Predicted Values and Residuals

The predicted value, $\hat{y}_i$, is the conditional mean or the average number of events, given the x_i. This is also denoted a λ_i and is typically not an integer number, whereas the observed value y_i

[4] A different way of stating this property is to note that the sum of the residuals equals zero. As for the classical linear regression model, this is not guaranteed without a constant term in the regression.

is a count. The use of the exponential function guarantees that the predicted value is non-negative. Specifically:

$$\hat{\lambda}_i = e^{x_i'\hat{b}} \tag{C-37}$$

The "residuals" are simply the difference between observed and predicted:

$$e_i = y_i - e^{x_i'\hat{b}} = y_i - \hat{\lambda}_i \tag{C-38}$$

Note that, unlike the case for the classical regression model, these residuals are not needed to compute estimates for error variance (since there is no error term in the model).

Estimation Steps

The well known Newton-Raphson procedure proceeds iteratively. Starting from a set of estimates $\hat{\beta}_t$ the next value is obtained as:

$$\hat{\beta}_{t+1} = \hat{\beta}_t - \hat{H}_t^{-1}\hat{g}_t \tag{C-39}$$

where $\hat{g}_t$ is the first partial derivative of the log-likelihood, evaluated at $\hat{\beta}_t$ and $\hat{H}_t$ is the Hessian, or second partial derivative, also evaluated at $\hat{\beta}_t$.

In the Poisson regression model,

$$g = \sum_{i=1}^{N} x_i(y_i - \hat{\lambda}_i) \tag{C-40}$$

$$H = -\sum_{i=1}^{N} \hat{\lambda}_i x_i x_i' \tag{C-41}$$

In practice, one can proceed along the following lines.

1. **Set initial values for parameters**, say $b_0[h]$, for h=1,...,K. One can set $b_0[1]=\bar{y}$, the overall average count as the constant term, and the other $b_0[h]$=0, for h=2,...,K.
2. **Compute predicted values** for each i, the value of $\hat{\lambda}_i = e^{x_i'b0}$.
3. **Compute gradient**, g, using the starting values. Note that $g[h]$ is a K by 1 vector. Each element of this vector is the difference between:

$$O_i = \sum_{i=1}^{N} x_{ih}y_i \tag{C-42}$$

$$N$$

$$P_i = \sum_{i=1} x_{ih}\lambda_i \tag{C-43}$$

$$g_i = O_i - P_i \tag{C-44}$$

Note that C-42 does not contain any unknown parameters and needs only to be computed once (provided there is sufficient storage). As the Newton-Raphson iterations proceed, the values of g will become very small.

4. **Compute the Hessian**, H, using the starting values. H is a K by K matrix (C-41) that needs to be inverted at each iteration in C-39. It is *not* the $X'X$ of the classical model, but rather more like $X'\Sigma X$, where Σ is a diagonal matrix. One way to implement this is to multiply each row of the X matrix by $\sqrt{\hat{\lambda}_i}$, e.g., $xs[i][h]=x[i][h]*\text{sqrt}(\hat{\lambda}[i])$, where xs is the new matrix (X^*), i is the observation (row) and h the column of X. The Hessian then becomes the cross product of the new matrices, or, $H=X^{*'}X^*$. This needs to be done at each iteration. There is no need to take a negative since the negative in C-41 and in C-39 cancel.

5. **Update the estimate** for the $b[h]$, say $b_1[h]$ is obtained using the updating equation C-39 except that the product $H^{-1}g$ is added to the initial value. In general, for iteration t, the new estimates are obtained as b_{t+1}. After checking for convergence, the old b_t is set to b_{t+1} and inserted in the computation of the predicted values, in step 2 above.

6. **Convergence**. Stop the iterations when the difference between b_{t+1} and b_t becomes below some tolerance level. A commonly used criterion is the norm of the difference vector, or $\Sigma_h (b_{t+1}[h] - b_t[h])^2$. When the norm is below a preset level, stop the iterations and report the last b_t as the result. The reason for not using b_{t+1} is that the latter would require an extra computation of the Hessian needed for inference.

Inference

The asymptotic variance matrix is the inverse Hessian obtained at the last iteration (i.e., using b_t). The variance of the estimates are the diagonal elements, the standard errors their square roots. The asymptotic t-test is constructed in the usual way, as the ratio of the estimate over its standard error. The only difference with the classic linear regression case is that the p-values must be looked up in a standard normal distribution, not a Student t distribution.

Likelihood Ratio Test

A simple test on the overall fit of the model, as an analogue to the F-test in the classical regression model is a Likelihood Ratio test on the "slopes". The model with only the intercept is nothing but the mean of the counts, or

$$\lambda_I = \bar{y}\,\forall \tag{C-45}$$

with $\bar{y}=\sum_{i=1}^{N} y_i/N$.

The corresponding log-likelihood is:

$$L_R=-N\bar{y}+\ln(\bar{y})(\sum_{i=1}^{N} y_i)-\sum_{i=1}^{N}\ln y_i!, \qquad \text{(C-46)}$$

where the *R* stands for the "restricted" model, as opposed to the "unrestricted" model with *K-1* slope parameters. The last term in C-46 can be dropped, as long as it is also dropped in the calculation of the maximized likelihood (C-35) for the unrestricted model (L_U), using $l_i=e^{x_i'b_t}$. The Likelihood Ratio test is then:

$$LR=2(L_U-L_R), \qquad \text{(C-47)}$$

and follows a χ^2 distribution with *K*-1 degrees of freedom.

References

Gentle, J. E. (1998). *Numerical Linear Algebra for Applications in Statistics*. Springer-Verlag, New York, NY.

Gentleman, W. M. (1974). Algorithm AS 75: Basic procedures for large, sparse or weighted linear least problems. *Applied Statistics*, 23: 448–454.

Greene, W. H. (2000). *Econometric Analysis, 4th Ed.* Prentice Hall, Upper Saddle River, NJ.

Miller, A. J. (1992). Algorithm AS 274: Least squares routines to supplement those of Gentleman. *Applied Statistics*, 41: 458–478.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1992). *Numerical Recipes in C. The Art of Computing (Second Edition)*. Cambridge University Press, Cambridge, UK.

Rao, C. R. (1973). *Linear Statistical Inference and its Applications*. Wiley, New York, 2nd edition.

Wooldridge, J. M. (2002). *Econometric Analysis of Cross Section and Panel Data*. MIT Press, Cambridge, MA.