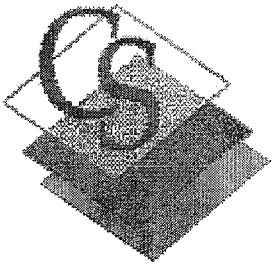
CrimeStat® II

VERSION 2.0

A Spatial Statistics Program for the Analysis of Crime Incident Locations



Ned Levine & Associates

Houston, TX

Pt. 1

The National Institute of Justice

Washington, DC

May 2002

195772

FINAL REPORT

Approved By: _____

Date: 6/24/02

195772
pt. 1

Table of Contents

Table of Contents (continued)

Table of Contents (continued)

Table of Contents (continued)

Table of Contents (continued)

Acknowledgments

CrimeStat II was developed under the direction of Dr. Ned Levine of *Ned Levine & Associates*, Houston, TX, with Grant No. 1999-IJ-CX-0044 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, the programmer for the project. Mr. Doan's brilliance in programming was essential to the development of the program; much of the innovation and efficiency of the program is due to his efforts.

2. Mr. Phil Canter of the *Baltimore County Police Department*, Towson, MD who co-sponsored the effort and provided support and data for analysis.

3. The dedicated professionals at the Mapping and Analysis for Public Safety Program (formerly the Crime Mapping Research Center) at NIJ: Ms. Elizabeth Groff, project director, now at the Institute for Law and Justice; Mr. Eric Jefferis; Dr. Robert Langworthy, now at the University of Alaska, and, for the first version, Ms. Cindy Mamalian; and Dr. Nancy LaVigne, now at the Urban Institute.

4. Professor Richard Block of Loyola University in Chicago and Mr. Dan Helms of the Las Vegas Police Department who served as criminal justice advisors to the project.

5. Dr. Carolyn Rebecca Block of the Illinois Criminal Justice Information Authority who allowed us to incorporate their STAC program in this version of *CrimeStat*.

6. Professor Eric Renshaw of the University of Strathclyde in Glasgow and Professor Luc Anselin of the University of Illinois at Urbana-Champaign for providing statistical advice to the project.

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

8. Mr. John DeVoe, now of Siebel Systems, but formerly of the Criminal Division, U. S. Department of Justice, who integrated *CrimeStat* into their Regional Crime Analysis Geographic Information System.

9. Professor Jim LeBeau of Southern Illinois University, Bryan Hill of the Glendale (Arizona) Police Department, and Martin Hittleman of Valley Community College in Los Angeles for providing extensive feedback in improving the program,

10. To the individuals who provided one-page applications 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; Silvana Amaral, Antônio Miguel V. Monteiro, Gilberto Câmara, and José A. Quintanilha of the Instituto Nacional de Pesquisas Espaciais, 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 University of Texas-Houston; Ron Wilson of the University of Michigan and the National Institute of Justice, Chaosheng Zhang of the National University of Ireland in Galway, Ireland, along with Richard Block, Carolyn Block, Phil Canter, Jim LeBeau, and Bryan Hill mentioned above.

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

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

License Agreement and Disclaimer

This project was supported by Grant No. 1999-IJ-CX-0044 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

Ms. Debra Stoe
Mapping and Analysis for Public Safety Program
National Institute of Justice
U. S. Department of Justice
810 7th St, NW
Washington, DC 20531
maps@ojp.usdoj.gov

CrimeStat II

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 2000*® 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.

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 2000*® 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 graphic 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 II includes routines for:

Spatial distribution

Mean center
Standard distance deviation
Standard deviational ellipse
Median center
Center of minimum distance
Moran's I spatial autocorrelation index
Geary's C spatial autocorrelation index

Directional mean and variance

Distance analysis

Nearest neighbor analysis
Ripley's K statistic
Intra- and Inter-distance matrices

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
Journey-to-crime calibration
Journey-to-crime estimation

Space-time analysis

Knox index
Mantel index
Correlated walk model

Many of these routines have variable parameters allowing 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.

What the Program Does and Does Not Do

CrimeStat provides descriptions of the spatial arrangements of crime incidents. There are a variety of tools that can be used to describe these arrangements from the analysis of central tendency and one- and two-dimensional dispersion to the analysis of the distances between incident locations to identification of collections of incidents that cluster together ('hot spots') to three-dimensional models of crime density to the analysis of serial events. These tools are useful in helping crime analysts detect patterns of crime and provide different perspectives on the arrangements. In this sense, it is a tool for analyzing one or, at most, two variables affecting crime incidence - the incidents themselves and a secondary variable that can be used for comparison.

On the other hand, *CrimeStat* is not a standard statistical package aimed at modeling correlates or determinants of crime incidents. It does not have a regression module nor other multivariate techniques quantifying the predictors of crime locations. Users who want to model determinants of crime can use specialized regression packages, such as *SpaceStat*© (Anselin, 1992) or *S-Plus*® (Insightful Corp., 2001).

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

CrimeStat runs on a *Windows NT*, *Windows 2000*, or *Windows XP* system; it is not hardware dependent so that any processor that can run *Windows NT/ 2000/ XP* will suffice. While it can run on a relatively slow computer (e.g., 75 MHZ clock speed) with limited RAM (e.g., 8 MB), it will run much better on a 800 MHZ Pentium III computer (or faster) with more than 128 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 large matrices (e.g., the distance from every point to every other point). 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 2000* system with 256 MB of RAM running a duel-processor Pentium III 800 mhz computer.

In addition, *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 NT, Windows 2000*, or *Windows XP Professional. Windows NT*, *Windows 2000 Professional*, and *Windows XP Professional* support two processors; *Windows 2000 Server* and, presumably, *Windows XP Server* support four processors, while *Windows 2000 Advanced Server* and *Windows 2000 Advanced Server* (Microsoft, 2000) support up to 32 processors. However, neither *Windows 95* (Microsoft, 1995) *Windows 98* (Microsoft, 1998c), nor *Windows XP Home* will recognize multiple processors. Thus, if there are two processors and *Windows NT* or *Windows 2000* is the operating system, *CrimeStat* will calculate routines in about half the time. If there are

four processors and *Windows 2000 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 800 MHZ single-processor *Pentium III* computer with 256 MB of RAM running *Windows 2000*, it takes about 14 minutes to complete a nearest neighbor analysis on 19,191 cases involving the calculating of distance from every point to every other point multiple times (for different neighbors). On a dual-processor *Pentium III* computer with 256 MB of RAM running *Windows 2000*, it takes about 7 minutes to complete the same task. Slower systems will produce correspondingly slower times. For example, on a single processor 133 MHZ Pentium computer with 48 MB of RAM running *Windows 95*, it takes about an hour and a half to finish this run. 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 NT/ 2000/ XP* operating system so it is better optimized for that system. In particular, *Windows NT/ 2000/ XP* 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 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 *NT* than in *98* which, in turn, will run more efficiently than *95*.

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.

Other Versions of Windows

While *CrimeStat* was designed in a *Windows NT/ 2000/ XP* environment, it will run properly in *Windows 98* and *Windows 95*. In Windows 98 and Windows 95, however, it will

not be possible to take advantage of multiple processors since those operating systems don't support multi-threading.

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 three 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.

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 and III. Part II presents a number of statistics for spatial description while part III presents a number of statistics for spatial modeling. 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.

Chapter 1 Endnotes

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*®, paralleling the online help menus in the program. Detailed instructions should be obtained from chapters 3-9 in the documentation. *CrimeStat* has four basic groupings in ten 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
7. 'Hot Spot' analysis

Spatial modeling

8. Interpolation
9. Journey to crime estimation
10. Space-time analysis

Options

11. Saving parameters, colors and options

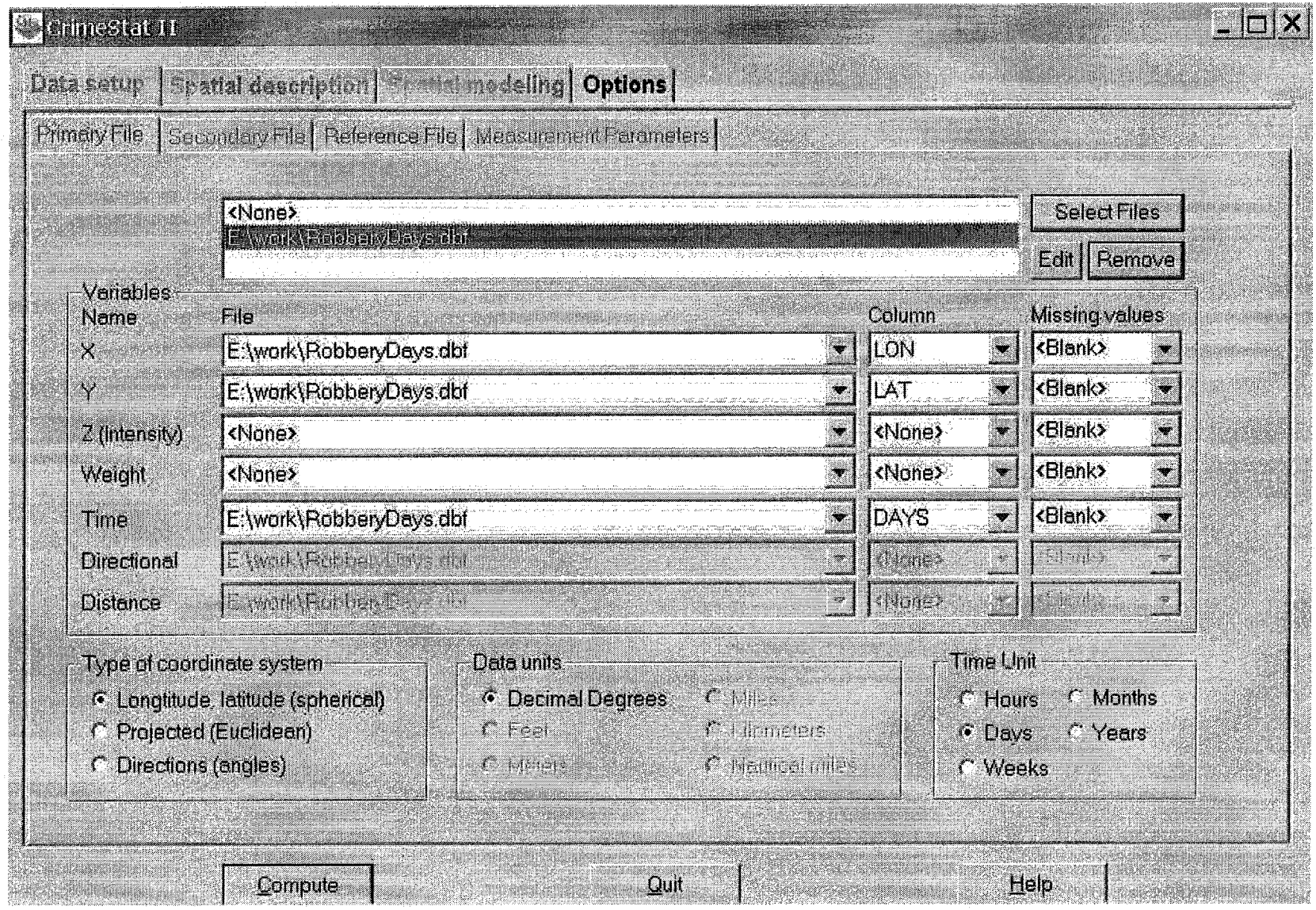
Figure 2.1-2.10 show the ten operational tab screens with examples of data input and routine selection.

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

Figure 2.1: Primary File Screen



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 (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.

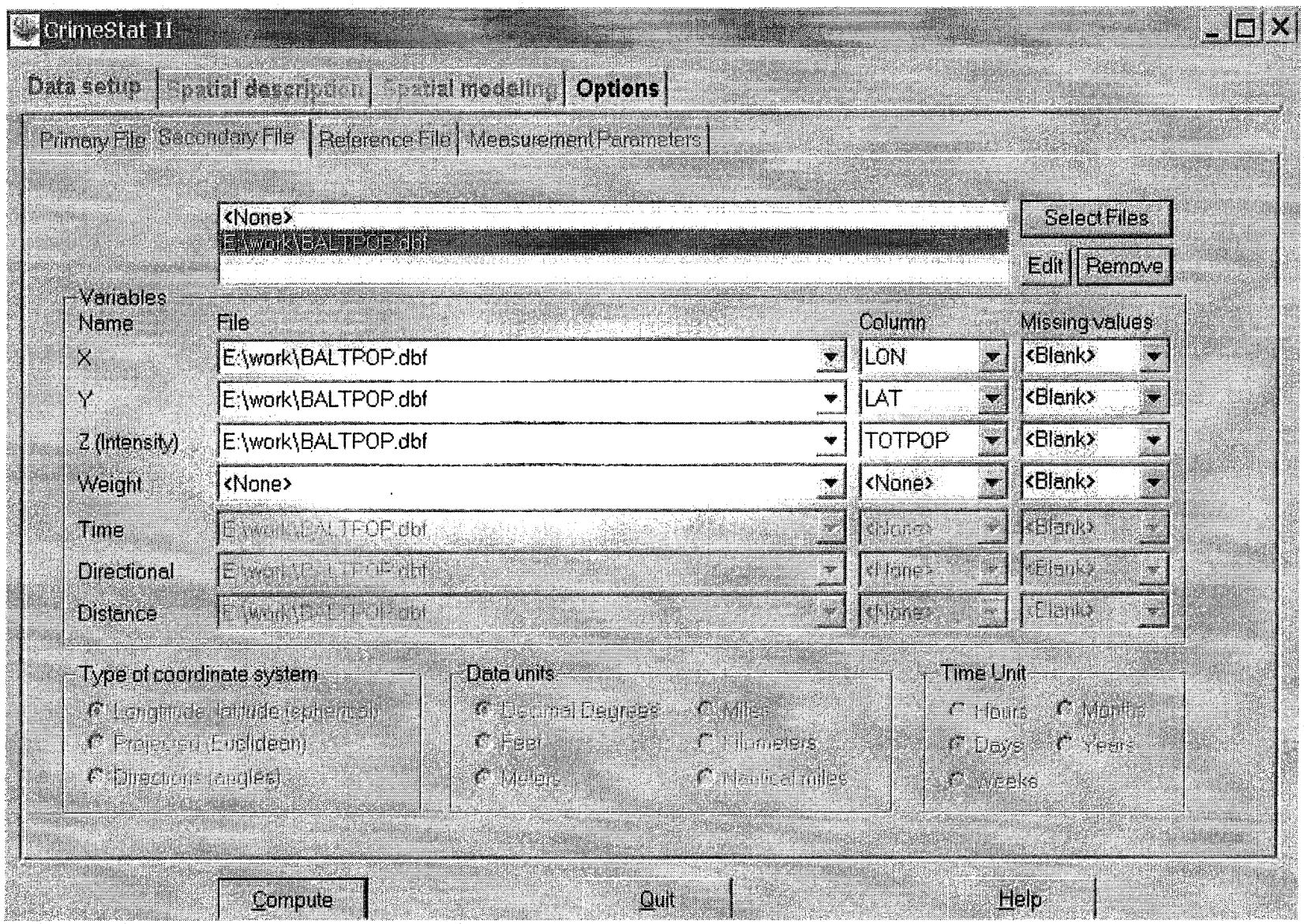
Secondary File

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 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

Figure 2.2: Secondary File Screen



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.

