An Efficient Algorithm for Constructing Delaunay Triangulation

Yu Jiang[1], Yin-tian Liu[1,2], Fan Zhang[1]
1 Intelligent Information Processing Laboratory, Chengdu Uninversity of Information Technology, Chengdu, China, 610225
2 College of Mathematics, Sichuan University, Chengdu, China, 610065
jiangyu@cuit.edu.cn

***Abstract*—An algorithm for fast constructing Delaunay triangulations was proposed. In iterations, this algorithm is to select the leftmost one point lying right some convex edges from set of points. The point and those convex edges construct new triangles, and add them to Delaunay triangulations. At same time, when new triangles do not meet Deaunay condition, optimize triangles based on LOP method, which is implemented in non-recursive way, and only needs computing and flipping. The average time complexity of the algorithm is analyzed, and it is O(n).**

Keywords-DEM; triangulated irregular networks; Delaunay triangulation; LOP optimizing

I. INTRODUCTION

A Delaunay triangulation of a set of *N* planar points is a triangulation with the property that the interior of the circumcircle of each triangle contains no points of the set [1]. If there are no four cocircular points in the set, then the Delaunay Triangulation is unique.

There has been much research based on computing them as well as research that makes use of their interesting properties in a variety of applications. The standard algorithms for finding the Delaunay triangulation of a point set in the plane as follows: A divide and conquer algorithm[2~6], In [2], this algorithm makes use of a quad-edge data structure, and an orientation and in-circle test are the only geometric queries involved. It runs in $O(N \log N)$ time. An algorithm based on sweepline technique computes the Delaunay Triangulation[7], this algorithm is largely concerned with computing the Voronoi diagram of a set of points in the plane directly, but only simple modifications are needed to compute the Delaunay triangulation instead. The running time is $O(N \log N)$.While a random incremental algorithm[1] has an average complexity of $O(N \log N)$ too. In [8~10], they proposed an algorithm respectively, the time complexity is about O(N).

At present, many algorithms use same steps to construct Delaunay triangulations. First, constructing triangulations; Second, optimizing triangulations based on **L**ocal **O**ptimization Process[11], LOP for short, which makes sure triangulations are Delaunay triangulations. And the time complexity for constructing triangulations is much less than LOP's. Therefore, LOP optimizing strategy determines the efficiency of constructing Delaunay triangulations.

In this paper, a right-hand rule is presented, which is convenient to establish correctly whether a given point lies left、on or right an oriented edge. And a non-recursive LOP algorithm is implemented. this algorithm only is a computing and flipping procedure. The time complexity of the algorithm in worse case is O(N). Based on those, an efficient algorithm for constructing Delaunay triangulation is proposed. At first, order the points according to x-coordinate and y-coordinate. Secondly, select a leftmost point from point set. Find out all convex edges from convex hull, and ensure that point lies right each convex edge. Thirdly, add new triangle to Delaunay triangulation. At same time, this algorithm using LOP method optimizes the quadrangle that is composed of the new triangle and another triangle having the common convex edge with the new triangle, which avoids searching Delaunay triangulation and improves algorithm efficiency.

978-1-4244-5265-1/10/$26.00 ©2010 IEEE

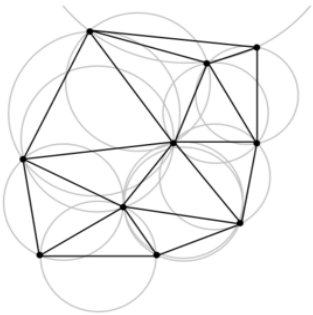
II. PRELIMINARIES

A. Delaunay Triangulation

In mathematics, and computational geometry, a Delaunay triangulation for a set P of points in the plane is a triangulation DT(P) such that no point in P is inside the circumcircle of any triangle in DT(P). Delaunay triangulations maximize the minimum angle of all the angles of the triangles in the triangulation; they tend to avoid skinny triangles.

Based on Delaunay's definition, the circumcircle of a triangle formed by three points from the original point set is empty if it does not contain vertices other than the three that define it (other points are permitted only on the very perimeter, not inside).

