ISSN 1746-7659, England, UK
Journal of Information and Computing Science
Vol. 1, No. 5, 2006, *pp.* 275-283

Contour Line Extraction from Paper-based Topographic Maps

Dongjun Xin [1+], Xianzhong Zhou [2], Huali Zheng [3]

[1]College of Automation, Nanjing University of Science &Technology, Nanjing 210094
[2]School of Management and Engineering, Nanjing university, Nanjing 210093
[3]Artillery and Air Defence Army Research Institute of Equipment & Technology, Beijing 100012

(*Received April 21, 2006, Accepted August 11, 2006*)

Abstract. Contour line is the main linear feature on topographic maps. Extraction of contour lines is tedious and time-consuming process, and is still an interesting problem. A novel method for extraction of contour lines from paper-based topographic maps is presented in this paper. In our approach, we firstly segment color topographic maps for achieving the binary image of brown color contour lines. Then, mathematic morphology method is used to filter the binary image. Next, Utilizing c-means algorithm to look for initial seed point on thinning contour lines. Fourthly, utilizing improved active contour model to extract non-thinning contour lines. Last, we have analyzed the directional field of contour lines near the gap, and then reconnected broken contour lines. The performance of the algorithm is tested on several topographic maps and comparing with other algorithms.

Keywords: contour line, extraction, c-means, active contour model, directional field

1. Introduction

Paper-based topographic maps vectorization is a tedious and time-consuming process. Many scholars have researching how to realize fully automatic extraction from scanned topographic maps. Contour line is very important information on topographic maps. Extraction of contour line is still an interesting problem. Today, techniques of fully automatic vectorization for topographic maps are still insufficient to provide a satisfying result.

Topographic map essentially consists of linear features and area features. Different feature is printed with different color. Brown is used to depict contour line. Contour line is smooth, continuous curve. On 300dpi resolution maps, contour line's width is approximate from 2 to 4 pixels. Contour lines of topographic maps are taken as approximate parallel continuous curves. They maybe overlap each other on precipitous cliff. Usually, contour lines are looked as independent each other and no overlap.

Because of colorific dispersion, a Paper-based topographic map after scanned own thousands of different colors. This comes into being the aliasing and false color. Further more, contour lines overlap with other features on map can also increase extraction difficulty.

In this paper, we first summarize the related works in section 2. Next, in section 3, color image segmentation, morphological filtering and contour lines thinning are discussed. In section 4, initial seed points are searched by c-means algorithm. In section 5, utilizing improvement snake model to extract contour lines, reconnecting broken contour lines basing on analyzing the directional field of contour lines. In last section, there are some experiments and analysis.

2. Related Works

Research on automated map recognition has been going on for many years, thus resulting in a huge amount of publications. Early researches focus mainly on monochrome image. For example, leberl[1] utilizes digitized binary image to vector clean contour and drainage/ridge sheets. Greenlee attempts to extract elevation contour lines on topographic maps[2]. Amin attempts to recognize lines and symbols[3]. Their

[1+] Corresponding author. Tel.: +86-25-84303047-8
E-mail address: xindj@msn.com.

procedure mainly include four steps: ① topographic map digitization by scanner, ② filtering and thresholding, ③ thinning and pruning the binary image, ④ raster to vector conversion of the resulting thinned lines. For color topographic maps, color information is essential for recognizing its features. More and more researches are on color-based maps. For example, in the early periods, Soille uses the mean and variance of the hue channel for discriminating soil types on a digitized soil map[4]. Recently, Loh performs color image segmentation and thins contour lines. At last, this paper utilizes the A^* search algorithm to reconnect contour lines[5].

Dupont uses external terrain elevation data to enhance the extraction of contour lines from a scanned topographic map[6]. The algorithm uses a watershed divide algorithm in RGB space to assign a pure map color to each pixel. An expert system is used to reconnect contour lines. This algorithm performs well for images scanned by high resolution and quality scanners and may not perform well when the image contains aliasing and false colors.

Gamba uses A^* search algorithm to extract discontinuous lines like symbols and dashed lines on topographic maps[7]. The algorithm is not optimal for continuous lines such as contour line.

Yamada uses a Multi-Angled-Parallel operation algorithm for the extraction of text and symbols on topographic maps[8][9]. The algorithm uses directional mathematical morphology method for extraction of contour lines.

