USGS
science for a changing world

PTCOUNT—A Fortran-77 Computer Program to Calculate the Areal Distribution of Mapped Data Points Using Count-Circle Methodology

by William Z. Savage[1], Jeffrey A. Coe[1], and Ronald E. Sweeney[2]

Open-File Report 01-0002

2001

U.S. DEPARTMENT OF THE INTERIOR
U.S. GEOLOGICAL SURVEY

[1]Denver Federal Center
Box 25046, M.S. 966
Denver, Colorado 80225

[2]Denver Federal Center
Box 25046, M.S. 973
Denver, Colorado 80225

PTCOUNT—A FORTRAN-77 Computer Program to Calculate the Areal Distribution of Mapped Data Points Using Count-Circle Methodology

Introduction

Count-circle methodology is typically used for stereonet analyses of structural data (for example, see Ramsay and Huber, 1987, p. 335). Many computer programs are available for stereonet analyses (for example, the Stereo module of Rockware98, 1998), but, to our knowledge, there were no programs available as of June 2000 for application of count-circle methodology to mapped point data.

Application of count-circle methods to mapped data leads to the creation of isopleth maps. Contours (isopleths) on an isopleth map (see Schmid and MacCannel, 1955) connect equal numbers or densities of mapped features that occur within a fixed area. A population density map is an example of an isopleth map. To manually create an isopleth map, an equally spaced grid is overlain on the geologic map. Then the center of a count circle of fixed area is sequentially placed at each grid node and the number of mapped features falling within the circle are counted and recorded. Finally, the number of mapped features at each grid coordinate is contoured to produce an isopleth map. For examples of manually created isopleth maps see Campbell (1973), Wieczorek and other (1988), and Coe and Godt (in review).

The PTCOUNT program described in this report automates the production of count-circle data from digital map data. Below, we describe the program and the automated point-counting procedures, present results from an application of the program, provide guidelines for determining grid spacing and count-circle size, and present a tutorial (Appendix A) that includes instructions on running PTCOUNT.EXE, a sample input file, and a sample output file. Lastly, two FORTRAN-77 listings of PTCOUNT.FOR are given in Appendix B. The first listing is that for PTCOUNT.EXE and the second is a UNIX version.

A brief description of the PTCOUNT program

The program carries out point counting procedure by the following steps. First, increments in x and y, for the rectangular grid shown in Figure 1 are obtained by respectively dividing the maximum and minimum x and y coordinates (xmin, xmax, ymin, and ymax in the program listing) by the number of x- and y-steps (nxs and nys) to give,

$$\Delta y = \frac{(y\max - y\min)}{nys}$$

$$\Delta x = \frac{(x\max - x\min)}{nxs}$$

Next beginning at x = xmin and y = ymin, that is, at the first grid node, i = 1 and j = 1 in Figure 1, the radii to all input data points in the arrays xcrd(k) and ycrd(k) where k = 1, npoint, are calculated by the Pythagorean theorem,

$$r = \sqrt{[xcrd\ (k) - x]^2 + [ycrd\ (k) - y]^2}$$

The calculated radii, r, are compared with the count-circle radius, ccrad, and the resulting number of points falling within the count circle are automatically counted and stored in the nz(i,j) array in the program. The program then moves the count-circle center to the next grid node given by i = 2 and j = 1 and the procedure for computing the number of data points within the count-circle radius is repeated. The program continues to move the count-circle center (from the left to the right and upward as shown in Figure 1) to each grid node and repeats the counting procedure until the maximum x and y coordinates (xmax and ymax, i.e., when i = nxs + 1 and j = nys +1) are reached. Grid coordinates are defined by x = x + ix and y = y + iy and the number of geologic features found by the counting procedure at each of these centers are stored in the nz(i,j) array.

