Metrol. Meas. Syst., Vol. XVIII (2011), No. 4, pp. 597–606.

PAN POLSKA AKADEMIA NAUK

METROLOGY AND MEASUREMENT SYSTEMS

Index 330930, ISSN 0860-8229

www.metrology.pg.gda.pl

M&MS

A NEW METHOD OF LINE FEATURE GENERALIZATION BASED ON SHAPE CHARACTERISTIC ANALYSIS

Hongshan Nie[1)], Zhijian Huang[2)]

1) ESSS Center. School of Electronic Science and Engineering, National University of Defense Technology, Changsha, China (nhs@nudt.edu.cn)

2) Institute of Software, Chinese Academic Science, Zhongguancun, Beijing, China (✉zhijian07@iscas.ac.cn, +86 152 0130 0594)

Abstract

This paper presents a piecewise line generalization algorithm (PG) based on shape characteristic analysis. An adaptive threshold algorithm is used to detect all corners, from which key points are selected. The line is divided into some segments by the key points and generalized piecewise with the Li-Openshaw algorithm. To analyze the performance, line features with different complexity are used. The experimental results compared with the DP algorithm and the Li-Openshaw algorithm show that the PG has better performance in keeping the shape characteristic with higher position accuracy.

Keyword: Line generalization; key point detection; shape characteristic analysis.

1. Introduction

Line feature generalization is a process from a large-scale map to a smaller-scale one, preserving bend characteristic and shape complexity [1]. It is an important process whenever there is a change of map scale or when data sets from different scales of a source document are mixed together [2]. For decades, it has attracted extensive attentions of many scholars and became an important issue in the field of cartography and geographic information systems. Many algorithms for line generalization have been proposed, but it is still a challenge, because of the variety and complexity of lines [1].

It is known that the performance of any generalization operation depends on line type and properties [3]. Though the proposed spatial line feature generalization algorithms are different from each other [4], generally a good algorithm should have the following properties:

(1) No spatial conflict, including self-intersection and cross-intersection.
(2) No starting point dependency.
(3) Preserving shape characteristic.
(4) Adaptive or no threshold.

In the last decades many efforts have been made to find an algorithm with no self-intersection, high position accuracy, which needs no tolerance threshold or adaptive tolerance threshold, and keeps well the shape characteristic [5]. The most influential one is the Dauglas-Peucker (DP) algorithm [6], presented in 1973. It approximates a curve by segments through points reduction. However, it still has some deficiencies such as self-intersection [7], starting point dependency, and using a single tolerance threshold which is difficult to apply to lines with different complexity.

Muller (1990) [8] developed some geometric procedure for removal of the spatial conflict. Visvalingham and Whyatt (1993) [9] presented an algorithm to improve shape distortion.

Article history: received on Apr. 4, 2011; received in revised form on Oct. 26, 2011; accepted on Nov. 17, 2011; available online on Dec. 12, 2011; DOI: 10.2478/v10178-011-0057-5.

Saalfeld (1999) [10] made use of dynamically updated convex hull data structure to efficiently detect and remove potential topological conflicts. Zhang and Liao (2001) [7] tried to improve self-interaction by gradually lowering the tolerance threshold until self-interaction is eliminated. Gold and Thibault (2002) [11] proposed an algorithm based on a skeleton, which can keep the topology of lines better and reduce the risk of self-intersection. Tong and Xu (2004) [12] improved DP approach with minimum variance line fitting. Shahriari and Tao (2002) [5] proposed a solution to set the tolerance adaptively by pre-analysis. Li and Openshaw [2, 13, 14] proposed an adaptive algorithm based on natural principle (Li-Openshaw). The line feature is generalized according to the minimum size of visual objects (SVO).

