International Journal of Database Theory and Application
Vol.8, No.1 (2015), pp.137-148
http://dx.doi.org/10.14257/ijdta.2015.8.1.15

A Novel Algorithm for Aggregating the Topological Nodes in Web GIS Network Management

Min Huang*[1], Likun Zhu[2], Pengfei Liu[1], Jingyang Wang[1] and Liwei Guo[1]

[1]*School of Information Science and Engineering, Hebei University of Science and Technology, Shijiazhuang, 050018, China*

[2]*Department of Enrollment and Employment, Hebei University of Science and Technology, Shijiazhuang, 050018, China*

**ever211@163.com, zhulikun@hebust.edu.cn, 1026106132@qq.com, jingyangw@hebust.edu.cn*

Abstract

The problems of the icons covering each other and slow data loading are very challenging when the GIS technology is applied to load a large number of topological nodes. A novel algorithm for aggregating the topological nodes in GIS based on the hierarchical clustering algorithm is presented in this paper. Comparison with the hierarchical clustering algorithm and partitioning algorithm is explored. In addition, the implementation of the algorithm is derived. The experiment demonstrates that the algorithm solved the problems of the icons covering each other and slow data loading effectively, realized the aggregation of the topological nodes preferably, and presented the distributed trend of the topological nodes and the topology relation between each other in the topology. The experimental results verify the effectiveness of the method.

Keywords: *aggregating the topological nodes; hierarchical clustering algorithm; topological node; distributed trend; geographic info system; network management*

1. Introduction

In recent years, with the rapid development of GIS (Geographic Information System) technology, domestic and foreign research institutions, network operators, software developers and other corporations have come to realize the necessity of combining GIS technology with network management, and they have managed the public network and professional network with GIS technology. We can solve many problems of analyzing and processing data in network management through GIS technology [1, 2]. GIS combines spatial information and associated attribute effectively, and it uses electronic map as the carrier. Further, it can accurately, visually reflect the operational status of the entire network, the distributed trend of the topological nodes and the topology relation between each other in the form of icons in real time. Not only can it help decision-maker grasp the overall macroscopically, but also can provide good basis for decision-maker to make accurate and effective decision in time [3, 4].

Since the entire topological graph uses the electronic map as the carrier, the icons of geographical location adjacent will be overlapped each other when the map is loaded firstly or the zoom level is changed. Obviously, if we don't deal with this phenomenon, the display effect of entire topological graph will be affected directly. In addition, with the rapid expansion of the network scale and the increasing number of various network management systems, the topological nodes need to be managed are in the number of a geometric growth too. So the problem which causes loading rate slowly by the large numbers of topological nodes may appear. These problems present a new challenge to the research and development of topology, we need to find a simpler and more effective way

ISSN: 2005-4270 IJDTA

to improve the display effect of the entire topological graph and solve the problem mentioned above.

In order to solve these problems, the paper introduces the hierarchical clustering algorithm and partitioning algorithm, which are two categories more general aggregation algorithm. Based on the advantages of the hierarchical clustering algorithm processing large data sets and a high conformity with the property of real data, we propose an aggregation algorithm for aggregating the topological nodes in GIS. The aggregation algorithm divides the screen into several grids firstly, then it realizes the aggregation of the topological nodes by judging the relation between the numbers of topological nodes in each grid and the predetermined threshold. It is proved by the experiment that this method can reduce the process time of browser greatly, and improve service quality of the entire network topology.

2. Aggregation Algorithm

The aggregation algorithm is to reduce the complexity factor and more convenient to analyze problems, and one class or multiclass objects with similar are aggregated into one class of objects in a particular set of data. The goal of the aggregation algorithm is to make the similarity of the same kind of objects as small as possible and make the similarity of the different kind of objects as large as possible. As the importance and particularity of aggregation algorithm in data processing, there have been many aggregation algorithms in the last few years. According to the different of the basic idea, the aggregation algorithm can be divided into four kinds roughly: the hierarchical clustering algorithm, the partitioning algorithm, the density-based method, the grid-based method and the model-based method [5, 6]. Now hierarchical clustering algorithm and partitioning algorithm is more common aggregation algorithm. The following paragraph will analyze and compare these two kinds of aggregation algorithm simply.

