AREA-LINE GEOMETRY CHANGES IN MODEL GENERALIZATIONS:
TRIANGULATION METHOD

H. Zahit Selvi* 1, Öznuğ Bildirci*

*Selcuk Univ., Faculty of Engineering and Architecture, Dept. of Geodesy and Photogrammetry, Kampus, Konya,
hzselvi@selcuk.edu.tr, jobbildirci@yahoo.com

KEYWORDS: Model Generalization, Geometric Change, Triangulation, Waterlining, Straight Skeleton, Thinning

ABSTRACT:
Reducing the real world reality to map scale makes generalization necessary. This reduction influences geometry and attributes of objects. In model generalization, area-line, area-point and line-point changes occur. In this study, area-line changes are handled. Since area objects have different geometric properties, there is no universal solution that is applicable to all kinds. Therefore there are several methods applicable to certain objects under certain circumstances. Priorities of methods are accuracy, processing time and optimal use of computer memory. We first discuss why we need area-line geometry changes, and try to explain situations that area-line geometry changes are necessary. Thereafter we talk about four area-line geometry change methods, triangulation, waterlining, straight skeleton and thinning, especially triangulation based approach in detail. Advantages and disadvantages of the methods mentioned are also explained. Finally an algorithm that is based on triangulation is explained in detail. The authors are developing a script based on this algorithm, which will work under NETCAD software. Preliminary results are also given.

1. INTRODUCTION

The concepts of digital mapping began to be important worldwide because of developments in computer technology. As a result, mapmaking on a computer is getting easy and common, and many studies are continuing either in academic areas or in other applicable areas because generalization that is one of main subjects of cartography is automated.

Defining generalization in cartography is a quite complex study, because objects in cartography are so various and they have extremely different structures from each other. Difficulties in defining cartographic objects and varieties of requirements prevent from creating cartographic generalization software that satisfies all demands. Therefore, cartographic objects are classed as area, line, and point objects and generalization patterns for each type of objects is to be developed.

Geometry changes in model generalization are classified as area-line geometry changes (i.e. defining street centerline), area – point geometry changes (i.e. defining city or town with a point ), line – point geometry changes (i.e. defining bridge with a point ). We discuss area – line geometry changes in detail.

We need area – line geometry changes because of two reasons. Particularly the objects like roads, rivers etc. can be represented with boundary lines at most in 1:5000 scale, in smaller scales centrelines, which are defined by area-line geometry change methods, are required. On the other hand, spatial analyses, which are very important for Geographical Informational Systems, make necessary these geometry changes, even in large scales.

In the second part of this paper, we talk about four methods on area-line geometry changes: triangulation, waterlining, straight skeleton and thinning. These methods are compared with each other and advantages and disadvantages are also discussed. In the third part, an algorithm that is based on triangulation is explained in detail. The authors are developing a script, based on this algorithm, which will work under NETCAD software. Preliminary results are also given. Finally, outcomes of this study and proposals for future are taken place.

2. AREA – LINE GEOMETRY CHANGES

2.1 Triangulation Method

This method is based on Voronoi diagram and Delaunay triangulation. Therefore we explain this two concept firstly. The Voronoi diagram for a set of point sites \((x_1, x_2, \ldots, x_n)\) in the plane is the decomposition of the plane into maximally connected regions that have the same set of closest sites. There are three types of regions. Each site \(x_i\) has a Voronoi cell, which consists of those points with \(x_i\) as the unique closest site. Voronoi edges, with two closest sites, are segments of the perpendicular bisector between the sites. Voronoi edges meet at Voronoi vertices, which are points that are equidistant to three or more closest sites. The Delaunay triangulation of a set of points \(P\) is the triangulation of \(P\) in which every triangle satisfies an empty circle property: the circumcircle of a triangle \(T\) does not contain any point of \(P\) in its interior. Mathematically, the Voronoi diagram and the Delaunay triangulation are duals (Figure 1).

Each diagram contains the same topological information and one diagram can be derived from the other (McAllister and Snoeyink, 2000).

