

A Fast Merging and Segment Insertion of Delaunay Triangulation

Hu Rui and Zhang Qun

*Donlinks School of Economics and Management, University of Science and
Technology Beijing
Hurui870929@hotmail.com*

Abstract

Digital Terrain Model is the most effective way to solve the terrain visualization problem. TIN is commonly used in DTM and the most preferable method in TIN is Delaunay Triangulation. In traditional Delaunay Triangulation, there are divide and conquer algorithm, incremental insertion algorithm and Growth triangulation algorithm. Our study is based on divide and conquer algorithm, and a fast merging and segment insertion strategy were applied. The simulation on test set indicated that this improved method has a 7% average time saving in generating a Delaunay triangulation than traditional way, which means it's an effective improvement and will result to efficiency on DTM problems.

Keywords: *Delaunay triangulation; DTM; divide and conquer algorithm; fast merging*

1. Introduction

Terrain Visualization is designed for three-dimensional display for real-world terrain. It takes the digital terrain model (DTM) as the basic framework, and builds more realistic reproduction of actual terrain scenes by color filling, shading, texture mapping, rendering and visibility judgment on each fundamental side of the frame model.

DTM was proposed by MIT professor Chaires L Milier in 1956, when he intended to design an automatic highway. DTM can be expressed by the following formula:

$$A_i = F(x_i, y_i) \quad i = 1, \dots, n \quad (1)$$

where A_i represents the value of specific information for any plane surface, mainly basic geomorphologic information such as elevation, aspect, slope, topography and other factors. Depending on the value of A_i , its name varies. For example, when A_i refers to soil distribution, the model is called the DSM; when A_i represents elevation, the model is named DEM.

DTM has multiple representations, including regular grid and irregular grid. Because the triangle is the simplest planar geometry, it is often used in an irregular grid, and called Triangulated Irregular Network (TIN) [1].

In all possible Triangulation, Delaunay triangulation has the best performance in terms of terrain fit. Therefore, it is often used to generate irregular triangular network. When there are pre-defined constraints generated in TIN such as ridge line with no intersection of the surface, the valley lines and break lines, we must consider with Constraint Delaunay Triangulation (CDT) [2].

Currently, there are three major methods to generate a Delaunay triangulation:

(1) Growth triangulation algorithm:

Growth algorithm was originally developed by Green and Sibson [3], then Brassel, McCaullagh, Maus and so???? has made improvements [4-5]. There are also a lot of scholars in China who has studied this algorithm. He *et al.* [6] proposed the concept of a closed point to improve triangulation growth algorithm. Zhai [7] brought up a block-by-block merge algorithm, which divided the process into three steps: firstly, dividing the

discrete point set into several sub-blocks, then taking each sub-block expansion with triangulation method network configuration, and then merge the neighboring sub optimization. The algorithm is not efficient when the number of sub-blocks and the midpoints are relatively large. Wu and Xu [8] proposed an algorithm called "arcs searching exclusion" to find the third point. It narrowed the search scope by zoning and reduced the searching scale by ordering the discrete points and source points. Thereby, the generation efficiency of Delaunay Triangulation has been increased.

(2) Divide and conquer algorithm:

This algorithm was proposed by Lewis and Robinson [9], and later, Andre, *et al.* carried out improvement [10]. After that, scholars tried to solve the problem in the construction of TIN / Delaunay triangulation with large amounts of data by grid-based block and quadtree algorithm [11]. This improvement made the overall time complexity of the algorithm decline, but the algorithm becomes more complicated.

(3) Incremental insertion:

Incremental insertion was presented by Lawson [12], which was followed by a number of scholars who made various modifications: Su, Wang, Lv, Wu and Li [13] used adaptive Hilbert curve to improve the efficiency of the algorithm when dealing with random data. Yang et al used the Qi algorithm to generate Delaunay triangulation and reduce the complexity of the algorithm [14]. Guo and Zhang [15] combined the divide and conquer and incremental insertion to the construct the triangulation, but it's not efficient when building a sub-block.