2.1. Hierarchical Clustering Algorithm

The hierarchical clustering algorithm is a simple and effective clustering algorithm which is designed for the large data sets. It uses the method of bottom-up or top-down, groups the objects at different levels, and forms a tree cluster structure. The algorithm only needs to traverse the dataset one time, and it will generate a better clustering result. What's more, the algorithm is developed early and applied widely [7].

Its implementation principle is as follows:

1) Define one object set of N which is needed to cluster. The similar matrix (distance matrix) N * N should be defined.

2) Each object is defined as a class, and then it will get N number of classes. Define for each class only contains one object. The distance between the various classes is the minimum distance between contained objects.

3) Traverse the N number of classes, and get two classes which have the minimum distance between them. The two classes should be merged into a new class.

4) Return to step 3, recalculate the minimum distance between the new class and all classes of unconsolidated, and continue to merge.

5) Repeat step 3 and 5 until all the class are merged into a class, this class will include N number of objects.

Depending on the difference of step 3, the hierarchical clustering algorithm can also be divided into single-linkage method, complete-linkage method, average-linkage method *etc*.

2.2. Partitioning Algorithm

The partitioning algorithm is not suitable for handing the large data sets compared with the hierarchical clustering algorithm, since this algorithm needs to put the data in memory generally. When we use the partitioning algorithm, we need to create an initial partition firstly, and then improve the quality of the partition through using a loop positioning technology and moving the object from a partition to another partition [8]. Its implementation principle is as follows:

1) Define i and j are two pointers, i and j point to the upper bound and lower bound of the unordered data set respectively, k represents the reference object, P[i] and P[j] represent the data object which the pointer i and j refers to.

2) The pointer i traverses the unordered data set from left to right until find the data object X, which is the first object less than the reference object, and then exchanges the position of data object X with P[i].

3) The pointer j traverses the unordered data set from right to left until the data object Y which is the first greater object than the reference object is found. Then the position of data object Y with P[j] should be exchanged.

4) Repeat step 2 and step 3, the pointer i and j continue to traverse the unordered data object until i = k, then it will complete a partition.

2.3. Comparison of the Hierarchical Clustering Algorithm and the Partitioning Algorithm

The hierarchical clustering algorithm is more suitable for handing a large data sets compared with the partitioning algorithm. So the hierarchical clustering algorithm has obvious advantages in the age of big data. The partitioning algorithm only uses a fixed algorithm to cluster, which may not achieve a better clustering result in some time. For example, when the shape of the clustering is irregular or the size of the clustering is different greatly, the algorithm may not be desired. Through the above analysis, we know that the time complexity of hierarchical clustering algorithm and partitioning algorithm are $O(n^2)$ and $O(n\log_2 n)$ respectively, so the partitioning algorithm has a more excellent executive efficiency. However, the hierarchical clustering algorithm is more in line with the characteristics of the real data in the algorithm; it has a better clustering result. In a practical application, the hierarchical clustering algorithm has the obvious advantages.

3. Aggregating the topological nodes in GIS

Now, the network is inseparable from our learning, life and work. Meanwhile, the public network and professional network have made a leap forward. However, the network has expanded rapidly, and the information services business has grown exponentially. The network architecture has become complex increasingly and the network resources are available in a great variety. As such, how to manage the network better is difficult. So, it is very important that how to manage the network and guarantee the network connectivity [9, 10, 11].

The GIS technology is applied to the network management, which can display the entire network topology to users and provide support decision-function for decision-maker. However, when we change the zoom level of the map, it is a critical problem that how to reflect the distributed trend of the topological nodes and the topology relation between each other and realize a large number of topological nodes be aggregated quickly and accurately.

