

DELAUNAY TRIANGULATION PARALLEL CONSTRUCTION METHOD AND ITS APPLICATION IN MAP GENERALIZATION

J. Shen^{a, b*}, L. Guo^{a, b}, L. Qi^c, W. Zhu^{a, b}

^a Institute of Geographic Science, Nanjing Normal University, 210046 Nanjing, China - (Shenjje, guolishuai, zhuwei)@njnu.edu.cn

^b Key Laboratory of Virtual Geographic Environment, Ministry of Education, Nanjing Normal University

^c Nanjing Institute of Geography and Limnology Chinese Academy of Sciences, 210008 Nanjing, China(qilin04043224@126.com)

Commission II, WG II/2

KEY WORDS: Delaunay Triangulated Irregular Network, parallel computing, data partitioning, dynamic strip, mesh simplification

ABSTRACT:

Delaunay triangulated irregular network (D-TIN) has been widely used in various fields and also played an increasingly important role on map generalization. But for massive data processing, current D-TIN algorithm is still not efficient enough to meet the requirements of map generalization. Data partitioning is an important step of parallel algorithm design. The load balance and efficiency of data partitioning is the precondition of improving parallel algorithm efficiency. For aggregated distributed point sets, the traditional Delaunay Triangulation parallel algorithm can't ensure the balance and the execution's efficiency of the partitioning result. The paper introduces a partitioning method using dynamic strips aiming to guarantee the computing load balance. We tested the speed-up of the D-TIN parallel algorithm using different type of point sets and the results of the experiments shows that the method of dynamic strips partitioning can help to get high and stable speed-up and the data distributional pattern and size has less influence to it. The paper realizes a mesh simplification algorithm based on parallel D-TIN and compares the efficiency based on parallel and serial D-TIN.

1. INTRODUCTION

As an important tool for geometry calculation, D-TIN (Delaunay Triangulated Irregular Network) has been widely used in various fields. D-TIN is very useful for analysing morphological structures of geometric figures because it has the properties of maximal and minimal angle, uniqueness. D-TIN has also played an increasingly important role on map generalization and gradually became an indispensable tool. D-TIN generation algorithm was first put forward by Delaunay in 1934 and has been matured within several decades of research. Tsai classified the D-TIN algorithms into three kinds, divide-and-conquer methods, incremental insertion methods and triangulation growth methods. Shamos and Hoey first introduces the divide-and-conquer method, Lewis put forward the incremental insertion method and Lee, Schachter carried out improvement and perfection, Green first realized triangulation growth method. Although so many D-TIN algorithms has been developed, but for massive data processing, current D-TIN algorithm is still not efficient enough to meet the requirements of map generalization, especially for on-the-fly map generalization. Recently, high performance computing techniques has been widely used in Geo-computing. The parallelization of D-TIN algorithms began in the late 1980s, among various methods that have been proposed over the years, the problem that how to divide the computing task in parts that can be run concurrently in parallel processors is still not solved.

The present paper describes a new parallel D-TIN construction method based on dynamic strips data partitioning. This method

proposed the division principle and can meet requirements of relatively balanced division results for different distributing types of points set data, including gathering point set data. Data partitioning is the most commonly used method and aims to guarantee the computing load balance, thereby it is an important premise for enhancing the performance of parallel algorithms. Data partitioning for traditional D-TIN parallel algorithms can obtain relatively balanced division results and high efficiency when it is used for the point set in uniform density. But the balance of division results can only be obtained at the expense of division efficiency under non-uniform density of point set situation. There are many map generalisation algorithms using DTIN because D-TIN is help for adjacency relation recognizing, conflict detecting and feature line extracting. The paper realizes a mesh simplification algorithm based on parallel D-TIN and compares its efficiency based on parallel and serial D-TIN.

The present study is organized as follows. Section 2 presents the dynamic strips data partitioning method for parallel D-TIN. Section 3 introduces the D-TIN parallel computing process and the D-TIN parallel algorithms speed-up. Section 4 presents computational results of mesh simplification based on parallel D-TIN and the concluding remarks of the paper are outlined in Section 5.

2. OVERVIEW OF D-TIN PARALLEL COMPUTING

The research of D-TIN parallel construction algorithm began in the late 1980s, after 20 years of exploration, scholars proposed many parallel methods, among the D-TIN parallel algorithms,

* Corresponding author.

the parallel algorithm based on data partition was the most commonly used method. With the development of computing technique, some D-TIN parallel algorithms were proposed based on different computing environment.