Just as Ai [15] pointed out that map generalization is a process of "information abstraction" rather than "data compression", line generalization should be an intelligent process associated closely with computer vision and pattern recognition. This paper presents a piecewise line generalization algorithm (PG) based-on key points detection. An adaptive threshold algorithm is used to detect corners, from which key points are selected. Segmented at the key points, line feature is generalized piecewise with the Li-Openshaw algorithm. Lines with different complexity are tested in the experiments and compared with the DP algorithm and the Li-Openshaw algorithm. To evaluate the performance quantitatively, three indices [16], Area-Difference, Bend-Ratio and Key-Point-Ratio, are employed.

This paper is organized as follows. In Section 2, the importance of key points on shape characteristic keeping is shown. An adaptive corner detection algorithm is introduced in Section 3. Key points are selected from corners during an optimization process described in Section 4. Results of experiments compared to the DP algorithm and the Li-Openshaw algorithm are shown in Section 5, followed by conclusions in Section 6.

2. Shape Characteristic Preserving with Key Points

When it comes to preserving the line's shape characteristic, vertexes of a line feature are not of same importance. Some are more important, such as corner points with high curvature, while others are less. As a matter of fact, only a few of them determine the basic shape of a line, which are called ***Key Points***. Once the key points are detected and preserved, the basic shape of a line feature will be kept. The previous algorithm, such as the DP algorithm and the Li-Openshaw algorithm, lost some key points, which inevitably affects the performance. So, analyzing the shape characteristic and detecting the key points before generalization, the result will be improved significantly and with less distortion.

The DP algorithm treats every point on the line as having the same importance, which inevitably results in line distortion. Such an example is shown in Fig. 1, where the dots denote key points. According to the DP algorithm, if the distance from point C to line BD is smaller than the tolerance threshold T, point C will be removed. Obviously, once it is removed, the line will distort severely. The example also shows that points A, C and F are all key points, but are removed unreasonably. The reason is that the DP algorithm does not distinguish vertexes of a line according to their different importance. In addition, the result of the DP algorithm depends on the starting point. That is to say, if the original line in Fig. 1 is generalized in a reverse order, the point C will be reserved, while point B will be removed.

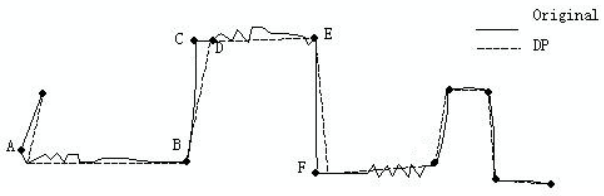


Fig. 1. An example with DP algorithm.

The Li-Openshaw algorithm also does not distinguish vertexes according to their different importance. It calculates the radius of SVO circle F_c according to the scale of source and objective maps, with formula (1).

$$F_c = S_t \cdot D \cdot (1 - \frac{S_f}{S_t}), \tag{1}$$

where: D is the minimum visual resolution, usually 0.4~0.6 mm [2]. S_f is the source map scale and S_t is the objective map scale.

As shown in Fig. 2, take point A as circle centre and take F_c as the radius, draw a SVO circle. The Li-Openshaw algorithm will generalize the curve A-B-C into the line A-D-C. The local maximum point B is removed, resulting in a distortion of shape of the line, which is not the purpose of 'good' generalization.

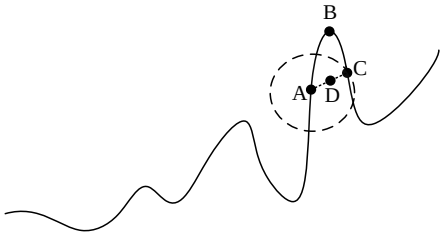


Fig. 2. An example with the Li-Openshaw algorithm.

Different curves have different shape characteristics, and different parts of the same curve also have different shape characteristics too [17]. If a curve is generalized with a unified threshold, it will inevitably lead to shape distortion. If the structural characteristic is analyzed before generalization and different parts are treated separately, it will achieve better results [17]. This paper presents a new method based on shape characteristic analysis. An adaptive threshold algorithm is used to detect corners, from which key points are selected during an optimization process. Segmented at the key points, the line feature is generalized piecewise with the Li-Openshaw algorithm.

