CALCULATING BISECTOR SKELETONS USING A THIESSEN DATA STRUCTURE

Robert G. Cromley
University of Connecticut
Storrs, Connecticut 06268

ABSTRACT

An important construct for analyzing the shape and structure of a polygon in Euclidean space is its bisector skeleton. A bisector skeleton partitions the area of a polygon into sub-polygons that are closer to one edge of the polygon or its internal linear extension than to any other one. While bisector skeletons are unique in their form and application, their cartographic elements are topologically equivalent to those of a Thiessen diagram. Consequently, procedures for calculating Thiessen diagrams may be adopted for calculating bisector skeletons, just as Thiessen procedures can be applied to the problem of Delaunay triangulation. This paper presents an algorithm for constructing bisector skeletons using a triangle data structure and the form of a procedure for identifying Thiessen diagrams within a convex boundary.

INTRODUCTION

An important class of cartographic problems is related to the partitioning of space based on proximity criteria. One proximity problem given a known point distribution, is to delineate the set of points on a surface that is closer to one known center than to any other points. This problem of constructing a Thiessen diagram has many applications in economic geography, quantitative techniques, and cartography. Another proximity problem is that of constructing the bisector skeleton of a given polygon. Bisector skeletons partition the internal area of a polygon into subpolygons that are closer to one edge of the polygon or its internal linear extension than to any other edge and its extension (Brassel, Heller, and Jones, 1984). The inclusion of internal edge extensions in its definition distinguishes bisector skeletons from earlier continuous skeletons (Montanari, 1969) and gives them their strictly linear appearance (see Fig. 1). It should be noted that while Thiessen polygons are convex, bisector polygons may be either convex or concave (again see Fig. 1).

Algorithms for analytically delineating Thiessen polygons have received more attention (Rhynsburger, 1973; Shamos, 1977; Brassel and Reif, 1979) than the newer developed problem of constructing bisector skeletons. Recently, an algorithm for identifying and storing a Thiessen diagram *within a convex boundary* has been proposed based on a triangulation data structure (Cromley and Grogan, 1985). The purpose here is to apply the design of this procedure to the problem of calculating bisector skeletons after showing the topological equivalence of respective cartographic elements of each diagram.

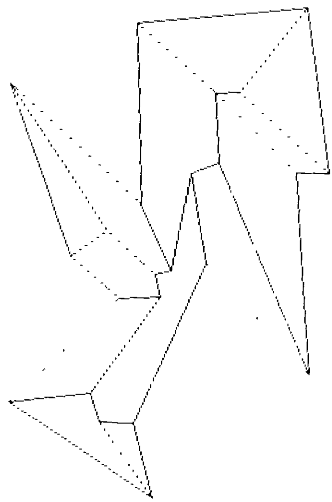

Figure 1. A Bisector Skeleton

CONCEPTUAL BACKGROUND

There is a clear analogy between the components of the bisector skeleton problem and identifying a Thiessen diagram within a convex boundary. For the Thiessen problem, a set of n points is given in a plane bounded by a convex polygon defined by a set of m edges. Each edge of the bounding polygon is a line segment connecting two boundary vertices; thus, the polygon is alternatively referenced by m vertices (Cromley and Grogan, 1985). It is assumed that the bounding vertices are sequentially numbered in a clockwise direction so that the Thiessen diagram is always to the right as one moves around polygon boundary. For the bisector problem a polygon composed of n edges is given in a plane. Each edge of the polygon is a line segment connecting two boundary vertices; again the polygon is alternatively referenced by these n vertices. It is also assumed that the polygon's vertices are sequentially numbered such that as one moves from vertex to vertex, the area of the polygon lies to the right of the connecting edge.

The set of n points are used in the Thiessen problem to generate convex polygons that are nearer to one point or

Thiessen centroid than to any other centroid. Likewise, the set of n edges of a polygon are used in the bisector skeleton problem to delineate subpolygons within the given polygon that are closer to one edge or its interior extension than to any other edge or its corresponding extension. Given an identical function within the context of the respective problem, each Thiessen centroid, C_i, is equivalent to each polygon edge, P_i.

