Constructing Shaded Maps with the DIME Topological Structure: An Alternative to the Polygon Approach

Michael E. Hodgson
Geography Department
University of South Carolina

ABSTRACT

Creating a choroplethic map with a vector plotter is routinely performed by filling the regions on the map with a regular set of lines of a specified angle and line spacing. The traditional approach to creating a shaded region map has been to shade each region individually. While considerable effort has gone into designing efficient algorithms to shade regions in a polygon format, little, if any, research has examined the possibility of creating choroplethic maps from other cartographic data structures. This paper presents an algorithm for shading a map of regions described by the Dual Independent Mapping and Encoding (DIME) topological data structure.

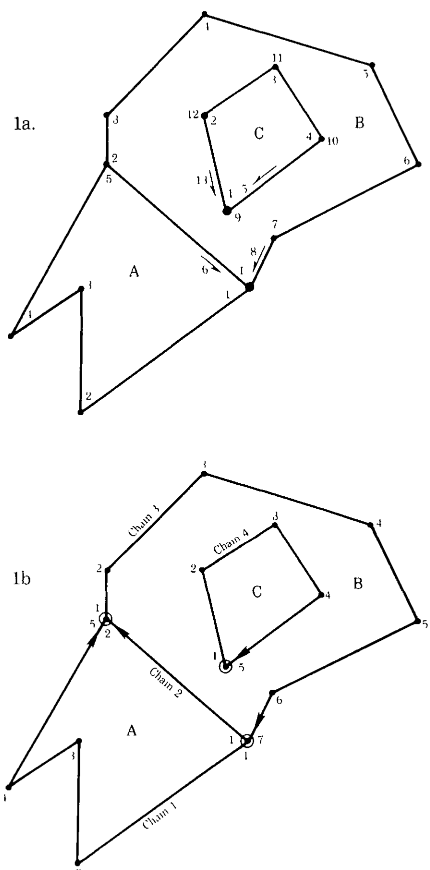
INTRODUCTION

Most modern polygon data bases begin simply as boundaries between regions in the digitizing phase; usually in the form of arcs, DIME segments, or DIME chains. These elemental entities are later linked in a "polygonization" step to form closed polygons, primarily for the purpose of mapping. However, polygonal data bases are inefficient in that they require more storage space (almost twice as much) than arc or "chain" data sets. If editing of the data base is necessary, as in the case of areas undergoing constant change (e.g., street patterns, land use/cover files, etc.), the updates are made using the original elemental boundary file. The boundaries must be linked into polygons after each update for choropleth mapping.

This traditional process of maintaining up-to-date digital cartographic files requires storing both the elemental boundaries and the polygonal files, as well as continuing the intermediate step of polygonizing the boundaries. Since the databases typically begin in the elemental boundary format, it seems unnecessary to polygonize the data merely for constructing shaded maps. Nonetheless, few, if any, efforts have been made for creating choroplethic maps directly from the elemental boundaries. This paper presents a method for constructing such a map using DIME boundaries.

POLYGON AND DIME BOUNDARY CARTOGRAPHIC DATA STRUCTURES

The creation of choroplethic maps with vector plotters are routinely performed by constructing the regular lines for each region individually. Regions are termed **polygons** in this approach, where the polygon is defined as an ordered sequence of coordinates, either clockwise or counter-clockwise around its boundary. Boundary **edges** are implicit in the order of the points. If any islands (holes) occur inside the polygon, they too must be similarly described with the containing polygon as a ordered sequence of coordinates (Figure 1a). Generally, the islands are defined by a closure opposite to the containing polygon. As defined, a polygon retains no information about its contiguous neighbors.



POLYGONAL FORMAT

Polygon A 6 points
$X_1Y_1, X_2Y_2, \ldots X_6Y_6$

Polygon B 13 points
$X_1Y_1, X_2Y_2, \ldots X_{13}Y_{13}$

Polygon C 5 points
$X_1Y_1, X_2Y_2, \ldots X_5Y_5$

DIME BOUNDARY FORMAT

Chain 1 5 points Left 0 Right A
$X_1Y_1, X_2Y_2, \ldots X_5Y_5$

Chain 2 2 points Left A Right B
X_1Y_1, X_2Y_2

Chain 3 7 points Left 0 Right B
$X_1Y_1, X_2Y_2, \ldots X_7Y_7$

Chain 4 5 points Left B Right C
$X_1Y_1, X_2Y_2, \ldots X_5Y_5$

Figure 1. Polygon and DIME boundary representations of an identical area.