Table 1 summarized the time complexity of three algorithms. As can be seen, the time cost in triangulation growth is high, so it rarely used at present. Divide and conquer algorithm has the highest efficiency, but it is also huge space consuming. Besides, its ordering is based on x coordinate, which could easily lead to elongated triangle. Incremental insertion occupies less memory, but with the increasing number of the insertion point, positioning the triangle and determining time increases. Its configuration of network is not as fast as divide and conquer algorithm.

Table 1. Comparison of Time Complexity of Three Delaunay Construction Algorithm

Class	Algorithm brought by	Average	Worst
2	Lawson 1977	$O(N^{4/3})$	$O(N^2)$
3	Green and Sibson 1978	$O(N^{3/2})$	$O(N^2)$
1	Lewis and Robinson 1978	$O(N \log N)$	$O(N^2)$
3	Brassel and Beif 1979	$O(N^{3/2})$	$O(N^2)$
3	MaCullagh and Ross 1980	$O(N^{3/2})$	$O(N^2)$
1	Lee and Schachlter 1980	$O(N \log N)$	$O(N \log N)$
2	Lee and Schachlter 1980	$O(N^{3/2})$	$O(N^2)$
2	Bowyer 1981	$O(N^{3/2})$	$O(N^2)$
2	Watson 1981	$O(N^{3/2})$	$O(N^2)$
3	Mirante and Weigarten 1982	$O(N^{3/2})$	$O(N^2)$
2	Sloan 1987	$O(N^{5/4})$	$O(N^2)$
1	Dwyer 1987	$O(N \log N)$	$O(N \log N)$
1	Chew 1989	$O(N \log N)$	$O(N \log N)$
2	Macedonio and Pareschi 1991	$O(N^{3/2})$	$O(N^2)$

* 1- divide and conquer; 2- Incremental insertion; 3- Growth triangulation algorithm

2. Construction of Delaunay Triangulation

Delaunay triangulation doesn't exist alone. In the region where Delaunay triangulation was generated, a complementary Voronoi (Thiessen polygons) can be identified. In Voronoi graph, perpendicular bisectors of each edge of the triangle intersect at one point which is the center of circumcircle of this Delaunay triangulation. Each triangle has three perpendicular bisectors, and connecting the perpendicular bisectors of all sides according to the principle of the nearest, we can find out that the area constituted Delaunay triangulation can also be split by its complementary diagram. The area can be seen as formed by the union of polygon composed by perpendicular bisectors, as shown in Figure 1:

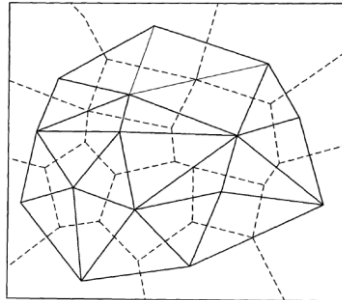


Figure 1. The Division of a Delaunay Triangle

Generate Delaunay triangulation must meet two important criteria:

First, blank circle criteria:

Delaunay triangulation is unique (any four points cannot be concyclic). There will not be a fourth point in circumcircle of a Delaunay Triangulation. As shown in Figure 2:

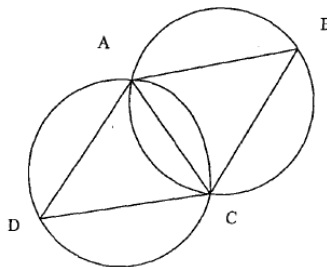


Figure 2. Blank Circle Criteria

As in Figure 2, point D is not included within $\triangle ABC$, likewise, B is not included within $\triangle ADC$. The formula we used to analyze if there's other point within the triangle circumcircle range is:

$$\begin{vmatrix} x^2 + y^2 & x & y & 1 \\ x_1^2 + y_1^2 & x_1 & y_1 & 1 \\ x_2^2 + y_2^2 & x_2 & y_2 & 1 \\ x_3^2 + y_3^2 & x_3 & y_3 & 1 \end{vmatrix} = 0 \quad (2)$$

