

TRIANGULATION FOR SURFACE MODELLING

D. Shojaee¹, H. Helali² AA. Alesheikh³

Dept. of Geomatics Eng.

Islamic Azad University, Maybod Branch, Yazd, Iran

¹MSc Student, dshojaee@gmail.com

²PhD Student, helali@alborz.kntu.ac.ir

³Assistant Professor, alesheikh@kntu.ac.ir

Abstract—Surface reconstruction by means of triangulation of point data leads to computational complex optimization problems. Delaunay triangulation is a common method for domain triangulation. There are other algorithms for triangulating a point set in two or three dimensions, but all of them are not suitable for surface modelling. In this paper all methods of triangulations are evaluated and their applications in visualization are discussed. Data dependent triangulation can optimise this problem. This type of triangulation uses the elevation of points and with these information can model the surface better than other triangulation methods. It represents that data dependent triangulations can improve significantly the quality of approximation.

Keywords-component; Delaunay, Triangulation Algorithms, TIN, Data dependent.

I. INTRODUCTION

Delaunay triangulation is common method for surface representation. Triangulation produces a continuous surface. Important problem in triangulation algorithm is dependency dispersion of points. In Delaunay triangulation, results are repeatable and predictable, but other methods depend on the starting point.

In Delaunay triangulation circum-circle of any triangle should not contain another point. These unique triangles are independent from starting points or any other parameters. Logically for height interpolation nearest point must be used, therefore Delaunay triangulation try to connect nearest points. It is shown that in data dependent triangulations, triangles are long and thin, which are traditionally avoided [5]. Some data dependent criteria for optimizing a triangulation are presented and compared to the Delaunay criterion.

Most important applications of triangulation are DTM generation, feature surface modelling, computer graphic, scientific visualization, robotic, computer vision, image synthesis, mathematic and natural science.

II. NON DELAUNAY ALGORITHMS

There are a wide variety of algorithms available to build a triangulation for a set of points. Some of these algorithms are investigated here.

A. Greedy Triangulation

This algorithm is suitable for triangulation of simple polygons. The objective is to minimize the total edge length in the triangulation. This is achieved by an iterative process that selects the shortest available internal diagonal at each stage. Each of these edges must be tested for intersection with the other edges in the triangulation. In the worst case, all $O(n^2)$ diagonals will be considered. This algorithm needs a big array of distance and did not show satisfactory results [4].

B. Garey and et.al. Triangulation

Garey method (1978) is suitable for the triangulation of simple polygons. With respect to time complexity, this algorithm is more efficient than greedy method. The algorithm proceeds by the first decomposing the polygon into monotone polygons and then triangulating each of the monotone polygons. Let us assume that monotonicity is with respect to the y-axis and that no two different vertices of the polygon have the same y-coordinate. Stage 1 of the algorithm decomposes the polygon into monotone polygons. This stage is essentially the regularization of the polygon. Extra edges are added between vertices of the polygon so that :

- Edges only intersect at polygon vertices.
- Apart from the highest vertex, every vertex is joined by a single edge to a higher vertex.
- Apart from the lowest vertex, every vertex is joined by a single edge to a lower vertex.

The triangulation procedure iterates over the vertices of the monotone polygon and runs in linear time [4].

C. Radial Sweep

This method was invented by Mirante and Weingarten in 1982. In this algorithm, central point in set of given points is connected to other points radially (figure 1-a). In the next step triangles are formed as radial edges (figure 1-b). Then convex hull of points is constructed (figure 1-c). Finally points from their angles are arranged.

Some of these triangles may have not good connections. For optimization of these triangles, their neighbouring triangles that have a common edge are found. Then a

tetrahedral is formed from each two triangles its diagonals are computed. Finally large diagonal is replaced with the short one. This procedure is repeated until no edge changes [1].

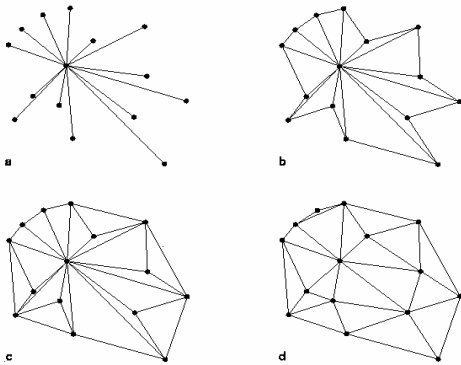


Figure 1. Radial sweep triangulation [1]

III. DELAUNAY TRIANGULATION

Delaunay triangulation is discussed by Delaunay in 1934. Delaunay triangulation maximizes the minimum angles in triangles and avoids skinny triangles.

A Delaunay triangulation T of P is a triangulation of P such that the circum-circle of any triangle belonging to T does not contain points of P in its interior. The Delaunay triangulation of a set P of points is unique provided that no four or more points of P are co-circular.