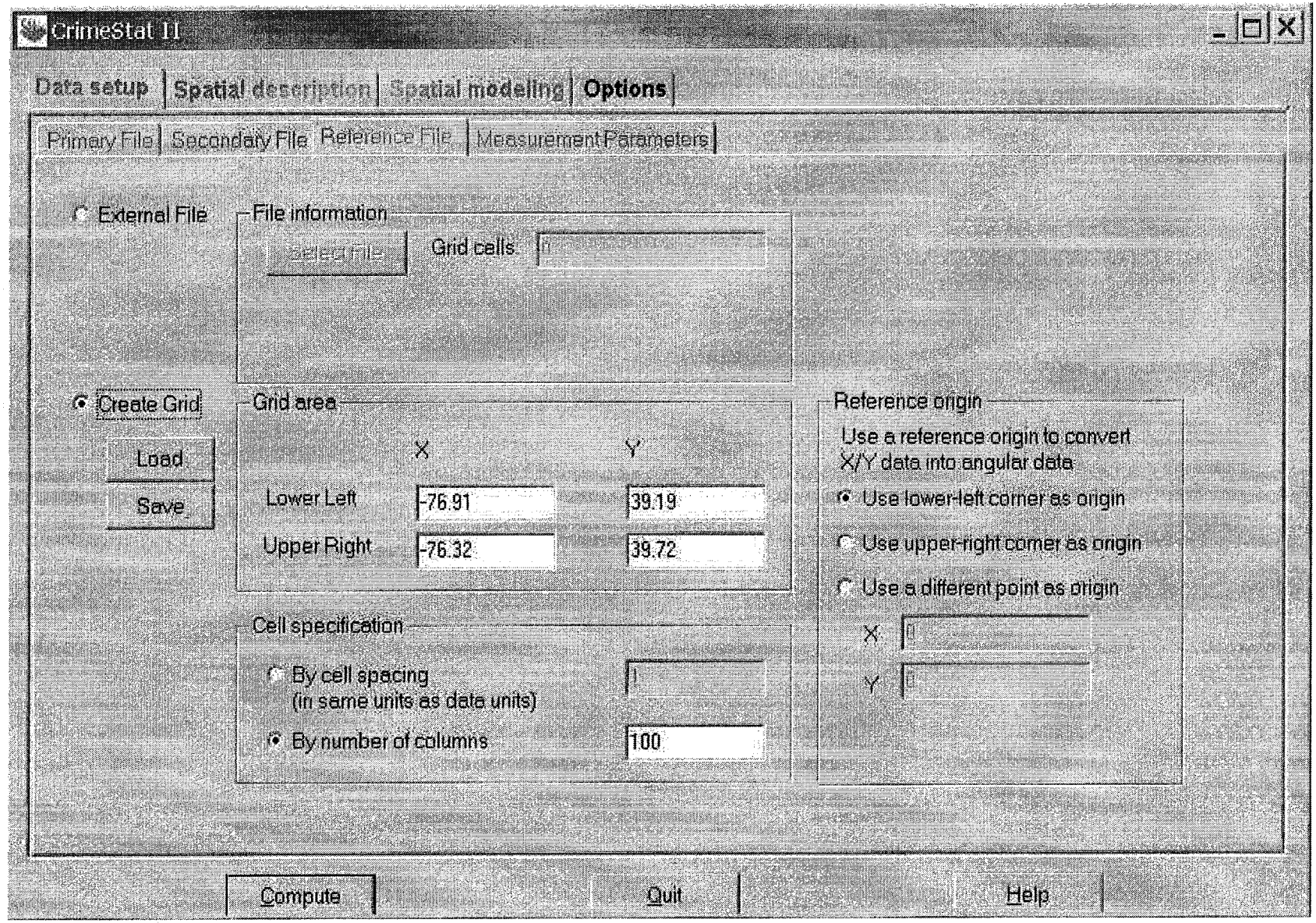
A reference file that is read into *CrimeStat* need not be a true grid (a matrix with k columns and l rows). However, an external reference file that is read in can only be output to *Surfer for Windows* since the other output formats *ArcView*, *MapInfo*, *Atlas*GIS*, *ArcView Spatial Analyst*, and 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

Figure 2. 3: Reference File Screen



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 define the measurement units of the coverage and the type of distance measurement to be used.

Area

Define the geographical area of the study area in area units (square feet, square meters, square miles, square kilometers, square nautical miles). 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* wi define a rectangle by the minimum and maximum X and Y coordinates.

Length of street network

Define the total length of the street network within the study area or an appropriate comparison network (e.g., freeway system) in distance units (feet, meters, miles, kilometers, or nautical miles). 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

Select the type of distance measurement to be used, direct or indirect.

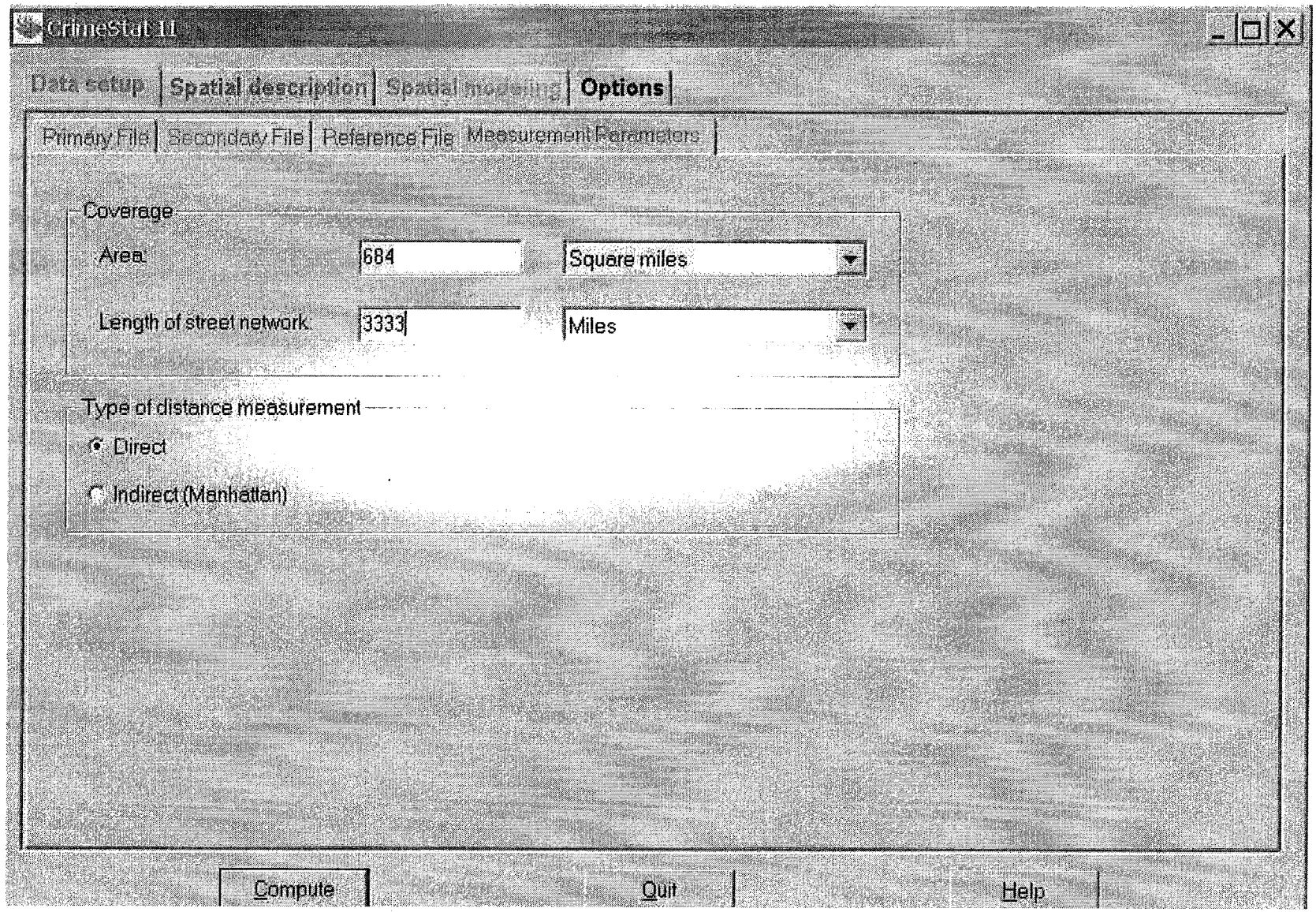
Direct

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.

Figure 2.4: Measurement Parameters Screen



If the coordinates are projected, then the shortest indirect distance is the right angle of a right triangle on a two-dimensional plane.

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.

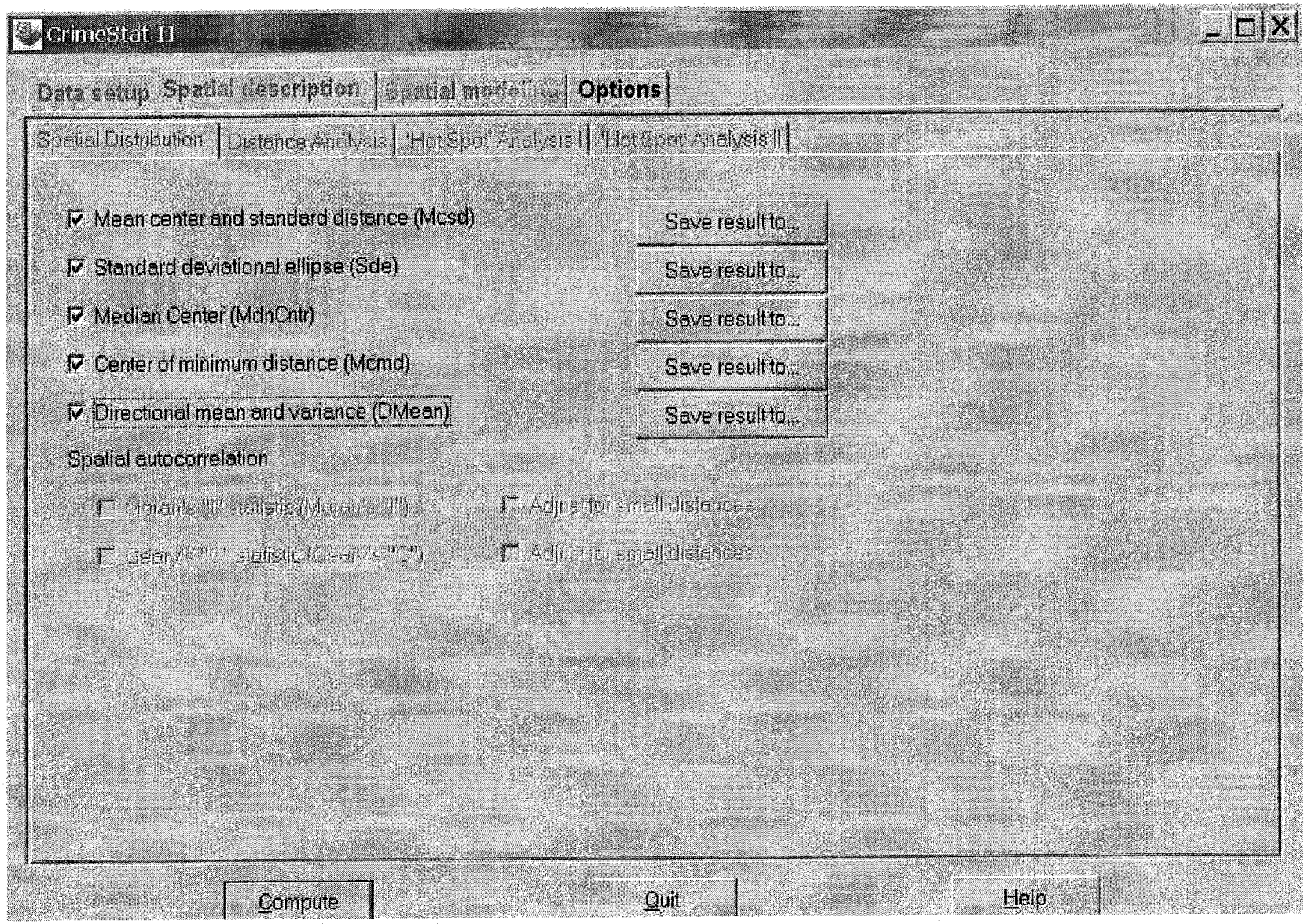
Mean Center and Standard Distance (Mcsd)

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
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',

Figure 2.5: Spatial Distribution Screen



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.

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.

Distance Analysis

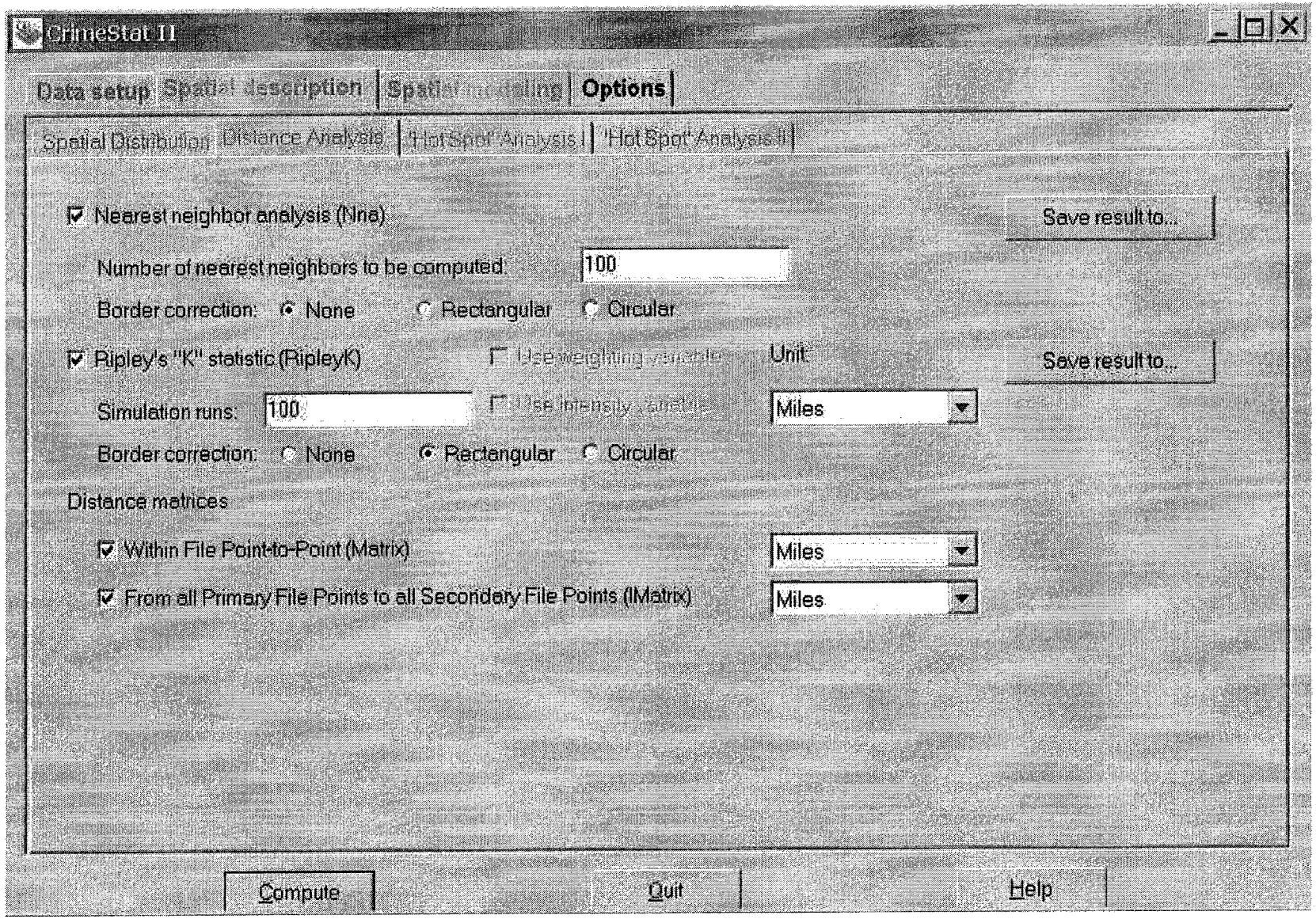
Distance analysis provides statistics about the distances between point locations. It is useful for identifying the degree of clustering of points. It is sometimes called second-order analysis. There are three routines for describing properties of the distances and there are two routines that output distance matrices.