3. Adaptive Corner Detection

He and Yung [18, 19] proposed an adaptive corner detection algorithm, which can detect almost all corner points on the curve without any thresholds. Mokhtarian and Mackworth (1992) [20] proposed a method to calculate curvature of points on the edge of a binary image. Curvature is defined as follows:

$$K(u,\sigma) = \frac{\dot{X}(u,\sigma)\ddot{Y}(u,\sigma) - \ddot{X}(u,\sigma)\dot{Y}(u,\sigma)}{[\dot{X}(u,\sigma)^2 + \dot{Y}(u,\sigma)^2]^{1.5}}, \tag{2}$$

where: $\dot{X}(u,\sigma) = x(u) \otimes \dot{g}(u,\sigma)$, $\ddot{X}(u,\sigma) = x(u) \otimes \ddot{g}(u,\sigma)$, $\dot{Y}(u,\sigma) = y(u) \otimes \dot{g}(u,\sigma)$, $\ddot{Y}(u,\sigma) = y(u) \otimes \ddot{g}(u,\sigma)$, $\otimes$ is the convolution symbol, $g(u,\sigma)$ is the Gaussian kernel function with width σ, $\dot{g}(u,\sigma)$ and $\ddot{g}(u,\sigma)$ are the first order and second order derivatives, respectively.

The corner detection algorithm includes three steps:

(1) Calculates the curvatures of all vertexes of the line, and selects points with local maxima curvature as the potential corners.

(2) Remove round corners.

(3) Remove false corners.

3.1. Round Corner Removing

A **round corner** is a point on an arc which has the local maximum curvature, but the change of curvature among its neighbours is gentle, shown as point A in Fig. 3. According to the formula (2), the curvature of pixels can be calculated. Examples of a round corner and true corner are shown in Fig. 3, where the upper half is a digital line and the bottom half is its curvature. Though the curvature of point A is the largest among its neighbours, the difference is not significant. As for the point B in Fig. 3, the curvature may have a smaller absolute maximum value, but the difference is obvious, so point B is a true corner and point A is a round corner.

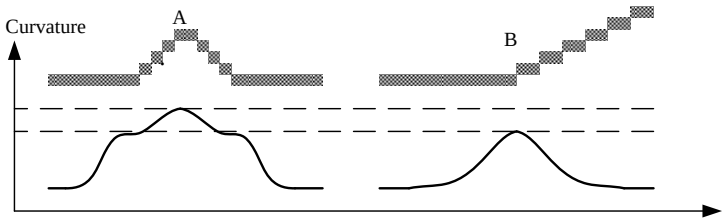


Fig. 3. Examples of round corner and true corner. **?**

An adaptive threshold method can be used to remove round corners. The threshold is defined as follows:

$$T(u) = R \times \frac{1}{L_1 + L_2 + 1} \sum_{i=u-L_2}^{u+L_1} |K(i,\sigma)|, \tag{3}$$

where R is a coefficient, 1.5 in general. L_1 is the length from the current maximum vertex to the last minimum vertex, L_2 is the length to the next minimum, and $K(i,\sigma)$ is the curvature at point i, which is defined by formula (2). The line between the adjacent minimum vertexes is called as Region of Support (ROS). In fact, $T(u)$ is an average curvature of all points in ROS. If the curvature is less than $T(u)$, the current point will be considered as a round corner. Because the curvature around a round corner declines slowly, the $T(u)$ is high with respect to that of true corner around which drops rapidly. It means that a round corner will be filtered out because of its high curvature value.

3.2. False Corner Removing

The ROS of this step is defined as the line between a potential corner and the next one. As shown in Fig.4, after points B and D are removed as a round corner, the ROS of point C is expanded from B-C-D to A-C-E. That is to say, point C will be tested in a larger scale.