Davy proposed a parallel algorithm based on divide and conquer algorithm for 2D point set D-TIN (Davy, 1989), this method is a data partition strategy, the efficiency is influenced by the load balance of partition. Lee proposed a data partition divide and conquer strategy, combined with the projection method for D-TIN parallel algorithm (Lee, et al., 1997). This segmentation method can guarantee no boundaries recovery and continuous processing in the merger step. But the computing task become more complex and difficult when it deals with large quantity of data. Cignoni proposes two methods of D-TIN parallel computing on 3D space (Cignoni, et al., 1993). First algorithm is DeWall algorithm which is a divide and conquer algorithm in Ed ($d > 2$) space expansion. Firstly, according to a certain mode split and use the incremental insertion method to construct the border triangle, sharing boundaries subsets involved in construction Triangulation. The performance of this algorithm is limited by the data size, the cost of additional computing will increase by increasing of the data size. Other a D-TIN parallel algorithm is called the InCode algorithm, is designed on the basis of the incremental insertion algorithm, the first set of points is divided into K rectangular areas, each region carry out the D-TIN on different processors points, then merge those subnets. The performance of this method is influenced by data partition and the characteristic of data.

The requirement for different computing environment D-TIN parallel construction is different, thus needing to design different D-TIN parallel algorithm. Kohout proposed a D-TIN parallel algorithm based on the circumcircle guidelines on shared memory systems (Kobout and Kolingerová, 2003). Zhang proposed a D-TIN parallel algorithm based on divide and conquer algorithm in distributed environment (Zhang, et al., 2000). Yi proposed a D-TIN parallel algorithm based on grid environment (Yi, et al., 2001). Li used the CORBA component to implement the D-TIN parallel algorithm in the cluster environment (Li, et al., 2008).

In summary, D-TIN parallel computing methods mainly concentrated in data partitioning, in this paper we will propose a data partitioning method based on the dynamic strips. With the popularity of the multi-core processor, OpenMP (Open Multi-Processor) programming model provides a new way to improve the efficiency of D-TIN parallel construction, we design the D-TIN parallel computing based on the OpenMP.

3. DESCRIPTION OF DYNAMIC STRIP PARTITIONING (DSP) METHOD

3.1 The principle of data partitioning method oriented parallel D-TIN

In order to ensure the accuracy and efficiency of parallel D-TIN construction, some special requirements for parallel D-TIN algorithm should be considered when carrying out data partitioning. The paper concluded the data partitioning principle oriented parallel D-TIN construction summarized as:

1) Data partition region must be a convex polygon. Because the outer boundary is convex hull which is the special characteristic of D-TIN, in order to avoid the D-TIN subnet across the

triangle edge, greater difficulties for the subnet merger or even can produce erroneous results, data partitioning need to ensure that every subset of the data are convex polygons surrounded and does not occur coincident.

2) The result of data partitioning must be balance. To ensure load balancing, each compute node of the computing tasks is relatively balanced, it is the important requirement for parallel algorithm design. The quantity of points in each point set directly related to the D-TIN algorithm's time complexity, therefore, the data partitioning results should ensure that the number of points contained in each region roughly to be equal.

3) Efficiency requirement. Data partitioning process is one part of the parallel algorithm, the execution time of the data partitioning included in the whole parallel execution time. In order to enable parallel D-TIN algorithm can get better acceleration, the data partitioning method should have a relatively high efficiency. Meet the above two principles, to minimize the amount of parallel D-TIN time, the time of data partitioning should be less.

3.2 The basic idea of DSP

The basic idea of the method is as follows: firstly, the minimum bounding rectangle of the point data set should be obtained, then the point set is split into m slim strips on the same direction ($m \in N$, larger than the number of processors for parallel computing). Secondly, the strips are merged into a partition region from the first strip in sequence, if the total number of points in the merged strips match the condition of load balance threshold, a new strip will no longer added to the current data region but beginning with a new one. In a special condition, the number of the last strip merged to one region is too large to meet the load balance requirement, thus the boundaries of some strips should be moved. It is the rule for the boundaries moving, the distance of each movement is half of the stripe width. The iteration stopped until the number of this strip meets the requirement of load balance. In order to facilitate the description, the variables involved in DSP algorithms as defined in the following table.

variables	Description of variables
N	The number of points in all point clusters;
K	The numbers of parallel compute node (or threads);
N_A	$N_A = N/K$, the average numbers of points for every compute node at balance. In fact, data partitioning does not guarantee that each node in the amount of data is strictly consistent, so should set a load threshold surround N_A upper and lower limits N_D , N_U ;
N_D	The lower limit of the threshold, that is, a single computing node load floor;
N_U	The upper limit of the threshold, that is, a single computing node load ceiling;
N_F	Number of points in the merger within the region.

