



Journal of  
**Software  
Engineering**

ISSN 1819-4311



Academic  
Journals Inc.

[www.academicjournals.com](http://www.academicjournals.com)



## Research Article

# Vector Curve Data Rapid Transmission over Internet Based on the Controlled Fractal Interpolation Model

Jiang Baode and Chen Zhanlong

National Engineering Research Center of Geographic Information System, China University of Geosciences, 430074 Wuhan, China

## Abstract

**Background:** With the dramatic increases in mobile GIS applications and limited network bandwidth, rapid transmission of vector map data is urgently need. The existing methods always use the progressive transmission to improve the response time of client side, but the total amount of transmission data is not reduced. **Objective:** To reduce the transmission data amount and improve the transmission performance, this study proposes a vector curve data rapid transmission method based on the controlled fractal interpolation model. **Materials and Methods:** First, on the server side, the vector curve data set is encoded into a bending data structure stream for rapid transmission over the internet. Then on the client side, to avoid transferring different resolution map from the server side as the users need and save network flow, a new reconstruction method based on the controlled fractal interpolation model is presented to decode the received data and derive different resolution curve map. **Results:** The experiment results demonstrate that this method can decrease the transmission data amount of vector curve map by encoding the curve data before transmission and moving the multi-scale representation from the server side to the client side. **Conclusion:** This study can decrease the data amount of vector curve map significantly and improve the transmission performance over the internet.

**Key words:** Internet transmission, fractal interpolation, controlled, vector curve data

**Received:** July 28, 2016

**Accepted:** September 12, 2016

**Published:** March 15, 2017

**Citation:** Jiang Baode and Chen Zhanlong, 2017. Vector curve data rapid transmission over internet based on the controlled fractal interpolation model. *J. Software Eng.*, 11: 202-209.

**Corresponding Author:** Chen Zhanlong, National Engineering Research Center of Geographic Information Systems, China University of Geosciences, 430074 Wuhan, China

**Copyright:** © 2017 Jiang Baode and Chen Zhanlong. This is an open access article distributed under the terms of the creative commons attribution License, which permits unrestricted use, distribution and reproduction in any medium, provided the original author and source are credited.

**Competing Interest:** The authors have declared that no competing interest exists.

**Data Availability:** All relevant data are within the paper and its supporting information files.

## INTRODUCTION

With the popularity of rich internet applications, web map will show its strong function at the client side as much as possible<sup>1</sup>. Users on the client can not only design the map display according to their needs and interests, but also do data analysis directly. The precondition of these applications is to ensure the rapid transmission of map vector data over the Internet with the limited network bandwidth<sup>2,3</sup>.

Due to the huge vector map data, it cannot wait the all detailed map data being transferred completely from the server side to the client side until to begin displaying the map, which will cause a bad user experience. The traditional method is to take the technology of vector data Internet progressive transmission<sup>4</sup>. That is first to transfer a coarser temporary version data from the server side to the client side and to display it immediately while the client side has received it, then, with the continuation of transmission, the higher resolution temporary map data is transferred to the client continually and the resolution of the map will be higher and higher, while more details are displayed at the client side. The reason for doing this is that on one hand, it can avoid the user to wait for a long time to see the contents of the map, which can improve the response time of client side. On the other hand, users need different detailed map, the data transmission can be suspended at any time by the users according to their needs. The whole transmission procedure consists of 3 phases<sup>1,4,5</sup>: (1) On the server side, preprocess the map data to establish a hierarchical model for data progressive transmission and organize the hierarchical map data into a linear data stream, (2) Transfer the map data stream in the network and (3) Reconstruct the map elements and display the map on the client side. The traditional methods are to construct a mathematical model for multi-scale representation of vector map on the server side in the first step, which can derive different levels of detailed map from the same data source. There are 2 typical types of multi-scale representation models for the server side. One is a multi-version model of which the initial version data is preprocessed into several coarse versions of different level resolution maps on the server side. Such as the progressive transmission model proposed by Bertolotto and Egenhofer<sup>4</sup>, the vertex decimation multi-scale model proposed by Yang *et al.*<sup>6</sup> and the geometric shapes similarity multi-scale model proposed by Tang *et al.*<sup>3</sup> are of this kind. The drawback of this model is that the map data are preprocessed into several different resolution map versions on the server side, which are quite redundant. What is more, the different resolution map data may need all to be transferred to the

