

A New Method of Breakpoint Connection using Curve Features for Contour Vectorization

Jianhui Zhao, Guozhong Liang, Zhiyong Yuan, Dengyi Zhang
 Computer School, Wuhan University,
 Wuhan, Hubei, 430072, P.R. China, phone: 86-27-68775669
 Suzhou Institute of Wuhan University,
 Suzhou, Jiangsu, 215123, P.R. China
 jianhui.zhao@whu.edu.cn

Abstract—Connection of break points is a vital and difficult problem during contour vectorization for digital image, such as scanned maps and engineering drawings. In this paper a novel algorithm is proposed for breakpoint connection using curve features. Three features including curve distance, tangent angle and average curvature, are defined and analyzed. Then the features are normalized and fused into one energy function to represent the connection probability. Based on the probability function, all candidate connections for one breakpoint are compared and the most suitable result is determined. With the help of image preprocessings including denoising, binarization and thinning, our method has been tested with experiments on real images, and the results prove its efficiency.

Index Terms— Breakpoint connection, Contour vectorization, Curve features, Object extraction

I. INTRODUCTION

Automatic object recognition from the scanned maps or engineering drawings is a useful way for data collection [1], during which the techniques of digital image processing, artificial intelligence and machine learning are usually adopted [2]. As one basic and important data type, contour vectorization is a key problem in object extraction [3]. Due to the fact that some broken curves often exist after image segmentation [4], determination of the proper connections among the breakpoints has become a vital step.

Until now, many researchers focus on this topic and the following several kinds of methods have been typically employed. (1) Method of the minimum value [5]: for one breakpoint, take the breakpoint with the minimum distance or direction difference as its connection. Obviously, some errors cannot be avoided, especially when the points are far away from each other. (2) Method of mathematical morphology [6]: connection is obtained by the basic operators such as erosion, dilation, opening and closing. It works well for the simple objects but cannot deal with complex situations. (3) Method of graph search [7]: build one graph based on the breakpoints,

and then determine the best connections by graph search considering the necessary physical constraints. Such method has very high computational complexity.

Different with the existing techniques, in this paper we present our further study on curve features of contour, and propose a novel algorithm for breakpoint connection.

II. IMAGE PREPROCESSING

The image, e.g. scanned maps or drawings, are firstly processed. For image denoising, adaptive median filter [8] is used because of its ability to preserve details and smooth non-impulsive noise. For image binarization, Otsu method [9] is adopted to automatically perform histogram shape based image thresholding, and reduce a gray level image into a binary image. For image thinning or skeletonization, Zhang's algorithm [10] is utilized to represent foreground region in the binary image as a skeletal remnant preserving the extent and connectivity of the region while throwing away the other pixels of the original foreground. After the preprocessings, only approximate contours remain from the discrete broken curves, therefore connection operation has to be performed subsequently for the existing break points, and then the contours can be extracted finally.

III. BASIC CURVE FEATURES

In our method, three basic features, i.e. curve distance, tangent angle and average curvature, are all considered to reflect the curve properties of contour.

A. Curve distance d

In the existing connection techniques, the linear distance between two breakpoints is normally used to evaluate the possible matching. However, since contour always has the form of closed curve, we present curve distance as a better inherent feature for the contour.

The curve distance for two breakpoints is calculated as: construct a curve, e.g. B-spline, based on all the related points, i.e. two breakpoints and the pixels connected with them; generate the virtual curve between two breakpoints with spline interpolation; compute the distance along the virtual curve, i.e. the sum of 2D distances between every two neighboring points on the virtual curve.

The connection probability of 2 breakpoints is inversely

Manuscript received March 12, 2012; accepted May 15, 2012.

This work was supported by National Basic Research Program of China (973 Program, No. 2011CB707904), Science and Technology Bureau of Suzhou Municipality (No. SH201115), Science and Technology Bureau of Wuhan Municipality (No. 201150124001), Natural Science Foundation of Hubei Province of China, and R&D Special Fund for Public Welfare Industry of China Meteorological Administration (No. GYHY201106047).

proportional to their curve distance, and two points with less curve distance have higher matching possibility.