3.1. Research Status

The aggregation algorithm can also be used in GIS. Firstly, we introduce two concepts: map generalization and point cluster. Map generalization refers to using the specified rules for converting a space collection to a new collection, which represents the main features of the space collection. Through map generalization, we can transfer the most valuable spatial information. As a the map generalization method, point cluster aims to solve the problems of the icons covering each other, taking up a lot of system resources and slow data loading, when we need to load a large number of topological nodes. Point cluster make it possible to display the map data distinctly by using a small number of pointers or icons to represent all pointers in the map. Although the common algorithm can be applied in GIS, efficiency is not high. Now, there are many methods of using the algorithm in GIS. For example, grid-based clustering, distance-based clustering, the clustering based on grid and distance, the clustering based on distance and limit the amount of the least numbers of point etc. Although some online maps have provided the method of point cluster (*e.g.*, Google Maps and Baidu Maps), the point cluster is very important for the second development and the map without the method of point cluster (*e.g.*, Map World). If we do not use point cluster, the large numbers of points can't be well reflected. So, it is very important that how to apply the aggregation algorithm in GIS better [12, 13].

In view of the network management, in order to make network management more effective the aggregation algorithm should be applied in GIS. The paper proposes a method, which realizes the aggregation of topological nodes based on GIS in network management. The method through a particular icon identifies the data elements with similarity in the topology, which can solve the problems of the icons covering each other and slow data loading effectively.

3.2. Basic Idea of the Algorithm

In view of the problem of aggregating the topological nodes, this paper proposes an aggregation algorithm based on the advantages of the hierarchical clustering algorithm. The basic idea of the aggregation algorithm is as follows: First, the screen is divided into M * N geographic grids, these grids does not exist in practice, we can know them through the coordinate position of the nodes in the topology and the size of the grid. Then, we calculate the position of each child object in the collection which contains the nodes to be aggregated, and the objects which are the same coordinate position will fall within the same grid. Finally, the number of the child object in each grid needs to be judged in turn. If the number of object exceeds the predetermined threshold, it is considered that this grid is an aggregation point. On the contrary, it is believed that this grid is a non-aggregation point.

At present, there are many kinds of aggregation algorithm in GIS. For example, we can realize aggregation the topological nodes by calculating the relation between the distance of each spatial element and the predetermined threshold. By contrast, we present an algorithm which is simpler to be used. Especially, when we use this algorithm to deal with the topological node which is the large data or the location is changed constantly, it is not necessary for the complex computations. As a result, it will reduce the pressure of the browser, and improve the loading speed.

3.3. Principle of the Algorithm

The implementation principle of this algorithm is shown in Figure 1, the main steps are as follows:

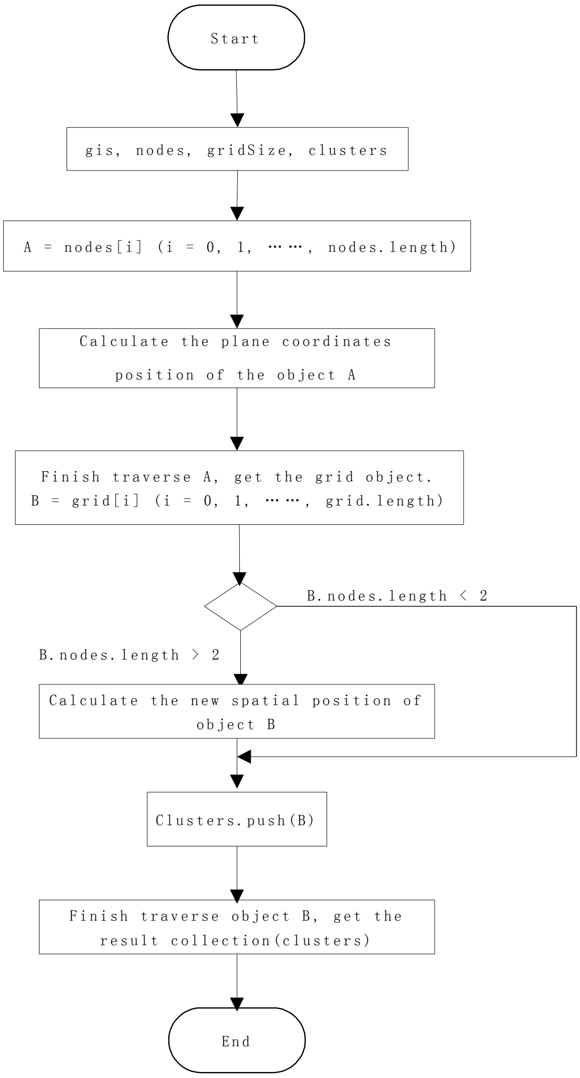


Figure 1. The Flow Chart of Aggregation Algorithm

1) Define the relevant parameters: gis represents a GIS object, nodes represent an element collection which contains the nodes to be aggregated in the topology, gridSize represents the size of each grid, clusters represent the element collection which contains the nodes have be aggregated.

2) Traverse the objects of the collection nodes successively, first A = nodes[i] (i = 0, 1… nodes.length).

3) Calculate the plane coordinate position of the object A. Because the GIS object may be different, so there are many methods of calculating the plane coordinate position of the object A. No matter which method we choose, it does not affect the result of the aggregation algorithm. Calculate the plane coordinate position of the object A refers to converting the spatial data of A to plane data, which means that we should convert the longitude and latitude of A to plane coordinates. Through custom the size of the grid and the plane coordinate of the object A, the coordinate of the grids which contain the

node object A is calculated, which means that we need to calculate the line number and column number of each node. After traversing the nodes object, we get a collection of plane coordinate position (grid).

4) Traverse the grid collection successively and get B = grid[i] (i = 0, 1… grid.length).

5) Determine the number of nodes object included in the object B, if the number is less than 2, the object B as a non-aggregation point added to the cluster set, then return to step 3. If the number is greater than 2, the object B will be considered as an aggregation point.

6) Calculate the new special position of object B. Because object B represents an aggregation point, we need to recalculate the special position of object B and get the more precise location of object B. We can get the new special position of object B through GIS object, the line number and column number of object B and the size of the grid. After getting the new spatial position of the object B, it will be added to the cluster set. Finally the algorithm returns to step 4.

7) After traversing the grid object, we get clusters collection. Then the aggregation algorithm is finished.

8) Show the node objects which has been aggregated through GIS technology.

3.4. Concrete Realization of the Algorithm

In order to describe the algorithm flow chart, the main code is shown as follows:

```
gis.aggregateNodes = function (nodes, gridSize, clusters) {
   var grid = {};
   //Traverse the topological nodes (nodes) and calculate the coordinate of the grids which
   contain //the topological nodes successively.
   for (var i = 0; i < nodes.length; i++)
   {
         var node = nodes[i];
         if(!node)
         {
            continue;
         }
         // only for the nodes which are visible to make calculations.
         if (node.isvisible  || node.isvisible == undefined)
          {
              var point = getXYByGIS(node.longitude, node.latitude);
```



```
            // calculate the line number and column number of each node
             var row = Math.floor(point.y / gridSize);
             var col = Math.floor(point.x / gridSize);
             var prop = row + "," + col;
             if (!grid[prop]){

                grid[prop] =
                {
                   "row" : row,
                   "col" : col,
                   "nodes" : []
                  };
                 }
              grid[prop].nodes.push(node);
         }
      }
// Traverse the grids and determine the grid is a non-aggregation point or an aggregation point.
for (var prop in grid)
   {
    var cluster= grid[prop];
     if (cluster.nodes.length < 2)
    {
          clusters.push(cluster.nodes[0]);
     }
    else
    {
         var newNode= {};
         newNode.children = cluster.nodes;
         // calculate the plane coordinate of the object newNode
         var x = cluster.row * gridSize + gridSize / 2;
         var y = cluster.col * gridSize + gridSize / 2;
         // calculate the spatial coordinate of the object newNode
         var gisLocation = getGISByXY(x, y);
         newNode.longitude = gisLocation.longitude;
         newNode.latitude = gisLocation.lat itude;
         clusters.push(newNode);
        }
     }
   return clusters;
};
// Get the plane coordinate of the topological nodes through the interface provided by TWaver.
gis.getXYByGIS = function(longitude, latitude)
 {
   this.map = new twaver.gis.Map(document.getElementById(domId), "twaverdiv");
   var XY =
   this.map.getScreenPointFromGeoPoint(new        twaver.gis.geometry.GeoCoordinate(latitude,
   longitude));
   return XY;
};
```