client, which lead to increase the burden of network transmission and spend a lot of network flow. Another is an incremental multi-scale model of which the initial version data is preprocessed into a coarser version data and the incremental data that formed by the difference between two consecutive representations. Such as the accumulation model proposed by Ai *et al.*<sup>1</sup> the river network progressive transmission method proposed by Ai *et al.*<sup>5</sup> and the model based on Fourier-descriptor proposed by Liu *et al.*<sup>7</sup> are of this kind. The drawback of this model is that the incremental data of different representations are also need to be transferred to the client progressively and the total amount of transmission data is not less than the initial version data. As a result, it still does not reduce the network flow.

In view of the above-mentioned problems, this study presents a rapid transmission method for vector curve data over the Internet based on the controlled fractal interpolation model aimed to reduce the transmission data amount. It first extracts the bending structure of the curve and represents it by a bending data structure stream on the server side, which can reduce the transmission data amount of vector curve. Then, transfer the bending data structure stream to the client side in proper order and reconstruct it into different scale curve map that the user needs base on the controlled fractal interpolation model. Different from the existing methods, the multi-scale representation of initial data is implemented on client side, which can avoid transferring different resolution version map data from the server side to the client side and save a lot of network bandwidth. The results demonstrate that this method can decrease the transmission vector curve data amount and improve the transmission performance significantly.

## MATERIALS AND METHODS

**Pre-processing curve on server side:** To decrease the amount of vector curve data on the server side, an available method is to take vector data compression technology. There are several compression methods for vector curve data, such as the well-known Douglas-Peucker algorithm<sup>8</sup>, Li-Openshaw algorithm<sup>9</sup> and etc. The critical issue of compression is how to maintain the geographical characteristics information of curve while doing simplification and easily reconstruct the curve from the compressed information on the client side. Because a curve on the map is not just a simple geometric line, it is an abstract expression of natural geographical feature, so it can represent the geographical characteristic of real world by its bending shape<sup>10</sup>. Bending is the basic characteristic of geographical curve, the geographical characteristic of curve is

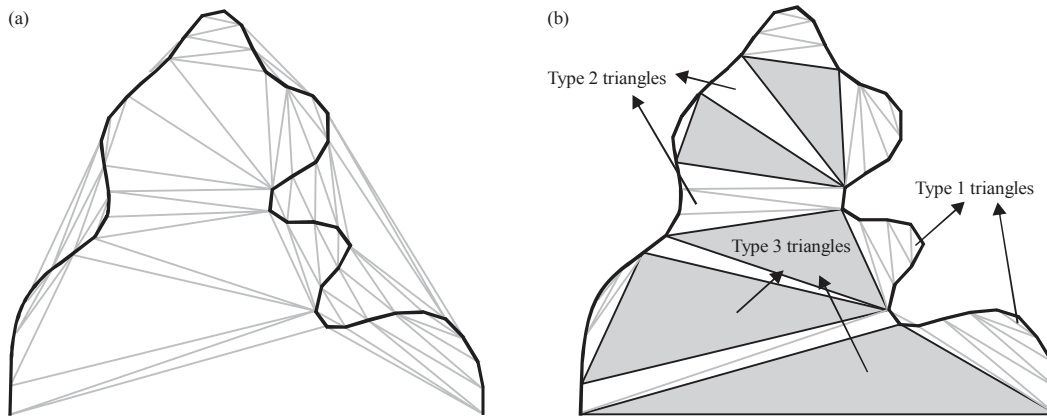


Fig. 1 (a-b): Pretreatment of curve data on the server side, (a) Construct CDT for curve and (b) Classify the triangles into three types to extract the bending structure of curve

implicated in its bending structure. Take coastline for example, the bending structure and bending degree of coastline can be used to represent the landform of coast, the bending on the land side of shoreline indicates the shoal head or peninsula etc. and the bending on the coastal sea side of shoreline indicates the bay. In order to keep the geographical characteristic of curve for compression and reconstruct the curve easily on the client side, the following will first introduce a method to extract the bending structure of curve and then, encode it by a bending data structure for the data stream transmission, at last, the data stream is decoded and reconstructed by a controlled fractal interpolation method.