A. Properties of Delaunay triangulation

1- Local empty-circle property:

The circum-circle of any triangle in Delaunay triangulation does not contain the vertex of the other triangle in its interior.

2- max-min angle property:

In Figure 2, PIPk satisfies the max-min angle property if PIPk is the diagonal of tetrahedral which maximizes the minimum of the six internal angles associated with each of the two possible triangulations of tetrahedral [1].

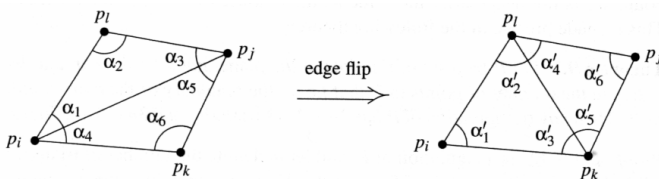


Figure 2. Angle criterion in Delaunay Triangulation [1]

3- Uniqueness:

There is a unique Delaunay triangulation from a set of points.

4- Boundary property:

External edges of Delaunay triangulation make the convex hull of the point set.

B. Delaunay Triangulation Algorithms

There are a wide variety of algorithms available to build a Delaunay triangulation for a set of points. Some of these algorithms have been presented.

1) Incremental Algorithms

The method of point insertion is very well known. It has been used, among others, by Lawson and Sloan in order to obtain an incremental algorithm that calculates the Delaunay triangulation of a set of points. This method is based on the following:

Let T be a Delaunay triangulation (DT) of a given set of points, S . Therefore, the insertion of one point $p \notin S$ inside the triangulated region, forming the new DT $S \cup \{p\}$, only modifies the triangles of T whose circum-circle contains the point p [1].

The derived algorithm begins by generating a triangle that contains all the points from which the DT can be obtained in order to guarantee that all the points will lie within the triangulation. The points are inserted within the triangulation one at a time; each inserted point implies making a series of changes in edges shared by two triangles, until all the triangles that contain the inserted point have been updated. Lawson demonstrated that this iterative process converges, after a finite number of steps, towards the new DT.

The bottleneck of incremental DT algorithm is the search for a triangle, into which the currently integrating point falls. Zalik and Kolingerova have transformed the problem into finding the closest point, which takes less processing time.

Three main steps of the algorithm are:

- Initialization
- Triangulation
- Finalization

To make the algorithm work fast, the searching structure has to be initialized.

The algorithm proceeds as follows:

- Add a point to the triangulation
- Find all existing triangles whose circum-circle contains the new point (Figure 3).

First the triangle which contains the new point has to be found, which is a proximity search taking time $O(\log n)$ for a suitable data structure (e.g., quadtree/octree). Then the neighbours of this triangle are searched and then their neighbours, etc., until no more neighbours have the new point in their circum-circle. This takes, on average, time $O(1)$ but in the worst case the circum-circles of all the existing triangles contain the new point, taking time $O(n)$.

- Delete these triangles, which create (always) a convex cavity.

- Join the new point to all the vertices on the boundary of the cavity (Figure 3)

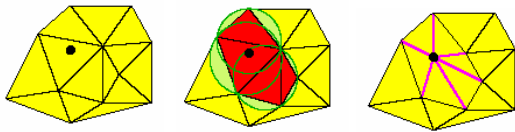


Figure 3. Building an Incremental Delaunay Triangulation

2) Step by step algorithm

The external edges in Delaunay triangulation form Convex hull and therefore we can start delaunization from these edges [4].

- The smallest edge in Convex hull is selected as Base line,
- Third point that can form a Delaunay triangle is found.
- The triangle is checked with a circle that passes from these three points. If there are more than three points in circle, the size of circle should be changed.
- Near points to bisector of base line are selected .

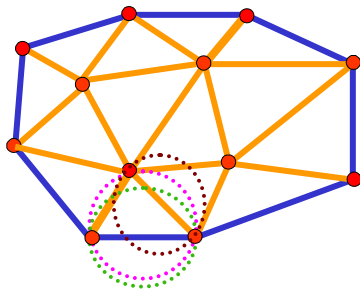


Figure 4. Step by Step algorithm

3) Flipping algorithm

This algorithm has two steps:

- 1- Construction of an arbitrary triangulation,
- 2- Optimization of the made triangulation to produce a Delaunay triangulation.

In the first step we build a triangulation with incremental approach and in second step we swap every edge with other diagonal of the tetrahedral that is not locally optimal. We iterate this process until not any other edge swap is needed [2].

4) Plane Sweep Algorithm

This algorithm is a gradual algorithm for Delaunay triangulation. In this method we sweep all points with a line. The points behind of the line are triangulated. As the line sweeps other points new triangles are built. This algorithm was introduced and developed for Voronoi diagram by Fortune in 1987 [3].