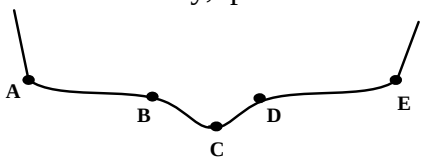


Fig. 4. Example of a false corner.

In order to remove false corner points, a tangent angle is defined as shown in Fig. 5. On one arm of ROS (from potential corner C to potential corner L, say), we can draw an arc passing point C, the midpoint M and point L, with circle centre Co. The tangent angle γ_1 is defined as the angle between the tangent line passing C and the horizontal line. The tangent

angle of the other arm from C to R is γ_2. Hence, the angle $\angle C$ between the two tangent lines is:

$$\angle C = \begin{cases} |\gamma_1 - \gamma_2| & if\ |\gamma_1 - \gamma_2| < \pi \\ 2\pi - |\gamma_1 - \gamma_2| & else \end{cases}. \tag{4}$$

The corner checking criterion is given as follows:

If $\angle C_i \le \theta_{obtuse}$, C_i is considered as a true corner, else a false corner. Parameter θ_{obtuse} is the maximum corner obtuse angle, range from 150° to 170°.

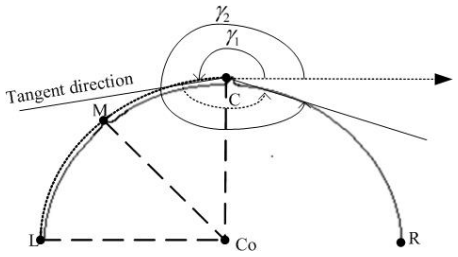


Fig. 5. Tangent angle definition.

Once removing the false corner points, the range of the ROS will enlarge and $\angle C$ will change correspondingly. So the process should go on iteratively until the number of the candidate corner points does not decline. Generally, the process converges after three- to five-fold iteration [18].

4. Key Point Detection

Because of quantization error of the digital line feature, there are still some corner points reserved after round corner removing and false corner removing. An example is shown in Fig.6, where points B and C are not necessary to represent the line. A method for key point detection is proposed in this section, which is virtually an optimization process under the criterion of least mean error. No threshold is needed in the process.

The object function is defined as follows:

$$F_{i,i+j} = E_{i,i+j} / L_{i,i+j}, \tag{5}$$

where $L_{i,i+j}$ is the distance between corner points P_i and P_{i+j}. $E_{i,i+j}$ is the sum of position error defined by formula (6):

$$E_{i,i+j} = \sum_{v_l \in V_{i,i+j}} d(v_l), \tag{6}$$

where $V_{i,i+j}$ is the set of all vertexes between corner points P_i and P_{i+j}. The $d(v_l)$ is the distance from vertex v_l to the line between corner points P_i and P_{i+j}. Seen from the formula (5), the object function $F_{i,i+j}$ means the position error of each unit distance between the corner P_i and corner P_{i+j}, called average error (AE). If the AE is decreasing, it means that the shape of the line does not change obviously. Otherwise, an obvious shape change will appear at the corner point where the AE is going to increase. The goal of the following process is to find this kind of corners.

Let $P_1, P_2, ..., P_n$ denote the sequence of corner points, the selection process is as follows:

Step (1) Let $i = 1$, $j = 1$, and select the beginning point P_1 as the key point

Step (2) Calculate $F_{i,i+j}$ with formula (5), and let $F_0 = F_{i,i+j}$

Step (3) Let $j = j+1$, and calculate $F_{i,i+j}$. If $i+j < n$, go to step(4), else go to step (6).

Step (4) If $F_0 = 0$ or $F_{i,i+j} > F_0$, then select corner point P_{i+j-1} as the key point, let $i = i+j-1, j = 1$, $F_0 = F_{i,i+j}$ and return to step (3)

Step (5) If $F_{i,i+j} \le F_0$, then let $F_0 = F_{i,i+j}$, and return to step (3)

Step (6) Select the end point P_n as the key point and stop.