**Extracting bending structure of curve:** The methods that extract the bending structure of curve have been researched a lot. The following will review a method based on Constrained Delaunay Triangulation (CDT)<sup>11</sup> which can extract the bending structure of curve robustly and effectively. The first step of this method is to construct CDT for the curve. Because the computation of CDT is well-studied in computational geometry and the computational aspect of the CDT algorithm is not the major focus of this study, so the algorithm of CDT will not be described in detail and this study will only give its result.

As seen in Fig. 1a, the CDT is constructed for a curve of coastline and there are many triangles on 2 sides of the curve. For convenience of analysis, only keep one side triangles of the curve and delete the other side triangles (Fig. 1b). Then, classify the remainder triangles into 3 types as following:

- Type 1 : The triangle that only has one adjacent triangle linked to it
- Type 2 : The triangle that has two adjacent triangles linked to it
- Type 3 : The triangle that has three adjacent triangles linked to it

Triangles of type 1 indicate the bottom parts of the curve, triangles of type 2 indicate the boundaries of the curve and triangles of type 3 indicate the branches of the curve. The method extracting the bending structure of curve is to search the all triangles of different types in order and record them by a binary tree data structure. The leaf nodes of the binary tree are the triangles of type 1 and the non-leaf nodes of the binary tree are the triangles of type 3. Triangles of type 2 do not create nodes in the binary tree. The leaf nodes of the binary tree correspond to atomic bending and the other nodes correspond to complex bending. The hierarchical relationships of the tree nodes describe the structure of the nested bending. Figure 2a shows the process of searching the bending structure of coastline and its corresponding binary tree representation is shown in Fig. 2b. More details about the algorithm can be found in the study<sup>11</sup>. When the bending structure of curve is extracted, the leaf nodes of the binary tree can form a bending unit set and the curve can be divided into several characteristic subsections by this bending unit set. For example, the coastline in Fig. 2a is extracted 6 bending units  $\{C_{2r}, C_{4r}, C_{6r}, C_{8r}, C_{10r}, C_{11}\}$  and the whole curve is divided into 11 characteristic subsections by the bending unit set, that is  $\{L_{1,2r}, L_{2,3r}, L_{3,4r}, L_{4,5r}, L_{5,6r}, L_{6,7r}, L_{7,8r}, L_{8,9r}, L_{9,10r}, L_{10,11r}, L_{11,12}\}$ .

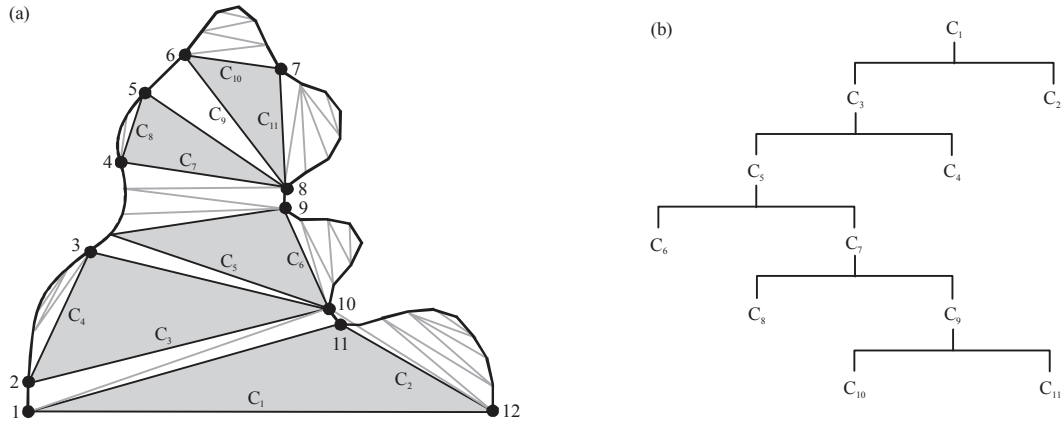


Fig. 2(a-b): Search the bending structure of coastline and with its binary tree representation, (a) Search the building structure and divide the curve and (b) Binary tree representation of bending structure

