EXTRACTION OF POLYGONAL INFORMATION FROM GRIDDED DATA

Scott Morehouse
and
Geoffrey Dutton
Laboratory for Computer Graphics
and Spatial Analysis
Harvard University
48 Quincy Street
Cambridge, Massachusetts 02138

Introduction

Rectangularly sampled, or gridded, spatial data such as classified LANDSAT imagery and Digital Terrain Models (DTM's), are increasingly ubiquitous sources of cartographic information. For environmental analysis, gridded data must be related to point, line, and polygon descriptions of phenomena, such as highway networks and soil zones. In combining such disparate forms of data, analysts must convert the irregular data structures to regular ones (or vice versa). That is, either points, lines and polygons describing objects may be gridded , or objects may be extracted from gridded representations of phenomena. This paper discusses integrating such data sources. Principally, issues involved in extracting polygonal features from grids will be addressed.

Characteristics of Gridded and Polygonal Data

Gridded data results from a systematic sample of a study area. Within a rectangular array of sample points (cells), the values of a geographic variable (such as terrain elevation or soil characteristics) are sensed and/or calculated for each cell.

Polygon data, in contrast, results from an exhaustive partitioning of a study area into discrete zones according to natural or administrative criteria (Peucker and Chrisman, 1975). The zones can either be defined _a priori_, or deduced from the distribution of phenomena under study. Frequently both types of polygons are of interest (e.g., county and soil zone boundaries). In either case, a patchwork of irregular zones will completely cover the study area. Each zone is regarded as internally homogeneous, with discontinuous transitions between zones. Within a computer, polygons are normally represented as strings of x,y boundary coordinates, using as many coordinates as one is willing or able to capture to characterize the behavior of the boundaries.

Despite the fundamental differences between them, gridded and polygonal data representations are interchangable, though not necessarily equivalent. The conversion of information from a grid cell system to polygonal representation involves several operations (Rosenfeld 1978).

Classification is required because in a data grid cell values can be measured on a continuous scale, and can represent samplings of a continuous data distribution. Terrain elevation data immediately comes to mind, but other continuous surfaces (such as proximity fields or rainfall measurements) are widely studied. To segment a continuous surface into subsets, each cell can be classified into one of a discrete number of data classes (as when employing a contour interval).

Once a classification has been defined, borders between unlike classes must be located. A number of assumptions can be made about the spatial behavior of the phenomena sampled. The first is that the value of the geographic variable varies continuously from place to place. This is a natural assumption for a variable such as terrain elevation. An alternative assumption is that there are discontinuities between places. This is always the case where data are collected in nominal categories, as are soil types.

This choice of assumptions will influence how polygon borders are located. If the variable is nominal or discrete, it is appropriate to locate boundary lines along cell boundaries; with continuous data, however, boundaries can be interpolated which cut through grid cells.

The third and final task is to assemble boundary segments into a polygonal representation. This involves identifying all polygons formed, smoothing or filtering borders, and ensuring that each polygon has proper closure in the face of boundary and resolution constraints.

Cell Classification versus Contour Interpolation

Two basic strategies are available for converting grids to polygons. In cell classification, each cell is assigned to a class based on its value. This classified matrix is then scanned and a border generated whenever a cell's value differs from a neighbor's. In this method, all elevation class boundaries follow cell borders.

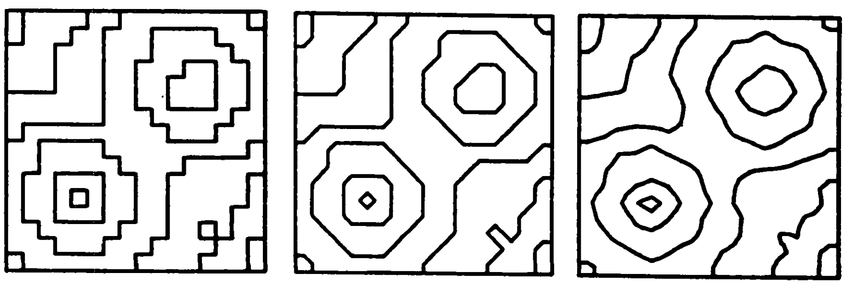

Fig. 1: (a) Cells Classified; (b) Edges Smoothed; (c) Interpolated Contours

This technique is satisfactory when the spatial resolution of the grid is fine enough. 'Staircase' lines resulting from following cell borders can then be smoothed.