A triangle net is a Delaunay triangulation if all the circumcircles of all the triangles in the net are empty (please refer to Figure 1). This is the original definition for bidimensional spaces.



Supported by the National Natural Science Foundation of China under Grant No. 60773169 and No. 60702075, Development Foundation of Chengdu University of Information Technology(KYTZ200811)

Figure 1. A Delaunay triangulation in the plane with circumcircles shown

B. *Local Optimization Process*[11]

Local optimization procedure was proposed by Lawson. For four points on the same circle (e.g., the vertices of a quadrangle), the Delaunay triangulation is not unique: clearly, the two possible triangulations that split the quadrangle into two triangles satisfy the Delaunay condition.

Example 1: Looking at two triangles ABD and BCD with the common edge BD (see figure 2). If the sum of the angles α and β is less than or equal to 180°, the triangle meets the Delaunay condition. This is an important property because it allows the use of a flipping technique. If two triangles do not meet the Delaunay condition(the sum of α and β is bigger than 180°,see figure 2), switching the common edge BD for the common edge AC produces two triangles that do meet the Delaunay condition(see figure 3).

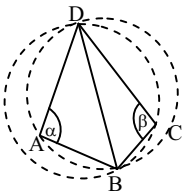


Figure 2. Those triangulations do not meet the Delaunay condition (the circumferences contain more than 3 points, the fourth point falls within the circle which is not allowed)

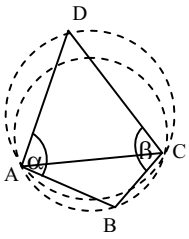


Figure 3. Switching the common edge BD for the common edge AC produces two triangles that do meet the Delaunay condition

C. *Right-hand rule*

In order to test mutual position of a point and an oriented edge, we can use right-hand rule. The definition of right-hand rule as follows:

Definition 1 Right-hand rule: Outstretch thumb and forefinger of right hand in the plane, and guarantee that thumb plumbs forefinger. Thumb and oriented edge overlap, and have same direction. The direction of forefinger is the right turn of oriented edge, on the contrary, is the left turn. (please refer to Figure 4)

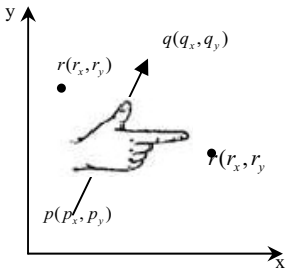


Figure 4. Right-hand rule

Right-hand rule can be computed as a value of determinant of a following matrix of dimension 3 or after some algebraic transformations by the matrix of dimension 2.

$$M(\overrightarrow{pq}, r) = \begin{vmatrix} p_x & p_y & 1 \\ q_x & q_y & 1 \\ r_x & r_y & 1 \end{vmatrix} = \begin{vmatrix} p_x - r_x & p_y - r_y \\ q_x - r_x & q_y - r_y \end{vmatrix} \quad (1)$$

The value of matrix M is positive if the point γ lies left. M is negative, if γ lies right, and zero if γ lies on $\overrightarrow{pq}$.

Theorem 1 Three vertexes of triangle are in counterclockwise order. If fourth point lies right two oriented edges of a triangle, this point must lie outside the circumcircle of the triangle.

Proof: if a point lies right two oriented edges of a triangle of which three vertexes are in counterclockwise order, this point must lie in the area closed in two broken lines (please refer to Figure 5). Obviously, this point lies outside the circumcircle of the triangle.

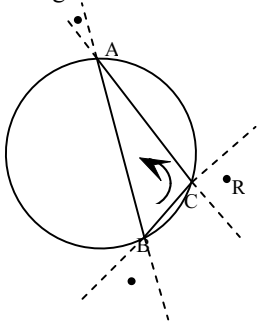


Figure 5. Point R lies right two oriented edges of a triangle

Property 1 If point R lies right the line through P and Q and oriented in the direction from P to Q, then three vertexes P、R and Q of ⊿PRQ are in counterclockwise order. If point R lies left the oriented edge <P,Q>, then ⊿PQR guarantees that three vertexes P 、 Q and R are counterclockwise.