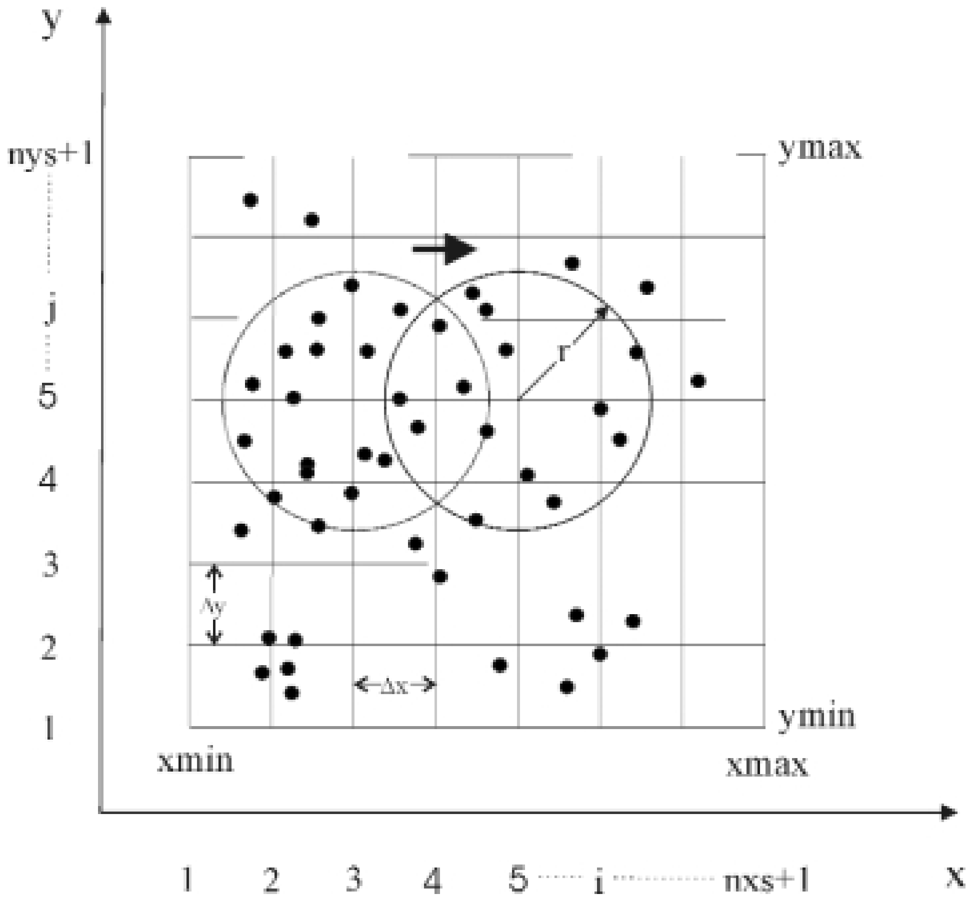

Figure 1. This figure is a sketch of the configuration for automated point counting of mapped data. Data points with x and y coordinates are indicated as black dots. The x,y coordinate system, the grid, the point counting circle, and an example of movement of the point circle center (from grid node i = 3, j = 5 to grid node i = 5, j = 5) are shown.

Output from the program gives the number of geologic features at each x,y grid coordinate in a form that allows these data to be contoured to produce an isopleth map. A contour (isopleth) map of landslide densities produced by using output data from PTCOUNT.exe in SURFER 7© (Golden Software, 1999) is shown in Figure 2. The original location data is from a database of 1358 Seattle-area landslides by Laprade, and others (2000). For this application of PTCOUNT.exe, x,y coordinates are in UTM format with xmin = 543600 m, xmax = 557200 m, ymin = 5260500 m, and ymax = 5286900 m. The grid is constructed using 68 xsteps and 132 ysteps. This choice of x and y steps, nxs = 68 and nys = 132, results in a grid where x and y are both equal to 200 meters. The radius of the point counting circle is 226 meters. This gives an area roughly 4 times the area of the largest landslides in the Seattle database. We made this choice of point-circle radius to minimize calculations. As we will see in the next section, it is larger than the radius used by Coe and others (2000) to create an isopleth map from the Seattle landslide data.