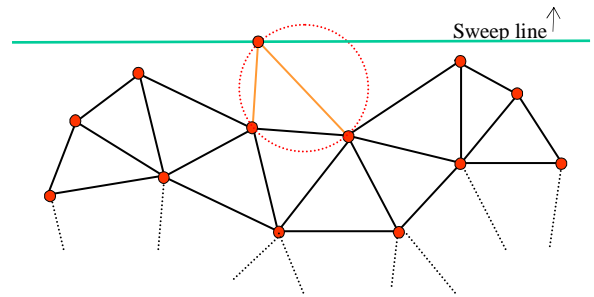


Figure 5. plane sweep Triangulation

5) Divide and Conquer Algorithm

The divide-and-conquer algorithm subdivides the area into two partial areas, computes recursively the Delaunay triangulation of the partial areas and merges finally both triangulations. The difficult section in this algorithm is to merge the parts. Technique to increase the speed of this algorithm should be used. Time complexity of this algorithm in the worst case is $O(n \log n)$ [3].

6) Dewal Algorithm

This algorithm presented by Cignoniet in 1998. In this algorithm a vertical line divide the points to right and left sections. Then, triangulation starts from the bottom of this line. Time complexity for this method is $O(n \log n)$. Figure 6 display this method.

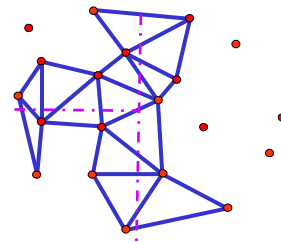


Figure 6. Dewal Triangulation

IV. DATA DEPENDENT TRIANGULATION

Notice that all the above mentioned optimal triangulations for use in interpolating $z = f(x,y)$ do not take the data values into account. This may not be desirable since some shape information "encoded" in the data values has great influence on the quality of the interpolation. In general, a data dependent criteria is one that takes into account more than just the locations of the vertices [7].

There are many methods of data-dependent triangulation for reconstructing a surface from irregularly spaced data. Various numerical experiments demonstrate the improvement in the quality of the approximation when certain data dependent triangulations are used [6].

This method introduced by Dyn, Levin, and Rippa and has proven capable of producing more pleasing reconstructions than axis aligned methods [8].

Given a set of distinct (and not all collinear) points $V = \{(x_i, y_i)\}$ in the x - y plane. A convex hull triangulation of

V that's a set $T=\{T_i\}$ of non degenerate triangles that satisfies the following conditions:

- Every triangle vertex is an element of V, and every element of V is a triangle vertex.
- Every edge of a triangle in T contains exactly two points from V.
- The union of all triangles in T is the convex hull of V.
- The intersection of any two different triangles in T is either empty, or is a shared edge or a shared vertex.

A convex hull triangulation proves the simplest way to produce a piecewise linear continuous surface that Interpolates a set of data points. If each point (x_i, y_i) also has a z coordinate z_i , then any surface $z = f(x, y)$ for which $f(x_i, y_i)=z_i$ is called an interpolating surface of $f(x, y)$. Thus, any triangulation of V whose vertices move from $(x_i, y_i, 0)$ to (x_i, y_i, z_i) is a piecewise linear interpolating surface [8].

Dyn et al. proposed numerous cost functions for data dependent triangulation. The four NC1 cost functions from Dyn et al. are as follows:

1- Angle Between Normals (ABN):

This cost function is the cosine of the 3D angle between normals. Figure 7 shows the normal vector for each triangle and angle between normals.

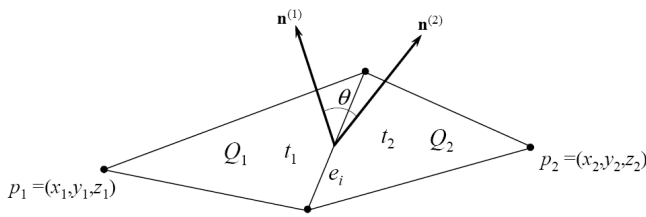


Figure 7. Angle between normal

$$\text{Cos } \theta = \frac{|n_1 \cdot n_2|}{|n_1| |n_2|} \quad (4.1)$$

2- Jump in Normal Derivatives (JND):

This is given by

$$C_{JND}(\Delta, e_i) = \left| (a_1 - a_2)n_x - (b_1 - b_2)n_y \right| \quad (4.2)$$

where (n_x, n_y) is a unit vector in the (x, y) plane orthogonal to the projection of edge e_i in the (x, y) plane.

