THE PROBLEM OF CONTOUR IN THE GENERATION OF DIGITAL TOPOGRAPHIC MAPS

Silvania Avelar
Remote Sensing Center
Federal University of Minas Gerais
30161-970 Belo Horizonte, Brazil
silvania@csr.ufmg.br

ABSTRACT:

Topographic maps are generated using Digital Terrain Models (DTMs), which provide the basis for numerical solutions of several important problems, such as the determination of contour lines of the terrain. Because DTMs do not address the question of the shape of the region worked with, in certain cases they may represent the region imprecisely. This work is concerned with the contour problem in the generation of topographic maps. By contour we mean a simple polygon that bounds a region containing all points gathered in the terrain. This paper presents a technique to determine a contour using geometric characteristics of the terrain data.

1 INTRODUCTION

Problems involving terrains are well documented in the literature (Van Kreveld, 1996). Work in this area, besides being useful for society, is especially interesting for computational geometry.

Several mathematical models have been used to represent the terrain numerically, but they usually do not take into account the shape of the region worked with. In general, algorithms for terrain modeling consider the convex hull (Preparata and Shamos, 1985) of the set of points In doing so, when the region of interest is not convex, they can induce wrong results. Although this question has been subject of intense research in computer science, in this scenario it remains without a suitable solution, to the best of the author's knowledge. The work presented in the following sections is a contribution in this direction.

This work address the contour problem in the generation of topographic maps. By contour we mean the polygonal curve, not necessarily convex, that bounds the polygonal region containing all points gathered in the terrain. We will focus the contour problem in the contour lines layer of topographic maps.

We begin by briefly presenting aspects related to the digital generation of topographic maps. Then, we describe a solution to the following problem: given n points $p_1, p_2, \ldots, p_n$ of the plane, compute the boundary polygon which fits the region containing these points better than the convex frontier. The aim is to minimize the imprecision in the representation of a terrain.

This paper is organized as follows. The next section discusses the Digital Terrain Models and the contour problem. Section 3 presents the methodology used to generate topographic maps considering contour determination. Section 4 describes a contour-generating algorithm. The following section makes some practical considerations about the implementation and shows one result obtained. Finally, concluding remarks are presented in the last section.

2 DIGITAL TERRAIN MODELS AND THE CONTOUR PROBLEM

For generation of digital topographic maps, a mathematical model describing the terrain is required, and this description must be as close as possible to the terrain's real aspect. In general, a Digital Terrain Model (DTM) is composed by points sampled from the region under study.

Digital Terrain Models are classified according to the mathematical model used. Interpolation models (network of points/tesselations) are usually preferred to approximation models (analytical equations) (Sakude, 1992). Based on the spatial distribution of the sampled points, the models can have a regular distribution (square, rectangular and triangular tesselations) or an irregular one. Despite their frequent use, regular tesselations do not yield a good representation of the variations of the terrain, because they are created artificially (Buys et alli, 1991). Tesselations of irregular distributions based on the original points gathered in the terrain can define more precisely the region in study. Because the triangle is the minimum polygon, irregular tesselations are usually triangulations.

There are many possible different triangulations for the same point set. Intuitively, a "good" triangulation for the propose of terrain modeling is the one in which triangles are as equiangular as possible. In other words, it is desirable to avoid long and thin triangles (De Floriani, 1987; Falcidieno and Spagnuolo, 1991; Buys et alli, 1991).

The Delaunay triangulation, a fundamental construction in Computational Geometry, is as equiangular as possible, and for this reason it is a standard tool in terrain description (Preparata and Shamos, 1985). Besides its very good capability of terrain modeling, it also saves on computation time with the choice of a suitable data structure. However, the domain of the Delaunay triangulation is the convex hull of the point set, and in certain situations the region of interest is not convex. Thus, another kind of frontier is necessary. In the case of sinuous regions like roads, for instance, this is a serious problem, because computations are extrapolated to places not known in the original region. In practice, this can invalidate the resulting topographic maps.

In this context, it is necessary to determine a contour to points set, minimizing extrapolation errors. We present a mechanism that minimizes this type of problem, while using the Delaunay triangulation. In the next sections we describe an algorithm which dynamically modifies the original convex hull to address this situation.