Three vertices of the triangle are A (x_1, y_1), B (x_2, y_2), C (x_3, y_3), expand the determinant and we obtained:

$$\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} (x^2+y^2) - \begin{vmatrix} x_1^2+y_1^2 & y_1 & 1 \\ x_2^2+y_2^2 & y_2 & 1 \\ x_3^2+y_3^2 & y_3 & 1 \end{vmatrix} x + \begin{vmatrix} x_1^2+y_1^2 & x_1 & 1 \\ x_2^2+y_2^2 & x_2 & 1 \\ x_3^2+y_3^2 & x_3 & 1 \end{vmatrix} y - \begin{vmatrix} x_1^2+y_1^2 & x_1 & y_1 \\ x_2^2+y_2^2 & x_2 & y_2 \\ x_3^2+y_3^2 & x_3 & y_3 \end{vmatrix} = 0 \quad (3)$$

Then the positional relationship between E (x4, y4) and the circumscribed circle of triangle ABC can be expressed by the following formula:

$$\det = \begin{vmatrix} x^2 + y^2 & x & y & 1 \\ x_1^2 + y_1^2 & x_1 & y_1 & 1 \\ x_2^2 + y_2^2 & x_2 & y_2 & 1 \\ x_3^2 + y_3^2 & x_3 & y_3 & 1 \end{vmatrix} \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} \quad (4)$$

When $\det > 0$, E is within the circumcircle range; when $\det < 0$, E point is outside the circumcircle; and when $\det = 0$, E point is located on the circumcircle.

Second: maximize the minimum angle:

In Deluanay triangle, any two adjacent triangles form a convex quadrilateral, and the shared line of these two triangles is the diagonal of the convex quadrilateral. A total number of six interior angles exist. If after changing the diagonal, the smallest interior angle is larger than the initial one, then this new form is taken as the final Delaunay triangle. This criterion causes the triangles within the network closer to equilateral triangle. As shown in Figure 3:

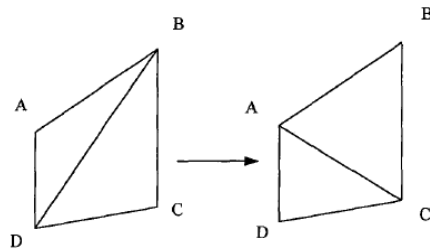


Figure 3. Maximizing the Minimum Angel

Theoretically, to construct a standard Delaunay triangulation, a regular triangulation can be ensured as a Delaunay triangulation after local optimization process (LOP). The basic approach of LOP is as follows:

- (1) The two sides of the triangle have a common synthesis of a polygon.
- (2) Guidelines for maximum empty circle checked to see whether it's the fourth vertex of the circumscribed circle of the triangle.
- (3) If the amendment diagonal reversed soon, complete the processing of local optimization process.

LOP process shown in Figure 4.

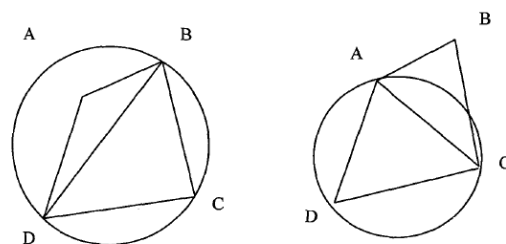


Figure 4. Exchanging the Diagonal Line

3. Improved Divide and Conquer Algorithm

3.1. Construction of the Convex Hull Triangulation

The first step of creating a Delaunay triangulation is to construct a convex hull triangulation.

In order to delineate the scope of all the discrete points and ensure the border of region can participate in the construction, a minimum convex polygon containing the triangulation has to be built firstly.

Traditional convex hull generation algorithm requires for the involvement of all discrete points throughout the region. With the large number of point set in a certain region, the calculation in constructing a convex hull is great. So an optimized convex hull generation algorithm based on incremental method has been proposed:

Step 1.1: sorting the points in ascending order in the observation region. The x coordinate is the primary key and y coordinate as the supplemented key. Adding the point sets with order marks into a list called Q.

Step 1.2 Identifying these eight points in the discrete set: the one with max x coordinate, the one with the min x coordinate, the one with the max y coordinate, the one with the min y coordinate, the one with the max x+y, the one with the min x+y, the one with the max x-y, and the one with min x-y. These eight points compose an initial convex hull. After this, adding these points with counterclockwise order into the convex border list which is L.