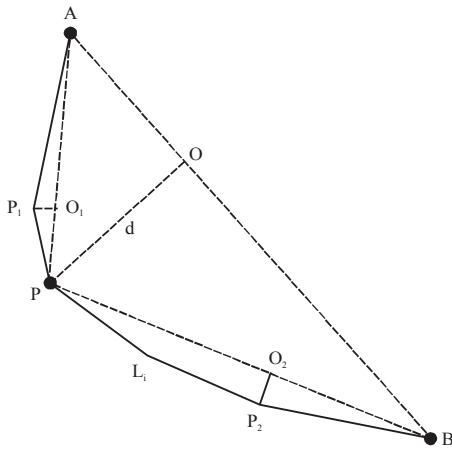


Fig. 3: Description of bending unit

**Encoding bending data structure:** In order to decrease the transmission data amount and facilitate the vector curve data transmission, each curve is represented by a divided bending unit set and each bending unit is encoded based on its bending characteristic. To define a bending data structure, take a bending unit as an example. As seen in Fig. 3, assume  $L_i$  is the bending unit, A and B are the endpoints of  $L_i$ , P is the most distant point on the bending curve to the line segment AB and O is the foot of perpendicular of line PO to line segment AB. Line segment AB is called the chord of bending unit and the distance of PO is called the bending depth. Thus, a bending unit can be described by its chord and bending depth.

In order to facilitate the fractal interpolation for reconstructing the curve on the client side, the bending data structure is defined as following.

```

Structure Bending Structure{Point A; // one of the endpoints of bending unit
Point B; // another endpoint of bending unit
Point P; // the most distant point on the curve to the line segment AB
double k; // deviation ratio of point displacement
int n; // number of coordinate points of bending unit}
(Pseudo code data structure of the bending unit)
    
```

The data structure not only encompasses the bending unit description parameters, such as endpoints of bending unit (A and B) and the most distant point on the curve to its chord (P), but also includes the fractal parameters, such as the deviation ratio of point displacement (k) and the number of coordinate points of bending unit (n). The chord length and bending depth of the bending unit can be calculated out by the distance of point A to point B and point P to point O, respectively. Parameter k is used for controlling the fractal interpolation offset of point displacement of bending unit, its value is set by the ratio of distance  $P_1O_1$  (when distance  $P_1O_1$  is longer than distance  $P_2O_2$ ) or  $P_2O_2$  (when distance  $P_2O_2$  is larger than distance  $P_1O_1$ ) to distance PO ( $P_1O_1$  and  $P_2O_2$  is the bending depth of arc  $AP_1P$  and arc  $PP_2B$ ). Parameter n is used for controlling the amount of fractal interpolation points and its value is set by the number of coordinate points of bending unit except the two endpoints. For example, if the number of coordinate points of bending unit is m, then,  $n = m - 2$ .

Expecting the bending units, the remanent characteristic subsections between 2 adjacent bending units can also be represented by this data structure. If the remanent parts are line segments, the parameter n and k is set 0. In this case, this bending unit is not need to do interpolation. Otherwise, the remanent parts are treated as bending units.

**Transmitting the curves:** Vector curve data transmission is initiated by client-side request. The server sends packets to the client. Each packet contains all parts of curve divided by bending units in a single row, insuring delivery of a complete representation of the vector feature. Each part is encoded by the bending data structure. Only 3 coordinate points and a double type parameter with an int type parameter is need to be transferred for each bending unit part, this insures efficient transmission times.

