

# RESEARCH ON MODELS MERGENCE ALGORITHM BASED ON DELAUNAY TRIANGULATION

CHEN Genggen<sup>a,\*</sup>, OUYANG Ping<sup>b</sup>, LIU Shaohua<sup>c</sup>, XIAO Genru<sup>a</sup>

<sup>a</sup>Faculty of Geoscience and Geomatics, East China Institute of Technology, Fuzhou, Jiangxi 344000, China - (pgcheng, grxiao)@ecit.edu.cn

<sup>b</sup>Dept. of Water Conservancy, Nanchang Institute of Technology, Nanchang 330099, China - nc\_oy@163.com

<sup>c</sup>Department of Geosciences, JianHan Petroleum Institute, Jingzhou Hubei, 434023, China - lsh811@126.com

## Commission WG I/5

**KEY WORDS:** DEM, Triangulation, Digital photogrammetry, Spatial modeling, Topographic mapping, Algorithms

### ABSTRACT:

Model merging is widely used in many domains such as geography modeling, 3D design of highway and railway route and so on, so it is essential to design an efficient model merging algorithm. The objective of this paper is to research on models merging algorithm based on Delaunay triangulation. A method of obtaining model merged intersection lines quickly and an algorithm of searching triangles on the inside of the intersection lines fast are proposed. For the purpose of implementing Delaunay triangulation models merge, two main steps, such as insert the merged intersection lines into the Initial Delaunay Triangulation (IDT), and cut the triangles inside of the intersection lines in IDT out, are adopted. The processes of models merging are expatiated detailedly. The proposed algorithms are implemented by using VC++ programming language and tested by some experiment data. The experimental result validated the correctness and feasibility about the proposed algorithms.

## 1. INTRODUCTION

Triangulation irregular net (TIN) is widely used in many domains such as Geographic Information System (GIS), Geo-Science, Computer Graphics, Virtual Reality and so on. It is also one of hot topics in the recently 20 years. Delaunay Triangulation (DT) has two important characters, i.e its smallest inner angle is maximization and it has the empty circum-circle. These characters are superior in all methods constructing triangulation, and results in DT become one of main way to construct the Digital Terrain Model (DTM). DTM that construct by TIN has high value in application. It is not only used in the analysis of gradient, cubage of excavation and fill, but also used in the design of 2D and 3D when two DT models are merged. For example, suppose a dam should be built in an area. Firstly, DTM models (named initial Delaunay triangulation) of this area should be constructed. Secondly, choose the appropriate address according to the situation and build the dam DT models (named designed Delaunay triangulation). Lastly, Merging these two kinds of models and we can get a new DTM models (named object Delaunay triangulation). Based on this new model, we can carry on all kinds of manipulation. Delaunay triangulation models merge have an important application value. The study of constructing triangulation has many mature algorithms at present (Wu, 1999; Liu, 2001; Xu, 2000; Lewis, 1978), but there are few algorithms about D-triangulation models merge. It is mentioned a algorithm of superposition about models (Pu, 2001), but only the merge with the designed models constructed by Coons Patch and the D-triangulation models. The objective of this paper is to research on models merging algorithms of D-triangulation models and the correlative algorithms are proposed. It can resolve the problem of merge among D-triangulation models effectively.

## 2. DATA STRUCTURE IN THE ALGORITHMS

There are close relations between the efficiency and the data structure in algorithm. An appropriate data structure can effectively manage the data, so the efficiency of the algorithm get improved. The correlative data structure in algorithm is as follows.

### Vertex data structure

```
typedef struct vertex
{
    int No; // the vertex number
    double x,y,z; // the 3D coordinate of the vertex
} VERTEX;
```

### Triangle data structure

```
{
    int NO; //the triangle number, it is the same with the index
        number of the triangle link
    int triNo[3]; //the 3 adjoining triangle numbers
    BOOL deleted flag; //the deleted flag of triangle
    struct vertex v[3]; //the 3 vertexes of triangle
} TRIANGLE;
```

It should be mentioned that the three vertexes of a triangle must be stored in counter-clockwise. If edge  $i$  (value is 0, 1, 2) has no adjoining triangle, then  $triNo[i] = -1$ . The structure of triangle contains a key information that the subscripts of the three edges are the same with the value of  $i$  in  $triNo[i]$ , the value is 0, 1 or 2.