B. Tangent angle τ

For one breakpoint of the curve, its tangent can be used to represent the curve's extension direction, as shown by vector AL from point A of curve L1 in Fig. 1. Suppose there are candidate curves to be connected, and vectors can be formed with their corresponding breakpoints, as shown by vector AB and vector AC.

Thus we define the feature of tangent angle as the angle between the extension direction of one breakpoint and the vector from it to another possible matching breakpoint. For example, angle θ_{AB} or angle θ_{AC} in Fig. 1 is the tangent angle between curve L1 and curve L2 or L3, which is also valuable to describe the essential property of the contour formed with the related curves.

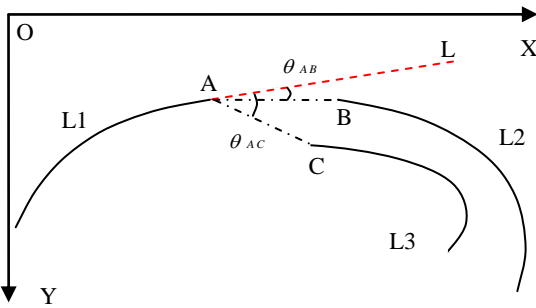


Fig. 1. Feature of tangent angle.

The connection probability of 2 breakpoints is inversely proportional to their tangent angle, and two points with less tangent angle have higher matching possibility. As illustrated in Fig. 1, since $\theta_{AB} < \theta_{AC}$, breakpoint B is more reasonable to be connected with point A.

C. Average curvature φ

As aforementioned, a curve such as B-spline can be generated between two breakpoints. Thus along the virtual curve, the accumulation of tangent differences from every two neighboring interpolated points on the curve can be computed as $\Delta\theta$, while the curve distance is d .

Then the feature of average curvature is defined as $\Delta\theta / d$, which is used to represent the curvature degree or the smoothness of the constructed contour. As illustrated in Fig. 2, the contour can be formed by curves L1 and L2, or curves L1 and L3, but they have different curved shape, i.e. different values of average curvature.

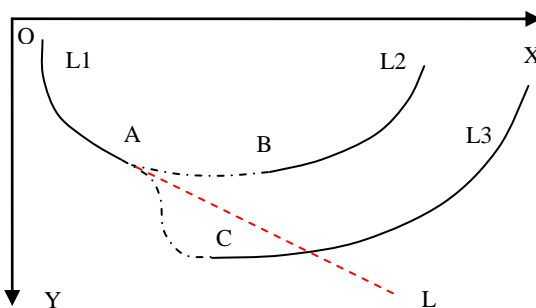


Fig. 2. Feature of average curvature.

The connection probability of 2 breakpoints is inversely proportional to their average curvature, and two points with less average curvature have higher matching possibility. As shown in Fig. 2, the contour made with L1 and L2 is much smoother, thus breakpoint B is more likely to be connected with point A.

IV. FUNCTION FOR CONNECTION PROBABILITY

To consider three basic features simultaneously, they are further normalized as follows.

A. Normalization of curve distance d_N

An empirical threshold value d_T is set for the curve distance d , if $d \geq d_T$, the related two breakpoints cannot be connected since they are far away from each other. The normalized curve distance d_N is defined as

$$d_N = 1 - d / d_T \quad (1)$$

Therefore d_N varies in range of $[0, 1]$, and is proportional to connection probability. When $d=0$, $d_N=1$, which means that the two breakpoints are already connected.

B. Normalization of tangent angle τ_N

For angle τ , i.e. the tangent angle, it varies in the range of $[0, \pi]$, and the less value of tangent angle corresponds to the higher connection possibility. Thus the normalized tangent angle τ_N is defined as

$$\tau_N = (1 + \cos \tau) / 2 \quad (2)$$

Then τ_N , varying in $[0, 1]$, is now proportional to the connection probability, and $\tau=0/\pi$ means the maximum/minimum connection probability, i.e. $\tau_N=1/0$. Of course, in practical applications, the empirical threshold τ_T (less than π) can be employed to reduce the varying range to $[0, \tau_T]$, and thus decrease the computing expense.

C. Normalization of average curvature φ_N

Based on the definition of average curvature, for two breakpoints A and B, φ_A can be computed as the average curvature from A to B along the interpolated virtual curve between A and B, in the same way φ_B can be computed from B to A, as illustrated in Fig. 3.