Table 1. Description of variables in DSP

3.3 The process of DSP

The process of dynamic strip partitioning method as follows (see Figure 1):

1) Calculation of the minimum bounding rectangle of point set, if the length of the rectangle in the X-axis direction is greater than the Y-axis direction, $|X_{\max}-X_{\min}| \geq |Y_{\max}-Y_{\min}|$, the point set will be divided according to the X-axis direction, otherwise the delineation of the Y-axis direction.

2) According to the number of compute nodes (or threads), the first point set of minimum bounding rectangle roughly divided into m strips (m is an integer, generally set for the number of computing nodes K are integer multiples), and order of band number in accordance.

3) Successive merging began from the band of the smallest number, then the number of points in the merging region is accumulated. If N_F (the number of points in the merging region) meet the load threshold range ($N_D \leq N_F \leq N_U$), new strip of points will no longer be added, these points will be marked and given a new unified number and will not be merged into other strip.

4) If the $i-1$ -th strip added points with N_F within the combined region, the number of points does not meet the minimum single node load threshold N_D , while point in the i -th strip to join the combined region, the N_F more than a single maximum node load threshold N_U , it need to move the border of i -th strip until the right amount of points allocated to the consolidated area.

5) Repeat steps 3), 4), until all bands have been assigned.

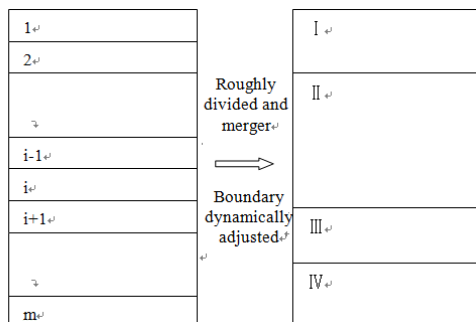


Figure 1. Process of dynamic data partitioning

In order to improve the efficiency of the algorithm, the frequency of boundary moving should be reduced, we use semi-mobile way to fine-tuning the boundaries of merge region. If produce the situation described in 4), the lower boundary moved inward (narrow strip range) $1/2$ width value of the region, then recount points in this strip. If N_F can meet the balance requirement, it will be added to the combined area; otherwise, it still needs to adjust the boulder. If N_F still more than the load threshold upper limit N_U , continue moving boundary inward; if the area points less than the load threshold lower limit N_D , moving boundary outward, the movement value is $1/2$ of the width of each border. This is an iterative process of implementation, which can make N_F reach the load threshold condition through a number of adjustments. This boundary fine-tuning rules of semi-mobile can make the combined region to minimize the number of the boundary adjustment to achieve the good effect of the Division of purpose.

4. D-TIN PARALLEL COMPUTING PROCESS

4.1 Experimental environment and Experimental data

Hardware: CPU: Intel(R) Core(TM) i5-2400 CPU @3.10GHz; Memory: 2 GB. Software: Windows 7, Visual Studio 2008, ArcGIS Engine 9.3, OpenMP, C++.

There are three types of the spatial distribution of point set such as uniform distribution, random distribution and clustered distribution. The size of each region of the point set which is uniform or random distribution is almost same, ESP (Equally Strip Partitioning) method can meet the need of the load balancing. But for clustered distribution, the point group always distributed around a single core or multi-core, the data is concentrated in different particular region, the ESP method cannot meet the load balancing requirement. In order to verify the rationality and applicability of the DSP algorithms, during the experiment we select clustered distribution point set data, including single-core, multi-core clustered by artificial simulation data and thematic point sets of Nanjing city (see Figure 2).