\* Corresponding author: Cheng Penggen, Doctor, Professor. email: pgcheng1964@163.com; phone 86 794 8258937; fax 86 794 825-8309.

### 3. DELAUNAY TRIANGULATION MODELS MERGENCE

The main idea of models mergence is follow. Firstly, construct the initial Delaunay triangulation model and the designed Delaunay triangulation model. Secondly, search the periphery edges (called merged intersection lines) of the designed Delaunay triangulation, and the periphery edge is a polygon. Thirdly, inset the periphery edges as the constrained lines into the initial Delaunay triangulation models. Lastly, delete all the triangles located in the periphery edge in the initial Delaunay triangulation model, thus the mergence of models can be obtained.

In the process of constructing initial Delaunay triangulation models interpolation point method is adopted, and the same time grid index is created for every triangles in the models. Because the grid index is established based on the triangle barycentera, so this method will reduce the time of judging which triangle is the vertex locate in greatly. It is convenient to insert the vertexes of the merged intersection lines into the initial Delaunay triangulation and determine the influence area of the merged intersection lines in the initial Delaunay triangulation model quickly, and the speed of models mergence algorithm will be increased consequently.

#### 3.1 Obtain the periphery edges of the designed triangulation models

Judging every triangle in the designed triangle models. If there is not adjoining triangle to a edge  $i$  in the triangle, i.e.  $triNO[i] = -1$ , it means this edge is a periphery edge. Although this method can judge all the periphery edges, but it is inefficient and the periphery edges are disjunct by getting order. Because there is a same vertex between two adjoining merged intersection lines, if the above method is adopted, then there are many repetitious computation in the process of inserting vertex of merged intersection lines into the initial Delaunay triangulation models, and every vertex will be used twice. Even if the vertex does not be inserted in the triangulation in second time, but it is time consuming in the process of searching which triangle is the vertex located in and checking whether the vertex has been insert in the triangulation, this will influence the efficiency of models merging algorithm. If we can develop an algorithm to obtain the adjoining periphery edges by order, then it will redound to the merged intersection lines inserted in the initial Delaunay triangulation. See figure 1, the algorithm is follow:

Firstly, find a periphery edge 13 (named target edge) in the designed Delaunay triangulation and store it into the periphery edge link list.

Secondly, start from the edge 13 and search the edges in the counter-clockwise in the target triangle  $T_1$ , if the edge is periphery edge, then change the next edge which subscript is bigger one than current edge into target edge. The detail process is as follows. Suppose edge 13 is a periphery edge and its subscript in the target triangle  $T_1$  is  $i$ . The next searching edge is the edge 32 which subscript is  $i+1$  in the target triangle  $T_1$ . If the value of  $i+1$  equal to 3 then let it equal to 0. If the edge is not the periphery edge then change the target triangle from  $T_1$  to  $T_2$  which is adjacent to the edge 32. Searching the next edge in counter-clockwise in the triangle  $T_2$ . Edge 34 can be obtained and judge it, if the periphery edge not be fond then change the target triangle to  $T_3$ . With the same way, the periphery edge 35 can be fond, and store it into the tail of the periphery edge link

list. Because edge 35 is the periphery edge, so the next target edge should be edge 54.

Repeat above process, all the periphery edges can be fond. They are stored in the periphery edge link list and are adjacent by order. Thus the merged intersection lines can be obtained. The algorithm end condition is that the target triangle equals the first target triangle  $T_1$  again.

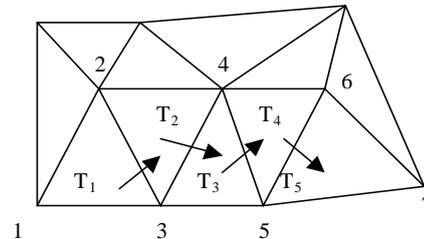


Figure 1. Obtain the merged intersection lines

#### 3.2 Inset merged intersection lines into the initial Delaunay triangulation

It is an important step to inset the merged intersection lines into the initial Delaunay triangulation models. This process is similar to the process of inseting constrained line segment into the unconstrained Delaunay triangulation. The process can be divided into two steps. Firstly, insert all the vertexes of the merged intersection lines into the triangulation. Secondly, find out all the influence areas of every merged intersection line and reconstruct the local triangulation.