3- Deviations from Linear Polynomials (DLP):

This expresses how well Q1 predicts the value of p2 and similarly, how well Q2 predicts the value of p1. DLP is given by C where

$$c_{DLP}(\Delta, e_i) = \left(|Q_1(x_2, y_2) - z_2|^p + |Q_2(x_1, y_1) - z_1|^p \right)^{\frac{1}{p}} \quad (4.3)$$

4- Distances from planes (DFP):

This measures the distance from p1 to the plane of Q2 and from P2 to the plane of Q1. This distance is given by C where

$$c_{DFP}(\Delta, e_i) = \left\| \begin{pmatrix} \text{dist}(Q_1, p_2) \\ \text{dist}(Q_2, p_1) \end{pmatrix} \right\| \quad (4.4)$$

$$\text{dist}(Q, p) = |Q(x_i, y_i) - z_i| / (a^2 + b^2 + 1)^{\frac{1}{2}} \quad (4.5)$$

Figure 8 shows a sample Delaunay and ABN triangulation.

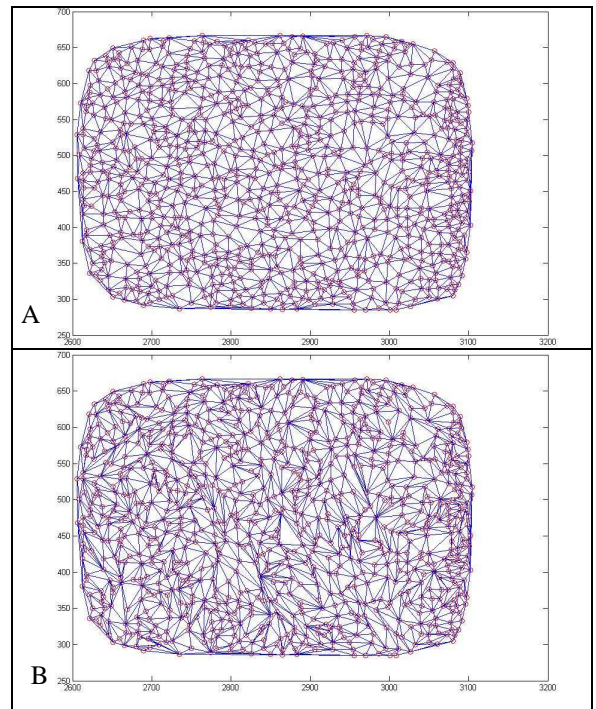


Figure 8. A: Delaunay Triangulation, B: ABN Triangulation

V. 5 CONCLUSIONS

In this article different algorithms for triangulation were investigated and compared together from the view points of accuracy. It was showed that Delaunay triangulation is the most practical algorithm as it provides unique triangles in different performances. Furthermore, Non-Delaunay triangulation algorithms are weak in surface modelling, and they are affected by starting point and density of points.

These procedures are not repeatable and predictable. Therefore their use is not recommended.

Delaunay triangulation algorithms are accurate and reliable. The only shortcoming of these algorithms is that they do not take into account the height of points so their robustness is not certain. In that respect an improved Delaunay triangulation that make use of points height in surface modelling is recommended data dependent triangulation.

Data dependent triangulation can model any surface better than Delaunay triangulation. Optimally triangulated surfaces are the basis for many practical applications. ABN method tested for three types of terrain(smooth and with various topography). Results shows that this method create a better triangulation and contour line with respect to Delaunay triangulation.

REFERENCES

- [1] A. Bowyer, (1981), Computing Dirichlet Tesselations, *Comp. J.*, 24 ,162.
- [2] Antoine Vigneron, (2004), Computing a Delaunay Triangulation, National of University of Singapore.
- [3] Jorge C Kramer, (1995), Delaunay triangulation in two and three Dimension. Wilhelm Schickard Institut, Graphische Interaktive Systeme,Fakultat fur Informatik, Eberhard Karls Universit at Tuingen
- [4] Michael F. Worboys, (1997), *GIS, A Computing Perspective*. Taylor & Francis, London.
- [5] Nira Dyn, David Levin, Samuel Rippa, (1990), Data Dependent Triangulations for Piecewise Linear Interpolation, School of Mathematical, Science, Sackler Faculty of Exact Sciences, Tel-Aviv University, Ramat Aviv, Israel.
- [6] Shmuel Rippa, (September 1992), Adaptive Approximation By Piecewise Linear Polynomials On Triangulations Of Subset Of Scattered Data, *SIAM J. SCI. STAT. COMPUT.* Vol. 13, No. 5, pp. 1123-1141.
- [7] Tiow Seng Tan, (1993), *Optimal Two Dimensional Triangulations*, University of Illinois at Urbana Champaign.
- [8] Xiaohua Yu, Bryan S. Morse, and Thomas W. (2001), Sederberg, Image Reconstruction Using Data-Dependent Triangulation, Brigham Young University.