Wu uses a multiplayer neural to extract characters and lines from color image[10]. Although this algorithm takes color intensity and gradient into account, it cannot overcome the problems of aliasing and false colors.

To overcome aliasing and false color problems, Hedley develops a gradient thresholding method[11]. The algorithm utilizes spatial and color space information to get rid of pixels in the high gradient set. Because most of the linear features on topographic map belonging to the high gradient set. Their scheme can't use for extraction of contour lines on topographic maps.

Arrighi uses mathematic morphology to process contour lines on binary image[12]. The algorithm utilizes propagation function to detect two extremities and then uses a skeletonization with anchor points to thin contour lines. At last, a combination of Euclidean distances between extremities, and differences between their directions are used joining the disconnected lines.

Eikvil uses china code tracing technique for reconstruction of contour lines[13]. In this approach, it is assumed that there is only one possible reconnection.

Frischknecht uses a hierarchical template matching procedure to extract text from topographic maps[14]. His algorithm is difficult to extract contour lines on topographic map.

Spinello uses local geometric properties to recognize the contour lines[15]. The algorithm is substantially based the global topology of a generic topographic map. The algorithm uses Delaunay Triangulation to thin and vectorize contour line.

In this paper, we present a novel extraction algorithm based on improved GGVF snake model and the directional fields of contour lines.

3. Color Image Preprocessing

3.1. Color Image Segmentation

Color information is very important for extraction of the features of topographic maps. There are many color spaces existence nowadays, such as RGB, CIE-LAB, HIS, HSV. The RGB color format is in common use in digital images. The primary reason for this is because it possesses compatibility with computer displays. However, the RGB space has the major drawback in that it is not perceptually uniform. Because of this, uniform quantization of RGB space gives perceptually redundant bins and perceptually uniform holes in the color space. Ordinary distance functions defined in RGB space will be unsatisfactory, because perceptual distance is a function of position in RGB space. Because transformation from RGB space to HSV space is non-linear but easily invertible. So we select HSV color space to segment topographic map. Transformation from RGB space to HSV space can be referred to image processing[16] On topographic map, usually, many of pixels are black or white pixels. The remainders all fall in the chromatic region of the HSV cone. Fig.1 is original topographic map. After thresholding, contour lines image that owns noisy pixels and holes is achieved (see Fig. 2).

3.2. Mathematical Morphological Filtering

As shown in Fig.2, there are many holes and noisy pixels on contour lines. If paper-based topographic map has poor quality, more holes and noisy pixels will appear on contour lines map.

To remove these holes and noisy pixels, we use mathematical morphological hit-or-miss transform and non-homotopic hit-or-miss operations with a family of composite structuring elements to process contour line image. Result image is shown in Fig.3.

3.3. Contour Line Thinning

Thinning operation can decrease amount of data, so we use Zhang's thinning algorithm to thin contour line before clustering. Zhang's thinning algorithm is a fast parallel algorithm[17].

Algorithm 1: Zhang's thinning algorithm

Let P_n is P's 8-neighbors pixel $(n=1,2,...,8)$. P_1 P1 is above P, and from P_1 to P_8 is clockwise arrangement. Point P's value is $I(i,j)$, and P_n 's value is I_n. If P_n is background pixel, $I_n=0$, else $I_n=1$.

Step1. signing boundary points which satisfy following conditions:

(1.1) $2 \le N(i,j) \le 6$;

(1.2) $Mod(i,j)=1$;

(1.3) $I_1 \bullet I_3 \bullet I_7 = 0$;

(1.4) $I_3 \bullet I_5 \bullet I_7 = 0$;

Where, *N*(i, j) is the number of nonzero pixel of P's 8-neighbors; *Mod*(i,j) is times of pixel value varying from 0 to 1 along clockwise in P's 8-neighbors. Deleting signed points after boundary points have been checked.

Step2. Processing is the same as Step1 besides following two steps:

(2.1) $I_1 \bullet I_3 \bullet I_7 = 0$;

(2.2) $I_1 \bullet I_5 \bullet I_7 = 0$;

Step3. If there is no point that satisfies conditions in Step1 and Step2, stop; else, go to Step1.