##### 3.2.1 Inset the vertexes of the merged intersection lines into triangulation:

The process of inseting the vertexes of the merged intersection lines into triangulation is similar to the algorithm of constructing triangulation by interpolation method (Wu, 1999; Liu, 2000; Liu 2003; Liu, 2001). Because of the grid index is adopted at the same time of building triangle during constructing initial Delaunay triangulation models, so this method could short the time of locating the vertexes into a triangle and speed up the whole process of inseting vertexes of merged intersection lines into triangulation.

##### 3.2.2 Judge influence areas and reconstruct local triangulation:

The key step of inseting the merged intersection lines into triangulation is to locate the influence area. Constructing the influence area need to carry on some intersection operation. These operations are began from the triangle in which the first vertex of merged intersection lines located and go ahead along with the merged intersection lines. The triangles intersected with the merged intersection lines are brought into the influence area. In fact, the influence area is a muster of the triangles intersected with the merged intersection lines and is a closed areas or a polygon. Because all the vertexes of the merged intersection lines have been inserted into initial Delaunay triangulation, and the influence area is divided into two parts by the merged intersection line  $ab$ , i.e. left part and right part, see figure 2. So the local reconstruction of the triangulation in the influence area should be done separately in the left part and right part. The process of triangulation local reconstruction is

the same in the left part ( $P_{left}$ ) and the right part ( $P_{right}$ ). Here, we take right part of the line  $ab$  as an example to describe the process of triangulation local reconstruction.

(1) At the right hand of line  $ab$ , find the vertex 4 which have the biggest angle formed by vertex  $a$  and vertex  $b$  in the vertexes set  $V$  of the influence areas. Constructing the first triangle  $\Delta a4b$  and the value of using time of three sides of the  $\Delta a4b$  plus 1, see figure 2.

(2) Taking the two new sides  $a4$  and  $4b$  of triangle  $\Delta a4b$  as the basic-side, repeat the step (1) to expand the triangle. When the expanded side is the periphery edge or the value of used time equals 2, the expand process is ended, for example,  $4b$  is the periphery edge, it need not to expand. If a new side is the periphery edge, which belong to the influence area, and it has two adjoining triangles, then the method of LOP should be used to deal with the two triangles. If the two triangles share a same circle, exchange their diagonal so as to get the optimal triangles and set up their topological relation. For example, in the new triangle  $\Delta a4b$ , the side  $4b$  is the periphery border, if the side  $4b$  have two adjoining triangles (i.e.  $\Delta a4b$  and  $\Delta 4cb$ ) then LOP algorithm should be used to deal with the adjoining triangles, otherwise, the LOP does not carry on.

Repeat the step (2), take the new side as the basic-side to expand triangle until all the sides have been processed once. Thus process of local reconstruction of the triangulation in the influence area is ended.

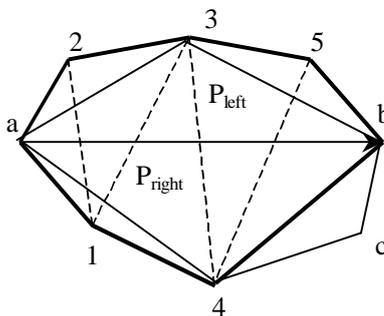


Figure 2. Influence area of the merged intersection lines and local reconstruct of triangulation

### 3.3 Cut out initial Delaunay triangulation

All In order to get a real mergence model, it should be done that delete the all triangles inside the area closed by the merged intersection lines in initial Delaunay triangulation. When all the merged intersection lines have been inserted into the initial D-triangulation models, the merged intersection lines construct a constrained boundary, and all the triangles inside the constrained boundary should be deleted. Obviously, How to find all the triangles is the key step. If judge all the triangles in the initial Delaunay triangulation models, it is obviously a complex process and it is take more time. Especially when there are huge number triangles the efficiency of this method is lower.