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)

Figure 2.6: Distance Analysis Screen



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. See chapter 5 for more information.

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. See chapter 5 for more information.

Distance Matrices

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 (feet, meters, miles, kilometers, or nautical miles). The Matrix output can 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 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.

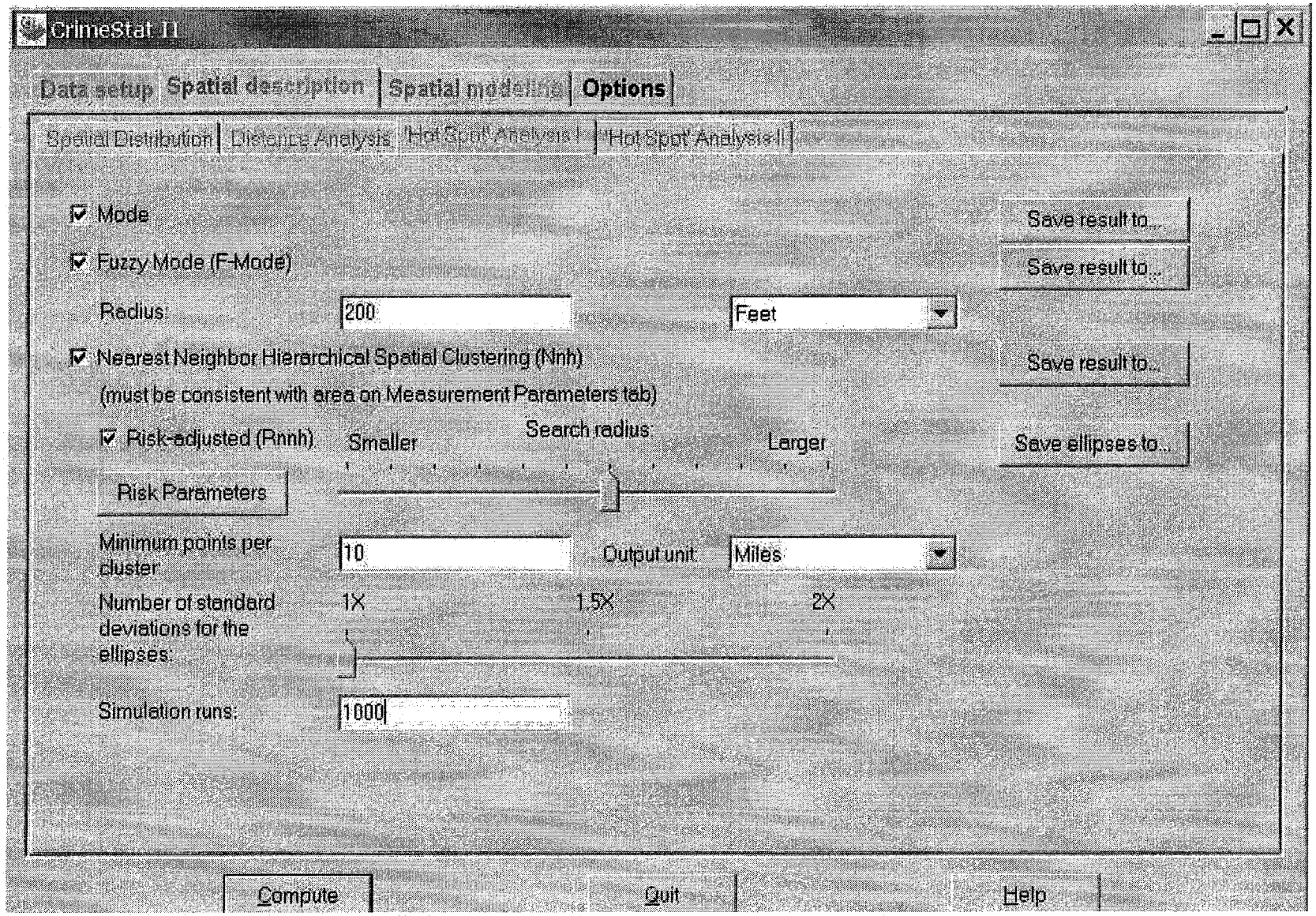
Fuzzy Mode

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 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 results can be printed, saved to a text file, output as a '.dbf' file, or output as ellipses to *ArcView* '.shp', *MapInfo* '.mif' or *Atlas*GIS* '.bna' files

Figure 2.7: **'Hot Spot' Analysis I 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 cluster
6. The density of the cluster (points divided by area)

Nnh threshold distance

The threshold distance is the confidence interval around a random expected distance for a *pair* of points. 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. Move the slide bar to the left to choose a smaller threshold distance and to the right to choose a larger threshold distance.

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.

Output size for Nnh 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.

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.

Nnh simulation runs

A Monte Carlo simulation can be run to estimate the approximate confidence intervals around the Nnh 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 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 results can be printed, saved to a text file, output as a '.dbf' file, or output as ellipses to *ArcView* '.shp', *MapInfo* '.mif' or *Atlas*GIS* '.bna' files

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 (rotation, length of 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 in 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 gives a larger threshold distance and, usually, more pairs that 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 of the 2500 grid cells, 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. 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.

Output size for Rnnh 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.

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.

Rnnh simulation runs

A Monte Carlo simulation can be run to estimate the approximate confidence intervals around the Rnnh 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 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.

'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 to *ArcView* '.shp', *MapInfo* '.mif' or *Atlas*GIS* '.bna' files

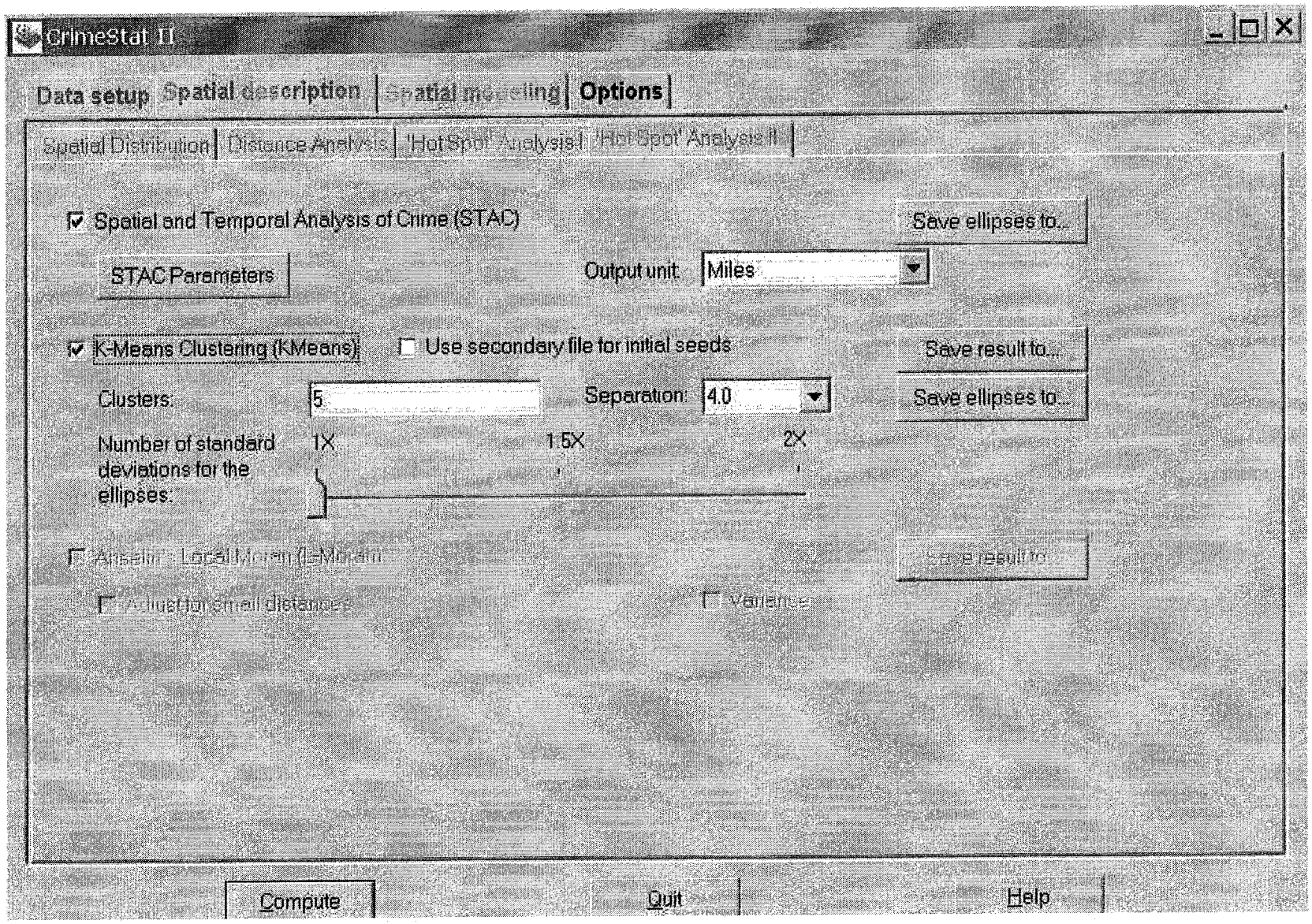
The routine outputs six results for each cluster that is calculated:

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.

Figure 2.8: 'Hot Spot' Analysis II Screen



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.

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.

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.

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.

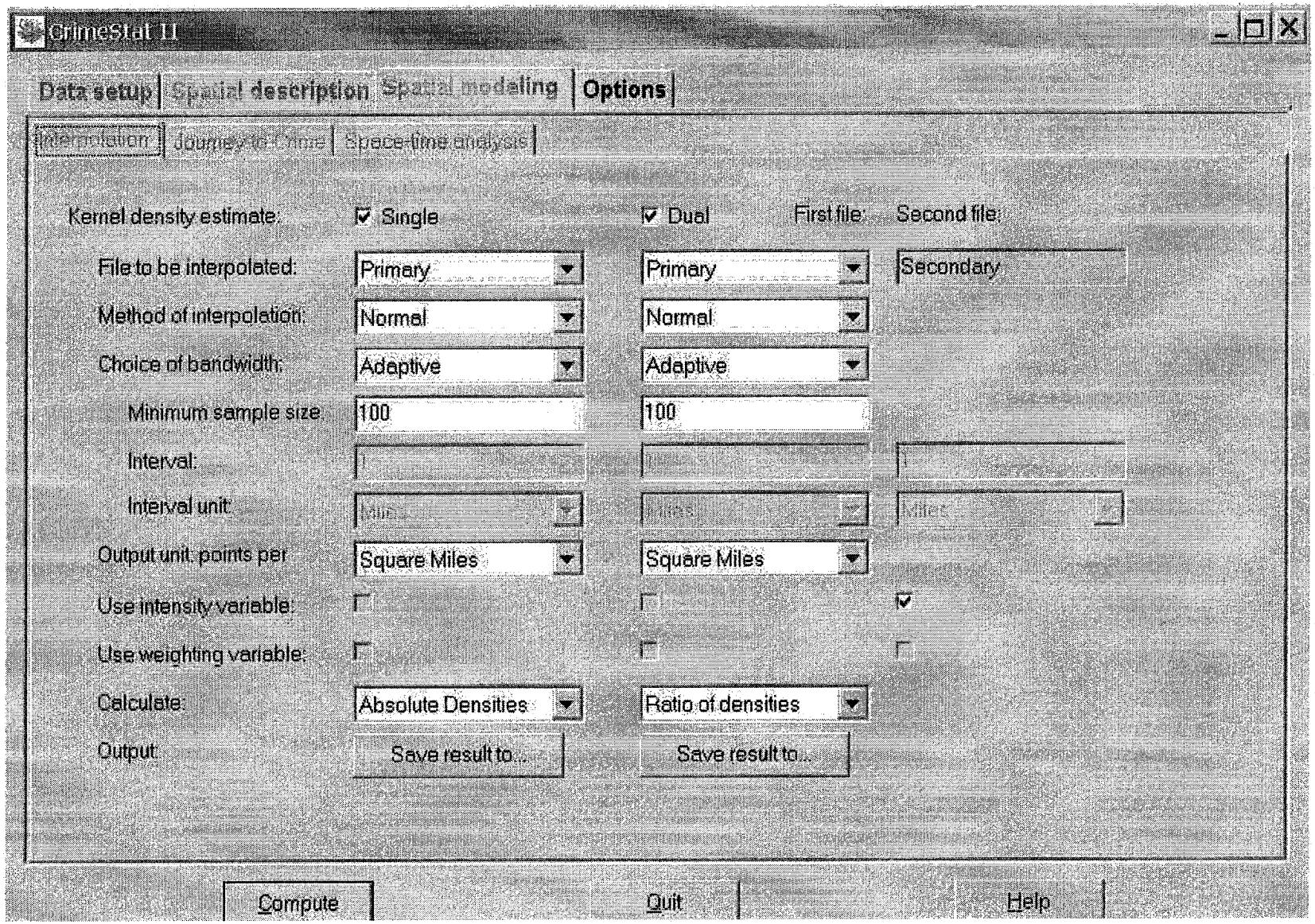
III. Spatial Modeling

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.

Figure 2.9: Interpolation Screen



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.

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*).

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.