The problem of identifying a corresponding polygon for each centroid is equivalent to finding a set of p points that are equidistant and closest to three centroids (Fig. 2); these points are called Thiessen vertices. Similarly, a skeleton vertex is a point equidistant and closest to three polygon edges or their interior linear extension (Fig. 3) (Brassel et al, 1984). Thiessen vertices and skeleton vertices are also equivalent as they represent junctions along the perimeter of local polygons where the generating centroids (polygon edges) change neighboring centroids (edges). Additionally, each Thiessen vertex that lies on the convex bounding polygon is called a boundary Thiessen vertex while the others are known as interior Thiessen vertices. Similarly, skeleton vertices will either lie in the interior of the polygon or on its boundary; in the latter case, the set of boundary skeleton vertices is identical to the original set of n polygon vertices.

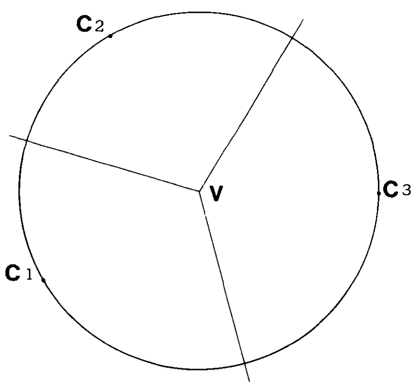

Figure 2. A Thiessen Vertex and Its Nearest Centroids

Finally, a Thiessen edge, E_i, is defined as the locus of points equidistant and closest to two centroids. A Thiessen edge will connect two Thiessen vertices that share two nearest centroids. Thiessen edges connecting two boundary Thiessen vertices are known as boundary Thiessen edges. For a bisector skeleton, a skeleton edge, S_i, is the locus of all points equidistant and closest to two polygon edges.

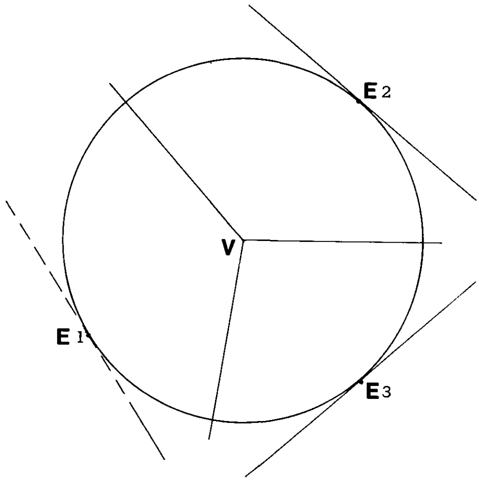

Figure 3. A Skeleton Vertex and Its Nearest Edges

Skeleton edges will connect skeleton vertices that share two closest polygon edges. Therefore, skeleton edges that connect two boundary skeleton vertices are also called boundary skeleton edges. In this case, the set of these boundary skeleton edges is the same as the set of polygon edges; in other terms, a polygon edge in the bisector skeleton problem is equivalent to two different elements in a Thiessen diagram with a convex boundary: a Thiessen centroid and a boundary Thiessen edge.

Additionally, one centroid is called a Thiessen neighbor of another centroid if the two centroids' polygons cobound the same Thiessen edge. It should be noted that some edges may connect two unique vertices that have the same cartographic location which gives the visual impression that one vertex has more than three nearest centroids and that some centroids share only a common vertex rather than an edge. However, only the centroids sharing the zero length edge are neighbors of each other (Cromley and Grogan, 1985). All centroids with a boundary Thiessen edge are neighbors of an imaginary background centroid, C_{n+1}. Analogously, one polygon edge is a skeleton neighbor of another if the respective subpolygons share a common skeleton edge. By definition, each polygon edge will be a skeleton neighbor of an imaginary background edge, P_{n+1}.

It is important to enumerate each Thiessen (skeleton) vertex and edge as an exact number of them exist as a function of the number of centroids (polygon edges). Cromley and Grogan have shown that a Thiessen diagram with n centroids will have 2(n-1) vertices and 3(n-1) edges. Similarly, a polygon with n edges will have 2(n-1) skeleton vertices and 3(n-1) skeleton edges. While it is unknown how many boundary Thiessen vertices there will be, it is always the case that there are exactly n boundary skeleton vertices and therefore n-2 interior skeleton vertices.