**Reconstruction of curve on the client side:** When the client has received the bending data packets from the server side, the bending data structure can be decoded and derived into multi-scale representation of vector curve map by fractal interpolation based on controlled one-dimensional random midpoint displacement method.

**One-dimensional random midpoint displacement method:** One-Dimensional Random Midpoint Displacement (RMD) method is a classic fractal interpolation method for geometric line object. It can generate a self-similar fractal curve by moving the middle point of line segment up or down by a random amount iteratively. The general interpolation equation of one-dimensional RMD is:

$$P_{mid} = \frac{P_i + P_{i+1}}{2} + \Delta_n \quad (1)$$

where,  $P_i$  and  $P_{i+1}$  are the two endpoints of a line segment and  $\Delta_n$  is the offset value of the midpoint displacement. The value of  $\Delta_n$  is relate to the self-similar parameter  $H$ , direction controlling parameter  $\tau$ , offset controlling parameter  $\delta$  and the number of iterations  $n$ , the relationship is as follows:

$$\Delta_n = \left(\frac{1}{2}\right)^{nH} \tau\delta \quad (2)$$

where, parameter  $H$  is used to control the smoothness of curve, the greater the  $H$  is, the more smooth the fractal curve is, otherwise, the smaller the  $H$  is, the more rough the fractal curve is. Parameter  $\delta$  can be set by a fixed value or replaced by a Gaussian function.

Though the one-dimensional RMD method can simulate the fractal characteristic of curve effectively, it takes the curve as a simple geometric line object and cannot keep the geographical characteristic of curve after fractal interpolation. So, it needs to control the procedure of fractal interpolation.

**Controlling fractal interpolation:** From the Eq. 1 and 2, it can be seen that the shape pattern of curve created by one-dimensional RMD method is controlled by the parameters of displacement seed point, displacement direction, coordinate offset, self-similar parameter and the number of iteration  $n$ . These parameters can be restricted by the constraints of each bending unit characteristics. The process of controlling fractal interpolation is illustrated as following.

First, control the displacement seed point. The traditional one-dimensional RMD method always takes the midpoint of the line segment as the displacement seed point for each segment of line, which leads the result shape of interpolation to be staid. The improved method is to select the definite proportion and division point of each line segment as the displacement seed point and the proportion is determined by each bending unit, for example, in Fig. 3, the proportion is equal to the ratio of distance AO to distance OB. Thus the displacement seed point can be automatically adjusted according to the characteristic of each bending unit and it makes the result of interpolation to be more self-similar.

Second, control the parameter of displacement direction. Because the displacement direction of traditional one-dimensional RMD is random up or down in the y direction. So, it cannot reflect the variation of geographical bending direction of curve and leads the result of interpolation to be self-intersection easily. The improved displacement direction for each line segment of bending unit is to select the direction that is perpendicular to the line segment and opposite to the opening direction of bending unit.

Third, control the parameter of coordinate offset. The coordinate offset parameter of one-dimensional RMD method is a fixed value or a random value, which leads the coordinates of fractal interpolation to be uncontrolled and does not satisfy the requirements of spatial data quality of GIS. The improved coordinate offset parameter is set by the bending depth of bending unit, it can be calculated out through the parameters of A, B and P in the bending data structure that transferred from the server side.

Fourth, control the self-similar parameter  $H$ . From the Eq. 2, it can be seen that the deviation ratio  $k$  that transferred from the server side is equal to  $(1/2)^H$ , so the value of  $H$  can be set- $\log_2(k)$ .