Step 1.3 As it can be seen in Figure 5, we name the point with minimum y coordinate P and draw a dividing line parallel with the Y axis at P. So the point set would be divided into two sub-sets: the ones with smaller x coordinate value than P and the ones with bigger x value than P. Accordingly, list M is established to contain the points after P in list Q.

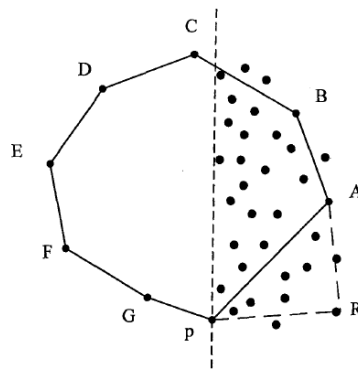


Figure 5. Generation of Convex Hull

Step 1.4 Locate the point that is farthest away from vector PA and we named it R. Then R will be insert into the list L between the point P and A. When finding R, all points that are on the right side of PA are considered to have a positive distance and the ones on the left side are with negative distances. All points with positive distance would be collected in list N.

Step 1.5 Similarly to 1.4, searching list N to locate the points farthest away from vector PR and PA and insert them between P, R and P, A.

Step 1.6 Repeating step 1.5 until all possible points in N were added to convex hull.

Step 1.7 Repeating steps 1.4 – 1.6 until all points that are capable of composing the convex hull on vectors AB, BC, CD, DE, EF, FG, and GP were found. And so far, the convex hull has finished building.

3.2. Initialization of Triangulation Net

The initial Delaunay triangulation was generated through the existing boundary of convex hull from outside to inside. Its general idea is:

Step 2.1 Selecting one random point in the convex hull list and locate the two points just before and after it according to the topology. Thusly, these three points can form a triangle.

Step 2.2 Testing the triangle with an empty circle detection criteria. If a fourth point were included, selecting the next point adjacent to the prior one.

Step 2.3 Repeating step 2.2 until a triangle that doesn't contain any other border point in convex hull and add this triangle into the triangulation list as the starter.

Step 2.4 Deleting the picked points in step 2.3 and update the convex hull list.

Step 2.5 Repeating steps 2.1 to 2.3 till the division of entire convex hull of triangulation is completed.

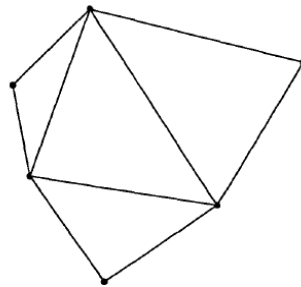


Figure 6. Triangle Section

3.3 Insertion of Discrete Points

The next step is step 3: to add the rest points within the boundaries of the convex hull into initial triangulation by incremental insertion method, as in Figure 7. Point 7, 8 and 9 were the added ones.

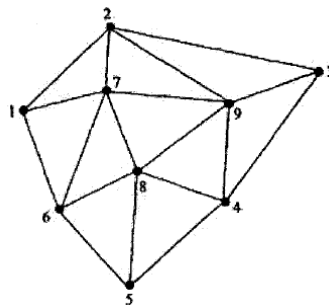


Figure 7. Delaunay Triangulation

3.4. Generation of Support Line

Step 4.1, finding support line. Firstly, defining a function *Position (Q1, Q2)* to determine the position relationship between the line and the vector. There are two arguments: the two endpoints of a line segment. The returning value is a BOOL variable. If the return value is 1, the point is on the left side of the vector segment; if the return value is -1, the point on the right side. As in Figure 8, line Q1Q2 is the bottom support line.

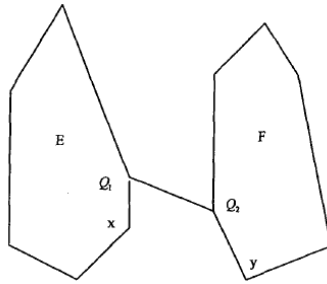


Figure 8. Searching for the Support Line

Step 4.2 In list E as shown in Figure 8, locating the next nearest point to Q_1 in clockwise direction, x . If the function Position gets -1 as feedback, replace Q_1 with x and xQ_2 is the new bottom support line.