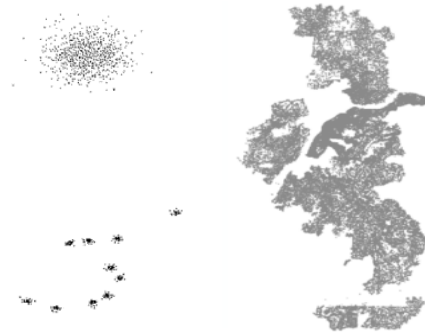


Figure 2. Example of experimental data

4.2 Dynamic data partitioning

This experiment used thematic point sets of Nanjing city, which contains 500,000 points. During the experiment, we compared the result of ESP and DSP methods, in accordance with division requirements of four processing nodes. Figure 3 shows the results of the data partitioning using ESP and DSP methods, Table 2, 3, 4 shows the data size of each region after data partitioning.

It can be known from Figure 3 and Table 2, 3, 4, the results of the ESP method shows serious data inequality, DSP method produced relative balanced quantity of data, and as the subset of distribution within the rectangular, the border of area across phenomenon can be avoided between subnets. More balanced subset of data obtained through the rough division, adjustment, and merging operations, these processes do not contain complex operations, such as global sorting, its time complexity is $O(n)$, according with the three principles of D-TIN construction data partitioning.

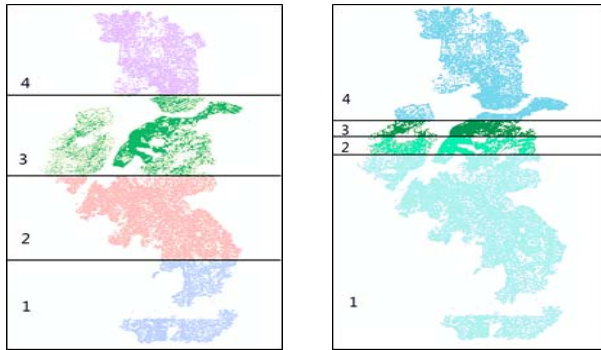


Figure 3. The result of data partition

Method	1	2	3	4
ESP	20930	396736	61289	21045
DSP	124497	124629	125676	125198

Tab.2 Result of data partition on thematic data of Nanjing using ESP and DSP for 4 nodes

Method	1	2	3	4
ESP	40930	186736	201289	71045
DSP	121497	127629	128676	122198

Tab.3 Result of data partition on single-core aggregated data using ESP and DSP for 4 nodes

Method	1	2	3	4
ESP	120930	166736	91289	121045
DSP	123497	125629	127676	123198

Tab.4 Result of data partition on multi-core aggregated data using ESP and DSP for 4 nodes

4.3 D-TIN parallel computing

D-TIN parallel computing includes three steps, data partitioning, parallel construction subnets and merger subnets. Serial implementation of the data division, the remaining two steps can be executed in parallel. During parallel construction subnet step, all threads simultaneously compute. During merger subnets, each of the two threads merge subnets at the same time, its time complexity is $O(\log^N)$. Figure 4. is the flow diagram of D-TIN parallel computing. Figure 5 shows the result of every step of D-TIN parallel construction using the data which is a part of the thematic point data of Nanjing city.

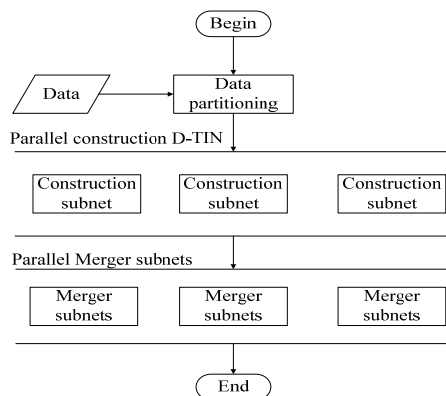


Figure 4. The flow diagram of D-TIN parallel computing

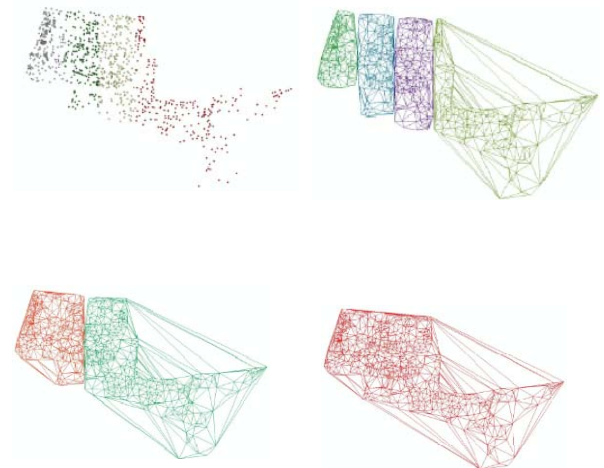


Figure 5. Processes of D-TIN parallel computing

To further validation the applicability of the DSP algorithm, we used single-core, multi-core aggregated data and the thematic data of Nanjing city to test the performance of D-TIN parallel algorithm (see Figure 6).