Property 2 Three vertexes of triangle are in counterclockwise order. If a point lies left three oriented edges of a triangle, this point must lie inside this triangle.

III. DELAUNAY TRIANGULATION ALGORITHM

The key step in the construction of the Delaunay triangulation of a finite set of planar points is to establish

correctly whether a given point of this set is inside or outside the circle determined by any other three points and an efficient data structure for storing triangles and edges. In two dimensions, one way to detect if point D lies in the circumcircle of ⊿ABC is to evaluate the determinant:

$$M(\triangle ABC,D)=\begin{vmatrix} A_x & A_y & A_x^2+A_y^2 & 1 \\ B_x & B_y & B_x^2+B_y^2 & 1 \\ C_x & C_y & C_x^2+C_y^2 & 1 \\ D_x & D_y & D_x^2+D_y^2 & 1 \end{vmatrix}=\begin{vmatrix} A_x-D_x & A_y-D_y & (A_x-D_x)^2+(A_y-D_y)^2 \\ B_x-D_x & B_y-D_y & (B_x-D_x)^2+(B_y-D_y)^2 \\ C_x-D_x & C_y-D_y & (C_x-D_x)^2+(C_y-D_y)^2 \end{vmatrix} \quad (2)$$

The value of matrix M' is positive if and only if point D lies inside the circumcircle of $\triangle ABC$. A、B and C are in counterclockwise order.

In Delaunay triangulations, a triangle has common edge with three triangles at most. So, an efficient data structure for storing triangles and edges as follows:

```
typedef struct{              typedef struct{
  double x;                    Point P1;
  double y;                    Point P2;
  double val;                  Point P3;
} Point;                     }triangle;

typedef struct{
   triangle Tri;
   int r1;
   int r2;
   int r3;
}TriangleNeighbor;
map<int,TriangleNeighbor> Triangle;
```

The values of r1、r2 and r3 are -1 or a nonnegative integer. For example, the value of r1 is -1 if and only if there is not a triangle having common edge $(Tri.p1,Tri.p2)$ with triangle Tri. On the contrary, there is a triangle having common $(Tri.p1,Tri.p2)$ with triangle Tri, and the index of this triangle in array Triangle is r1.

A. Algorithm Implementation

Local Optimization Process makes sure all triangles meet Delaunay condition. The vertexes of all triangles are in counterclockwise order.

LOP(⊿PDQ)

{

- SP⇐{⊿PDQ};
- If SP=∅ then goto(11);
- ∀⊿ABC∈SP;
- If there is no a triangle has common edge AB with ⊿ABC then goto(6); else assuming this triangle is ⊿RBA.
- If $M'(\triangle ABC,R)>0$ or $M'(\triangle RBA,C)>0$ then switching the common edge AB for CR produces two triangles ⊿RBC and ⊿RCA that do meet the Delaunay condition. SP⇐SP-{⊿ABC}, SP⇐SP∪{⊿RBC, ⊿RCA}, goto(2);
- If there is no a triangle has commond edge BC with ⊿ABC then goto(8); else assuming this triangle is ⊿RCB.
- If $M'(\triangle ABC,R)>0$ or $M'(\triangle RCB,A)>0$ then switching the common edge BC for AR produces two triangles ⊿RCA and ⊿RAB that do meet the Delaunay condition. SP⇐SP-{⊿ABC},SP⇐SP∪{⊿RCA , ⊿RAB },goto(2);
- If there is no a triangle has commond edge CA with ⊿ABC then goto(10); else assuming this triangle is ⊿RAC.
- If $M'(\triangle ABC,R)>0$ or $M'(\triangle RAC,B)$ then switching the common edge CA for BR produces two triangles ⊿RAB and ⊿RBC that do meet the Delaunay condition. SP⇐SP-{⊿ABC},SP⇐SP∪{⊿RAB , ⊿RBC },goto(2);
- SP⇐SP-{⊿ABC};goto(2);
- Algorithm end.