At last, control the parameter of fractal interpolation iteration number  $n$ . The parameter  $n$  determines the precision of interpolation. The bigger the  $n$  is, the higher the fractal precision is and the corresponding map scale is larger. The parameter  $n$  can be deduced by the number of coordinate

points of bending unit that transferred from the server side. If the number of coordinate points of bending unit is  $m$ , there are left  $m-2$  coordinate points to be interpolated. On the other hand, if a line segment has done  $n$  times fractal interpolation, it creates  $(2^n-1)$  coordinate points, so, there is:

$$2^n - 1 = m - 2 \quad (3)$$

Because the iteration number is an integer, so:

$$N = \lceil \log_2 (m-1) \rceil \quad (4)$$

By this point, the key fractal parameters of one-dimensional RMD have been constrained and each bending unit can be reconstructed into variable target scale by this controlled one-dimensional RMD fractal interpolation method doing different iteration of fractal interpolation progressively. When each characteristic subsection unit has been finished interpolation, link these interpolation results together in proper order and the different resolution approximate curve map can be got.

## RESULTS

To test the transmission performance of vector curve data over the Internet, a set of vector curve data of elevation lines with an accuracy of 16 bytes per vertex is taken for internet transmission experiments. It is supplied by the National Data Center of China and grouped in 1:100 k scale map sheets and the experiments are undertaken based on the client/server architecture.

When the server side accepts a request from the client side, the bending structure of requested curves are extracted and encoded with the bending data structure on the server side prior to transmission. Then, the server sends the packets to the client and each packet contains the all divided parts bending data structure of a curve in a single row. When the encoded bending data structure packets reach the client side one by one, the reconstruction operator based on controlled one-dimensional RMD fractal interpolation model will response to construct a coarse resolution vector curve map on the client side first and the user can freely manipulate the vector map data. Then, after the client side has received the all encoded curve data and finished the first round of reconstruction operation, another higher resolution curve map begins to be built by the controlled fractal interpolation method on the client side and it will recover the first low resolution map. Until the map resolution is equal to the original map or the user cancels the operation, the fractal interpolation reconstruction operation will be terminated.

Table 1: Transmission data amount before and after encoded vector curve data

Parameters	Vertices number (N)	Data amount (Bytes)	Compression ratio (%)
Initial curve	3825	61200	No compression
Encoded curve	535	23512	61.58

Figure 4 shows the reconstruction procedure of fractal interpolation. Figure 4a is the initial set of vector elevation lines, Fig. 4b-d are 3 different resolution curve map data results of first to third round of fractal interpolation reconstruction, the corresponding fractal interpolation iteration number  $n$  is 1, 2 and 3, respectively. In map visualization's view, it is clear that with the increase of the number of fractal interpolation iteration, the resolution of vector curve map data becomes higher and higher from Fig. 4b-d and the first coarse version still maintain the key geographical characteristics of the initial curve data. Moreover, there are no self-intersections in the multi-resolution results of vector curve data. In transmission performance's view, for a fixed network bandwidth, the smaller the amount of vector map data is, the better transmission performance of this method has. So, the transmission data amount is compared with that of the original data version. Table 1 lists the data amount of original version data and the encoded transmission data. The original version data have 3825 coordinate points to be transferred and each point needs 2 double type parameters to store and occupy 16 bytes. So, the original version data amount is 61200 bytes. For encoded curve, after the bending structure of curves being extracted and encoded, it only needs to transfer 535 coordinate points, plus the fractal interpolation parameters of the bending data structure, the encoded transmission data amount is 23512 bytes and the compression ratio reaches 61.58% (the compression ratio equals  $(M-M')/M$ ,  $M'$  is the data amount of encoded transmission data and  $M$  is the data amount of original data). It is clear that the encoded transmission data has smaller amount of vector data than the original version and it dramatically decreases the data amount of transmission. In addition, the bigger the bending is, the higher the compression ratio is. As less data amount over the internet requires less network bandwidth and transmission time. Therefore, this vector curve data rapid transmission method based on the controlled fractal interpolation model can significantly improve the transmission performance.