According to the process above, the line shown in Fig. 6(a) is tested. The points ABCDEF are corners detected with the method in Section 3. Starting from the point A, only points ADF are selected as key points. The result is shown in Fig. 6(b).

(a) (b)

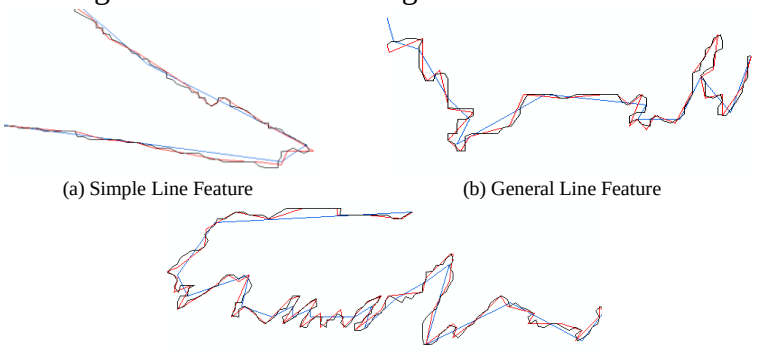
Fig. 6. Example of key point detection.

5. Results and Analysis

In order to analyze the performance of the method, lines with different complexity are used in the experiment to compare with the results of the DP algorithm and the Li-Openshaw algorithm.

5.1. Qualitative Analysis

The results compared with the DP algorithm are shown in Fig. 7, while results compared with the Li-Openshaw algorithm are shown in Fig. 8.



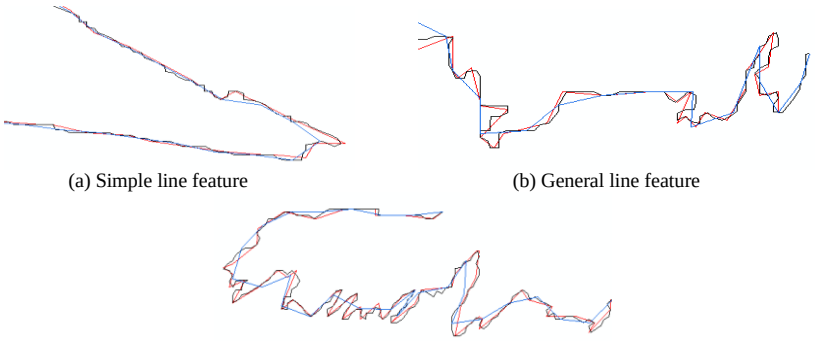
(a) Simple Line Feature (b) General Line Feature

(c) Complicated line feature

Fig. 7. PG results compared with the DP algorithm. (Blue lines denote the result of the DP algorithm, red lines denote the result of PG algorithm, and black ones denote the original lines).

Qualitative analysis results show that the performance of the method is improved more or less on lines with different complexity. As shown in the following figures, the results of the DP, Li-Openshaw and PG algorithm are almost the same in a simple case. But in a complicated case, the PG algorithm performs obviously better. Especially when the line bends rapidly, it keeps the shape of line feature better, with higher position accuracy. The reason lies in two aspects: first, almost all key points are preserved which always represent the shape characteristics of lines; second, non-key points are not removed simply as in the DP algorithm, but are made full use of. The risk of self-intersection is reduced greatly, because the line is divided before generalization. At the stage of piecewise generalization, the PG algorithm uses the same method as in the Li-Openshaw algorithm, in which the scale is

calculated based on nature law SVO. The threshold is calculated automatically according to the original map scale and target map scale.



(a) Simple line feature (b) General line feature

(c) Complicated line feature

Fig. 8. PG results compared with Li-Openshaw. (Blue lines denote the result of the Li-Openshaw algorithm, red lines denote the result of PG algorithm, and black ones denote the original lines).