3 METHODOLOGY

Using the coordinates x, y and z of the points gathered in the terrain, the adopted method to the generation of topographic maps, considering their contours, consists in five fundamental steps (Figure 1).

1. Partition of the region into triangles, using the Delaunay triangulation of the sampled points.
2. Determination of a polygon smaller than the convex hull contouring the point set.
3. Elimination of the Delaunay triangles outside the new contour.
4. Computation of the points that will constitute the contour lines for the representation of the relief.
5. Design of the topographic map: B-Spline interpolation of the constituting points of the contour lines and insertion of information of other required layers.

A number of well-known algorithms exist to implement each of the steps above, except for step 2, in which we will work.

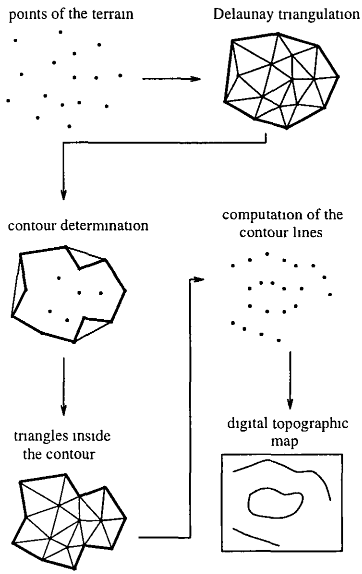

Figure 1: The applied methodology.

4 THE CONTOUR-GENERATING ALGORITHM

4.1 Definitions

For a given finite set of points of the plane, we wish to determine a simple polygon, with a smaller area than the convex hull, which bounds the polygonal region containing the points.

A polygon is defined as an ordered sequence of n ($n \geq 3$) points in the plane, $p_1, p_2, \ldots, p_n$, and the edges $p_1p_2, p_2p_3, \ldots, p_{(n-1)}p_n$ and p_np_1 formed by them. A simple polygon is a polygon with the restriction that non-consecutive edges do not intersect (Shermer, 1992). The convex hull of the point set defines the convex polygonal region with the smallest area that contains the points.

4.2 Description of the algorithm

The algorithm determines the contour departing from the known convex hull of the points. The idea is to dynamically modify the convex frontier, looking for candidate points to constitute the new edges of the searched contour. A circle is used to determine the candidate points to be analysed.

Beginning at one of the edges of the convex hull, a circle having this edge as diameter is drawn. The points of the set that lie within this circle are the candidate points. A candidate edge for the new contour is obtained by joining the candidate point closest to one of the two vertices of the edge under consideration to that vertex to which it is closest. To verify whether this is an acceptable edge, we form a triangle with the two vertices that defined the circle and the candidate contour point. Should the triangle so formed not contain any other point, the chosen point is accepted and the candidate edge will form part of the final polygon sought. In this way, the edge which was being worked with can be discarded and replaced by two new ones, reducing the area delimited by the contour.

This process is repeated recursively to one of the generated edges until the circle drawn contains no other points In this case, the edge is kept and the process goes on to the next edge of the convex hull. On completion of the process, the contour sought is produced. The convex polygon may be processed in a clockwise or anti-clockwise direction. The direction in which the polygon is processed affects the shape of the final contour.

A more formal description of the contour-generating algorithm is:

1. Let $c_1, c_2, \ldots, c_n$ be the vertices of the determined convex hull, whose edges are ordered as $c_1c_2, c_2c_3, \ldots, c_nc_1$. The coordinates of the vertices $c_1, c_2, \ldots, c_n$ are $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ respectively.
2. Each of the edges of the convex hull is worked on separately. To start the algorithm, one vertex of the convex hull is chosen as c_1 (for example, that vertex with the smallest y coordinate value) and a direction (clockwise or anti-clockwise) is chosen for proceeding the algorithm Starting from the edge c_1c_2, for example in the anti-clockwise direction, the circle C is determined whose diameter is given by the length of the edge being worked with and whose center lies at the mid-point of this edge.
3. The circle C will establish a region for analysis equivalent to a half-circle in which may lie points which will determine reentrances, defining edges different from the previous one. Then it is determined which points (x, y) lie within C.
4. Considering the points inside C, it is determined the closest one, called p_t, to one of the extremities c_1 or c_2:
   (a) The candidate edge is formed by p_t and the vertex nearest to p_t.
   (b) It is determined whether points exist within the triangle formed by c_1, c_2 and p_t.
   (c) Should any point lie within $c_1c_2p_t$, the point p_t is eliminated from the analysis and the candidate edge is not accepted. The next point meeting the condition set in item 4 is then identified
   (d) Should $c_1c_2p_t$ not contain any other point in its interior, p_t will form part of the solution polygon, defining an edge with the vertex to which it is nearest, c_1 or c_2.
5. A new circle C is drawn, with diameter equal to the edge formed by p_t and the anterior vertex which it is not nearest.
6. Steps 4 and 5 are repeated until no point lies within the circle, which means that the edge that defined the circle, in the context in question, can not be further reduced.