In this paper, according to the topological relation of triangles, a new algorithm is adopted so as to simplify the process. For the purpose of implement and management the algorithm, set up an array  $D$  to store the triangles located inside the constrained boundary,  $TT$  is the total of triangles in array  $D$ , and  $TU$  is the

recorder of a triangle be used time. Suppose  $T$  is a triangle which located inside the constrained boundary and is stored in array  $D$ , and  $TT=1$ ,  $TU=0$ . The algorithm of searching all the triangles located inside constrained boundary is illustrated as follows.

(1) In the initial Delaunay triangulation models, judge the three sides of triangle  $T$  which number is  $TU+1$  in array  $D$ . See figure 3, if it is not the side of constrained boundary (the bold side in figure 3) and the deleted flag of triangle adjoining this side is FALSE, then store the side into array  $D$  and  $TT$  plus 1, and change the deleted flag of this triangle into TRUE in initial Delaunay triangulation model. When the three sides of triangle  $T$  have been judged,  $TU$  plus 1. See figure 3, for example, in the initial triangle  $T$ , because side  $ab$  and side  $bc$  are constrained boundary, so the two adjoining triangle  $T_a$  and triangle  $T_b$  which correspond to side  $ab$  and side  $bc$  separately are needless to be stored in array  $D$ . Side  $ac$  is not the constrained boundary, the triangle  $T_1$  which adjoining  $T$  should be stored into array  $D$ ,  $TT$  plus 1, and set the deleted flag of triangle  $T_1$  to TRUE,  $TU$  plus 1.

(2) Judge if  $TU$  equals  $TT$ . If  $TU < TT$  then repeat the step (1). If  $TU = TT$  then the algorithm conclude. And this time, all triangles located inside the constrained boundary are stored in array  $D$ , and the deleted flag of all triangles located inside constrained boundary in initial Delaunay triangulation models are TRUE.

The algorithm begins from triangle  $T$  located inside the constrained boundary. It is time consuming to find a triangle  $T$  inside constrained boundary. To avoid this step, a simplified the algorithm can be adopted. Because the periphery edge of the designed Delaunay triangulation is obtained in the counter-clockwise direction, so the adjoining triangle on the left of the merged intersection lines should be deleted. When insert the last merged intersection line into the initial Delaunay triangulation, Record the first triangle, which is created by Delaunay triangulation in the left part of the merged intersection line, and let it as the start triangle, i.e. triangle  $T$ . For example, triangle  $\Delta ab3$  in figure 2.

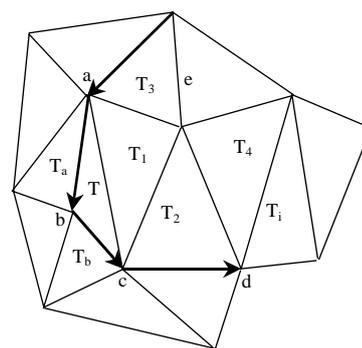


Figure 3. Search the triangles located inside constrained boundary

Delete all the triangles whose deleted flag are TRUE in the initial Delaunay triangulation model and the process of cutting out the initial Delaunay triangulation is finished.

After the above process, the muster of the initial Delaunay triangulation model and the designed Delaunay triangulation model are the mergence model.

#### 4. EXPERIMENT AND CONCLUSION

The algorithms proposed in this paper are implemented by using VC++ 6.0 programming language and are tested by some experimental data, the experimental result are showed in figure 4 - figure 7.

Figure 4 demonstrates an initial Delaunay triangulation model and a designed model before the models merging. According to the steps of the algorithms designed, firstly, insert the vertexes of the merged intersection lines into the initial Delaunay triangulation model showed in figure 5; secondly, insert the merged intersection lines into the initial Delaunay triangulation model showed in figure 6; delete all the triangles which locate inside the constrained boundary and get new Delaunay triangulation model showed in figure 7. The model merging process could be finished momentarily. The experimental results demonstrated the correctness and feasibility of the proposed algorithms. It can be used in many engineering application domains, sun as 3D design of highway and railway, geography modeling and so on, and have well application value.