In the Dual Independent Mapping and Encoding (DIME) system of boundary representation, the topological relationships are maintained. The basic element of the DIME structure is a directed line segment defined by two points, a beginning and ending point. The **DIME segment** represents a boundary or portion of a boundary between two regions. Codes for the regions on the left and right sides of the segment are included in its description. A **DIME "chain"** is similar to a DIME segment in that it also is the boundary between two regions, but it may be composed of more than two sequential points in defining the boundary (Puecker and Chrisman, 1975) (Figure 1b). Like the elemental DIME segment, the DIME chain has a beginning and ending point and includes codes for the regions on the left and right sides. Regardless of the boundary complexity, a single DIME chain can adequately describe it. For the purpose of this paper, both the DIME segments and DIME chains will be referred to as **DIME boundaries.**

TYPICAL POLYGONAL SHADING ALGORITHM

Several different procedures have been developed for improving the efficiency of the single-polygon shading algorithm (Brassel and Feagas, 1979; Lee, 1981; Cromley, 1984). The procedures for shading the polygon are essentially the same. The typical polygon shading algorithm requires the complete set of coordinates for the containment polygon and any islands along with the parameters for the shading line construction (e.g., line spacing, pen number, angle). The shading lines for a polygon are constructed either independently or as an entire set. For the immediate discussion, the shading lines will be restricted to horizontal lines.

In the "independent approach," the intersections for a single shading line and all boundary edges of the polygon are computed and stored. They are then sorted by their X-coordinate, and implicitly linked and plotted in a "move-to-odd intersection number, draw-to-even" manner. As the intersection coordinates for any given shading line are sorted from left (lowest X-coordinate) to right (largest X-coordinate), moving the pen to the first intersection begins the shading line. The pen is down while moving to the next intersection point to the right, and then back up to the following intersection, and so on. The same procedure for computing intersections, sorting, and plotting is performed for all other shading lines.

In the "entire set approach," the intersections of all the shading lines with every boundary edge of the polygon are computed and stored. They are then sorted by their Y and then X-coordinates, implicitly linked, and plotted. As the shading lines are horizontal, sorting the intersections by their Y-coordinates groups intersections together on their common shading line. Advantages of the "entire set method" is that each edge of the polygon need only be examined once. However, this approach requires more memory and sorting.

In either approach to polygon shading, shading lines at a certain rotation angle may be used if the coordinate points of the polygon are first rotated so that the shading line is parallel with the X-axis (Figure 2). (It is much easier to calculate the intersection of a horizontal line with any

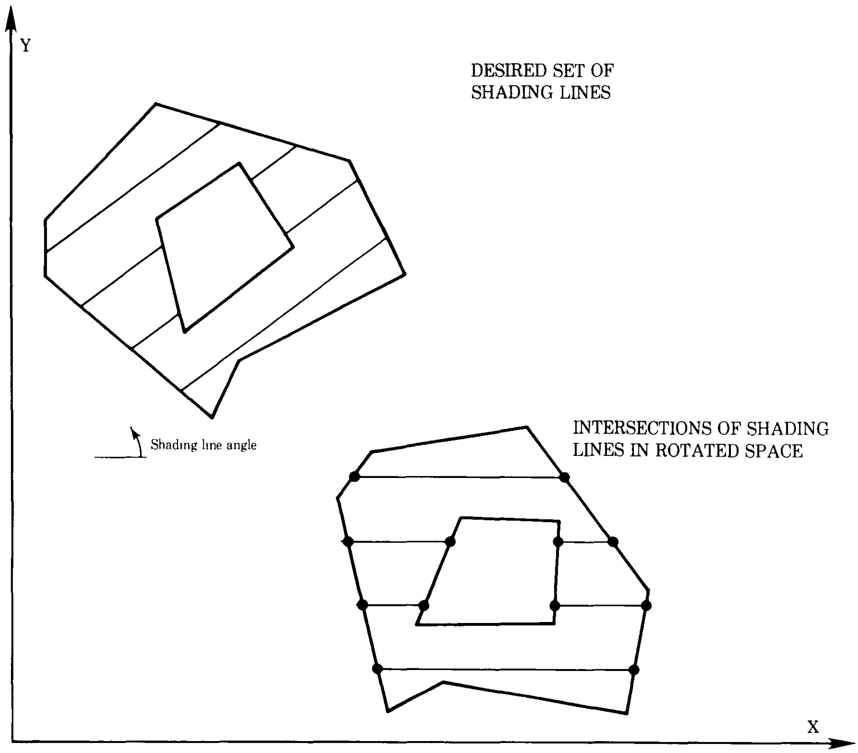


Figure 2. Polygon rotation for calculating shading line intersections.

other line than the intersection with two arbitrary lines.) After computing the intersections, the intersection points are sorted by their X-coordinate and are rotated back into the original space before plotting.

