THE METHODS OF TRIANGULATION

M. Varshosaz, Assistant Professor,
Faculty of Geodesy & Geomatics Eng., K.N. Toosi University of Technology
K.N. Toosi University of Technology, Vali_Asr St, Tehran , Iran, 19697
Email: varshosazm@kntu.ac.ir

H. Helali, PhD Student,
Faculty of Geodesy & Geomatics Eng., K.N. Toosi University of Technology
Email: helali@alborz.kntu.ac.ir

D. Shojaee, MSc Student,
Faculty of Geodesy & Geomatics Eng., K.N. Toosi University of Technology
Email: dshojaee@gmail.com

Abstract

A common method for reconstruction of a geometric figure for a given point set is the use of a triangulation algorithm to connect the points. Triangulation a set of sample points is an important method with many applications. There are some algorithms for triangulating a point set in two or three dimensions, but just few of them can solve the problem of optimizing the shape of the triangles e.g. Delaunay. Delaunay is used to reconstruct 3D features, more complicated than 2D ones. The potential of TIN-based methods has been partly exploited in two dimensions but hardly works in three dimensions. In this paper we discussed Delaunay and non-Delaunay algorithms.

Keywords: Delaunay, Triangulation, TIN, Algorithms

1- Introduction

Triangulation is a method for tessellation of domain. In fact triangulation is a common method for surface representation and for building a TIN. Triangulation produces a continuous surface. Important problems in triangulation algorithm are independency from starting point or dispersion of them. Further result must be repeatable and predictable. Therefore the triangulation is made, the aim is to maximize the minimum angles and establish circle condition.

In Delaunay Trinangulation circum-circle of any triangle should not contain another point[4]. These unique triangles are independent from starting points or any other parameters. Logically for height interpolation we use nearst point, therefore in triangulation, we connect near points.

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

2- 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.

2-1- 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[5].

2-2- Triangulation of Garey et.al

The algorithm of Garey et.al. (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 :

1. Edges only intersect at polygon vertices.
2. Apart from the highest vertex, evry vertex is joined by a single edge to a higher vertex.
3. Apart from the lowest vertex, evry 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[5].

2-3- 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 (fiqure 2-1-a). In the next step triangles are formed as radial edges are connected together (fiqure 2-1-b). Then convex hull of points is constructed (fiqure 2-1-c). Finally points from their angles are arranged.

Some of these triangles may have not good connections. For optimization of these triangles, their neighboring 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 [8].

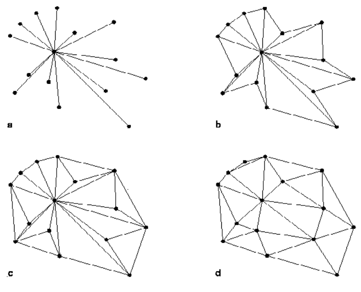


Figure 2-1 : Radial sweep triangulation[8]

3- Delaunay Triangulation Algorithms

There are a wide variety of algorithms available to build a Delaunay triangulation for a set of points. Delaunay triangulation is discussed by B.Delaunay in 1934. Delaunay triangulation maximizes the minimum angles in triangles and avoids skinny triangles.

3-1-Delaunay Triangulation

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 uniqe provided that no four or more points of P are co-circular.

3-2-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[1].

2- max-min angle property:

In Figure 3-1 P_lP_k satisfies the max-min angle property if P_lP_k is the diagonal of tetrahedral which miximizes the minimum of the six internal angles associated whit each of the two possible triangulations of tetrahedra[11].

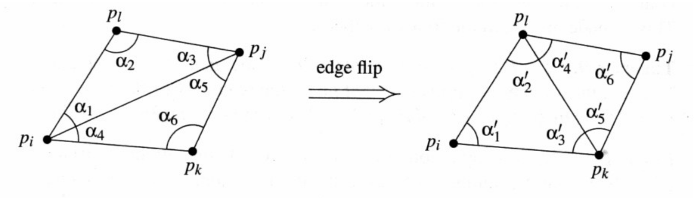


Figure 3-1 : Angle criterion in Delaunay Triangulation[11]

3- Uniquness:

There is a unique Delaunay triangulation from a set of points[9].

4- Boundary property:

External edges of Delaunay triangulation make the Convex hull of the point set[2].

3-3-Delaunay triangulation algorithms

3-3-1-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 proposition:

Proposition 1. Let T be a 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 circumcircle contains the point p [3].

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 Delaunay triangulation [6].

The bottleneck of incremental DT algorithm is the search for a triangle, into which the currently integrating point falls. Zalik and Kolingerova [6] 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[12].

The algorithm proceeds as follows :

- Add a point to the triangulation
- Find all existing triangles whose circumcircle contains the new point (Figure 3-2). 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 circumcircle. This takes, on average, time $O(1)$ but in the worst case the circumcircles of all the existing triangles contain the new point, taking time $O(n)$.
- Delete these triangles, which creates (always) a convex cavity.
- Join the new point to all the vertices on the boundary of the cavity (Figure 3-2)

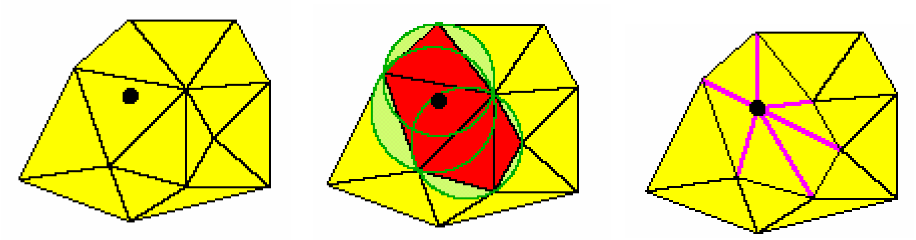


Figure 3-2 : Building an Incremental Delaunay Triangulation[13]

3-3-2- Step by step algorithm

The external esdges in Delaunay triangulation form Convex hull and therefore we can start delaunization from these edges[10].