Guidelines for determining grid spacing and count-circle size

The choice of grid spacing and count-circle radius affects the overall smoothness and compactness of output data and final contours on the isopleth map. A small grid spacing produces smoother output data than does a large grid spacing. Large count circles spread the output data over a more regional area than do small count circles. The only rule that should be followed to ensure consistent counting of input data is that there should be at least 50% overlap between count circles placed at successive grid locations. To achieve 50% overlap or greater, the grid spacing should be less than or equal to the circle radius. The individual user should carefully consider the constraints of their input data and the look they hope to achieve with their final product before selecting a grid spacing and circle radius. For example, in the version of Figure 2 published by Coe and others (2000) we use a grid spacing of 25 m. This grid spacing is roughly equivalent to an average residential lot in Seattle. The count-circle size of 40,000 m^2 (radius of 113 m) is used because it is roughly equivalent to the mapped area of the largest landslides in Seattle. Examples of additional grid spacings and circle radii used for various landslide projects are given in Table1. In general, the best approach is to experiment with various grid-spacings and radii before deciding which to use.

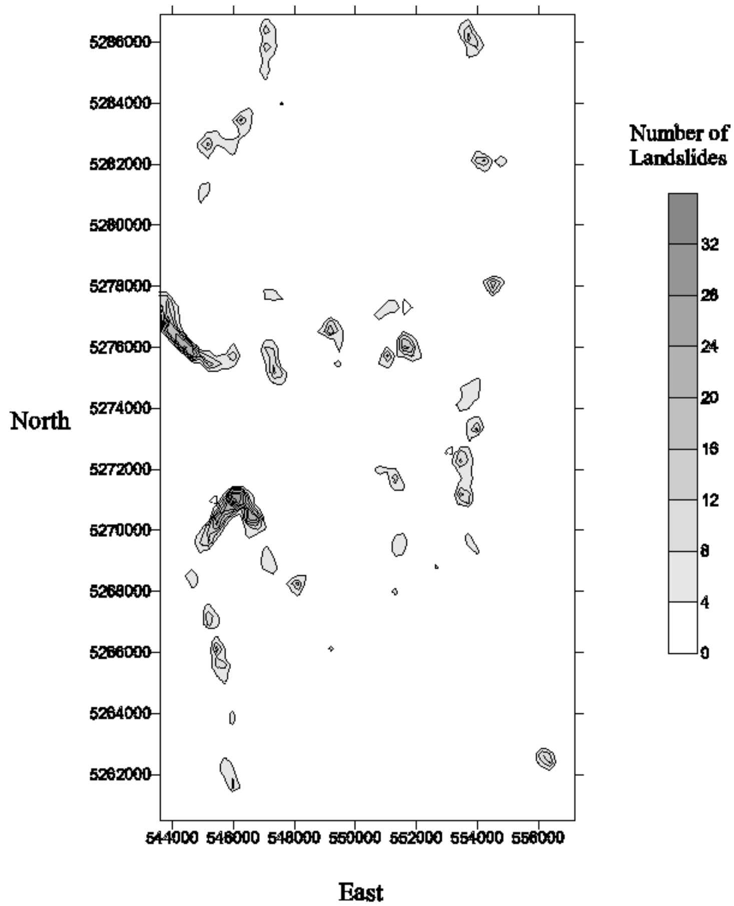

Figure 2. This figure shows a contour (isopleth) map of landslide densities in the Seattle area produced by using output data from PTCOUNT.exe in SURFER 7©. Coordinate values are in UTM format.

Table 1. Examples of previously used grid spacing and count-circle radii.

Reference	Scale of isopleth map	Grid spacing	Count-circle radius	Ratio of Grid Spacing to Circle radius
Campbell (1973)	24,000	1000 ft.	1128 ft.	89%
Wright and Nilsen (1974)	125,000	5208 ft.	5208 ft.	100%
Coe and Godt (in review)	24,000	250 m	282 m	89%
Coe and others (2000)	25,000	25 m	113 m	22%

References

Campbell, R.H., 1973, Isopleth map of landslide deposits, Point Dume Quadrangle, Los Angeles county, California, an experiment in generalizing and quantifying areal distribution of landslides: U.S. Geological Survey Miscellaneous Field Studies Map MF-535, scale 1:24,000.