Variants to these basic concepts have been presented by Brassel and Fegeas who decompose the polygon into convex subpolygons in the form of trapezoids during the shading procedure (Brassel and Feagas, 1979). Lee and Cromley each provide additional improvements to the trapezoidal extraction step (Lee, 1981; Cromley, 1984). Convex polygons have the property that a shading line may only intersect the polygonal boundary at two points. This decomposition procedure thus eliminates sorting the intersection points along a shading line by their X-coordinates. It also allows the shading line intersections to be computed in unrotated space. This method uses an incremental displacement for both the X and Y directions.

DIME SHADING ALGORITHM

One alternative to shading regions, other than from polygons, is to use DIME boundaries, as suggested here. The algorithm to create a map of shaded regions from DIME boundaries is graphically presented in Figure 3. Basically, the method involves two general steps: 1) computing the entire set of intersections points for all of the boundaries and storing them, and then 2) sorting the entire group of intersection points with an angle-Y-X sort, linking the intersection points into shading lines and plotting.

Similar to most traditional choroplethic mapping programs a cross-reference table of shading parameters (i.e., angle, spacing) for each region code has been constructed. This table matches data values to be displayed to a set of shading parameters. As the DIME boundary contains region codes for the left and right sides, the shading parameters for a side are found by looking into the cross-reference table.

Arbitrarily beginning with the left side, the boundary is first rotated counterclockwise by the angle of the referenced shading line so that the shading line set is horizontal to the X-axis (Figure 2). The boundary is then decomposed into segments and each segment is then oriented so that it points downward (Figure 4). Each segment now has a beginning Y-coordinate which is greater than or equal to the ending point. Beginning with the first boundary segment, the slope of the segment is calculated. Next, the possible range of shading lines are found by the minimum and maximum Y-coordinates of the points of line segment are found. For each of the shading lines from the maximum Y-coordinate downward, the intersection points are computed.

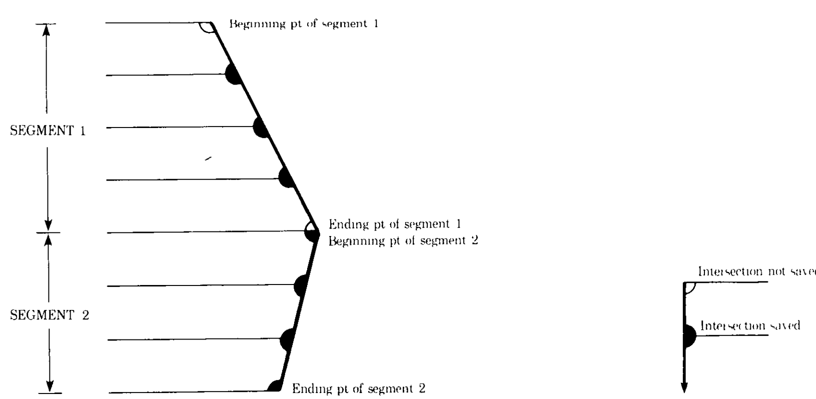


Figure 4. Example of oriented segment and inclusion rule for intersections.

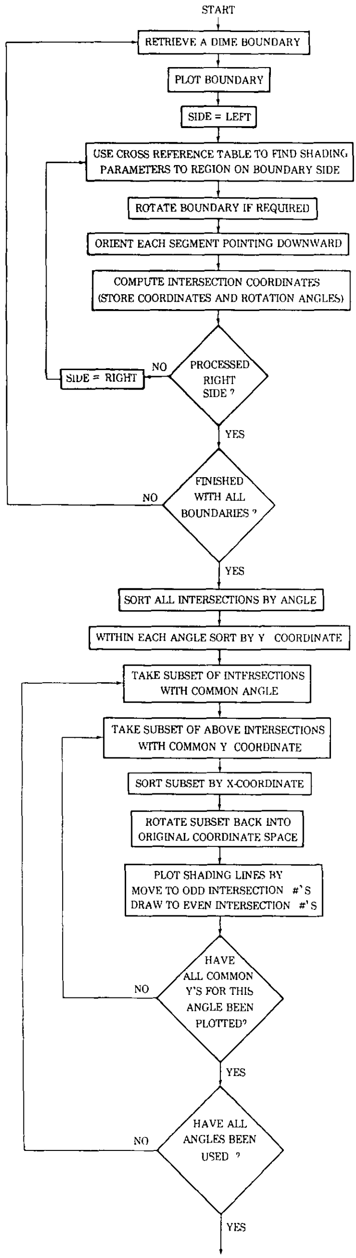


Figure 3. DIME boundary shading algorithm.