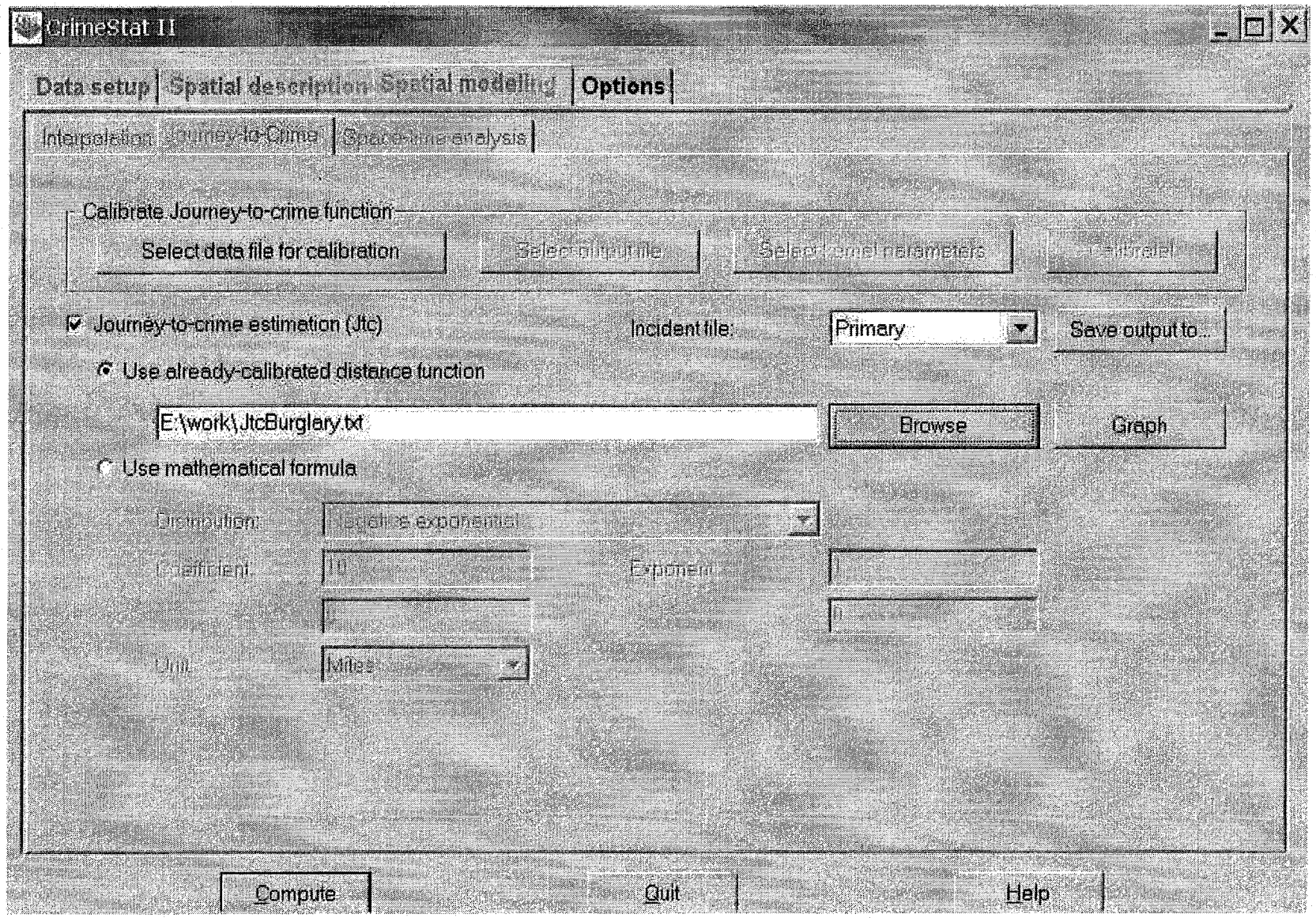
Select data file for calibration

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.

Variables

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

Figure 2.10: Journey to Crime Screen



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.

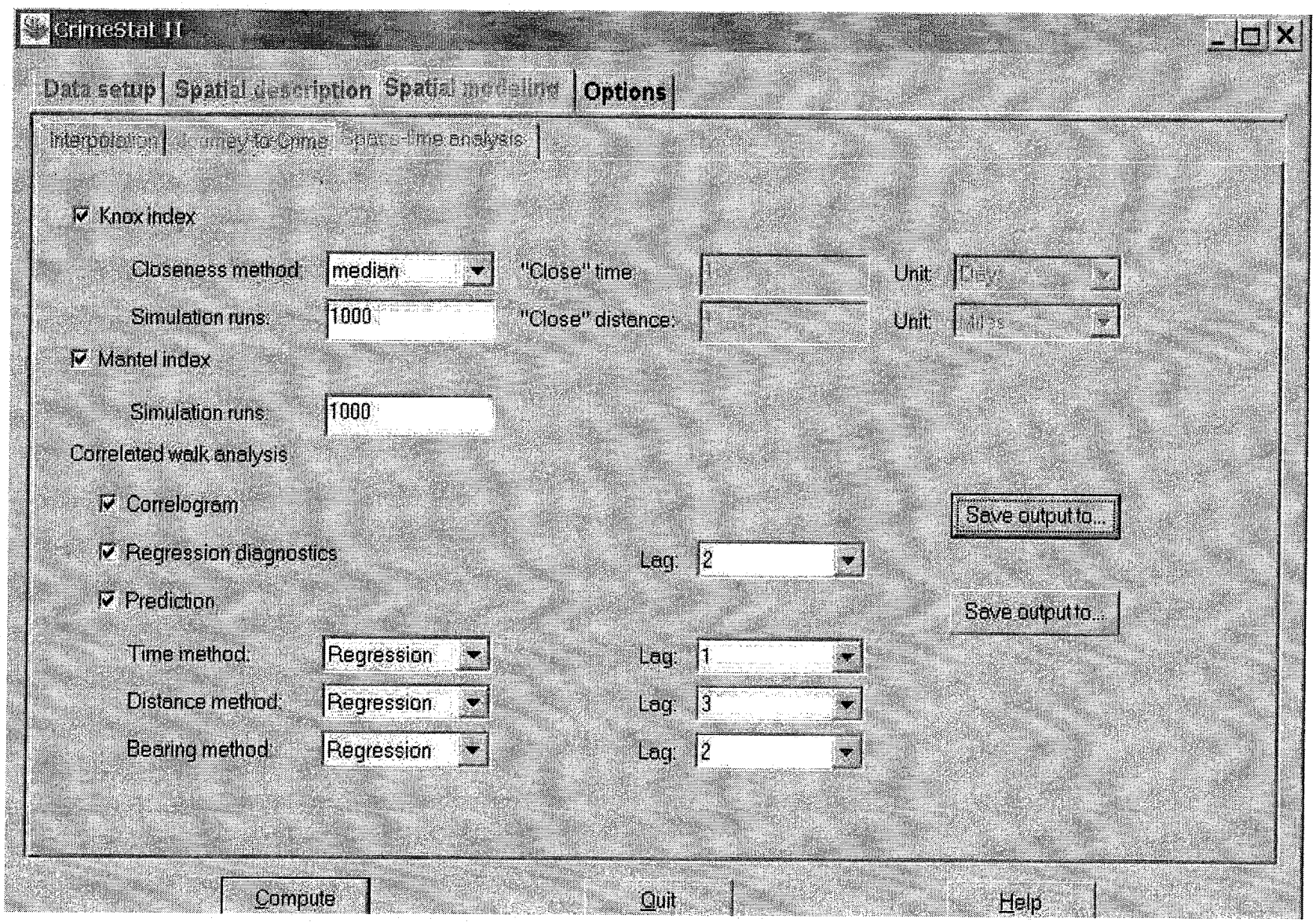
Output

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.

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.

Figure 2.11: Space-time Analysis



Third, 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.

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.

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

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.

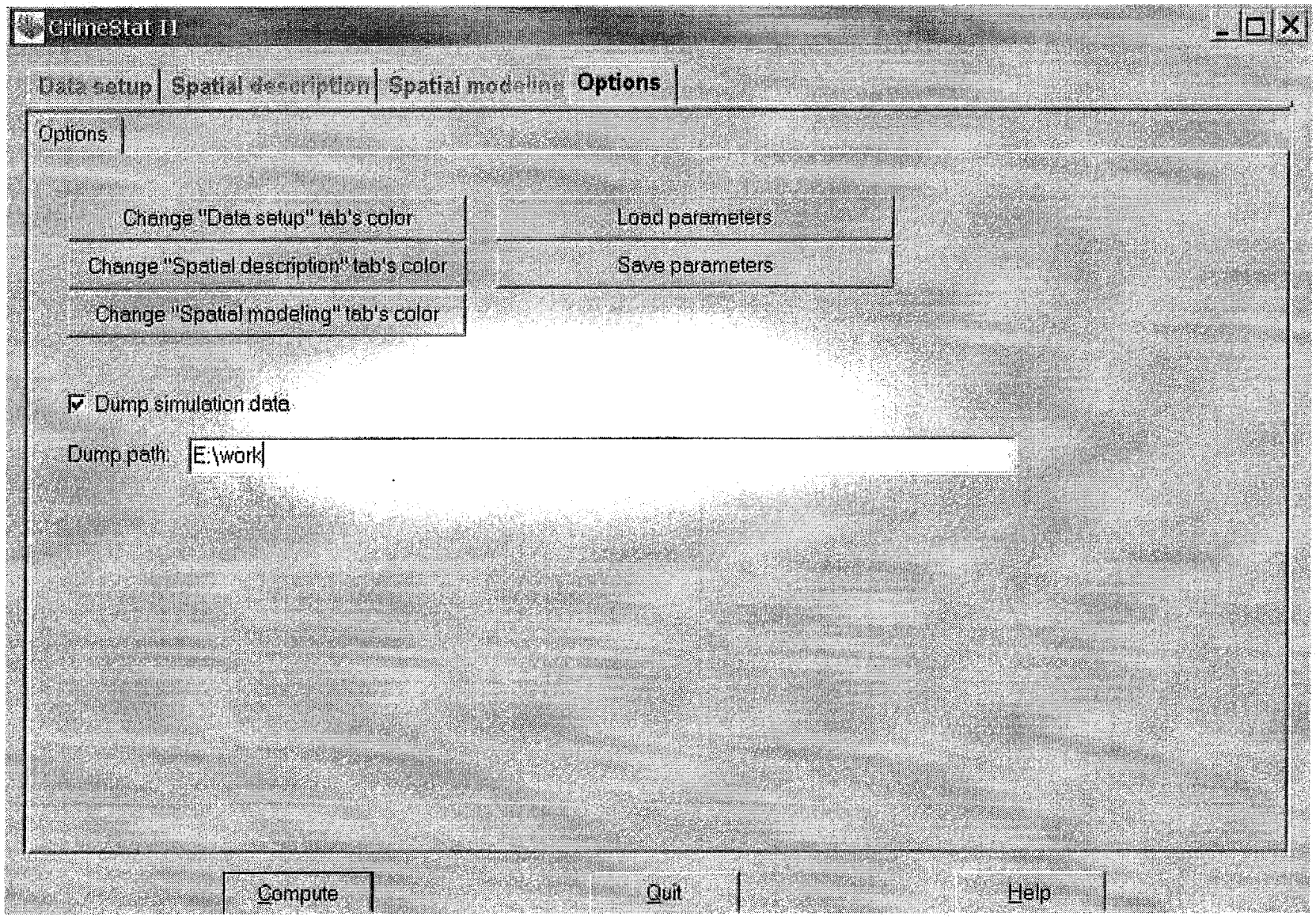
Colors

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).

Figure 2.12: **Options**



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.

Chapter 3
Entering Data into *CrimeStat*

The graphic user interface of *CrimeStat* is a tabbed form (figure 3.1). There are four groups of functions: Data setup, Spatial description, Spatial modeling, 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	Characteristics of the distances between points
'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

Options

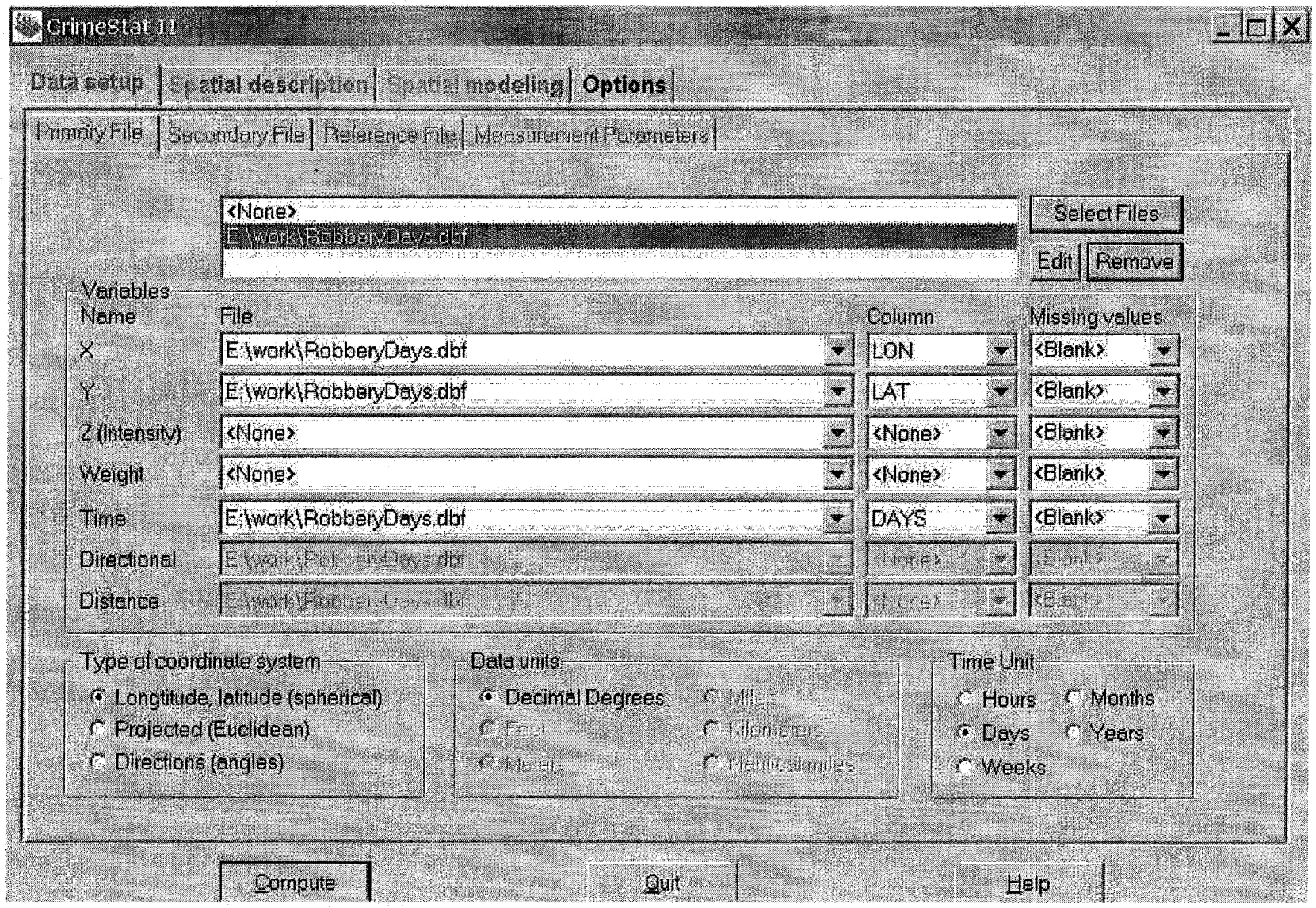
Save parameters	Save the data setup parameters
Load parameters	Load already-saved parameters file
Colors	Change the color of tabs
Simulation	Output simulation data

This section discusses the Data Setup tabs.