Figure 1: The Voronoi diagram (dashed) and Delaunay triangulation (solid) of points (McAllister and Snoeyink, 2000)

The centerline of a polygon is a subset of the Voronoi diagram whose elements are the open edges and vertices of the polygon. The edges of the Voronoi diagram that lies inside a polygon \(P\) contains the centerline of \(P\), so, the
Christensen developed water lining method (Christensen, 1996). He thinks defining polygon edge with line segment is difficult when the shape is complex, and he offers polyline for defining polygon edges.

The major difficulty in the centerline determination is the same one found in most spatial problems: How to make the computer see the whole picture? No one has so far succeeded in presenting the computer with such a complete picture. The best currently available solution is to organize the data into smaller and more understandable pieces. So Christensen considers using polyline is the best idea and uses waterlines defined like buffer operator in Geographic Information System (GIS) software for creating centerline.

This algorithm learns the shape in question in a gradual manner, through a regulated reduction in size of the perimeter that yields simpler but more numerous shapes. The reduction in size is achieved by the generation of lines parallel to the original shape (Figure 3).

The waterlining of a shape is the first step in the gradual approach to the construction of a centerline. The process of obtaining a single waterline from the previous one is similar to the buffering included in the command repertoire of most vector GIS. The first property of waterlines is that any waterline is at a constant distance from the boundary of the shape. However, distance from one waterline to the next may increase (Figure 3). The waterlines running parallel to the shore until they collide with other waterlines coming from the opposite shore. The colliding waterlines then merge into polygons that show changes of direction at the collision points. Joining collision points would then create a structure that could be taken as an approximation to the centerline of the shape.

The method accepts the input data that are unstructured polylines. The polylines are then automatically organized for the waterlining step. The result of this step is a set of files, one for each region. These files contain closed polygons, carrying the geometry, the total distance to the perimeter, the distance to the previous and to the next polygon, as well as a variety of other data elements.

During the second step the collision points are automatically detected. The collision points are then associated in links, single straight segments that inherit properties from the waterlines. Because of needing very large computer memory, very large data sets must be partitioned for this method. When we partition the data, a narrow overlap band must be extracted from the adjacent sheets for ensuring continuity. This band carries additional information to the adjacent sheets. Most of the processing time is taken by the waterlining. The other steps are finished less time.

The most important advantage of this method is its flexibility. Changes on a part of the shape, affect only that part of

中心线可从Voronoi图推导出来。McAllister和Snoeyink (2000) 谈了三个不同的近似方法来创建基于Voronoi图的中心线 (Figure 2)。

The first approximation uses the subset of the Voronoi diagram induced by the Voronoi vertices that correspond to Delaunay triangles. Two adjacent Delaunay triangles correspond to two Voronoi vertices that are joined by a Voronoi edge. It means that if we join the centers of circumcircles of adjacent triangles, we can construct the centerline. This approximation is called “Voronoi approximation”. But, unfortunately, if the discretized points along a boundary edge are far from one another, then this approximation has a zig-zag pattern rather than the expected smooth centerline (Figure 2).

The second approximation, which is called the “centroid line approximation”, joins the centroids of adjacent and Delaunay triangles into paths. Given a triangle with vertices a,b, and c, the centroid (a+b+c)/3 always lies inside the triangle. Although the centroid is a natural choice as a representative point for a triangle, the paths between centroids are not smooth. When the triangles have one side much smaller than the other two sides, then the centroid line approximation zig-zags again (Figure 2).

The third approximation is called “the midpoint line approximation”. The midpoints of the edges of Delaunay triangles, which are not on the boundary of the polygon, constitute the centerline. If the points on the boundary of the polygon are scattered appropriate especially the third approximation gives good results. But if the shape of polygon is complex, application of this method is very difficult. And also if the polygons have holes, the resulting centerline must be check manually. An application based on this method will share in the last part of this paper.

2.2 Waterlining Method

Since the triangulation method is poor on complex shape, Christensen developed waterlining method (Christensen, 1999). He thinks defining polygon edge with line segment is
centerline, not affect all part of it. Since the process stores all of the data, we can reconstruct the polygon from the centerline.