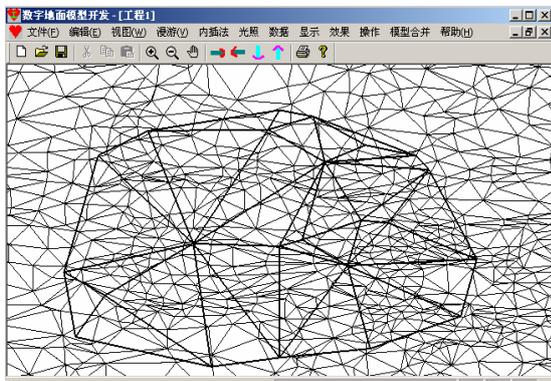


Figure 4. Before the models merge

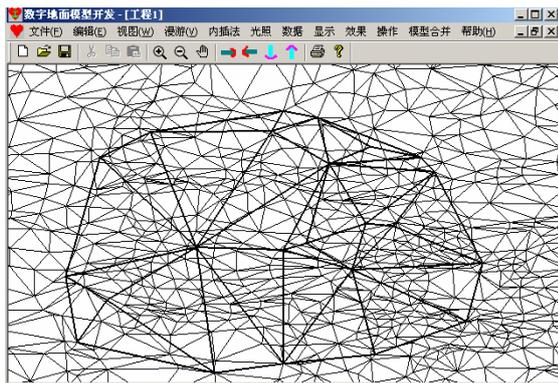


Figure 5. Insert the vertexes of the merged intersection lines

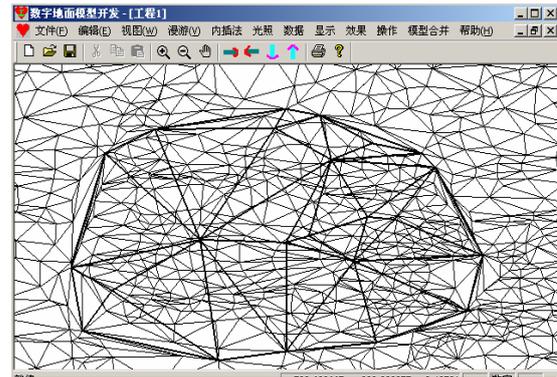


Figure 6. Insert the merged intersection lines

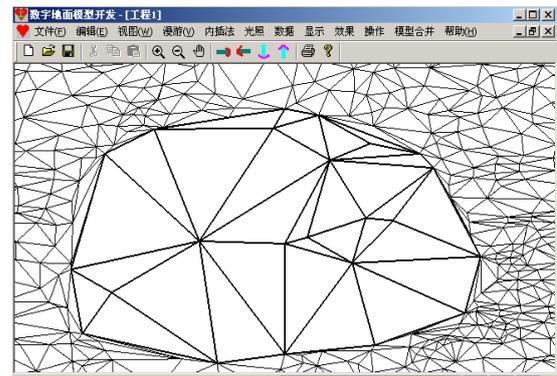


Figure 8. Result after the models merging

#### REFERENCE

- Wu X., Wang S., 1999. A study of algorithm for constructing Delaunay triangulation. *Acta Geodaetica et Cartographica Sinica*, 28(1), pp. 28-35.
- Liu X, Fu X., 2000. A study of algorithm for fast constructing triangulation irregular net (TIN). *China Journal of Highway and Transport*, 13(4), pp. 16-20.
- Xu Q, Chang G., 2000. The algorithm for TIN based on the way to divide it into few parts. *Journal of Image and Graphics*, 5(6), pp. 461-465.
- Lewis B., Robinson J., 1978. Triangulation of Planar Regions with Applications. *The Computer Journal*, 21(4), pp. 324-332.
- Pu H., Song Z., 2001. The route 3D models based on Delaunay triangulation. *Journal of the China railway Society*, 23(4), pp. 81-87.
- Liu S., Cheng P., 2002. The study of algorithm for inserting points into the Delaunay triangulation. *Journal of east China geological institute*, 25(3), pp. 254-257.
- Liu X., Gong J., 2001. Delaunay triangulation of constrained data set. *Acta Geodaetica et Cartographica Sinica*, 30(1), pp. 82-88.