Coe, J.A. and Godt, J.W., in review, Debris flows triggered by the El Niño generated rainstorm of February 2-3, 1998, Walpert Ridge and vicinity, Alameda County, California, U.S. Geological Survey Miscellaneous Field Investigation Series Map, scale 1:24,000.

Coe, J.A., Michael, J.A., Crovelli, R.A., and Savage, W.Z., 2000, Preliminary map showing landslide densities, mean recurrence intervals, and exceedance probabilities as determined from historic records, Seattle, Washington: U.S. Geological Survey Open-File Report 00-303, 25 p., scale 1:25,000.

Golden Software, Inc., 1999, Surfer 7: Contouring and 3D surface mapping software by Golden Software, Inc, Golden, CO.

Laprade, W.T., Kirkland, T.E., Nashem, W.D., and Robertson, C.A., 2000, Seattle Landslide Study, Shannon and Wilson, Inc. Internal Report W-7992-01, 164 p.

Ramsay, J.G. and Huber, M., 1987, The techniques of modern structural geology: Academic Press, London, 700 p.

Rockware, Inc., 1998, RockWorks98: A suite of computer programs by RockWare, Inc., Golden, CO.

Schmid, R.H. and MacCannel, 1955, Basic problems, techniques, and theory of isopleth mapping: Journal of the American Statistical Association, v. 50, no. 269, p. 220-239.

Wright, R.H. and Nilsen, T.H., 1974, Isopleth map of landslide deposits, southern San Francisco Bay region, California: U.S. Geological Survey Miscellaneous Field Studies Map MF-550, scale 1:125,000.

Appendix A: Tutorial

For immediate application, we have included an executable version of the program PTCOUNT.exe. The first FORTRAN-77 listing in Appendix B is the source code for PTCOUNT.exe.

The executable version of the program PTCOUNT.exe requires specification of the number of x- and y-steps (nxs and nys), the number of data points (npoint), maximum and minimum x and y coordinates (xmin, xmax, and ymin, and ymax), the radius of the count circle (ccrad), and the coordinates of all data points (the arrays xcrd(k) and ycrd(k) where k = 1, npoint). These variables must be included in an input file named point.dat. Input data is placed in the file in the following order,

nxs,nys,npoint
xmin, xmax, ymin,ymax,ccrad
xcrd,ycrd

Note that the input data are in free format, i.e., commas separate input values.

The executable version of PTCOUNT.exe allows the grid to have 2001 x and 2001 y values, that is, nxs and nys can have maximum values of 2000. Also the executable version allows a maximum of 50,000 data points, that is, npoint ≤ 50,000. Output data, placed in an output file named grid.dat, are in the order i,j,x,y,nz(i,j). The Format of the output file is 2i5,2f15.3,i10.

A sample input file listing coordinates of landslides modified from Coe, and others (2000) is included in **program**. Execution of the program creates the output file grid.dat and contouring of the results with SURFER 7© or equivalent contouring software should lead to the isopleth map shown in Figure 2.

PTCOUNT.exe was compiled with MicroSoft Developer FORTRAN© on a PC running MicroSoft Windows 2000©. It should run on most PC's with a Window's operating system and 64 MB of RAM for array allocation. To download and run the test case , first, create a directory on your PC for the executable program file PTCOUNT.exe, input data file point.dat, and output data file grid.dat. Next click on **program** and then double click on PTCOUNT.exe to download the executable program file to your new directory. You will also need to download the data file point.dat to your new directory in the same way. To execute the program double click on ptcount.exe in your directory. The program will then run and create the output file grid.dat that can then be contoured using a contouring package such as SURFER 7©

Appendix B: FORTRAN-77 Listings of PTCOUNT.FOR

Two listings are given. In the first listing the user inputs the maximum and minimum x and y coordinates of the input data. The second listing, written for a UNIX operating system, automatically calculates maximum and minimum x and y coordinates from the input data.