The ultimate thinning result is shown in Fig.4.

4. Using C-means Algorithm to Look for Initial Points of Contour Lines

There are many contour lines in topographic map. For tracking contour lines utilizing active contour model algorithm, it is necessary to point out some key points as initial boundary points. We use hard c-means algorithm to finish this task.

4.1. C-means Algorithm

The c-means algorithm classifies a data set $X=\{x_1,x_2,\cdots,x_n\} \subset R^p$ into *c* homogeneous groups represented as sets $\tilde{F}_1,\tilde{F}_2,...,\tilde{F}_c$. The objective is to obtain the c-partition $\tilde{F}=\{\tilde{F}_1,\tilde{F}_2,...\tilde{F}_c\}$ by minimizing the function *J*:

$$J(u,v)=\sum_{i=1}^{c}\sum_{k=1}^{n}u_{ik}\,\|x_k-v_i\|^2 \tag{1}$$

where u_{ik} is the membership degree of data x_k to the cluster $\tilde{F}_i$. Pattern matrix is $U=(u_{ik})_{c\times n}$, $V=\{v_i\}\subset R^p$ is a vector comprised of the centroids of the cluster set $\tilde{F}$. $\|x_k-v_i\|^2$ denotes the Euclidean norm between x_k and v_i. By iteratively updating the membership with formula (2) the centroids with formula (3), the algorithm converges to a local minimum of *J*.

$$u_{ik}^{(l+1)}=\begin{cases}1, & i=\arg\min\{\|x_k-v_i^{(l)}\|\}\\ 0, & \text{others}\end{cases} \tag{2}$$

for all *i*, *k*,

$$v_i^{(l+1)}=\frac{\sum_{k=1}^{n}u_{ik}^{(l+1)}x_k}{\sum_{k=1}^{n}u_{ik}^{(l+1)}} \tag{3}$$

for all *I*, where $u_{ik} \in \{0,1\} \quad i=1,2,...,c; k=1,2,...,n$

4.2. Analysis of Clustering

In this section, we discuss some questions existing in clustering process.

A. Choice of size of binary image block

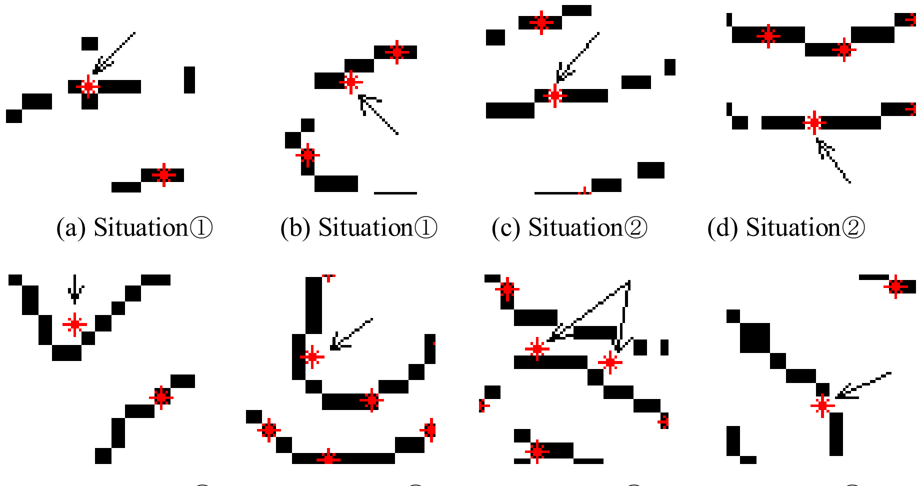
For the centroid of a cluster, distances between the pixels within small region around the centroid and the centroid need to be calculated, because only these pixels belong to this cluster. In fact, distances between all pixels and every centroid are all calculated once iteration. Unfortunately, topographic map is a large sample set. Apparently, if a big image is divided into some small sub-images and clustering on sub-images, amount of participant pixels to calculate distance to a centroid will be reduced. This decreases greatly computation. So we divide a big image into many sub-images which sizes are less than 100×100.