Figure 3 illustrates the working of the contour-generating algorithm applied to the edge c_1c_2 of the convex hull of the set of points in Figure 2.

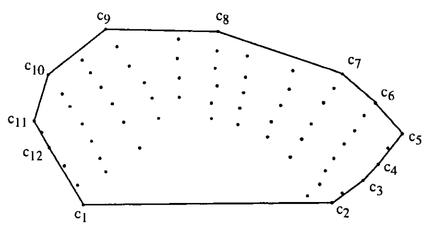

Figure 2. The convex hull of a finite set of n points.

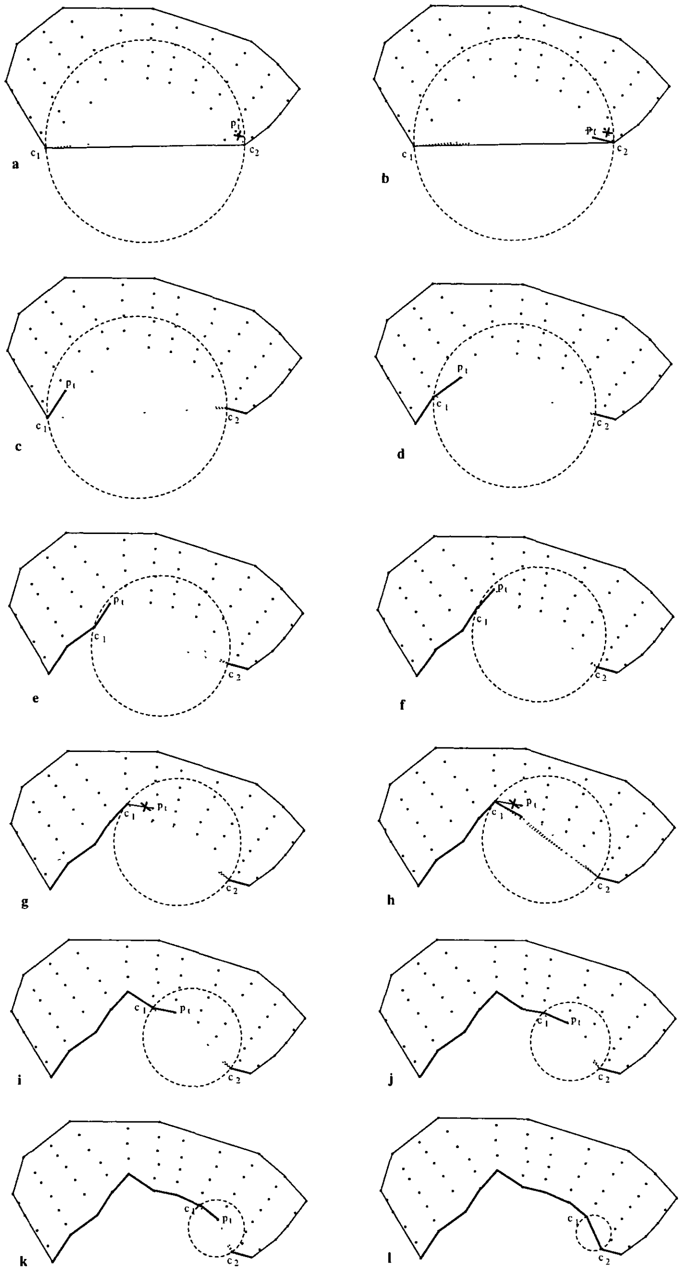

Figure 3: Contour generating for the edge c_1c_2 of the convex hull.