Step 4.3 Repeat step 4.2 till the Position function achieve 1.

Step 4.4 repeat step 4.1-4.3 to find the top support line.

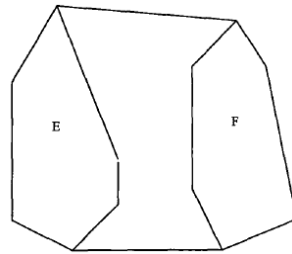


Figure 9. Convex Hulls with Two Supporting Lines

3.5. Merge Two Sub-triangulation

Step 5.1 Determining the consolidation area, which is the area between the top line and the bottom support line (Figure 9).

Step 5.2 the end points of the bottom support line is two vertices of a new triangle. Locating the third vertices as in Figure 10.

Step 5.3 Gradually find the other triangles in the region and add to the triangle list, until a new built triangle meet the top support line. The emerging process is completed there.

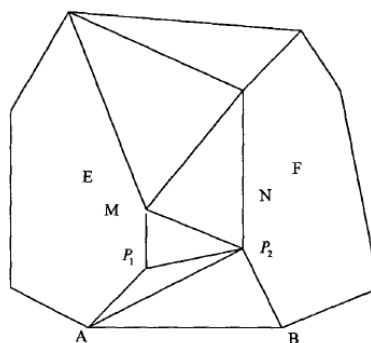


Figure 10. Merging of Two Sub-Triangulation

This merging algorithm differs from the traditional merging algorithm because that traditionally, when selecting a third point, the one smaller radius of circumradius is chose.

Thusly, the calculation of the circumscribed circle and triangle are large. Compared with it, this algorithm made the selection based on maximizing the minimum interior angle criteria. This method has a small amount of calculation.

4. Comparison of Fast Merging Algorithm and Traditional Algorithm

To verify the efficiency of this fast merging algorithm, we compared the time cost of this new algorithm and the traditional algorithm as in Table 2 below:

Table 2. Comparison of Fast Merging Algorithm and Traditional Algorithm

Time: second		Traditional algorithm	Fast merging algorithm
Vertices	10	1.52	1.67
	30	12.41	11.82
	100	19.76	15.38
	1000	183.29	161.05

The number of vertices was decided by table 1 in reference 16 [16].

As we can see from the table, this new algorithm has a better performance when the vertices are large in number. When there are 10 and 30 nodes in the area, these two algorithms are roughly the same on time consuming. About 7% of time saving exists in average, and if large amount numbers are considered only, this savings can reach 17%.

5. Conclusion

Delaunay triangulation is a typical method while building a DTM. As most graphical algorithms, Delaunay triangulation is also time-costing. This study proposed a fast merging and segment insertion algorithm. This algorithm is based on divide and conquer algorithm and made improvement on the merging of two sub-triangles. There's about 7% of average time advantage in this new algorithm. Its performance is better when the vertices scale is larger. When the vertices scale comes to a level above 10^2 , the time consumption will conserve about 17%. The improved algorithm has been verified in the lab, in the future, it has a broad application on DTM area.