When the spatial resolution of the grid is poor, additional height values can be interpolated within each cell. These heights can be used to locate borders with finer resolution than the original cell possessed. This interpolation may involve resampling the grid with finer resolution, using a distance weighted average to interpolate values for the new cells. This new matrix can then be classified and processed.

An alternative to grid resampling is to interpolate the position of borders within the original cells. As grids tend to be large (and redundant) , resampling adds further bulk (hence processing effort) to the data. By allowing contours to pass through cells, smooth borders can be produced without resampling.

A Conversion Algorithm

Many algorithms for generating contours exist (see, for example, Calcomp 1968; Dayhoff, 1968; Murray, 1968; Coulthard, 1969). Most existing algorithms follow contours throughout a data grid (or randomly accessable subsets of a data grid). While this directly produces the desired result -- contour lines in the form of strings of coordinates -- random access memory requirenments may be burdensome. As contour lines may start and end anywhere in the grid, large sections of the grid must be available for processing in random access memory for the contour following approach to be efficient. Since all cells of the grid must be scanned for possible contours anyway, it seems sensible to minimize random access memory requirements by separating the segment generation process from the segment linking process. With such a strategy, each process can be performed using sequential scans of the data.

The conversion algorithm has two principle steps -- contouring and structuring. In the first step, the data grid is scanned and processed to generate a contour segment file. For each segment, the elevation class on the left and right sides of the segment are recorded as well as the endpoints of the segment. In the second step, structuring, the segment file is processed to generate a polygon boundary file. This involves linking together segments to form borders and identifying the polygons thus created.

Contouring Phase

Contour segments are generated on a row by row basis using the grid values in the current row and the the two ajacent rows. within each row, each cell is processed independently to locate and output contour segments. All such segments which fall in the cell will be identified before passing on to the cell in the next column. Segments are derived by decomposing the rectangular cell into eight triangles:

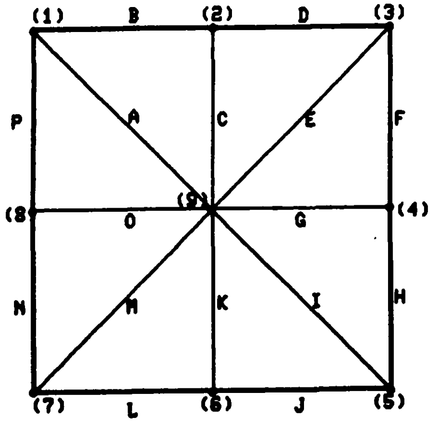

Fig. 2: Triangulation of a Grid Cell with Nodes numbered (1) - (9) and Edges Labelled A - P

The network of triangles imposed on the cell has nine nodes, interconnected by sixteen edges. Each of the nodes is assigned a z-value which is the average of the classified values of all cells touching that node. Thus, in the above figure, node 9 is assigned the value of the cell , nodes 2,4,6 and 8 are assigned averages of the cell with values in the top, right, bottom and left cell neighbors, respectively, and nodes 1,3,5 and 7 are assigned z-values averaged using the cell and the 3 other cells touching each node. Cells with missing values are excluded from this averaging process.

First, the minimum and maximum z-values are checked to determine if they fall into different classes. If both are in the same class (which will normally be true), no contours exist in the cell, and the next cell in the row is processed. Otherwise, a scan of the 16 edges is performed, referencing a table of the z-values of the nodes bounding edges A through P. Should both nodal values for an edge lie in the same contour interval, the edge has no contours crossing it, and the next edge can be inspected. Otherwise, the intersection of the contour with the edge is interpolated. The resultant coordinate marks the location where the contour crosses into a triangle. Each such intersection is entered in a table with the identifier of the edge and the class break involved.

Should this table already contain an entry for the same class break, a contour segment may have been found. Whether the current intersection is linked to the prior intersection, depends on whether both intersections are located in the same triangle. If the edge index for the prior intersection is either (a) one less than the current one or (b) two less than the current one (for interior edges), then a contour segment exists and is output. In any case, the table is updated, replacing the prior intersection's coordinates with those of the current one. Although no search of the table is necessary, the number of contours allowed to cross an edge is limited by the table's length. A 100-word table is sufficient to index up to 33 countours intersecting a cell.

Edges are examined in clockwise sequence around the cell, ending by processing the first edge once again. The procedure ensures that all segments of contours which are within the cell will be identified, located, and output. Correct "left" and "right" class identifiers are also output with each segment's coordinates. The six-word-long segment records (x1,y1,x2,y2,left,right) are accumulated in an output file as the grid is scanned. This file is then processed in the concatenation phase, in which polygon boundaries are constructed from the segments.