4 PRACTICAL CONSIDERATIONS

The described algorithm was coded as a module in a system designed to generate topographic maps in engineering projects related to roads With this, it was possible to do tests with real data and a more realistic model. It was used in the generation of topographic maps in the project of duplication of the Brazilian interstate road BR-381 between the states of Sao Paulo and Minas Gerais Figure 4 shows the contour lines generated using the described method with contour determination for a topographic map from this project

The system was implemented in C and the compiler used was the Gnu C. It can work in DOS and Unix The algorithm to implement the Delaunay triangulation was based on the divide-and-conquer approach (Lee and Schachter, 1980), using the winged-edge data structure (Baumgart, 1975)

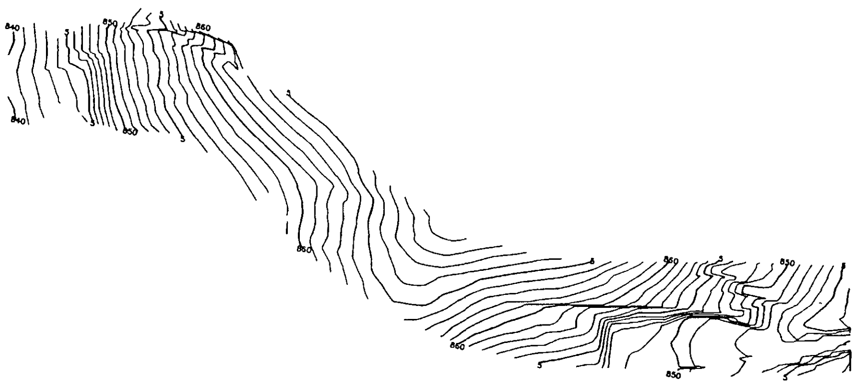

Figure 4. An example of contour determination.

5 CONCLUDING REMARKS

Many applications in Geographic Information Systems and other areas, as computer graphics and robotics, require a polygonal form closer to the region of points being dealt with, that is, they require that a non-convex contour of the points be generated In order to meet the needs of such applications and minimize imprecision in representation of the set of points, this algorithm can be used In most cases, the described method proved to be appropriate, finding a smaller polygon than the convex one. It also presented a good tradeoff between the quality of the solution and time, being able to solve complex instances on a PC compatible microcomputer (Avelar, 1994).

As the direction in which the polygon is processed affects the shape of the final contour, it can be considered that there is an alternative solution which can be compared with the first one, chosing the more convenient contour to the worked region.

This solution is not, in general, the optimal one, because of the very strong convexity hypothesis. The optimality criteria depends on each particular application Of course, if the polygonal region is convex, the convex hull is the right answer. But, in general, this is not true, as illustrated in Figure 3. The new shape has vertices and edges that do not belong necessarily to the original polygon

BIBLIOGRAPHY REFERENCES

Avelar, S (1994) *Representation of Relief using Geometric Algorithms* (In Portuguese) Master's Thesis, Federal University of Minas Gerais, Dept. of Computer Science.

Baumgart, B.G. (1975). A Polyhedron Representation for Computer Vision. *AFIPS Proceedings*, Vol.44, pp 589-596.

Buys, J., H.J. Messerschmidt and J.F. Botha (1991) Including known discontinuities directly into a triangular irregular mesh for automatic contouring purposes. *Computer & Geosciences*, 17(7). 885-881.

De Floriani, L (1987) Surface representations based on triangular grids. *The Visual Computer*, 3(1): 27-50.

Falcidieno, B and M Spagnuolo (1991). A new method for the characterization of topographic surfaces. *Int. J. Geographical Information Systems*, 5(4): 397-412.

Lee, D.T. and B. Schachter (1980). Two algorithms for constructing Delaunay triangulations. *Journal of Computer and Information Science*, 9(3): 219-242.

Preparata, F.P and M.I. Shamos (1985). *Computational Geometry - An Introduction*, Springer-Verlag, New York

Sakude, M T.S. (1992). Terrain Modeling with Bezier triangular surfaces (In Portuguese) In *Sibgrapi V*, pp. 213-222.

Shermer, T C. (1992). Recent Results in Art Galleries, *Proceedings of the IEEE*, Special Issue on Computational Geometry, 80(9). 1384-1397

Van Kreveld, M. (1996) Digital Elevation Models overview and selected TIN algorithms. Course Notes, CISM Advanced School on Algorithmic Foundations of Geographic Information Systems, Udine