It is also important to enumerate each Thiessen or skeleton vertex because respective vertex reference files can be constructed based on the topological relationships between vertices and centroids (polygon edges). Because a Thiessen diagram is the dual of a Delaunay triangulation, a Thiessen data file is based on Elfick's triangle structure (Cromley and Grogan, 1985). Each record of the file contains six entries corresponding to the neighborhood information of each unique vertex and two entries for its coordinates. The first three neighborhood values contain the references of the three adjoining vertices recorded in a counter-clockwise order. The next three entries are the reference values of the corresponding centroid (polygon edge) whose generated polygon is the right-hand neighbor of the edge connecting the current vertex to an adjoining vertex. Thiessen or skeleton edges are not retained in this file as they are line segments connecting vertices and would be redundant information. Once all 2(n-1) records have been completed, a digital representation of a bisector skeleton or a Thiessen diagram is complete.

ALGORITHM DESIGN

Cromley and Grogan have presented a two stage method for constructing the vertex reference file for a Thiessen diagram. In the first stage, all Thiessen boundary vertices are enumerated by walking around the outline of the bounding polygon in a clockwise direction. As each boundary vertex is found, its three centroid neighbors and the two adjoining vertex neighbors that are also boundary vertices are identified. Only the third vertex which is an interior vertex remains to be identified. In the second stage, the interior vertices are found by moving along the boundary of each individual Thiessen polygon in a clockwise manner. This process starts by first enumerating those Thiessen polygons that have a boundary Thiessen edge and then continues in an inward spiral until all polygons and vertices have been found. As the boundary of an individual polygon is completed, its generating centroid is removed from the list of potential centroid neighbors for new vertices.

A similar procedure can be applied to the bisector skeleton problem. In this case, the first stage is trivial as the set of boundary skeleton vertices is the same as the given set of polygon vertices. The second stage is also less complicated as the spiral process terminates when the last subpolygon having a boundary skeleton edge is completed as there are no subpolygons in a bisector skeleton that are completely interior to the bounding polygon.

While the overall design of the bisector skeleton procedure is the same as a Thiessen procedure, there are many technical details that differ. First, the edge that partitions the subpolygons is formed by the bisector of the angle between two polygon edges or their extensions rather than the perpendicular bisector between two centroids. Second, Brassel and Reif's circle test for finding centroids that are closer neighbors cannot be used. Instead, the third neighbor (a polygon edge in this case) for each interior vertex is found by sequentially testing each polygon edge in a clockwise order. A half-plane test is used to determine if the last vertex lies in the same half-plane as the edge being tested or in the half-plane of the current potential edge. The current potential edge is updated whenever the vertex is in the half-plane of the new edge. Finally, as one proceeds around the boundary of each subpolygon, only those edges that are subsequent in the clockwise order of the last polygon edge neighbor need to be tested as potential neighbors for new interior vertices of the current subpolygon. This algorithm has been implemented in FORTRAN 77 and used to construct Fig. 1.

SUMMARY

The algorithm presented here has shown that calculating bisector skeletons is analogous to that of calculating Thiessen diagrams. Although many cartographic entities have very different forms and functions, their digital form is often quite similar. This enables digital methods to be more integrated than their manual counterparts.

REFERENCES

Brassel, K., M. Heller, and P. Jones, 1984, The Construction of Bisector Skeletons for Polygonal Networks: Proceedings, First International Symposium on Spatial Data Handling, Vol. 1, pp. 117-126.

Brassel, K. and D. Reif, 1979, A Procedure to Generate Thiessen Polygons: Geographical Analysis, Vol. 11, pp. 289-303.

Cromley, R. and D. Grogan, 1985, A Procedure for Identifying and Storing a Thiessen Diagram within a Convex Boundary: Geographical Analysis, Vol. 17, pp. 167-175.

Elfick, M., 1979, Contouring by Use of a Triangular Mesh: The Cartographic Journal, Vol. 16, pp. 24-29.

Montari, U., 1969, Continuous Skeletons from Digitized Images: Journal of the Association for Computing Machinery, Vol. 16, pp. 534-549.

Rhynsburger, D., 1973, Analytic Delineation of Thiessen Polygons: Geographical Analysis, Vol. 5, pp. 133-144.

Shamos, M., 1977, Computational Geometry, Ph.D. dissertation, Yale University, 1977.