Structuring Phase

The concatenation of contour line segments was performed by the WHIRLPOOL program of Harvard's ODYSSEY family of geoprocessing modules (Dutton, 1978; Dougenik, 1979). This program, although designed to perform overlay of polygon coverages, is also capable of finding intersections of a single coverage with itself. That is, given a minimal amount of topological information (in this case, the left and right elevation classes of each contour segment), WHIRLPOOL can link segments, as well as discover errors in topological and geometric descriptions. In addition to serving the present purpose, this capability allows manually-digitized coverages to be verified as planar graphs, making automatic adjustments for small discrepancies in coordinates (part of the so-called "fuzzy overlay" problem). Not all of WHIRLPOOL's power is required to link segments, although one portion of the problem would have been been troublesome without it. This has to do with contours which, instead of forming closed loops, simply disappear at the edge of the grid. In order to form polygons, such contours must be connected to one another at the points where they terminate. This was approached by generating a "rim" contour, by writing out the coordinates of the cell edges at the border of the study area. Because of the nature of the contour generation algorithm, contours can cross cell edges at any point. All such intersections must be found with the rim contour in order to ensure that all contours are part of some polygon, and that all polygons are completely and consistently described. While a much simpler program could accomplish this limited objective, WHIRLPOOL performed the task without any modification, using about as much computation time as the contour generation phase preceeding it. The resultant contour polygons are then available for plotting contour maps in which elevation classes can be shaded, and for overlay with other kinds of data zones for analytic purposes.

REFERENCES

Calcomp (California Computer Products, Inc.) (1968). General Purpose Contouring Program. Company Publication, Anaheim, California.

Coulthard, W. J. (1969). Contouring a Grid: Programme Description (UBC CNTOUR). U. of British Columbia Computing Centre, Vancouver, B.C. Canada.

Dougenik, J. A. (1979). WHIRLPOOL: A Geometric Overlay Processor for Polygon Coverage Data. Paper presented at Auto-Carto IV, Reston, Va.

Dayhoff, M.O. (1968). A Contour-map Program for X-ray Crystallography, Communications of the A.C.M. vol 6.

Dutton, G. H. (1977). An Extensible Approach to Imagery of Gridded Data, Computer Graphics, vol 11, no. 2, pp. 159-169.

Dutton, G. H. (1978). Navigating ODYSSEY, Harvard Papers on Geographic Information Systems, vol. 2. Reading, Mass: Addison-Wesley.

Murray, F. W. (1968) A Method of Objective Contour Construction. Santa Monica, Calif.: Rand Corp.

Peucker T. K. and N. R. Chrisman (1975). Cartographic Data Structures, American Cartographer, vol. 2, pp. 55-69.

Rosenfeld, A. (1978) Extraction of Topological Information from Digital Images, Harvard Papers on Geographic Information Systems, vol. 6. Reading, Mass: Addison-Wesley.

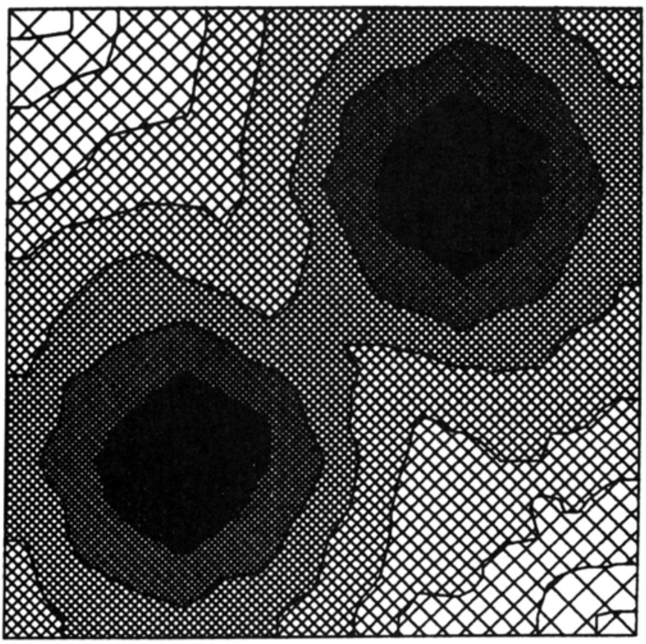

Fig. 3. Shaded Map of Polygons Extracted from a Coarse (16x16 cells) Digital Terrain Matrix