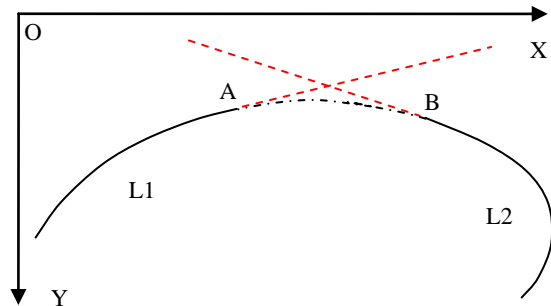


Fig. 3. Normalization of average curvature.

We define the normalized average curvature φ_N as

$$\varphi_N = 1 / 2 + (\cos \varphi_A + \cos \varphi_B) / 4 . \quad (3)$$

Thus φ_N varies within $[0, 1]$, and is proportional to the connection probability. When the virtual curve between A and B is very smooth, both φ_A and φ_B tend to be 0, and φ_N approaches to 1, i.e. the maximum probability.

After normalizations of three basic features, they are fused into one function to represent the connection probability for a pair of breakpoints

$$P = \mu_1 d_N + \mu_2 \tau_N + \mu_3 \varphi_N , \quad (4)$$

where μ_1 , μ_2 and μ_3 are the weighting parameters for three normalized features. When the function is used to determine the connection of breakpoints, the empirical threshold P_T can be used for filtering. Only if $P > P_T$, the related two breakpoints can be further considered.

Based on the experiments, it can be found that τ_N is the most important feature, while d_N usually has the least effect, thus the values of weighting parameters should be set as $\mu_2 > \mu_3 > \mu_1$. For example, $\mu_1 = 0.2$, $\mu_2 = 0.5$, $\mu_3 = 0.3$ are a set of suitable assignments.

V. PROCEDURE OF BREAKPOINT CONNECTION

For the input digital image after preprocessing steps of denoising, binarization and thinning, the whole procedure of our breakpoint connection algorithm is:

(1) Location of breakpoints:

(1.1) Check each pixel of image, where black pixels in objects while white pixels in the background;

(1.2) Regard current black pixel as one breakpoint, if only one of its eight neighbors is also black pixel;

(1.3) Store each breakpoint into the set of breakpoints.

(2) Determination of connections:

(2.1) Take one point from the set of breakpoints, and search for its connection from the other points in the set;

(2.2) For the current breakpoint, use the curve distance threshold d_T to filter its candidate points for connection;

(2.3) For the current breakpoint, use the tangent angle threshold τ_T to further filter its candidate points;

(2.4) For the current breakpoint, compute all its connection probabilities with each of its filtered candidate points using Function (4), then filter its candidate points again based on probability threshold P_T ;

(2.5) Determine the breakpoint for connection by:

(2.5.1) If there are several filtered candidates, take the one with maximum probability as connection point;

(2.5.2) If there is only one filtered candidate, take it as connection point;

(2.5.3) If there is no filtered candidates, current breakpoint cannot be matched and just remain in the set;

(2.6) Remove the current breakpoint and its matched point from the set if the connection exists;

(2.7) Jump to (2.1) and deal with the next breakpoint;

(2.8) Stop if the set of breakpoints is empty, or jump to (3) if there are still breakpoints remained in the set.

(3) Processing of the remained breakpoints:

(3.1) Consider one of the remained breakpoints, and search for its connection from the other points;

(3.2) Increase the threshold d_T of curve distance;

(3.3) Take the constraints off from the tangent angle threshold τ_T and the connection probability threshold P_T ;

(3.4) For the current point, calculate all its connection probabilities with the other breakpoints using Function (4);

(3.5) Take the breakpoint with maximum probability as the connection point, then remove these two matched points from the remained breakpoints;

(3.6) Jump to (3.1) and deal with the next breakpoint until all of the remained breakpoints are processed.