5.2. Quantitative Analysis

In order to evaluate the performance quantitatively, three indices [16], Area-Difference, Bend-Ratio and Key-Point-Ratio, are employed.

Area-Difference is the area of the polygon formed by line feature before and after generalization. It reflects the sum of error on position.

Bend-Ratio is the length of line generalized divided by that of the original line. The value is between 0 and 1. Actually, it reflects the difference in length before and after generalization.

Key-Point-Ratio is the number of the key points detected by the PG algorithm divided by that of those detected by a human. The ratio value is also between 0 and 1. It means that the shape characteristic of line feature is preserved well when the value is high.

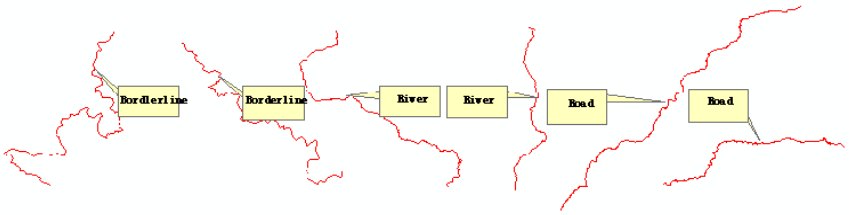
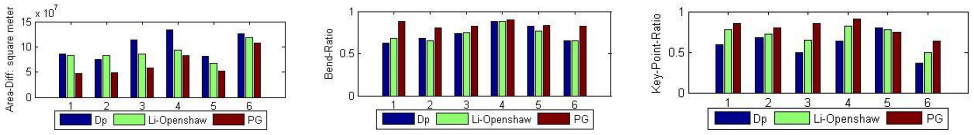


Fig. 9. Original line features.

In the experiment, three kinds of line features are used, downloaded from the national fundamental geographic information network, at http://nfgis.nsdi.gov.cn/nfgis/chinese /c_xz.htm. As shown in Fig. 9, line features with different complexity consist of regional borderlines, rivers and roads. Results of the PG algorithm, compared with the DP algorithm and the Li-Openshaw algorithm, are shown in Fig.10. It can be seen that the PG algorithm produces the minimum of area difference, while the DP algorithm produces a maximum. The difference of the result depends on the length and complexity of the line feature. The PG algorithm has a maximum Bend-Ratio, which means it has the best performance to keep a shape characteristic. At the same time, the Key-Point-Ratio of the PG algorithm is the highest, which indicates that the shape characteristic is preserved well.

As for the 5th experiment data, the Key-Point-Ratio of the PG algorithm is smaller than in the DP algorithm and Li-Openshaw algorithm, shown in Fig. 10c, it is because in the ***"Key Points Detection"*** section the PG algorithm throws off more corner points and selects out fewer key points. But the shape of the line is well kept, and the Area-Difference and Bend-Ratio parameters are better than in the DP algorithm and the Li-Openshaw algorithm.



a) Area-Difference b) Bend-Ratio c) Key-Point-Ratio

Fig. 10. Results of accuracy estimation.

6. Conclusions

So far, automatic line generalization is still a challenge because of the variety and the complexity of line features. This paper presents a piecewise generalization (PG) algorithm based on key points detection. An adaptive threshold algorithm is used to detect all corners, from which key points are selected. A line, divided by these points, is generalized piecewise with the Li-Openshaw algorithm. To analyze the performance, line features with different complexity and different kinds are used. The results compared with the DP algorithm and the Li-Openshaw algorithm show that the PG algorithm has better performance of shape characteristic preservation and higher position accuracy.

References

[1] Li, Z. (2007). Digital Map Generalization at the Age of Enlightenment: A Review of the First Forty Years. *The Cartographic Journal*, 44(1), 80-93.