By orienting the boundary segments downard before computing intersections, the unique cases where a shading line intersects a beginning or ending point of a segment are handled by the following rule. Intersections created by a shading line with the beginning point of the segments are disregarded. The directional nature of the oriented segments insures the top point of the next segment in the boundary will pick up the intersection of the shading line at this point, and thus, only one intersection will be found (Figure 4). Similarly, horizontal segments are not processed for the same reasons. This inclusion rule also takes care of the cases where a shading line intersects a local peak or pit or a junction of three or more DIME boundaries. As they are computed, the intersection points are then stored along with the appropriate shading angle depicting the shading line set for that boundary. All segments in the boundary are processed in the same manner. The right side of the boundary is then processed.

After all intersection points for the entire map have been computed, they are sorted and plotted. First, they are sorted by their corresponding angle. For each angle, the intersections are sorted by their Y-coordinate and then X-coordinate withing common Y-coordinates. Processing each group of intersections with an identical angle and Y-coordinate, the points are rotated back into the original coordinate space, linked together into shading lines and plotted in. Like the "entire set" approach to shading single polygons, the intersection points are implicitly grouped as shading line segments because of the sorting.

If multiple pens are used (as in the case of pen colors), the pen number for each point would also have to be saved. The pen intersection points should then be sorted by pen number first, then angle, Y-coordinate, and X-coordinate.

An example map using the elementary DIME street segments from the U.S. Bureau of the Census is shown in Figure 5. Note the street segments terminating at a point instead of an intersection. The nature of the DIME shading algorithm allows terminal segments to exist in the regions.

SUMMARY

This algorithm has been presented not as a "better" or more efficient algorithm for shading single polygons, but as an alternative to polygonizing the elemental boundaries simply for the purpose of creating a shaded map. As the DIME boundary shading algorithm is radically different from polygon shading algorithms, constraints common to the traditional shading algorithms are avoided. Typical shading algorithms have limits on the number of islands, the order and closure direction of polygons and islands, and inability to handle terminal segments such as in DIME street files. Region boundaries are only processed once, thus sliver lines are eliminated.

Furthermore, the algorithm is directly applicable to 'coloring' polygons in raster graphics. In such an application, the shading lines would be horizontal and analagous to a scan line on the display screen. Memory requirements for storing intersection coordinates would be significantly less as there is a finite number of scan lines on a display screen.

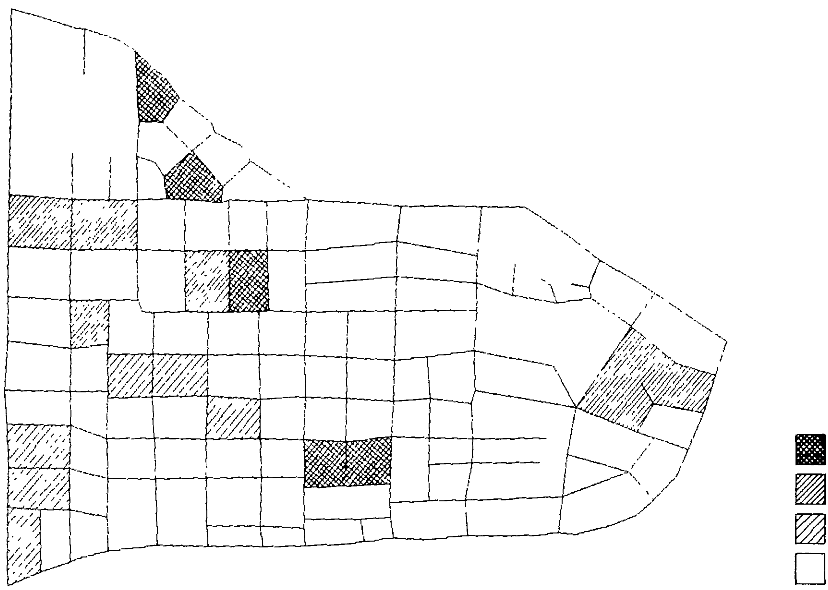


Figure 5. Example map produced using DIME shading algorithm and DIME street segments.

REFERENCES

Brassel, Kurt E., and Robin Fegeas, 1979. "An Algorithm for Shading of Regions on Vector Display Devices," Computer Graphics, vol. 13, pp. 126-133.

Cromley, Robert G., 1984. "The Peak-Pit-Pass Polygon Line-Shading Procedure," The American Cartographer, vol. 11, pp. 70-79.

Lee, D.T. 1981. "Shading Regions on Vector Display Devices," Computer Graphics, vol. 15, pp. 37-41.

Puecker, Thomas K., and Nicholas Chrisman, 1975. "Cartographic Data Structures," The American Cartographer, vol. 2, pp. 55-69.