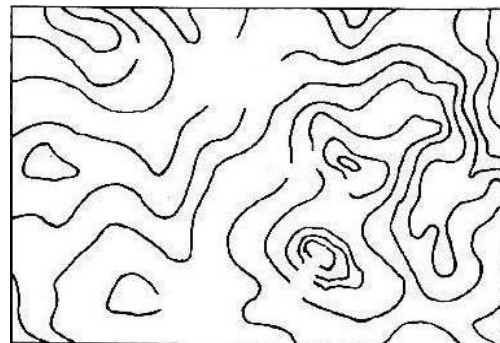
(4) Connection of the matched breakpoints.

VI. EXPERIMENTAL RESULTS

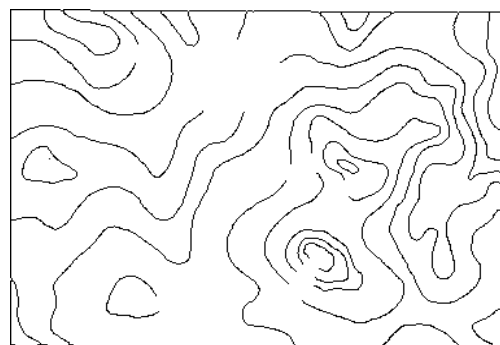
Our algorithm has been tested with real images. As shown in Fig. 4, the 1st row is the original image; the 2nd row is the preprocessed image after denoising, binarization and then thinning; the 3rd row is the connected results from the existing minimum value based approach; while the 4th row is the connection results from our method. From the results it can be found that the proposed method generates more accurate connections among the breakpoints.

Based on connection of breakpoints, we implemented the complete application program for contour vectorization from image. As shown in Fig. 5, the 1st row is the input image; the 2nd row is the skeletonized result with connected breakpoints using our new algorithm; and the 3rd row shows one of the vectorized contours.

After connection of breakpoints, the problem of contour vectorization turns into a simple task to track a series of connected pixels along one skeleton.



a)



b)

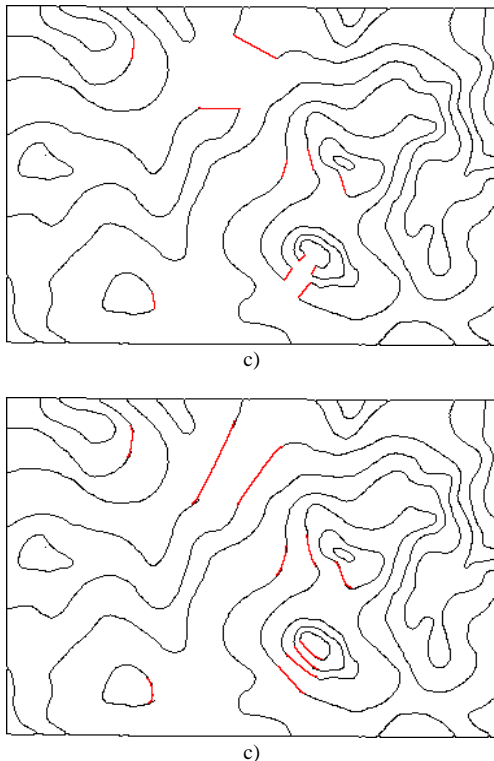


Fig. 4. Connection results of breakpoints, 1st row: the input image, 2nd row: image after preprocessing, 3rd row: result of minimum value method, 4th row: result of our method.

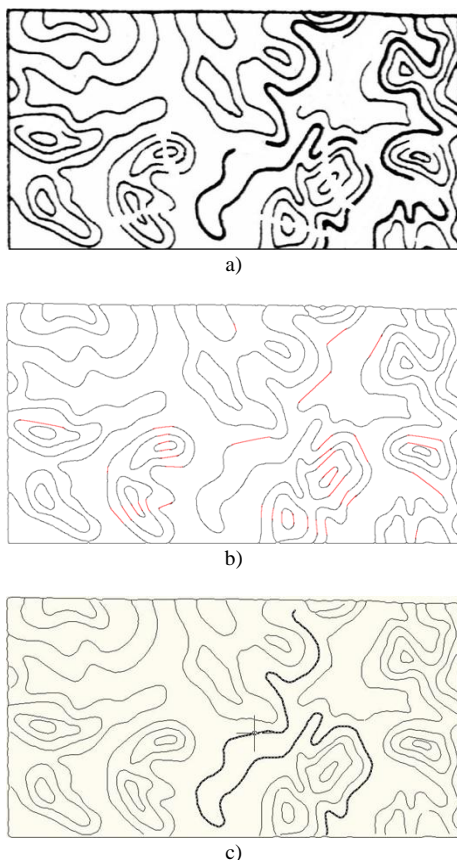


Fig. 5. Application on contour vectorization, 1st row: the input image, 2nd row: skeletonized image with connected breakpoints, 3rd row: one vectorized contour.