The algorithm is written by JavaScript [14-17]. Firstly, traverse the element collection (nodes) which contains the nodes to be aggregated, convert the spatial coordinate of each node object (node) to plane coordinate through the interface provided by TWaver. The plane coordinate (x, y) of node object (node) is included in the object point. Through custom the size of the grid and the plane coordinate of the node object (node), the line number (row) and column number (col) of each node object (node) is calculated. The variable prop which includes the line number (row) and the column number (col) will represent the position of the grid. The attribute of the node object (node) will be added into the grid. The node object (node), which has the same row and col will be distributed in the same grid.

Then, traverse the grids. If the numbers of node object (cluster.nodes) is less than the threshold 2 in a grid, it is believed that the grid is a non-aggregation point. The node object and its attribute data will be added to the element collection (clusters) which contains the nodes have been aggregated. If the number of node object is greater than the threshold 2, then the grid is an aggregation point. Redefine a new node object (newNode) which represents all node objects in the grid. Define an object (newNode.children) which points to attribute data of all nodes. Through the line number, column number and size of the grid, the plane coordinate (x, y) of the object newNode is calculated, and then the spatial coordinate (longitude, latitude) of the object newNode by the gis object is also calculated. Finally, add the object newNode to the clusters collection.

Finally, return the clusters collection back to other programs and continue to execute.

3.5. Advantages of the Algorithm

When the map is loaded firstly or the zoom level is changed, the topological nodes of geographical location adjacent will be overlapped each other and cause load the data slowly. This algorithm resolve these problems mentioned above. The algorithm has the following advantages:

1) Simplicity. The implementation of the algorithm is simple and easy to understand, and it does not require complex mathematical or logical operations.

2) Language independence. The algorithm does not use the features of any kind of programming language, so we can use all programming language to implement it. It can be used in the client, through the browser realizes the aggregation of the topological nodes. Meanwhile, it also can be applied to the server.

3) High efficiency. Because the algorithm is simple and efficient, so it has obvious advantages in the age of big data. It is suitable for the demand of network management.

4) Accuracy. The algorithm is suitable for processing the topological node which is the large data or the location is changed constantly. As such, if the algorithm is too complex, the lag time between the topological nodes and network management system will increase, and the location information showed in the topology may be different from the actual location of the topological nodes.

In a word, the algorithm will further improve service quality of the entire network topology, and it also provides a good technical support for realizing manage the network accurately, visually. It is more convenience to manage the network resources scientific. The aggregation algorithm plays a key role in promoting manage the network resources.

4. Experiment

In order to illustrate the necessity of using aggregation algorithm for displaying the topology relation in the network management, and explain the feasibility of the aggregation algorithm presented in this paper. The experiment is shown as follows. The experimental hardware environment is a computer with 2G memory, 500G hard drive and Intel processor. The software environment is SuperMap iServer Java 6R, MyEclipse and Google browser [18-21].