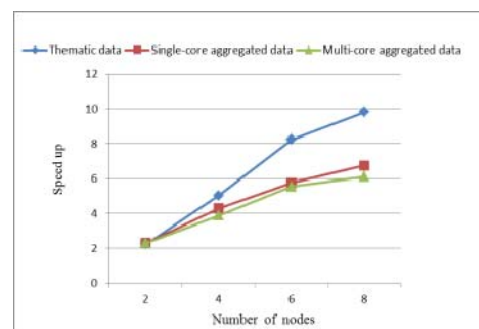


Figure 6. DTIN parallel algorithms Speed-up

The figure 6 presents the speedup of the D-TIN construction for 3 types of point set. The observed speedups, increase with the number of nodes, are almost equal between single-core and multi-core aggregated data. The speedups of the thematic data of Nanjing city are better than the other two datasets. After analyzing the spatial feature of the partitioning data, we found that some boundary lines of the strips crossing the center of the single-core and the multi-core aggregated data, therefore it would need more merger time than the thematic data of Nanjing city.

5. THE APPLICATION OF D-TIN PARALLEL COMPUTING IN MESH SIMPLIFICATION

5.1 Mesh simplification

Mesh simplification is original mesh optimization. Hoppe et al. (1993) referred that mesh optimization can be put to use effectively in mesh simplification (the reduction in the number of vertices in an initially dense mesh of triangles). Burghardt et

al. (2007) used the mesh simplification for the typification of buildings.

This part we will use the mesh simplification to simply point feature. Mesh simplification is composed of two steps, positioning and representation. The positioning step determines the number and position of the point features based on D-TIN, the representation step is used to calculate the distance for the replacement point features, then reconstruction the D-TIN (see Figure 7).

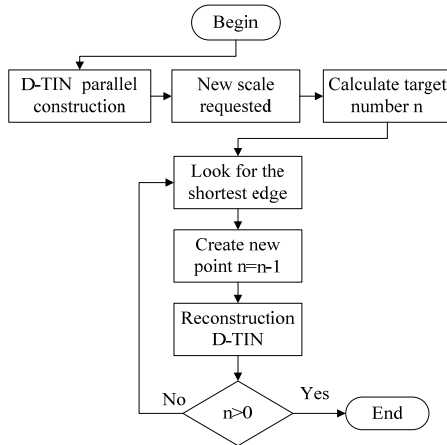


Figure 7. The flow diagram of mesh simplification

5.2 Experiment and analysis

Figure 8 is an example of this approach. On the left-hand side of figure 8, a dense mesh made up of large number of vertices describes the buildings of Nanjing. On the right-hand side, we used mesh simplification to generation the buildings, maintaining it outline and characteristic.

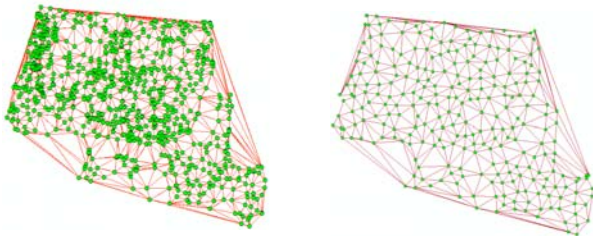


Figure 8. Example of the mesh simplification for simplifying point feature

The efficiency of this algorithm is affected by data characteristic and the performance of D-TIN construction, thus, in order to test the this algorithm's adaptability, we still use three types data to test its speed-up (see, Figure 9), reduced number of the points is the 30% of the original data.