References

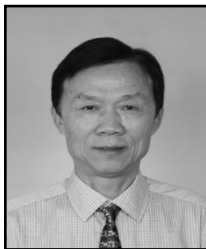
- [1] S. Ying, L. Li and M. C. Wang, "Applications of computational geometry in map generalization", *Science Of Surveying And Mapping*, vol. 3, no. 30, (2005).
- [2] C. L. Shao, P. Hu, C. Y. Huang and Q. Peng, "The expatiation of delaunay algorithms and a promising direction in application", *Science Of Surveying And Mapping*, vol. 6, no. 29, (2004).
- [3] P. J. Green and R. Sibson, "Computing Dirichlet tessellations in the plane", *The Computer Journal*, vol. 2, no. 21, (1978).
- [4] K. E. Brassel and D. Reif, "A procedure to generate Thiessen polygons", *Geophysical Analysis*, vol. 11, no. 2, (1979).
- [5] M. J. McCullagh and C. G. Ross, "Delaunay triangulation of a random data set for isarithmic mapping", *The Cartographic Journal*, vol. 2, no. 17, (1980).
- [6] Q. B. He, L. G. You and J. Wen, "Improved Algorithm for Delaunay triangulation", *Computer and Digital Engineering*, vol. 5, no. 34, (2006).
- [7] Z. L. Zhai and M. Y. Pang, "Integrating particle swarm algorithm and artificial fish swarm algorithm to optimize BP algorithm", *International Journal of Control and Automation*, vol. 7, no. 8, (2015).
- [8] J. Q. Wu and A. L. Xu, "An improved method of Delaunay triangulation growth", *Surveying and Mapping*, vol. 3, no. 37, (2012).
- [9] B. A. Lewis and J. S. Robinson, "Triangulation of planar regions with applications", *The Computer Journal*, vol. 4, no. 21, (1978).

- [10] A. L. Moura, J. R. Camacho and S. C. Guimarães, “A functional language to implement the divide-and-conquer Delaunay triangulation algorithm”, Applied mathematics and computation, vol. 1, no. 168, (2005).
- [11] B. Wang, B. C. Khoo, Z. Q. Xie and Z. J. Tan, “Fast centroidal Voronoi Delaunay triangulation for unstructured mesh generation”, Journal of Computational and Applied Mathematics, vol. 5, no. 280, (2015).
- [12] C. L. Lawson, “Software for C1 surface, Rice Journal Mathematical Software III Pasadena, (1977); California, US.
- [13] T. Y. Su, W. Wang, Z. H. Lv, W. Wu and X. F. L, “Rapid Delaunay triangulation for randomly distributed point cloud data using adaptive Hilbert curve”, Computers & Graphics, vol. 2, no. 54, (2016).
- [14] Q. M. Yang, X. G. Zhou and J. Zhang, “Insert Qi algorithm based on Delaunay triangulation point by point method”, Remote Sensing Information, vol. 1, (2011).
- [15] G. Z. Sheng and D. R. Zhang, “An improved algorithm to efficiently generate Delaunay triangulation”, Remote Sensing Information, vol. 1, (2005).
- [16] R. S. Jonathan and C. B. Brielin, “Fast segment insertion and incremental construction of constrained Delaunay triangulations”, Computational Geometry, vol. 8, no. 48, (2015).

Authors



Hu Rui, was born in Sept. 1987, Shan Xi Province, China. PhD candidate in Donlinx School of Economics and Management, University of Science and Technology Beijing. She received her B.S. and M.S. degree in the field of Management Science Engineering at the same university in 2009 and 2011. She is majored in industrial economy, operation management and system dynamics. She has been working on the “Low-carbon production mode in China’s metallurgical enterprises based on system dynamics” project, which was sponsored by National Natural Science Foundation of China, since 2011. She has had the main roles in data-purifying, steel production forecasting, model building and analysis. She also took part in several other project: the optimization of charge composition in converter process for Ma steel, 2014; the Research and Application of key technologies in the large-scale retail terminal management based on GIS, 2013; Research on the Optimization of Logistics process for Baotou Tobacco company, 2012; Research on the Development of New Formats of Business in Producer Service Sector, 2012. At the same time, she had published 10 papers in the past four years.



Zhang Qun, was born in Huang Gang, Hu Bei Province, in 1950. He received the B.S. and M.S. degrees in metallurgy from the University of Science and Technology Beijing, China, and the Ph.D. degree in Aachen University, Germany, in 1993. From 1995 to 2010, he was the Dean of Donlinx school of Economics and Management in University of Science and Technology Beijing. He is the author of “operation management”, one of the bestselling books in operation management field. He’s the editor in chief of another two books and over 120 articles. Meanwhile, he’s the expert in chief in the “low-carbon operation institution for enterprises in Beijing”. Mr. Zhang won the “National Fine-designed Books” by his book “Operation Management” in 2009. He also received the first grade prize for the science and technology progress set by the ministry of education in 2010, the third prize of science and technology in He Bei Province.