The experimental results demonstrate that this vector curve data rapid transmission method has several advantages: (1) Decrease the transmission data amount of vector curve data for the internet delivery significantly, (2) Implement the multi-resolution model of vector curve data on the client side with great flexibility and (3) Maintain the geographical characteristic of vector curve data in the multi-resolution

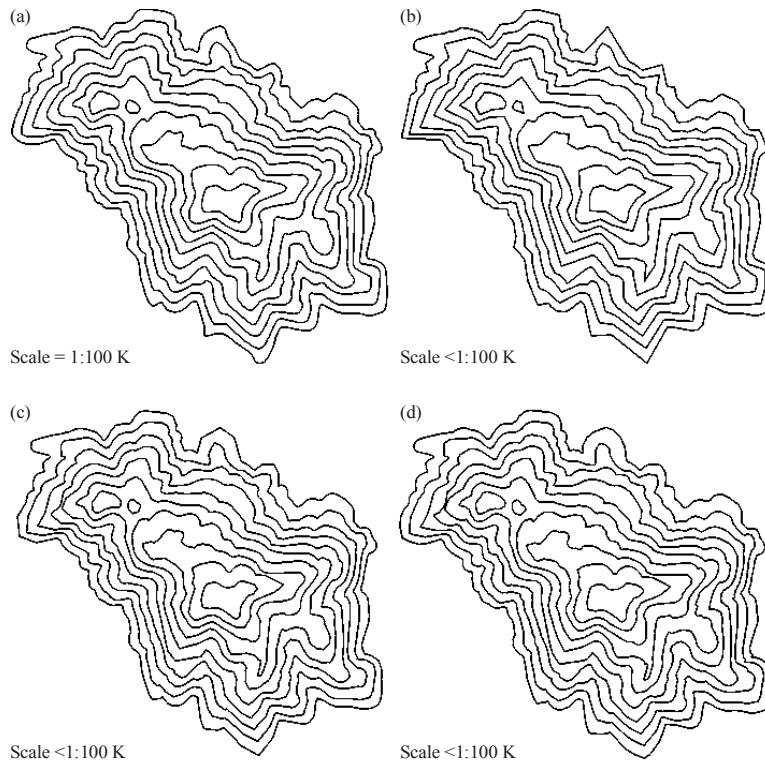


Fig. 4(a-d): Test set of vector elevation data and fractal interpolation results, (a) Initial vector curve map data set, (b) Results of first round of fractal interpolation reconstruction, iteration number = 1, (c) Results of second round of fractal interpolation reconstruction, iteration number  $n = 2$  and (d) Results of third round of fractal interpolation reconstruction, iteration number  $n = 3$

model on the client side. Therefore, the vector curve data rapid transmission method is viable for the rapid transmission of vector curve map data over the internet when the network bandwidth is much narrow and data amount is heavy.

## DISCUSSION

The rapid transmission of vector map data over the internet is a bottleneck of spatial data delivery and visualization in web-based environment because of the increasing data amount and limited network bandwidth. In order to improve the transmission performance of vector map data over the internet, this study presents a vector curve data rapid transmission method based on the controlled fractal interpolation model.

Compared with the other methods discussed previously, the progressive transmission based on geometric shapes similarity proposed Tang *et al.*<sup>3</sup> and vertex decimation method proposed by Yang *et al.*<sup>6</sup> construct the multiple-resolution representations of vector map data on the server side, which need to transfer the different resolution map data to the client side and lead to increase the spends of network flow for the wireless mobile devices. While this study only needs to

transfer the encoded bending data structure set of vector curve data to the client side only once, it can save a lot of network flow and improve the transmission performance of vector curve map data. The incremental multi-scale model proposed by Ai *et al.*<sup>1</sup> and Fourier-descriptor model proposed by Liu *et al.*<sup>7</sup> although only transferring a coarser version data and its incremental data that formed by the difference between 2 consecutive representations, the amount of total transmission data is still much larger than this method.

The most contribution of this study is: (1) Define a bending data structure for condensed representation of vector curve, which can reduce the data amount of curve map before transmission, (2) Propose a new method of multi-resolution vector curve map reconstruction based on the controlled fractal interpolation, which is no longer need to transfer different resolution map or incremental data from the server side to the client side and save a lot of network bandwidth and (3) The vector curve set is transferred through the encoded bending data structure stream, but not the coarse version map data, which is not mentioned in the previous proposed methods, it can compress the initial curve data a lot and decrease the total amount of transmission data dramatically.