First listing of PTCOUNT.FOR

```
         dimension nz(2000,2000)
         dimension xc(2000,2000),yc(2000,2000)
         double precision xcrd(50000),ycrd(50000)
         double precision xmin,xmax,ymin,ymax,ccrad
         open(10,file='point.dat')
         open(11, file='grid.dat')
         read(10,*)nxs,nys,npoint
         read(10,*) xmin,xmax,ymin,ymax,ccrad
         xns=float(nxs)
         yns=float(nys)
         xinc=(xmax-xmin)/xns
         yinc=(ymax-ymin)/yns
         do 1 n=1,npoint
         read(10,*) xcrd(n),ycrd(n)
 1       continue
         y=ymin
         do 30 j=1,nys+1
         x=xmin
         do 20 i=1,nxs+1
         ncount=0
         do 5 k=1,npoint
         r=sqrt((xcrd(k)-x)**2+(ycrd(k)-y)**2)
         if(r .le. ccrad) then
         ncount=ncount+1
         nz(i,j)=ncount
         else
         goto 5
         end if
 5       continue
         write(11,1200) i,j,x,y,nz(i,j)
         xc(i,j)=x
         yc(i,j)=y
 20      x=x+xinc
 30      y=y+yinc
 1200    format(2i5,2f15.3,i10)
         close(10)
         close(11)
         end
```

Second listing of PTCOUNT.FOR

```
      dimension nz(2000,2000)
      double precision xcrd(50000),ycrd(50000)
      double precision xmin,xmax,ymin,ymax,ccrad
      double precision x,y,xpt,ypt,xns,yns,xinc,yinc
        character*80 infil,iofil
        write(6,1)
  1     format(' Enter input x,y point filename:  ',$)
        read(5,2) infil
  2     format(a)
        open(unit=10,file=infil,status='old',form='formatted')
        write(6,3)
  3     format(' Enter output ASCII grid filename:  ',$)
        read(5,2) iofil
        open(unit=11,file=iofil,status='new',form='formatted')
        write(6,4)
  4     format(' Enter number of columns & rows for output grid:  ',$)
        read(5,*) nxs,nys
        write(6,5)
  5     format(' Enter length of radius for point count circle:  ',$)
        read(5,*) ccrad
c
c   Calculate xmin, xmax, ymin, ymax for input point data.
c
        npoint=0
  8     read(10,*,end=10) xpt,ypt
        npoint=npoint+1
          xcrd(npoint)=xpt
          ycrd(npoint)=ypt
          if(npoint.eq.1) then
            xmin=xpt
            xmax=xpt
            ymin=ypt
            ymax=ypt
          else
            if(xpt.lt.xmin) then
              xmin=xpt
            else if(xpt.gt.xmax) then
              xmax=xpt
            end if
            if(ypt.lt.ymin) then
              ymin=ypt
            else if(ypt.gt.ymax) then
              ymax=ypt
            end if
          end if
        go to 8
 10     close(10)
```

```
c
c   Round off xmin, xmax, ymin, ymax.
c
        ixmin=idint((xmin-50.d0)*.01d0)*100
        ixmax=idint((xmax+150.d0)*.01d0)*100
        iymin=idint((ymin-50.d0)*.01d0)*100
        iymax=idint((ymax+150.d0)*.01d0)*100
        write(6,11) npoint,ixmin,ixmax,iymin,iymax
  11    format(' Number of points in input point file = ',i10,/,
   1          '    xmin = ',i12,/,
   2          '    xmax = ',i12,/,
   3          '    ymin = ',i12,/,
   4          '    ymax = ',i12)
       xns=dfloat(nxs)
       yns=dfloat(nys)
       xinc=dfloat((ixmax-ixmin))/xns
       yinc=dfloat((iymax-iymin))/yns
       y=iymin
       do 30 j=1,nys+1
       x=ixmin
       do 20 i=1,nxs+1
       ncount=0
       do 15 k=1,npoint
       r=dsqrt((xcrd(k)-x)**2+(ycrd(k)-y)**2)
       if(r .le. ccrad) then
       ncount=ncount+1
       nz(i,j)=ncount
       else
       goto 15
       end if
  15   continue
       write(11,1200) x,y,nz(i,j)
  20   x=x+xinc
  30   y=y+yinc
 1200  format(2f15.3,i10)
       close(11)
       end
```