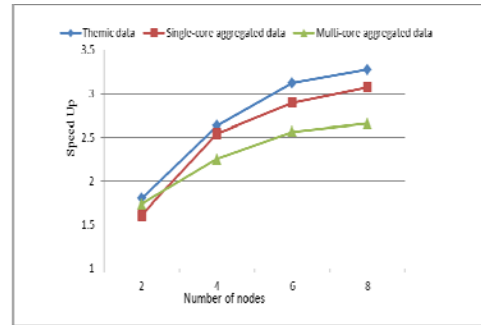


Figure 9. Mesh Simplification Speed-up

The figure 9 shows the speedups of the mesh simplification algorithm, obviously the speedups is increasing with the increasing the number of nodes. The performance of thematic data is better than the rests, the single-core aggregated data is senior than the multi-core aggregated. The reasons of this phenomenon are determined by D-TIN parallel efficiency and the cost of the position and rebuild the topological relation of the new D-TIN after mesh simplification on the different characteristic data.

6. CONCLUSION

The paper reviewed the DTIN construction algorithms and also the parallel algorithms. Based on the review, we found that data partitioning has been the main method of D-TIN parallel computing. The paper introduces a partitioning method using dynamic strips aiming to guarantee the computing load balance. We tested the speed-up of the D-TIN parallel algorithm using different type of point sets and the results of the experiments shows that the method of dynamic strips partitioning can help to get high and stable speed-up and the data distributional pattern and size has less influence to it. The paper realizes a mesh simplification algorithm based on parallel D-TIN and compares the efficiency based on parallel and serial D-TIN.

Acknowledgements

The research is financed by the National Natural Science Foundation of China (41071288, 41171350). The authors would like to thank Prof. Naijie Gu and the others of network computing and advanced algorithm laboratory of USTC (university of science and technology of China) for their help and discussion when we visited the laboratory.

REFERENCES

- Ai T, Guo R, 2000. Extracting center-lines and building street network based on Constrained Delaunay Triangulation. ACTA GEODAEITICA ET CARTOGRAPHIC SINICA, 29(4).pp.348-354.
- Burghardt D,Cecconi A,2007.Mesh simplification for building typification.Internal Journal of Geographical Information Science,21(3).pp.283-298.
- Cignoni P, Montani C, Perego R, et al., 1993. Parallel 3D Delaunay Triangulation. Citeseer.

- Davy J R, Dew P M, 1989. A note on improving the performance of Delaunay triangulation. *Proceedings of Computer Graphics International*, pp. 209-226.
- Green P J, Sibson R, 1978. Computing Dirichlet tessellations in the plane. *The Computer Journal*, 21(2). pp. 168.
- Hopple H, Derose T, Duchamp T, et al, 1993. Mesh optimization. *Computer Graphics*, pp. 247-254.
- Kohout J, Kolingerová I, 2003. Parallel Delaunay triangulation based on circum-circle criterion. *ACM*.
- Lee S, Park C I, Park C M, 1997. An improved parallel algorithm for Delaunay triangulation on distributed memory parallel computers. Published by the IEEE Computer Society.
- Lee D T, Schachter B J, 1980. Two algorithms for constructing a Delaunay triangulation. *International Journal of Parallel Programming*, 9(3). pp. 219-242.
- Lewis B A, Robinson J S, 1978. Triangulation of planar regions with applications. *The Computer Journal*, 21(4). pp. 324.
- Li J J, Li C Y, 2008. Parallel Algorithm for Delaunay Triangulation and Its Implementation on Computer Clusters. *Journal of Northeast Forestry University*, 36(10). pp. 83-85.
- Tsai V, 1993. Delaunay triangulations in TIN creation: an overview and a linear-time algorithm. *International Journal of Geographical Information Science*, 7(6). pp. 501-524.
- Shamos M I, Hoey D, 1975. Closest-point problems. *IEEE*. pp. 151-162.
- Tang G, Liu X, LV G, 2005. Digital elevation model and its geological analysis principle and method. Science Press, Beijing.
- Yan H, Weibel R. 2008. An algorithm for point cluster generalization based on the Voronoi diagram. *Computers and GeoSciences*, 34(8). pp. 939-954.
- Yi F L, Li Q H, Yang W W, 2001. Parallel algorithm of Delaunay Triangulation dividing. *Mini-micro Systems*, 22(4). pp. 450-452.
- Zhang M M, Pan Z G, et al., 2000. A Distributed Parallel Algorithm for Delaunay Triangulation of Scattered Data Points. *Journal of Computer Aided Design & Computer Graphics*, 12(7). pp. 44-47.
- Zhou T, Long Y, Tang G A, Hu L D, 2010. Research on the Hybrid Index Structure for Aggregated Spatial Data. *Geography and Geo-Information Science*, pp. 7-10.