- The smallest edge in Convex hull is selected as Base line,
- Third point that can form a Delaunay triangle is found.
- The traingle 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 .

3-3-3-Filliping 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[7].

3-3-4- 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 [13].

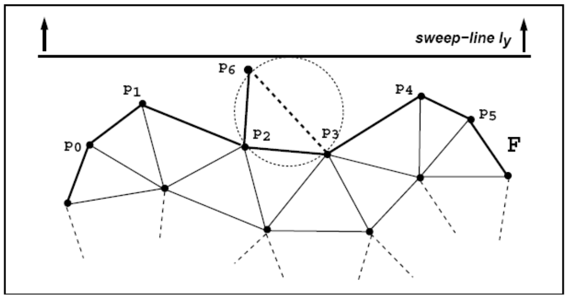


Figure 3-3 : plane sweep Triangulation[13]

3-3-5- 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(nlogn)[13].

4-Evaluation of these Algorithms

There are some criteria for comparison and investigation of these algoritms. These criteria are:

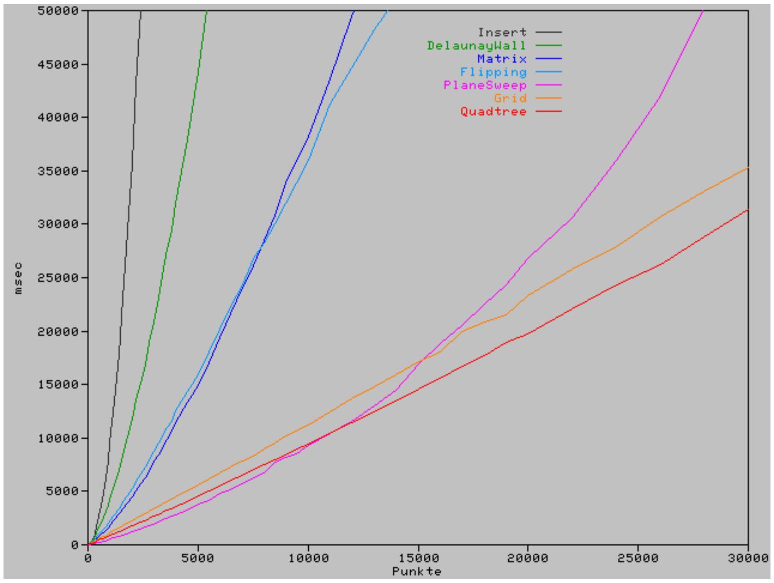
1- Rapidity: Rapidity criterion is one of the most important criterion for triangulation algorithm because usually large amounts of data are available to be triangulated and therfore time efficiency is crucial.
2- Accuracy and precision: As we need a reliable surface, triangulation algorithms should be accurate and precise.
3- Stablity: we mean by stability the uniquness of results in different performances. Because multiple performances are essential in TIN editting this is an important property of the algorithm.

A comparison is made between triangulation algorithms with respect to time complexity by Kramer in 1995[13]. These algorithms are:

- Insert: randomized insertion algorithm
- Dewall: Delaunay wall algorithm without searching structre

- Matrix: with sparse matrix
- Grid: with regular grid
- Flipping: flipping algorithm plus computation of the start triangulation
- PlaneSweep: planesweep algorithm
- Quadtree: randomized insertion algorithm using a quadtree for locating the enclosing triangle

In Graph 3-1 , horizontal line represents the number of points and vertical line shows the time.



Graph 3-1 : A comparison between triangulation algorithms[13]

As the graph shows, palne sweep algorithm is the most efficient when the number of points are less than 11000 but for more points incerementaI algorithm performs better. It shows that data structure play an important role in time efficiency of the algorithms[13].

5-Conclusions and Recommendations

In this article different algorithms for triangulation were investigated and compared together from the view points of rapidity, robustness, reliability and accuracy. It was showed that Delaunay triangulation is the most practical algorithm as it provides unique trianles in different performances.Furthermore, it was showed that data structure is the most important factor that influences the time efficiency of triangulation algorithms.

Non-Delaunay triangulation algorithms are weak in surface modelling, and they are affected by starting point and density of points. These procedures are not repaetable and pridictable. Therefore their use is not recommanded.

Delaunay triangulation algorithms build a uniqe triangulation from a set of points. They are accurate and reliable. The only shortcomming 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 for further study. This is what we intend to implement.

6-References

1-R.Lattuada and J.Raper, Application of 3D Delaunay algorithms in geoscientific modelling.

2-S.R. Idelsohn, E.Oñate, N.Calvo, F.Del Pin, Meshless Finite Element Ideas.

3-Stephan Wise, GIS BASIC, First Published 2002

4-Christopher B.Jones, Geographical Information Systems and Computer Cartography, 1998

5-Michael F. Worboys, GIS, A Computing Perspective, 1997

6-Marc Vigo Anglada, An improved incremental algorithm for constructing restricted Delaunay triangulations

7-Antoine Vigneron, Computing a Delaunay Triangulation, National of University of Singapore, 2004

8-George Kontopoulos, Triangulation in 2D space–2D quality meshing algorithm by Ruppert .

9-Leila De Floriani, Delaunay Triangulation, 2003

10- Mantena V. Raj u, A Parallel algorithm for Delaunay Triangulation, 2001

11-Glenn Eguchi, Delaunay Triangulations, Computational Geometry, 2001

12-Marc Vigo Anglada, An improved incremental algorithm for constructing restricted Delaunay triangulations.

13-Jorge C Kramer, Delaunay triangulation in two and three Dimension, 1995