[2] Li, Z., Openshaw, S. (1992). Algorithms for Automated Line Generalization Based on a Natural Principle of Objective Generalization. *International Journal of Geographical Information Systems*, 6(5), 373-389.

[3] Ariza, L.F.J., Garcia, B.J.L. (2008). Generalization-Oriented Road Line Segmentation by Means of an Artificial Neural Network Applied over a Moving Window. *Pattern Recognition*, 41, 1610-1626.

[4] Lei, W., Liu, D., Tong, X. (2005). Discussion About Uncertainty of Spatial Line Feature Generalization Algorithms. *Engineering of Surveying and Mapping*, 30(6), 20-22.

[5] Shahriari N., Tao V. (2002). Minimissing Positional Errors in Line Simplification Using Adaptive Tolerance. *Symposium on Geospatial Theory, Procesing and Application*, 4(3), 213-223.

[6] Douglas, D., Peucker, T. (1973). Algorithm for the Reduction of the Numbers of Points Required to Represent a Digitized Line or Its Caricature. *The Canadian Cartographer*, 10(2), 112-122.

[7] Zhang, Q., Liao, K. (2001). Line Ganeralization Based on Structure Analysis. *Acta Scientiarum Naturalium Universitatis Sunyatseni*, 40(5), 118-121.

[8] Muller, J.C. (1990). The Removal of Spatial Conflicts in the Line Generalization. *Cartography and Geographic Information Systems*, 17(2), 141-149.

[9] Visvalingham, M., Whyatt, J. (1993). Line Generalization by Repeated Elimination of Points. *The Cartographic Journal*, 30(1), 46-51.

[10] Saalfeld, A. (1999). Topologically Consistent Line Simplification with the Douglas–Peucker Algorithm. *Cartography and Geographic Information Science*, 26(1), 7-18.

[11] Gold, C., Thibault, D. (2002). Map Generalization by Skeleton Retraction. *ICA Workshop on Map Generalization*, Ottawa, 78-85.

[12] Tong, X.-H., Xu, G.-S. (2004). A New Least Squares Method Based Line Generalization in Gis. *IEEE International Geoscience and Remote Sensing Symposium Proceedings*, 5, 2912-2915.

[13] Li, Z., Openshaw, S. (1993). A Natural Principle for Objective Generalisation of Digital Map Data. *Cartography and Geographic Information Systems*, 20(1), 19-29.

[14] Li, Z., Openshaw, S. (1994). Linear Feature's Self-Adapted Generalization Algorithm Based on Impersonality Generalized Natural Law. *Translation of Wuhan Technical University of Surveying and Mapping*, 1, 49-58.

[15] Ai, T. (2007). The Drainage Network Extraction from Contour Lines for Contour Line Generalization. *ISPRS Journal of Photogrammetry and Remote Sensing*, 62, 93-103.

[16] Chen, B., Zhu, K., Xue, B. (2007). Quality Assessment of Linear Features Simplification Algorithms. *Journal of Zhengzhou Institute of Surveying and Mapping*, 2(24), 121-124.

[17] Zhu, K.-P., Wu, F., Wang, H.-L., et al. (2007). Improvement and Assessment of Li-Openshaw Algorithm. *Cehui Xuebao/Acta Geodaetica et Cartographica Sinica*, 36, 450-456.

[18] He, X.C., Yung, N.H.C. (2004). Curvature Scale Space Corner Detector with Adaptive Threshold and Dynamic Region of Support. *Proceedings of the 17th International Conference on Pattern Recognition*, 2, 791-794.

[19] He, X.C., Yung, N.H.C. (2008). Corner Detector Based on Global and Local Curvature Properties. *Optical Engineering*, 47(5).

[20] Mokhtarian, A.K., Mackworth, A.K. (1992). A Theory of Multi-Scale Curvature-Based Shape Representation for Planar Curves. *IEEE Trans. Pattern Anal. Mach. Intell*., 14(8), 789-805.