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*®, *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.

Figure 3.1: *CrimeStat* User Interface



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.[1] 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, meters, miles, kilometers, or nautical miles 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.[2] 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.

Intensities and weights

CrimeStat has two ways to weight an observation. 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 (i.e., in the context of X, Y and Z values. This is not to be confused with a Z-test). 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). Many 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'.

The terminology is historical but, essentially, these are two different weights. 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] For example, if population is used as both an intensity *and* a weight, it will be double-counted in any statistics which use either of these weights variables. That is, any statistic which allows weight (e.g., the mean center) will weight each observation by the weight x the intensity (or, in this case, Pop x Pop = Pop^2). This will have the effect of weighting locations with large populations much more heavily that if the population weight was used only once.

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 that 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. Only the primary file can allow a time variable. However, it has 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. Thus, many records have *missing values*.

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.

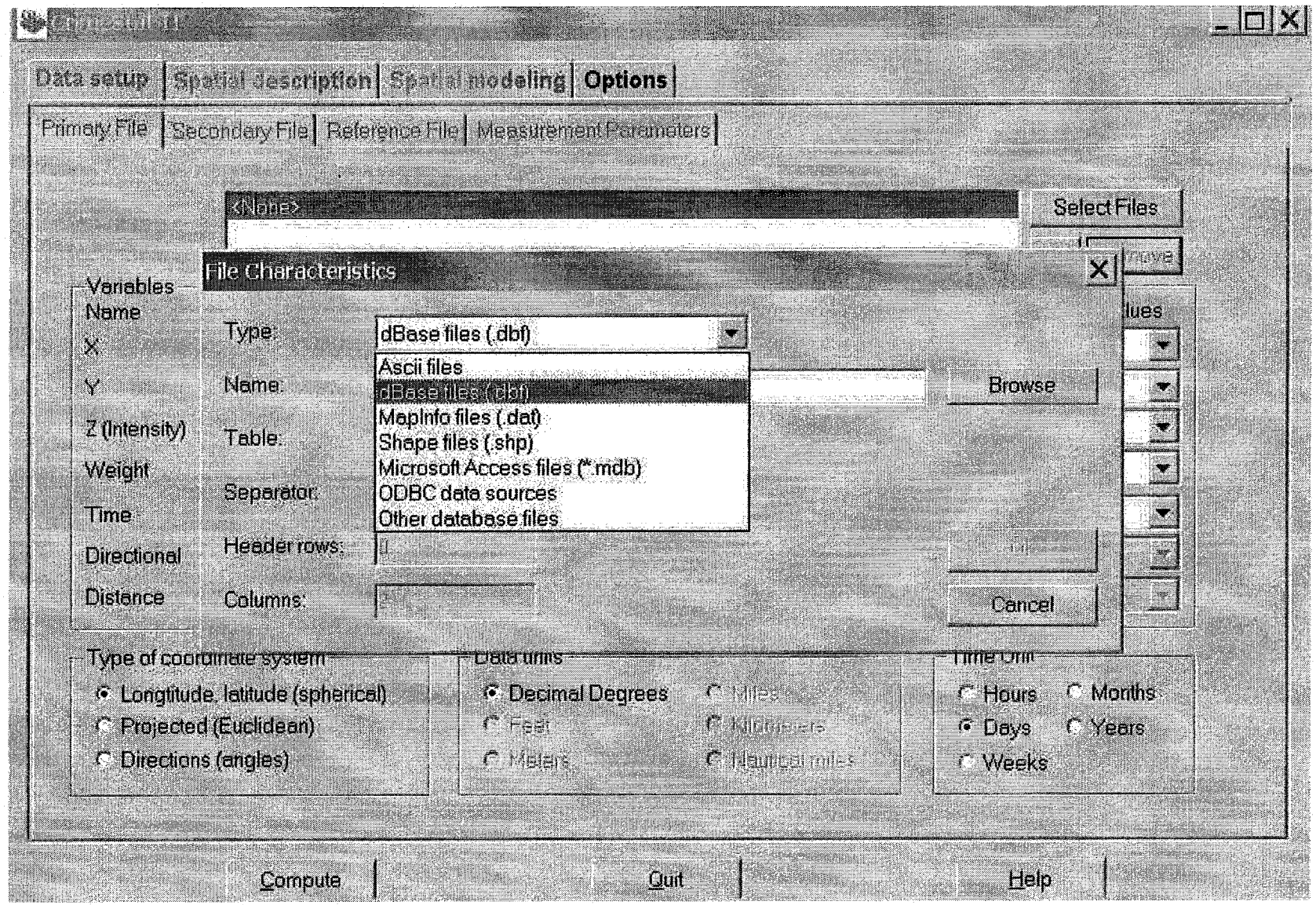
In developing this program, we have targeted it towards users of *ArcView*, *MapInfo* and *Atlas*GIS*. These GIS programs either store their attribute data in *dBase III/ IV* format in a file with a 'dbf' extension (e.g., precinct1.dbf) or can read and write directly 'dbf' files. Many other GIS programs,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

Figure 3.2: **File Format Selection**



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 that is similar to 'dbf'. This type of file can be read 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.

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 (or polar) 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*.

ODBC

Similarly, *CrimeStat* can read any file that uses Open Database Connectivity (ODBC). ODBC is a Microsoft standard interface for programs that use Structured Query

Linking *CrimeStat II* to *MapInfo*

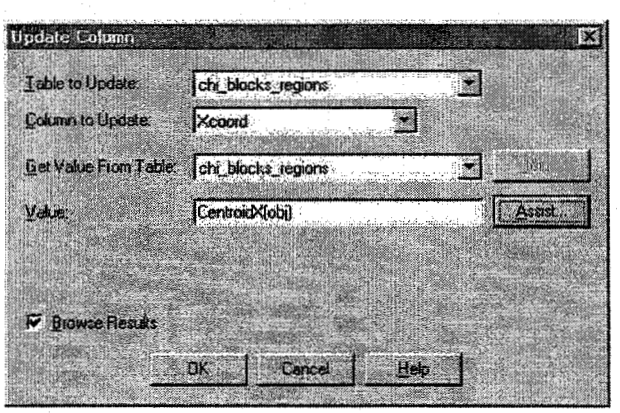
Richard Block
Professor of Sociology and Criminal Justice
Loyola University of Chicago

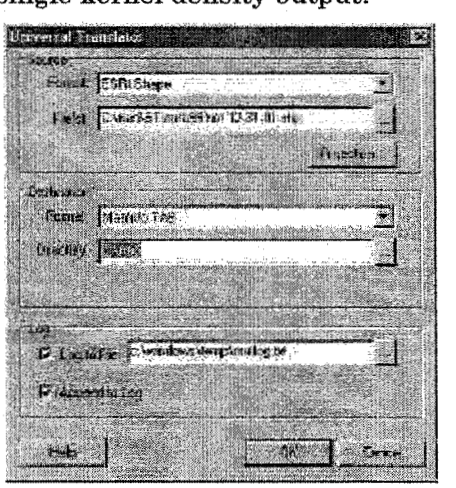
MapInfo point 'dat' files can be inputted to *CrimeStat* as primary or secondary files. However, x and y coordinates need to be added to the file. If the point data are in latitude/longitude, this is easily done with a free extension, *Table Geography*, available through the Directions Magazine website 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.





Language (SQL) as a data access standard, such as Excel®, Paradox®, Microsoft Access®, or FoxPro®. A file in one of these programs can be read by *CrimeStat* if it is first defined as an ODBC file. The sidebar below gives instructions for doing this.

Microsoft Access

Microsoft Access® 'mdb' files (from 1997 or earlier versions) can be read directly by *CrimeStat* if there is an Indexed Sequential Access Method (ISAM) driver installed. There has to be an X/Y coordinate associated with each record.

Other Database Formats

Similarly, if ISAM drivers are installed, *CrimeStat* can read files from other indexed database programs such as dBase, FoxPro, Paradox, or Lotus 1-2-3®, if ISAM drivers are installed. ISAM is a method for managing how a computer accesses stored records and files. There has to be an X/Y coordinate associated with each record.

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.

Reading Data into *CrimeStat* from an ODBC Data Source

Long Doan
Doan Associates, Falls Church, VA

CrimeStat has the ability to read files from an ODBC (Open Database Connectivity) data source. Any file for which there exists an ODBC driver - Microsoft Access®, Excel®, FoxPro® and Paradox® files, to name a few, can be read by *CrimeStat* once it has been defined as an ODBC data source. The steps are:

First, be sure you have the latest Microsoft ODBC drivers. They should have been installed with Microsoft Office, Visual Basic, or Visual C++. Otherwise, you can download them at http://www.microsoft.com/data/download.htm. Second, within *CrimeStat*, click "Select File" and choose ODBC as the file type (figure A). Third, click "Browse". You will be presented with the ODBC interface. Fourth, click on "Machine Data Sources" followed by "New" (Figure B).

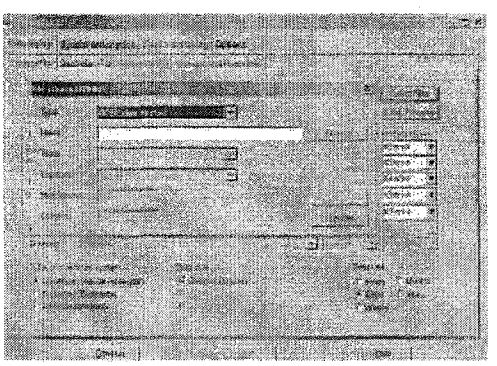


Figure A

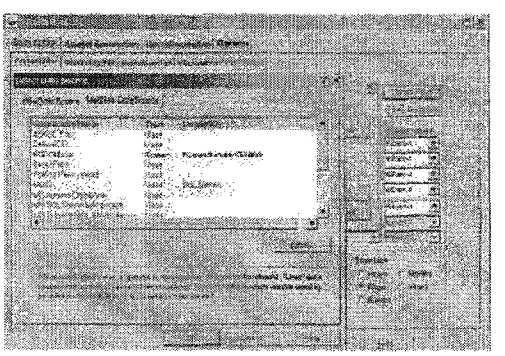


Figure B

Fifth, select "User Data Source (Applies to this machine only)" and click "Next". Sixth, select the driver (e.g., "Excel Driver"), click "Next", followed by "Finish". Seventh, enter the name of the driver, a description and click "OK" (Figure C). Eighth, select the newly created data source and click "OK". Finally, select the particular file and click "OK" (Figure D). The file will be loaded into *CrimeStat*.

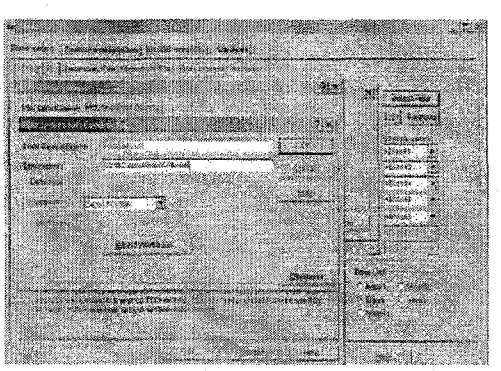


Figure C

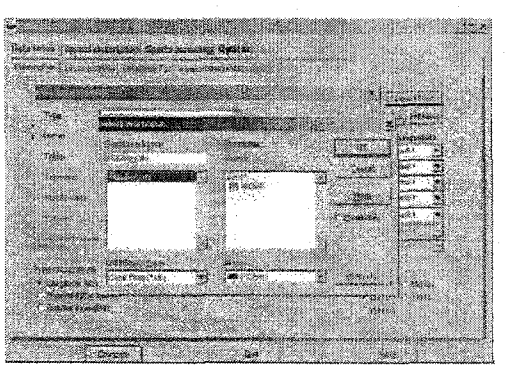
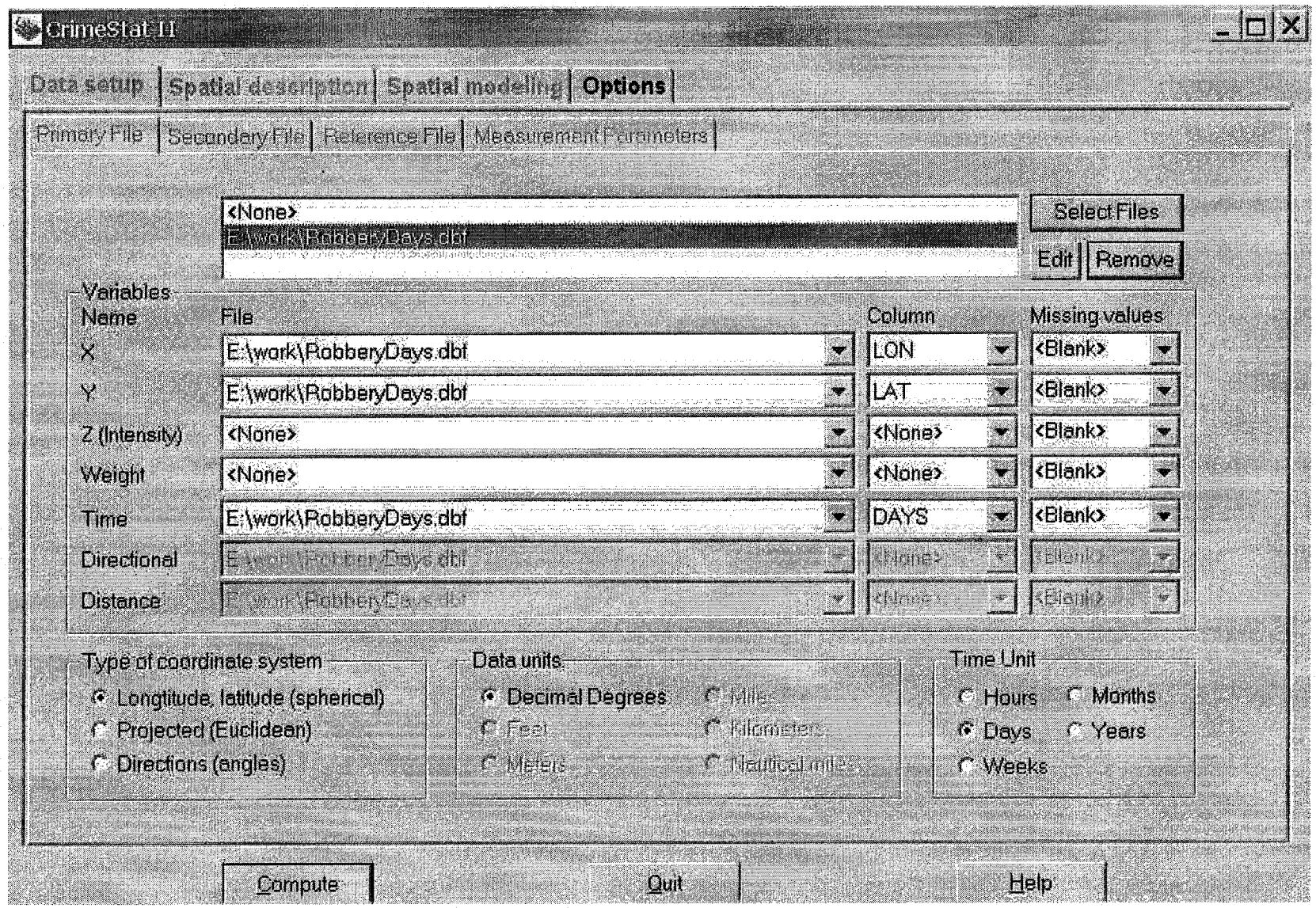


Figure D

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.