4.1. Compare the Time of the Map Load the Different Numbers of Topological Node, when we do not Use any Aggregation Algorithm and Use this Aggregation Algorithm

The experimental data uses the nodes which are selected the longitude from 73 to123, and the latitude from 20 to 50 randomly. The map's zoom level is 4. Through 50 times experiment, we get the time of the map loads a different numbers of topological nodes, when we do not use any aggregation algorithm and use this aggregation. We also get the time needed for this aggregation algorithm and the time required to render the topological nodes, which are showed in Table 1 and Table 2. As shown in Table 1 and Table 2, this aggregation algorithm will reduce the time to load topological nodes and improve the loading speed greatly.

In addition, if you use a different GIS technology, the time of rendering the topological nodes may be different. So we list the Table 1 which illustrates the advantage and necessity of using the aggregation algorithm in the topology. If we use other GIS technology, the time may be shorter.

Table 1. Time Comparison Table

Node number (a)	The time of don't use any algorithm (ms)	The time of using this algorithm (ms)
1000	319.86	280.14
2000	452.66	346.86
3000	718.54	370.38
4000	892.42	389.04
5000	1052.22	399.92

Table 2. Time Table of this Algorithm and Render Nodes

Node number (a)	The time of this algorithm (ms)	The time of rendering nodes (ms)
1000	17.54	262.60
2000	20.3	326.56
3000	27.1	343.28
4000	28.34	360.70
5000	39.78	360.14

4.2. Effect of Before Aggregation and after Aggregation with this Aggregation Algorithm

The experimental data is the nodes which are selected randomly. In order to show the experimental results clearly, we use two different icons as the aggregation point and the non-aggregation point in the topology. The blue icon represents a non-aggregation point, and the red icon represents an aggregation point. We get the effect graphs of before aggregation and after aggregation with this aggregation algorithm. The experiment results are shown in Figure 2 and Figure 3.

The Figure 2 shows the effect of not using the aggregation of the topological nodes with a zoom level of the map. The Figure 3 shows the effect of topological nodes whose location are close take place aggregation, when we reduce the map's zoom level. Through compare the two pictures, it can be seen that the topological nodes of location nearly are aggregated when change the map's zoom level.

Figure 2. Effect of Not Using the Aggregation

Figure 3. Effect of using Aggregation

Through the experiment, it is proved that this aggregation algorithm can reduce the numbers of the topological node needs to be rendered effectively, when the map first loads the data and we change the map’s zoom level. Meanwhile, the aggregation algorithm improves the service quality and level of the entire network topology, solves the problems of the icon covering each other and slow data loading effectively, and provides a good technical support for the research and development of topology.

5. Conclusion

The paper presents an aggregation algorithm for aggregating the topological nodes in GIS based on the hierarchical clustering algorithm. The algorithm solves the problems which cause the icons covering each other and slow data loading by applying GIS technology to load a large number of topological nodes effectively. The algorithm has the merits of simplicity, language independence, high efficiency, accuracy *etc*. It provides a solution for aggregating the topological nodes effectively. However, the algorithm has also the following deficiencies:

1) When the zoom level of the map is changed, all the topological nodes on the map will execute the aggregation algorithm. The algorithm should execute the algorithm only for the topological nodes which are displayed on the screen.

2) We focus on the relationship between the size of the grid and the size of the node icon, and achieve a better aggregation result in the future.

We will continue to study the aggregation algorithm and make the algorithm more available.

Acknowledgments

The authors would like to thank the editor and referees for their careful review and valuable critical comments. This work was supported by the Science Fund of Hebei of China No. 2011228 and the Initial fund for doctors of Hebei University of Science and Technology No.QD201223.