### 2.3 Straight Skeleton Method

Since the idea of combining triangulation and waterlining methods seems beneficial, straight skeleton method was developed by.. Straight skeleton is solely made up of straight line segments which are pieces of angular bisectors of polygon edges. In this method, the polygon edges are moving in constant speed inward the polygon and they are changing their lengths. The polygon vertices move along the angular bisectors as long as the polygon changes its topology (Figure 4). The changes of topology define straight skeleton. Two possible types of changes are described by Felkel and Obdrzalek (1998):

**Edge event:** An edge shrinks to zero, making its neighbouring edges adjacent.

**Split Event:** A reflex vertex runs to this edge and splits it, thus split the whole polygon. New adjacencies occur between the split edge and each of the two edges incident to the reflex vertex.

The straight skeleton of the polygon is defined as the union of the pieces of angular bisectors traced out by polygon vertices during the shrinking process. After either type of events, we are left with a new, or two new, polygons which are shrunk recursively if they have non zero area. The algorithm of this method is defined by Felkel and Obdrzalek (1998) as below:

The basic data structure used by the algorithm is a set of circular lists of active vertices (SLAV). In the case of convex polygon, it always contains only one list, in the case of simple non – convex polygon, it stores a list for every sub polygon (split event ). All the vertices in the SLAV have references to both neighbors (vertices of the polygon ) in the circular lists stored in SLAV. The angular bisectors are computed in every polygon vertices. For each vertex compute the nearer intersection of the bisector with adjacent vertex bisectors and if it exists, store it into a priority queue according to the distance to the edge. When we finished to compute intersection of the bisectors in every vertices, look at priority queue and for every bisectors research intersection points, if two or more intersection points exist on a bisector, we keep related polygon edge, if not edge event is done (Figure 5). After the edge event SLAV is reconstructed and algorithm continues until the centerline is created.

The principle of the method for the straight skeleton computation is in the case of non – convex polygons similar. But a reflex vertex may lead into a polygon splitting in non – convex polygons, and split event is done, so it is necessary to split the polygon into two parts at split point. Splitting of the polygon also implies splitting of the SLAV into two parts, and the algorithm continues with two polygons as in the case of convex polygon.

If this method is compared with the other area – line change methods, straight skeleton contains only straight-line structure, not crooked structure. The straight skeleton is more sensible to changes of the shape. Adding a reflex vertex with very small exterior angle may alter the skeleton structure completely. According to Christensen most disadvantages of this method is similarity of centerlines, which belong different polygons.

### 2.4 Thinning Method

Thomas (1998) offers rasterization – skeletonization – revectorization approach. So we can create centerline using thinning method based on vectorization. Process of the method also included rasterization of vector map is examined by Thomas (1998) as below:

Run – Length Encoding (RLE) process translates the vector map into a binary image by casting a grid over the map and identifying those cells of the grid that cover the geometric elements of the map. After this, each pixel is labeled for representing the region. The errors, i.e., overshooting and layer misplacing, will be corrected after region labeling. Run Length Encoded version of a vector map contains a set of closed lines, which divides the plane into connected regions. In order to separate these regions the image is first inverted. The lines thus become the background, and the resulting connected regions in the foreground. After these processes image is ready for skeletonization that our aim is.

The desired skeleton, being one – pixel width, must preserve homotopy to ensure that the net topology is properly captured. A skeleton is one pixel width when each of its points is necessary to keep connectivity. In order to obtain the desired skeleton, thinning process consisting of a successive removal of the outer pixels of the regions while retaining those necessary to keep connectivity, until no more progress is made is applied. The deletion or retention of a (black) pixel P would depend on the configuration of pixels in a local neighbourhood containing P (Figure 6).

Crossing number is very important expression for thinning algorithms. Rutovitz and Hilditch define crossing number deferently. Rutovitz first proposed this useful measure of
connectivity as the number of transitions from a white point to a black one. This crossing number can be defined as,

\[
X_R(P) = \sum_{i=1}^{\text{\# of transitions}} |x_{i+1} - x_i|
\]  

(1)

If \(X_R(P)\) is equal to 2, then \(P\) can be deleted. Hilditch defines the crossing number \(X_H(P)\) as the number of times one crosses over from a white point to a black point when the points in \(N(P)\) are traversed in order.

\[
X_H(P) = \sum_{i=1}^{\text{\# of times crossed}} h_i
\]  

(2)

If \(X_H(P)\) = 1, \(P\) can be deleted.