Converting Dates to Integer or Real Variables

Since *CrimeStat* requires the data be entered as integer or real variables, there are several ways to convert it to these. For example, if the date is stored as a formatted variable (e.g., May 1, 2002), most spreadsheet program can convert this to an absolute number (e.g., with the @value function in *Lotus 1-2-3* or the =value function in *Excel*). The two most common systems in place are the 1900 system, whereby the reference date is January 1, 1900, or the 1904 system, whereby the reference date is January 1, 1904. Thus, May 1, 2002 is 37377 days since January 1, 1900. An initial date can be written with a 'Date' function (e.g., =date(2002,5,1) and then converted to an absolute number with a value function (e.g., =value(date(2002,5,1))). This will work for days, but not for other time units, such as years or months.

Figure 3.4: Primary File With Intensity Variable Defined

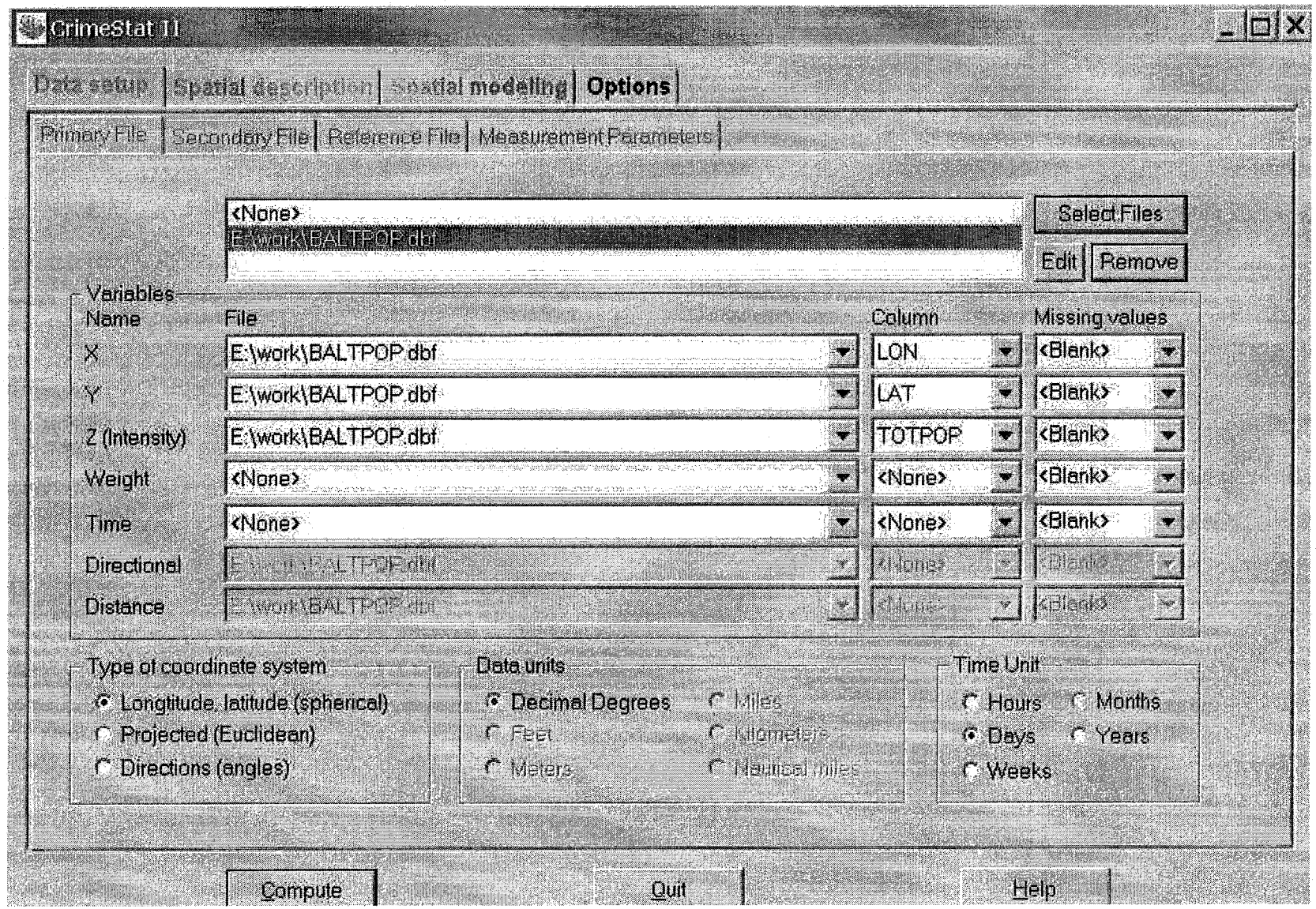
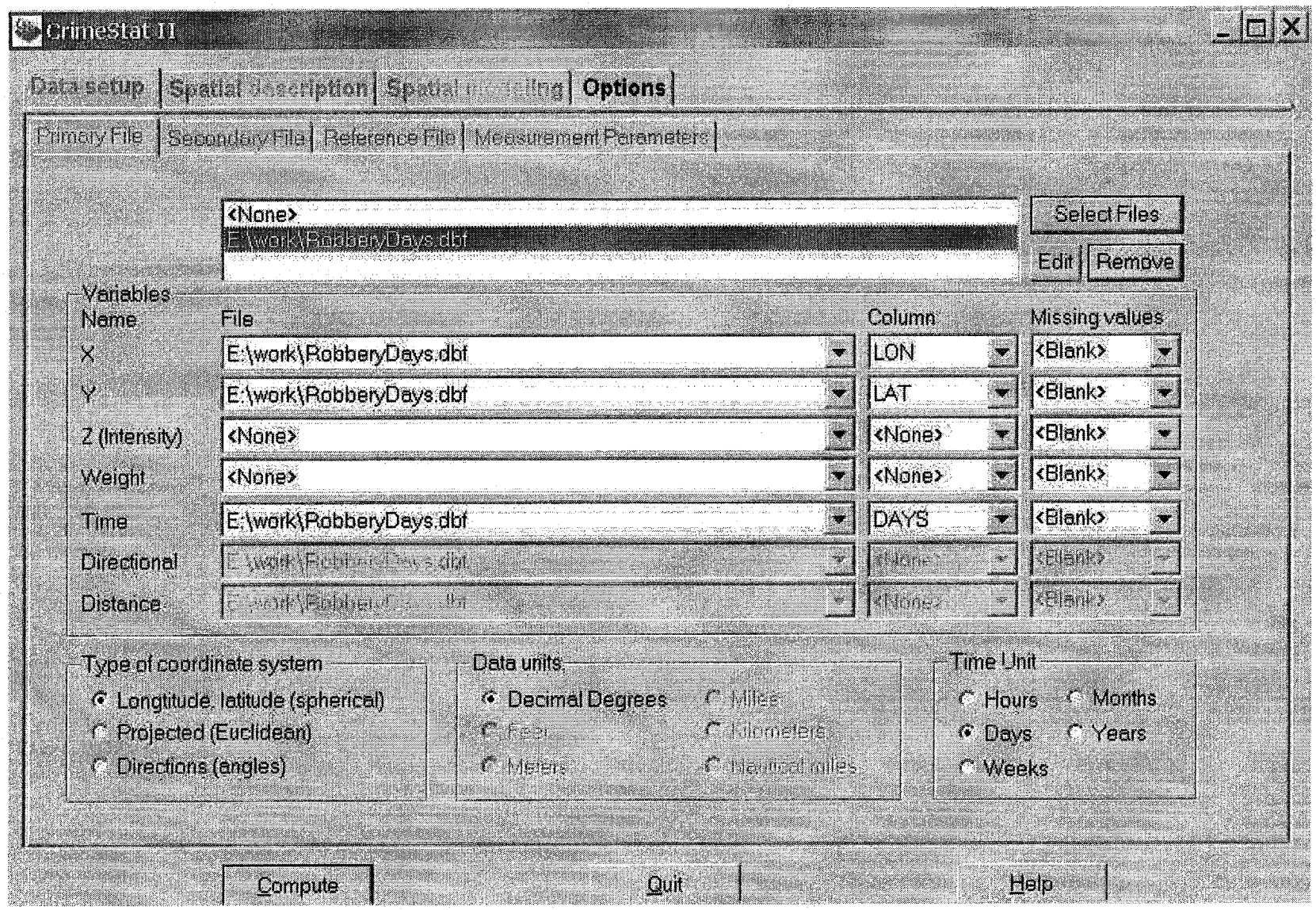


Figure 3.5: Time Variable Definition



A second example would be to convert all time units into partial days. For example, if both days and hour of the day are considered, the hour can be converted in a partial day with 24 hours. If midnight is given the value 0, then 2 AM is 0.083 (i.e., 2/24), 6 AM is 0.25 (i.e., 6/24), 3 PM is 0.625 (i.e., 15/24), and 11:30 PM is 0.979 (i.e., 23.5/24). The hours can be then added to absolute days to produce a real variable indicating time. Thus, 12:30 PM on May 1, 2002 is 37377.5208 ('37377' for May 1, 2002 and 0.5208 for 12.5/24 hours). In this case, time is defined as a real variable (i.e., with fractions).

A third method is to define the earliest date as 1 and then reference the remaining dates to the starting one. For example, if the units are months and the earliest event occurred in January 1999, then that month receives a value 1. If a record had an event occur in May 1999, then that month receives a value of 5. If another record had an event occur in December 2001, then that month receives a value of 35. The scale is a relative one since one has to know when the first event occur to interpret the remaining ones.

The point is, there are different ways to reference dates. For *CrimeStat* to use those dates, however, they must be defined consistently as integer or real numbers.

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]

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 usually defined in units of feet or meters.[9] However, *CrimeStat* allows projected coordinates to be defined with feet, meters, miles, kilometers, or nautical miles.

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, meters, miles, kilometers, or nautical miles.[10]

Directional (or polar) 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 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.

Figure 3.6: File Definition With Angles (Directions)