}

Algorithm 1: An efficient algorithm for constructing Delaunay triangulation.

1) P={$m_1,m_2,\ldots,m_n$} is the set of points. The convex hull CH(S) of a set S of points in the plane is the smallest convex polygon containing S. S ⇐∅, CH(S) ⇐∅.

2) Sort the points in set P by their x-coordinate and y-coordinate when x-coordinates are equal, in increasing order.

3) Select the leftmost two points A and B in set P . Connect A to B form a oriented edge $\overrightarrow{AB}$ .P⇐P－{A,B}, S⇐ S∪{A,B}.

4) Select the leftmost one point in set P, and this point does not lie on $\overrightarrow{AB}$ .Assuming this point is C, and guarantee that three vertexes of ⊿ACB are counterclockwise order (according to property 1). Assuming A,C and B to lie counterclockwise.

5) P⇐P－{C}, S⇐ S∪{C},CH(S) ⇐{$\overrightarrow{AC},\overrightarrow{CB},\overrightarrow{BA}$};

6) if(P=∅) then goto(11).

7) Select a leftmost point D from set P. Find out all convex edges from CH(S), and ensure that point D lies right each convex edge. Assuming CE is composed of all convex edges that D lies right.

8) P⇐P－{D}, S⇐ S∪{D}, CH(S) ⇐ CH(S)－CE.

9) For each convex edge $\overrightarrow{PQ}$ in CE

Begin

- Construct ⊿PDQ, P,D and Q lie counterclockwise order;
- If $\overrightarrow{DP}\notin CH(S)$ then
  $CH(S) \Leftarrow CH(S)\cup\{\overrightarrow{PD}\}$
  Else $CH(S) \Leftarrow CH(S)-\{\overrightarrow{DP}\}$;
- If $\overrightarrow{QD}\notin CH(S)$ then
  $CH(S) \Leftarrow CH(S)\cup\{\overrightarrow{DQ}\}$
  Else $CH(S) \Leftarrow CH(S)-\{\overrightarrow{QD}\}$;
- if ⊿PDQ don't meet Delaunay condition then LOP(⊿PDQ);

End.

10) Goto(6).

11) Algorithm end.

B. Algorithm Complexity Analysis

The time complexity of Algorithm1 is composed of quick sort time and local optimization procedure (LOP) time in constructing Delaunay triangulation. In each iterations, |CH(S)| is much less than |S|(please refer to Table 1). So the time complexity for searching convex edges meeting condition form CH(S) is much less than LOP time complexity. And the time of quick sort is much less than LOP's time too. Therefore, the time complexity of Algorithm1 is mainly decided by the time of LOP in constructing Deaunay triangulations.

In each iterations, LOP only is computing and flipping procedures, which makes sure that forth point is not inside the circumcircle of any triangle in Delaunay triangulations, and |SP|≤|S|. The time complexity of LOP is O(|S|). Here, S is the point set of current Delaunay triangulation. Therefore, the average time complexity of Algorithm1 is as follows:

$$O(4)+O(5)+\ldots+\ldots+O(n)/n<O(1+2+3+4+\ldots+\ldots+n)/n=O(n(n+1)/2n)=O((n+1)/2)$$

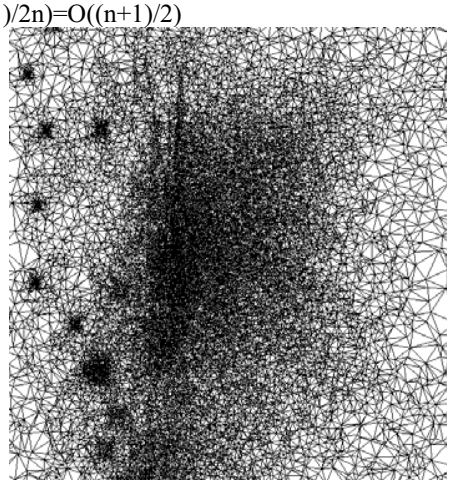


Figure 6. Constructing Denaunay Triangulation based on Argorithm1

In a word, the average time complexity of Algorithm1 is O(n).Table 1 shows the experiment result of Algorithm1.