But crossing number properties are not enough for deleting a pixel. The different conditions must be done for creating desired skeleton. What are these conditions and in which situations are these conditions important, are explained by Lam et al. (1992) in detail. In this paper, we do not explain the rules of pixel deletion in detail.

Because the obtained skeleton is one pixel width, each pixel has only one or two neighbours unless it is placed at a crossing area. Thus, crossings can be obtained by labelling each pixel according to its number of neighbours. The result can then be seen as a set of separate skeleton segments and crossings, as shown Figure 7.

3. A NEW TRIANGULATION APPROACH

3.1 Data Structure in Turkey

In Turkey it is common that the roads are not explicitly defined in most digital data sets. In other words, there are no polygon objects for roads. They can be extracted by subtracting all the polygons from the background. This is the case for large-scale maps, such as 1:1000 and 1:5000. We introduce a triangulation-based algorithm for this common data structure. We are also developing a script with this algorithm. This Visual Basic script will be run under the NETCAD software, which is Turkish CAD and GIS software.

![Figure 8: Data structure of large-scale maps in Turkey](image)

3.2 Algorithm

Before the actual process we first add new points to the foreground polygons because optimum point dispersion is required for preventing undesired triangles. The criterion for adding new points is the parameter “\(d\)” that is given by users, which can be thought as average road width. We assume that the maximum distance between points along polygons should be smaller than 1.5*\(d\). The new points are created in a temporary layer and are given special names such as 102/1, where 102 denotes the polygon number, 1 the point number (Figure 9).

![Figure 9 Adding new points to closed areas](image)

After the intensification of points, all are triangulated. In this stage constrained Delaunay Triangulation is applied because...
we desire that triangles coincide the polygons. In other words, the edges of triangles and the edges of polygons cannot intersect. Then, the triangles that are within the polygons are deleted. The remaining network of triangles represents the road network, and these triangles are called “road triangles” (Figure 10).

For constituting desired centerlines, we use midpoint line approximation, which is explained above in triangulation method. In this approach, the edges of road triangles, which are not on borders of polygons (closed areas), are used for defining centerlines.

We define three types of triangles for this algorithm.
- Type A: The triangle, which has two neighbour road triangles (Figure 11).
- Type B: The triangle whose three vertices belong to three different polygons (Figure 11).
- Type C: The triangle that has only one neighbour road triangle, or one of the neighbour triangles is of Type B (Figure 11). It can have two neighbour triangles at most.

The steps of the algorithm can be explained as follows:

- Algorithm starts with any Type C triangle. It examines the triangle, which edge connects different polygons (i.e. 101/8, 102/3), the midpoint of this edge is a point on the centerline. After finding such two edges, the triangle is marked as processed.
- An edge of Type C triangle is coincided with neighbour Type A triangle, so the algorithm moves to the neighbour triangle. The midpoints of suitable edges are saved as points of the centerline. Algorithm continues to examine all triangles like this and marks all. When it comes to a Type B triangle, this centerline is finished.
- Algorithm moves to next unmarked Type C triangle, and does same processes until it finds a Type B triangle.
- When the algorithm finishes to examine every triangle, desired centerlines are obtained (Figure 11).
- At the end, all triangles and temporary points, are deleted.

4. CONCLUSION

Different experts offer different methods for area line changes. Unfortunately the common GIS (Geographic Information Systems) software does not include sufficient modules for this purpose. In many cases operators are drawing centerlines manually. Therefore, we experience insufficient data, high cost and too much working hours. Because of these reasons we need to automate area line changes. In this paper, we introduced four different methods, and tried to explain advantages and disadvantages of them. A new triangulation approach, which is suitable for Turkish data, is suggested. In near future, we will have implemented the algorithm as a VBScript under NETCAD software.

ACKNOWLEDGMENT

This authors wish to acknowledge for the cooperation and assistance of Hasan Cevik and Yaser Bozbıyık and the financial assistance given by the Scientific Research Found (BAP) of Selcuk University.

5. REFERENCES


Cacciola, F., 2000. A CGAL implementation of the Straight Skeleton of a Simple 2D Polygon with Holes, is not published.