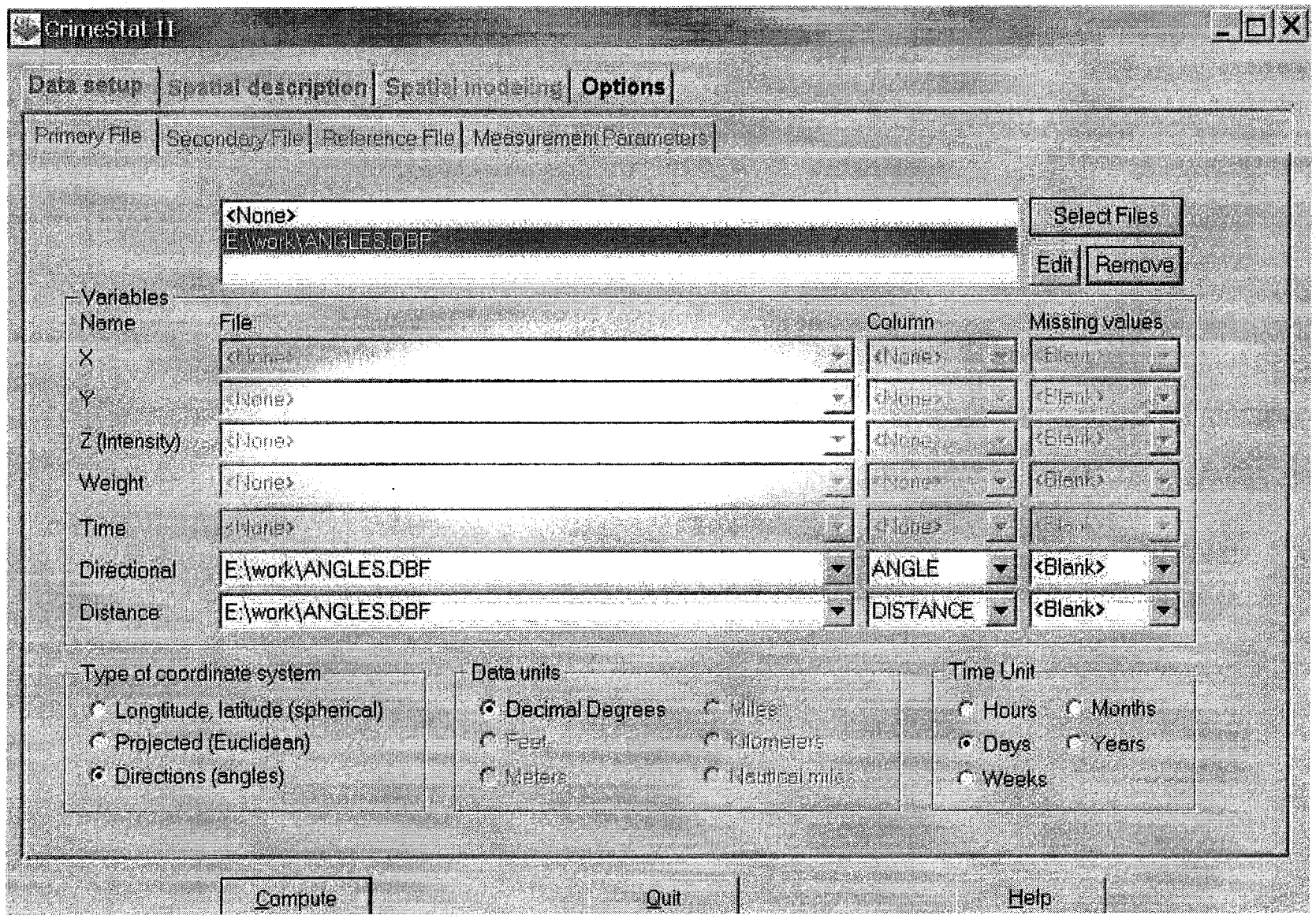


Figure 3.7: **Ascii File Selection of Secondary File**

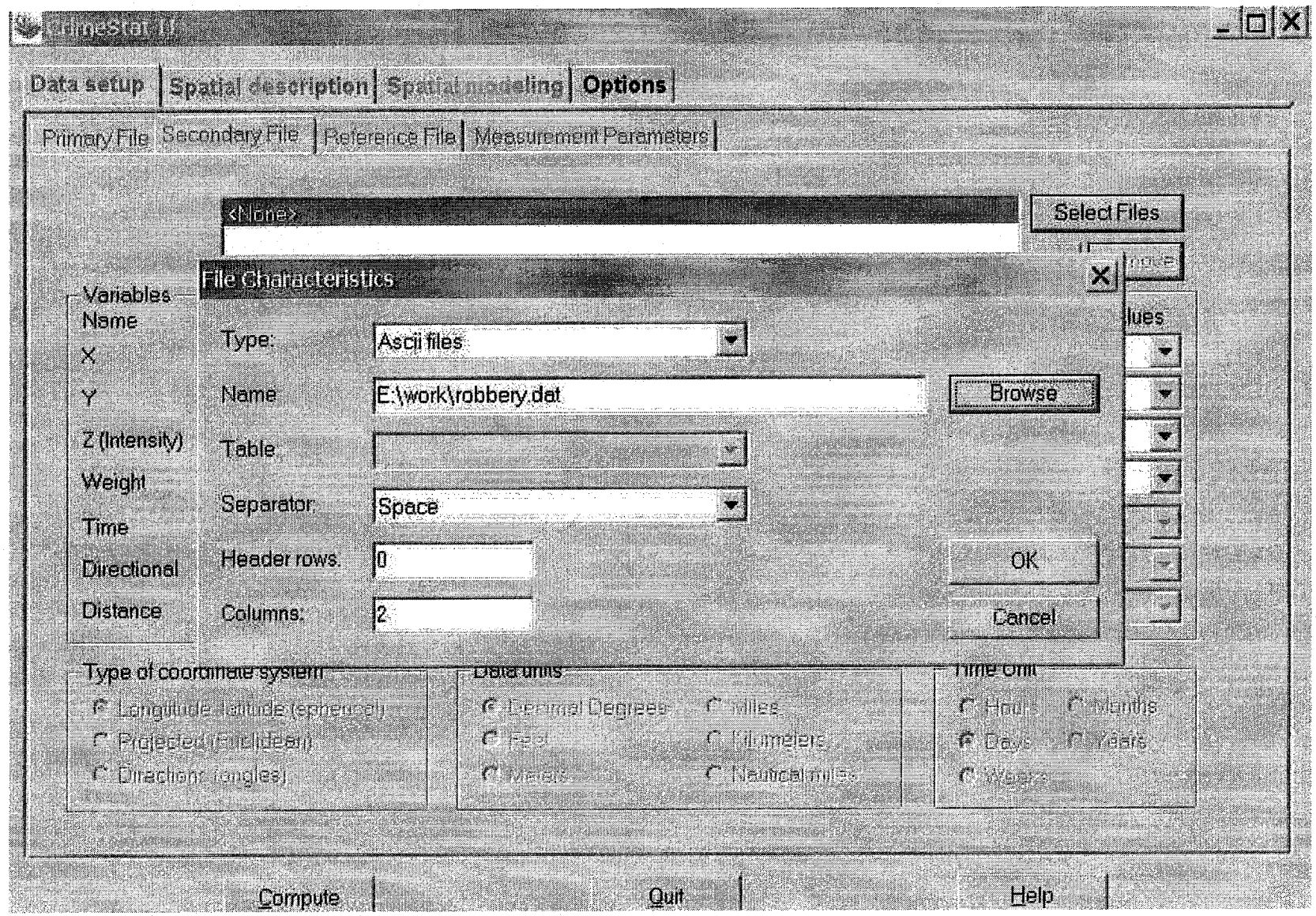


Figure 3.8: Secondary File Definition

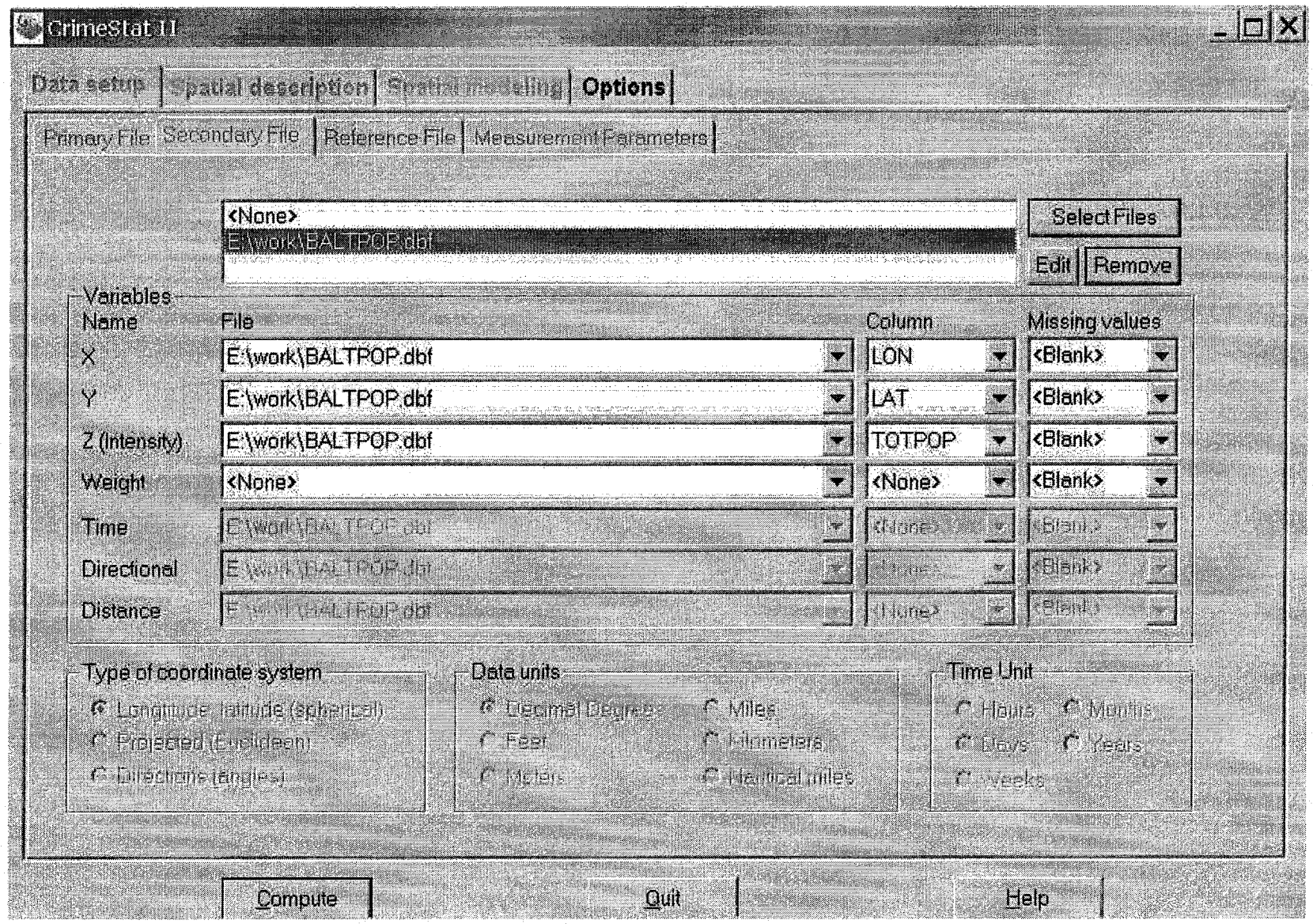
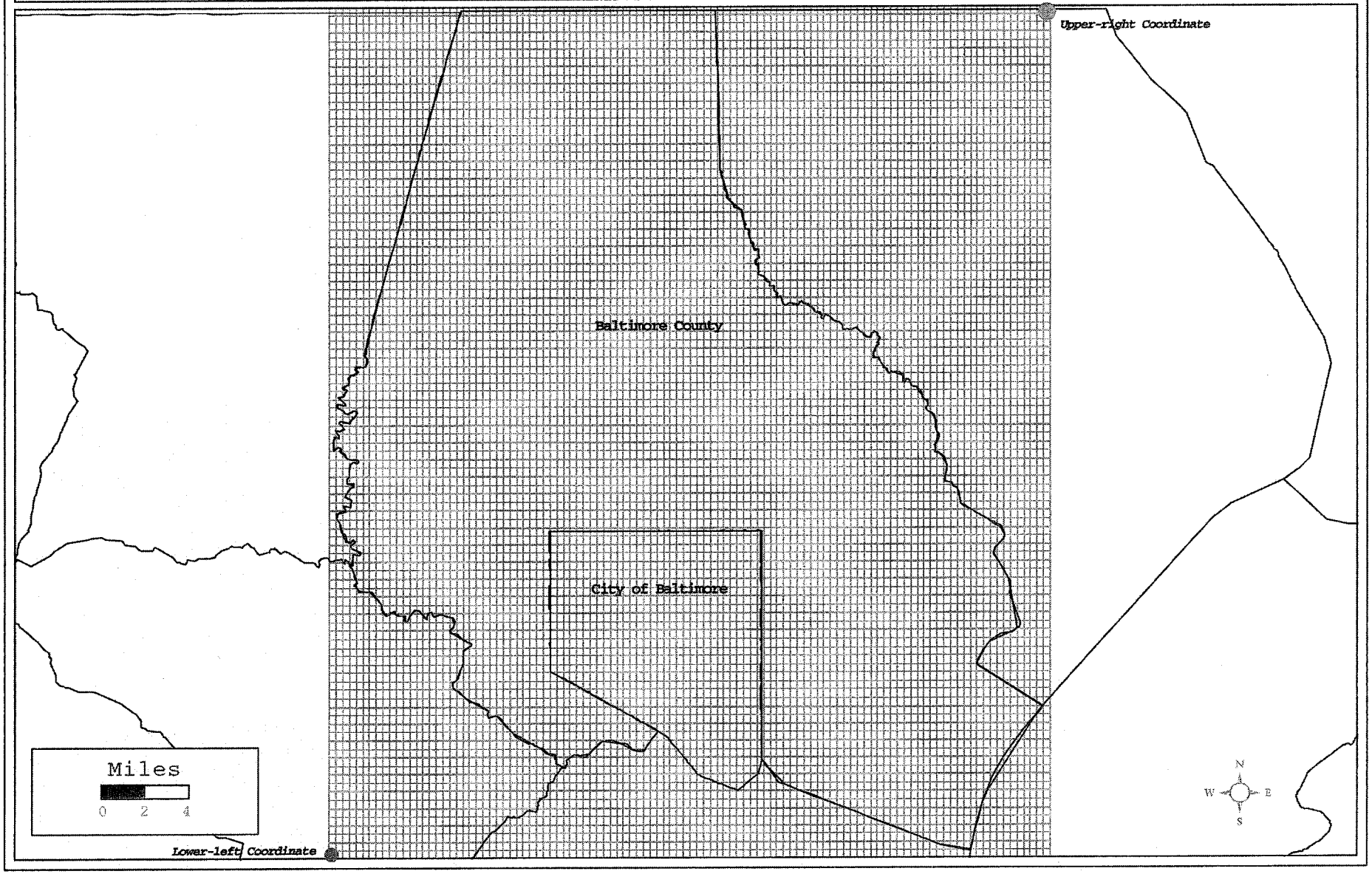


Figure 3.9: Grid Cell Structure for Baltimore Region

108 Width x 100 Height Grid Cells



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.

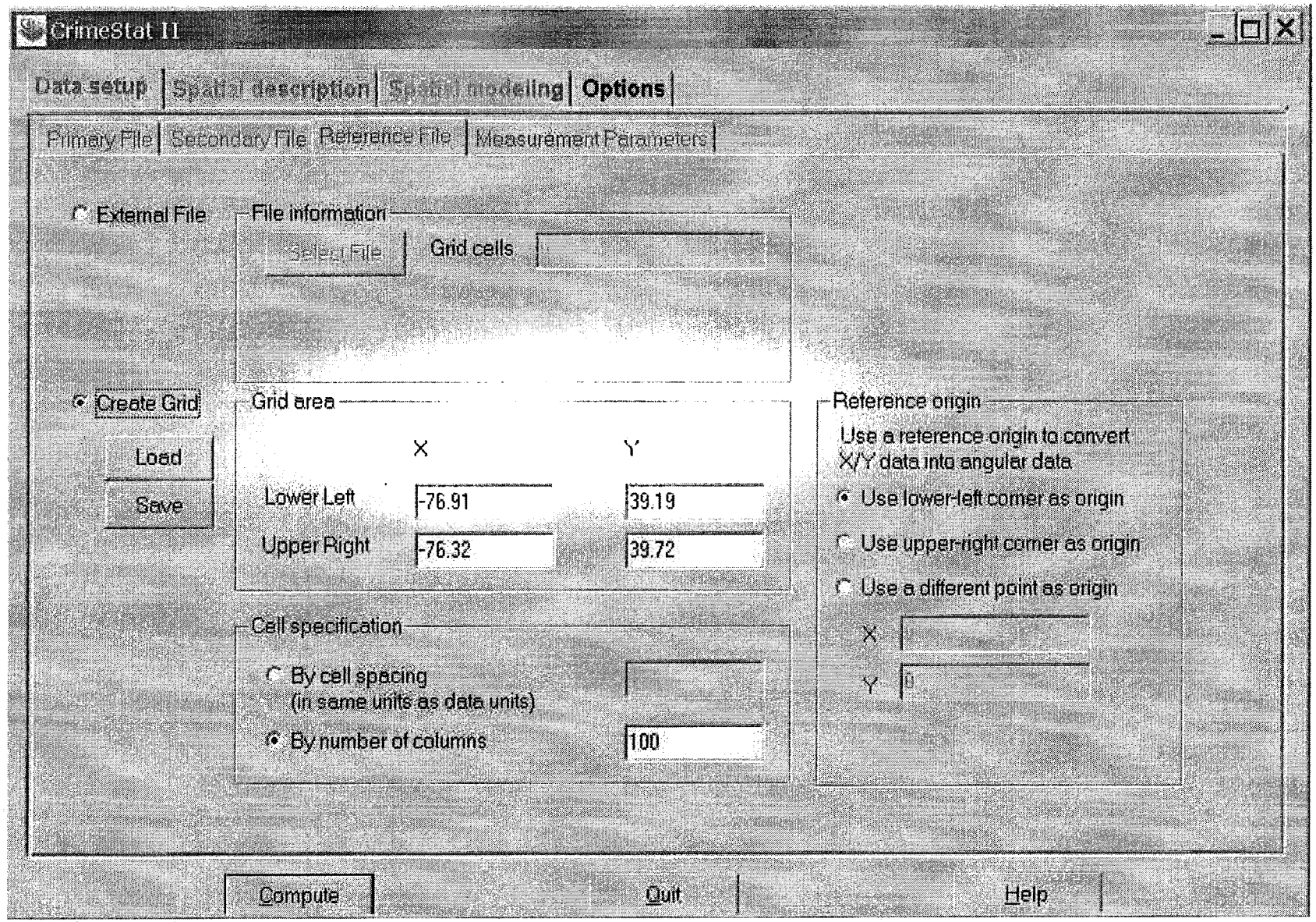
Thus, if the primary file is using spherical (lat/lon) coordinates, then the grid file coordinates must also be spherical. Conversely, if the primary file coordinates are projected, then the grid file coordinates must also be projected, using the same measurement units (feet, meters, miles, kilometers, or nautical miles). The lower-left and upper-right coordinates are those from a grid that 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 *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, meters, miles, kilometers, or nautical miles).

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.

Figure 3.10: Create Reference Grid Setup



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 Save <filename>. The coordinates and column sizes will be saved in the system registry. To load an already defined reference file, type Load 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 Load 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 that 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.