And then the vectorized results can be stored in a file with standard format such as DXF, which can be further processed with other softwares, e.g. AutoCAD.

VII. CONCLUSION

The existing techniques for connection of breakpoints for contour vectorization from digital images have disadvantages of less accurate, narrow application, or costly computation. In our paper, a novel algorithm for breakpoint connection is proposed based on the basic curve features of contour. Our approach considers all features of curve distance, tangent angle and average curvature, thus is more accurate than the minimum value methods. Our approach uses preprocessing to denoise, binarize and skeletonize the objects, thus has more wide applications than the mathematical morphology methods. Our approach deals with the filtered breakpoints and avoids the global search, thus spends less computing cost than graph search methods.

The proposed method has been tested by experiments on images of scanned maps or engineering drawings. It is also taken as one key step of contour vectorization, and the whole program works well on images with the output of satisfying extracted contours, which can then be used by many other possible applications.

REFERENCES

- [1] A. Pezeshk, R. L. Tutwiler, „Automatic Feature Extraction and Text Recognition from Scanned Topographic Maps“, *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 12, pp. 5047–5063, 2011. [Online]. Available: <http://dx.doi.org/10.1109/TGRS.2011.2157697>
- [2] J. M. Saragih, S. Lucey, J. F. Cohn, „Deformable Model Fitting by Regularized Landmark Mean-Shift“, *International Journal of Computer Vision*, vol. 91, pp. 200–215, 2011. [Online]. Available: <http://dx.doi.org/10.1007/s11263-010-0380-4>
- [3] W. Tao, „Iterative Narrowband Based Graph Cuts Optimization for Geodesic Active Contours with Region Forces“, *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 284–296, 2012. [Online]. Available: <http://dx.doi.org/10.1109/TIP.2011.2160955>
- [4] X. Hilaire, K. Tombre, „Robust and accurate vectorization of line drawings“, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, iss. 6, pp. 890–904, 2006. [Online]. Available: <http://dx.doi.org/10.1109/TPAMI.2006.127>
- [5] T. Lu, C. L. Tai, H. Yang, S. Cai, „A Novel Knowledge-Based System for Interpreting Complex Engineering Drawings: Theory, Representation, and Implementation“, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 8, pp. 1444–1457, 2009. [Online]. Available: <http://dx.doi.org/10.1109/TPAMI.2008.161>
- [6] D. Tuija, F. Pacifici, M. Kanevski, W. J. Emery, „Classification of Very High Spatial Resolution Imagery Using Mathematical Morphology and Support Vector Machines“, *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 11, pp. 3866–3879, 2009. [Online]. Available: <http://dx.doi.org/10.1109/TGRS.2009.2027895>
- [7] J. A. Fernandez, J. Gonzalez, „Multihierarchical graph search“, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp.103–113, 2002. [Online]. Available: <http://dx.doi.org/10.1109/34.982887>
- [8] S. Akkoul, R. Ledee, R. Leconge, R. Harba, „A New Adaptive Switching Median Filter“, *IEEE Signal Processing Letters*, vol. 17, no. 6, pp. 587–590, 2010. [Online]. Available: <http://dx.doi.org/10.1109/LSP.2010.2048646>
- [9] G. Y. Zhang, G. Z. Liu, H. Zhu, B. Qiu, „Ore image thresholding using bi-neighbourhood Otsu's approach“, *Electronics Letters*, vol. 46, no. 25, pp. 1666–1668, 2010. [Online]. Available: <http://dx.doi.org/10.1049/el.2010.2962>
- [10] T. Y. Zhang, C. Y. Suen, „A fast parallel algorithm for thinning digital Patterns“, *Communications of the ACM*, vol. 27, no. 3, pp. 236–239, 1984. [Online]. Available: <http://dx.doi.org/10.1145/357994.358023>