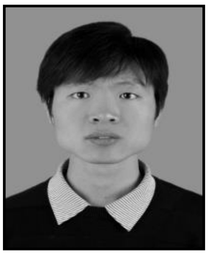
References

[1] M. Gao, "GIS-Geographic Information System", China Science & Technology Panorama, vol. 2, no. 239, **(2012)**.
[2] G. Zhao and Y. Chao, "Geographic Information System Analysis and Applications", Publishing House of Electronics Industry, Beijing, **(2010)**.
[3] N. Schuurman, "Critical GIS", International Encyclopedia of Human Geography, **(2009)**.
[4] M. Wang, "Design and Realization of Management and Decision-making System for Military Communication Network Pipeline Resource Based on GIS", Northeastern University, **(2008)**.
[5] H. Li, "The Research and Realization of Spatial Clustering Algorithm in GIS", Guizhou University, **(2008)**.
[6] P. Lv, "Research and Application of a Generic Aggregation", Computer Technology and Development, vol. 17, no. 219, **(2007)**.
[7] E. Lv, "Application of Clustering Algorithm in the Power GIS", Electric Power IT., vol. 10, no. 37, **(2012)**.
[8] W. Wang and H. Liu, "A GIS Spatial Data Oriented Clustering Approach", Computer Simulation, vol. 24, no. 66, **(2007)**.
[9] H. Xiong, "Research of mobile communication network quality monitoring and analysis system Based on GIS, Jiangxi University of Science and Technology, **(2012)**.
[10] J. Xu, "Study on a Military Communication Network Resource Management Information System Based on GIS", Chongqing University, **(2006)**.
[11] D. Wang, "The Research and Realization of GIS-based Military Information Network Resource", North China Electric Power University, **(2011)**.
[12] Habibullayevich, G. Kholmuradov, X. Chen and H. Shin, "Efficient Filtering and Clustering Mechanism for Google Maps", Journal of Advanced Management Science, vol. 1, no. 1, **(2013)**.
[13] Q. Guo, Y. Huang, C. Zheng and Y. Cai, "Spatial reasoning and incremental map generalization", Wuhan University Press, Wuhan, **(2007)**.
[14] J. Zhang, "Design and realization of spatial data publish system based on WebGIS", Hunan University, **(2006)**.
[15] S. Dewen, "Design and implementation of GIS course based on SuperMap", The 2012 IEEE Symposium on Robotics and Applications, (2012), June 552 – 554, Kuala Lumpur, Malaysia.
[16] D. Flanagan, "JavaScript: the definitive guide", O'Reilly Media, Inc., **(2002)**.
[17] N. C. Zakas, "Professional javascript for web developers", John Wiley & Sons, **(2011)**.
[18] L. Xu, "Twaver Technology in Telecommunication Operation Support System of the Application", Software Guide, vol. 11, no. 71, **(2012)**.
[19] Z. Zhang, H. Hu and G. Zhong, "SuperMap GIS Application and Development Tutorial", Wuhan University Press, Wuhan, **(2006)**.
[20] http://www.supermap.com.cn.
[21] http://twaver.servasoft.com.

Authors

Min Huang, Associate Professor, born in 1979. He received the B.Eng. degree in automatic control from Hebei University of Science and Technology, China, in 2000. He received the M.Sc. degree in computer science from Beijing Institute of Technology, China, in 2003. His main research interests include network and communication, system modeling and identification, image processing and distributed computing.

Likun Zhu, born in 1979. She received the B.Eng. degree in automatic control from Hebei University of Science and Technology, China, in 2002. She received the M.Sc. degree in automatic control from Huazhong University of Science & Technology, China, in 2009. Her main research interests include network and communication, system modeling and automatic control.



Pengfei Liu, born in 1988, He received the B.Eng. degree in computer science and technology from Hebei University of Science and Technology, China, in 2012. Now, he is a postgraduate of computer technology from Hebei University of Science and Technology, China. His main research interests include GIS technology, network and communication, and network management.

Jingyang Wang, Professor, born in 1971. He received the B.Eng. degree in computer software from Lanzhou University, China, in 1995. He received the M.Sc. degree in software engineering from Beijing University of Technology, China, in 2007. Now he is working in School of Information Science and Engineering, Hebei University of Science and Technology, Shijiazhuang Hebei, China. His main research areas include transmition control, image processing and distributed computing.

Liwei Guo, Professor, born in 1956. He received the M.Sc. degree in automatic control from Harbin University of Science and Technology, China, in 1988. His main research interests include network and communication, system modeling and automatic control.