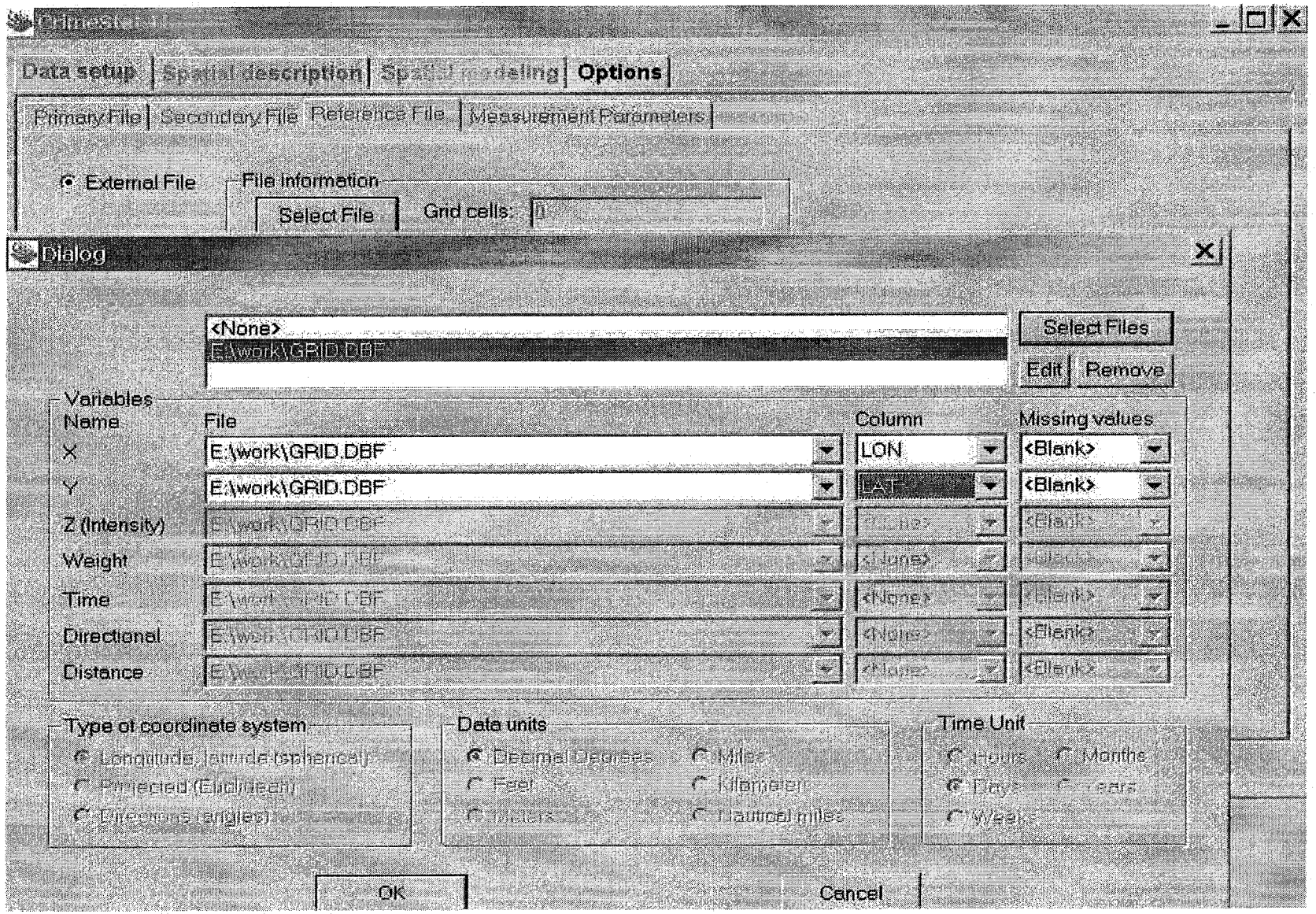
3. If it is a true grid, the number of columns should be entered. *CrimeStat* will automatically count the number of records in the file and place it in the *Cells* box. When the number of columns is entered, *CrimeStat* will automatically calculate the number of rows.

Figure 3.11 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

Figure 3.11: Reference File Definition With An External File



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, kilometers, or nautical miles). 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.

Direct and Indirect Distance

CrimeStat can calculate both direct and indirect 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 is 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

Figure 3.12: Measurement Parameters Page

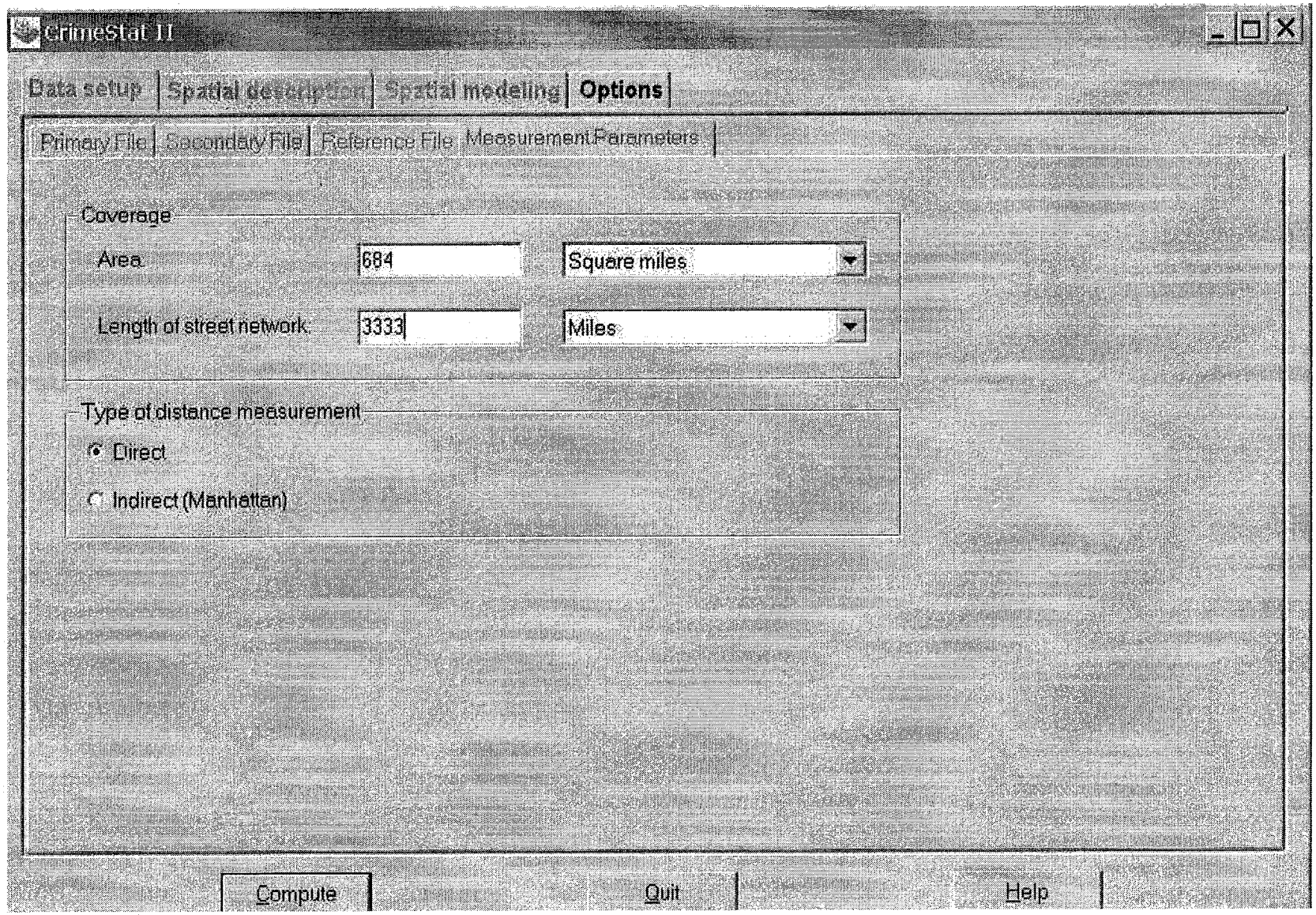
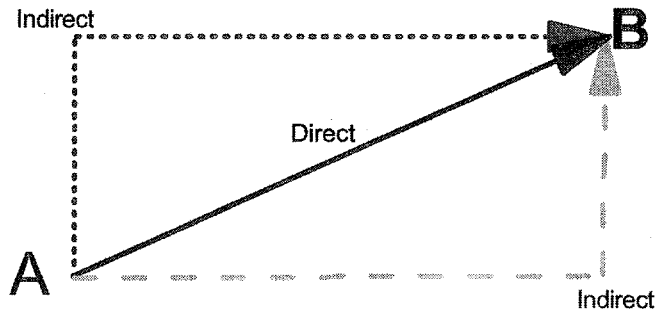


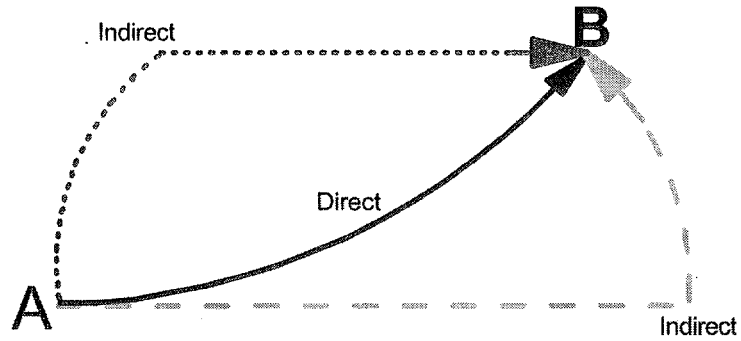
Figure 3.13: Direct and Indirect Distances

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')

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.

For some purposes, it may be useful to calculate distances that approximate an actual travel pattern rather than assume the shortest distance between points. 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.[11]

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.

$$d_{AB} = \sqrt{(X_A + X_B)^2 + (Y_A + Y_B)^2} \qquad (3.1)$$

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:

$$\text{Radians } (\phi) = \frac{2\pi\,\phi}{360} \qquad (3.2)$$

$$\text{Radians } (\lambda) = \frac{2\pi\,\lambda}{360} \qquad (3.3)$$

Then, the distance between the two points is determined from

$$d_{AB} = 2^* \text{ Arcsin } \{\text{Sin}^2[(\phi_B - \phi_A)/2] + \text{Cos } \phi_A{}^*\text{Cos}\phi_B{}^*\text{Sin}^2[(\lambda_B - \lambda_A)/2]^{1/2} \} \qquad (3.4)$$

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.

$$d_{AB} = (X_A - X_B) + (Y_A - Y_B) \quad (3.5)$$

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.

$$d_{AB} = \frac{[d_{AB}(1) + d_{AB}(2)]}{2} \quad (3.6)$$

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.

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 allows 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.

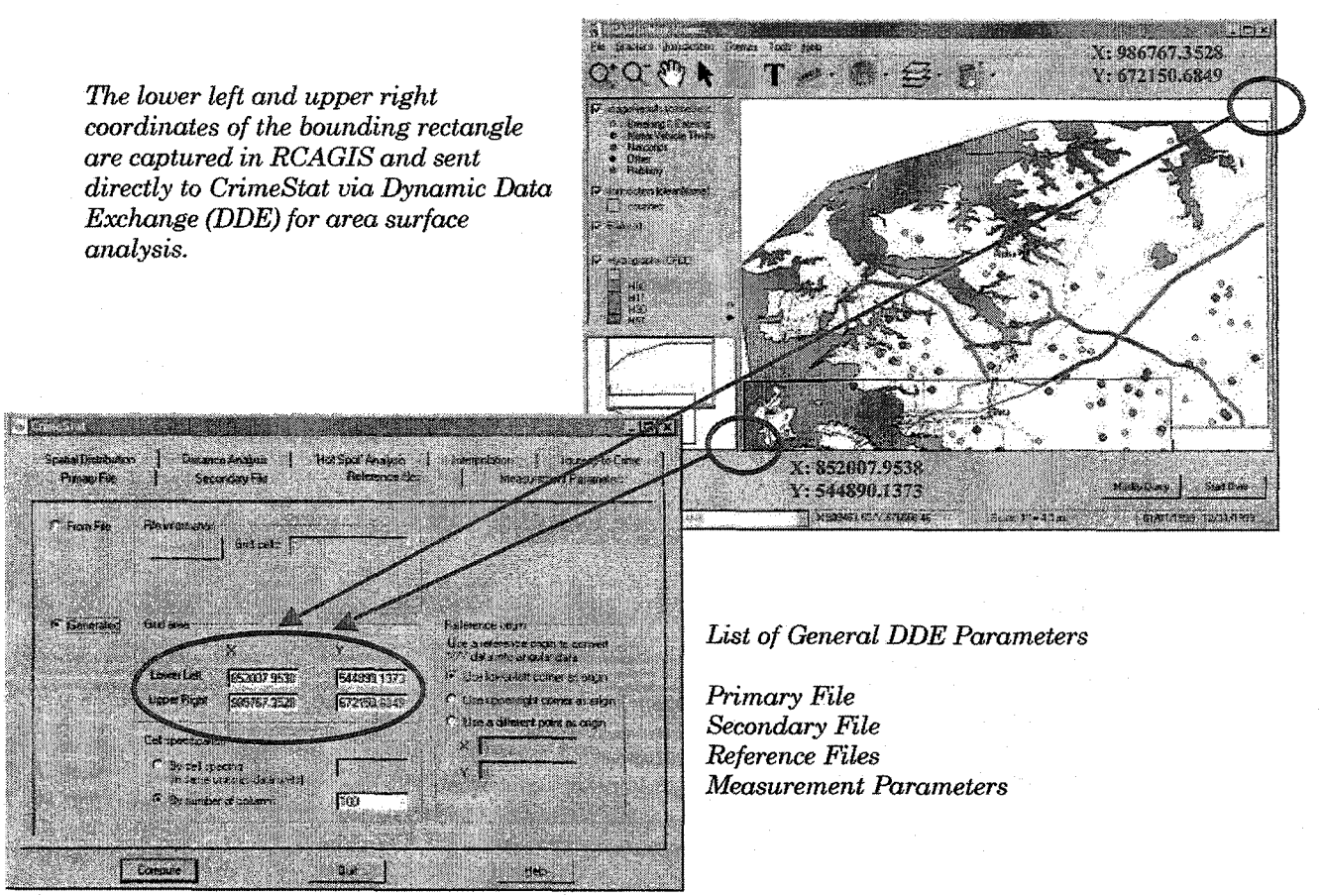
Using Dynamic Data Exchange (DDE) to Develop Software for Interfacing with *CrimeStat*

Ronald E. Wilson
Inter-university Consortium for Political & Social Research; University of Michigan
Mapping and Analysis for Public Safety Program, 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

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 Data Setup 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.

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.

A. 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).

B. 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. 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.

Endnotes for Chapter 3

1. Some *MapInfo* users in the United Kingdom have found difficulty in directly reading MIF/MID files from *CrimeStat* and converting them to the British National Grid. 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*."

2. 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.

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. The minimum should be 4 decimal places with 6 being more accurate.

   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

Decimal degrees = Degrees + Minutes/60 + 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 80th 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. If you don't know the projection system of the data you are working with, ask your administrator or the person from whom you received the data. While *CrimeStat* can work with any projected data, it will be necessary to know from what projection system the data came from if you try to use the graphical objects in a GIS package.

11. 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.*