TABLE I. SOME EXPERIMENT RESULT OF ALGORITHM1

Number of Points	Quick Sort Time	Max(CH(S))	Number of Triangles	Running Time of Algorithm(s)
5000	0.0	27	9978	1.062
10000	0.0	27	19980	2.390
15000	0.0	28	29976	3.812
20000	0.015	28	39983	5.313
25000	0.015	30	49979	6.734
30000	0.016	32	59976	8.062

Running Platform: Pentium 1.73GHz, 256M and VC6.0

IV. CONCLUSION

In each iterations, Algorithm1 adds new triangles to Delaunay triangulations continually, if the new triangles don't meet Delaunay condition, this algorithm using LOP method optimizes triangulations make sure triangulations are Delaunay triangulations, which avoids searching Delaunay triangulation like tradition incremental algorithms[8~10] and improves algorithm efficiency. Figure7 shows running time of algorithm1 in this paper and tradition incremental algorithm(Algorithm2 in [9])which construct triangulations firstly and then change triangulations into Delaunay triangulations based on LOP method.

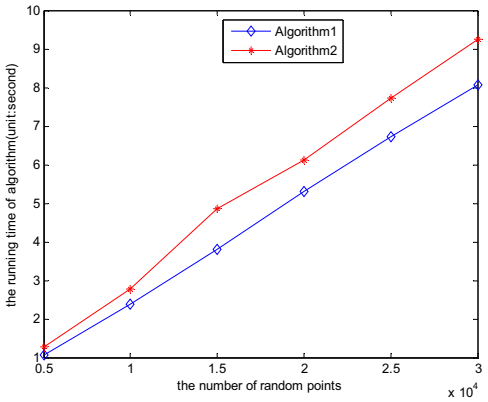


Figure 7. Running time of algorithm in this paper and tradition incremental algorithms.

REFERENCES

[1] M. de Berg, M. van Kreveld, M. Overmars, ea tl. Computational geometry. algorithms and applications, 2nd ed., Springer, 2000.

[2] Leonidas Guibas and Jorge Stolfi, Primitives for the manipulation of general subdivisions and the computation of Voronoi, ACM Transactions on Graphics (TOG) **4** (1985), no. 2, 74–123.

[3] M I Shamos, D Hoey. Closet-point problem. In: Proceedings of 16th IEEE Symposium on Foundations of Computer Science, Berkeley, California, 1975, (151) : 162.

[4] Lee D T, SchachterB J. Two algorithms for constructing a Delaunay triangulation.International Jounal of Computer and Information Science, 1980, 9 (3) : 219～242.

[5] Rex A Dwyer. A fast Divide-and-Conquer algorithm for constructing Delaunay triangulations. Algorithmica,1987, (2) : 137～151.

[6] [6] JIANG Hong-fei , TU Peng , LI Guo-zhong. Study on growth algorithm for generating constrained Delaunay triangulation. Journal of Highway and Transportation and Development ,2004 ,21(12) :38 - 41.

[7] S. Fortune. A sweepline algorithm for Voronoi diagram, Al gorithmica **2** (1987), 153–174.

[8] HU Jingxing, PAN Mao, MA Zhaoting, et al. Study on faster algorithm for constructing Delaunay triangulations DTM. Acta Scientiarum Naturalium Universitatis Pekinensis,2003,39(5):736-741.

[9] LIU Yonghe, WANG Yanping, QI Yonggan. Horizontal ExpandingMethod—A Quick Algorithm for Generating Delaunay Triangulation from Points in the Plane. Geo-Information Science,2008,10(1):20-25.

[10] MA Zhimin, LUO Bin. Entire optimized triangulation algorithm of Delaunay triangle network for DEM construction. Journal of Chang'an University,2008,28(3):44-48.

[11] Lawson. Software for C Surface Interpolation . In Mathematical Software III (J. R. R ice. ed) , Academic Press, New York , 1977, 161—194.