The proposed rapid transmission method of this study can be used for the vector curve data transmission in the mobile GIS environments, which can save a lot of network flow. Researchers who are focusing on vector spatial data transmission, delivery and visualization in web-based environment may be interested in this study.

### **CONCLUSION**

The rapid transmission method for vector curve map data based on the controlled fractal interpolation model has been analyzed and discussed. It aims to fulfill the multiple-resolution representations of vector curve map data on the client side and makes full use of the powerful computation performance of the client. The model first extracts the bending structure characteristic of curve by CDT and then divides the curve into several parts and encodes them with a fractal interpolation bending data structure at last, transfer the bending data structure to the client side and reconstruct them by the controlled fractal interpolation method. It can derive different multiple-resolution representations from the bending data structure on the client side. The results show that it can compress the initial curve data a lot and decrease the total amount of transmission data dramatically. So, it can save a lot of network bandwidth and significantly improve the transmission performance of vector curve data over the Internet.

### **SIGNIFICANCE STATEMENT**

- The increasing mobile GIS applications and limited network bandwidth calls for a rapid transmission method for vector map data
- To reduce the transmission data amount, a new bending data structure is defined to compress the curve data on the server side
- To avoid transferring different resolution version map data from server side to client side and save network bandwidth, a new method of multi-resolution vector curve map reconstruction is presented based on the controlled fractal interpolation
- The results show that this study can decrease the amount of transmission vector curve data significantly

### **ACKNOWLEDGMENT**

This study is partially supported by research funds for the Central Universities Basic Special Projects (No. CUGL150823) and project funded by China Postdoctoral Science Foundation (No. 2015M572223) and funded by State Key Laboratory of Geo-information Engineering (No. SKLGIE2014-M-4-3) and National Natural Science Foundation of China (41401443).

### **REFERENCES**

1. Ai, T.H., Z.L. Li, Y.L. Liu and Y. Zhou, 2009. The changes accumulation model for streaming map data transferring over web. *Acta Geodaetica et Cartographica Sinica*, 38: 514-519.
2. Yang, B., 2005. A multi-resolution model of vector map data for rapid transmission over the Internet. *Comput. Geosci.*, 31: 569-578.
3. Tang, L., X. Zhang, Z. Kan, B. Yang and Q. Li, 2014. Spatial data Internet progressive transmission control based on the geometric shapes similarity. *Int. J. Control Automation Syst.*, 12: 1110-1117.
4. Bertolotto, M. and M.J. Egenhofer, 2001. Progressive transmission of vector map data over the world wide web. *Geoinformatica*, 5: 345-373.
5. Ai, B., T.H. Ai and X.M. Tang, 2010. Progressive transmission of river network. *Geomatics Inform. Sci. Wuhan Univ.*, 35: 51-54.
6. Yang, B., R. Purves and R. Weibel, 2007. Efficient transmission of vector data over the Internet. *Int. J. Geographical Inform. Sci.*, 21: 215-237.
7. Liu, P.C., X. Li, W.B. Liu and T.H. Ai, 2015. Fourier-based multi-scale representation and progressive transmission of cartographic curves on the internet. *Cartography Geographic Inform. Sci.*, (In Press). 10.1080/15230406.2015.1088799.
8. Douglas, D.H. and T.K. Peucker, 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Can. Cartographer*, 10: 112-122.
9. Li, Z. and S. Openshaw, 1993. A natural principle for the objective generalization of digital maps. *Cartography Geographic Inform. Syst.*, 20: 19-29.
10. Huang, Y.F., T.H. Ai, Y.L. Liu and H.F. Zhang, 2013. Geographic-feature oriented ria coastline simplification. *Acta Geodaetica et Cartographica Sinica*, 42: 595-601.
11. Ai, T.H., R.Z. Guo and Y.L. Liu, 2001. A binary tree representation of curve hierarchical structure in depth. *Acta Geodaetica et Cartographica Sinica*, 30: 343-348.