B. Choice of c

In our experiments, the result is relatively insensitive to the choice of *c*. Because density and curvature of contour line in difference topographic map is different, we choice $c=\lfloor Num(\text{data}) / m \rfloor$, $m \in [2\ 15]$. Where *Num*(data) is the number of samples in each cluster. If contour lines density is great or curvature of contour line is big, *m* is evaluated a relatively large value, otherwise, *m* is evaluated a small value.

C. Analysis of clustering mistake

In our cluster result, considerable centroids locate on contour lines, but there are some exceptions. Result is shown in Fig.6(a). These exceptions comprise of following circumstances: ①For 4-neighbors pixels of the centroid, there is not less than two pixels locating on contour lines. The mark '*' denotes centroid in Fig.5 (a), (b). Centroid that arrowhead points to is not locating on contour lines;



(a) Situation① (b) Situation① (c) Situation② (d) Situation②

(e) Situation③ (f) Situation③ (g) Situation③ (h) Situation③

Fig.5. some situations of centroids don't locate on contour line

②Centroids locate on gap of contour line.Fig.5(c),(d); ③Another situations besides the foregoing two kinds of situations. Shown in Fig.5(e),(f),(g),(h). For the foregoing two kinds of situations, we need not to do anything. But for the third situation, it is to be clustered again. To the third situation, first, those pixels belonging to the cluster $\tilde{F}_j$ which corresponding to third situation are got together; Second, new cluster number *c* is chosen as double of the number of centroids belonging to the third situation; Third, repeating cluster until the third situation not appears.

4.3. The Procedures of the C-means Algorithm

The detail procedures for the c-means are listed as follows:

Algorithm 2: Cluster algorithm

Step1. Segmenting objective image into sub-images which size are less than 100×100.

Step2. Initializing vector $v^{(0)} = \{v_1^{(0)}, v_2^{(0)}, \cdots, v_c^{(0)}\}$, parameters *c*, maximal iteration times *T* and threshold ε.

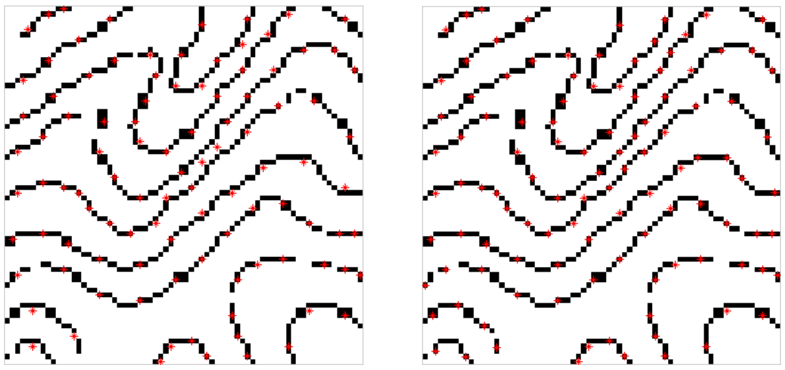
Step3. According to equation (2) updating membership $u_{ik}^{(l+1)}$

Step4. According to equation (3) updating centroid $v_i^{(l+1)}$

Step5. If $\max_i \| v_i^{(l+1)} - v_i^{(l)} \| < \varepsilon$ or $l>T$, stop and label contour pixels of contour lines, Else $l=l+1$, go to Step3.

Step6. Those pixels which accord with situation③ in 4.2 section are reclassified. At the same time, adjusting these pixels's label accordingly.

The ultimate centroids are regarded as initial contour points of contour lines. The result are shown in Fig.6(a), (b). Fig.6(a) is the middle result before centroids reclassified. Fig.6(b) is the ultimate result. In figure, black blocks denote pixels and asterisks denote centroids.



(a) The middle result (b) The ultimate result

Fig.6: Cluster Centroids

5. Extraction of Contour Lines Based on GGVF Algorithm

On topographic maps, it is inevitable existing different element overlap, intercross and aliasing and false colors. Through preprocessing, the resulting lines include many gaps or disconnections. Classical techniques first skeletonize the lines and then try to find relevant extremities of the skeleton for recovering the missing parts of the lines. There are some problems with this approach because the skeletonization is very sensitive to noise and usually creates small branches at the extremities of the lines. These can be easily removed by pruning. However, by doing so, the resulting lines are shorter than the original ones, and therefore some relevant information is lost. To utilize sufficiently topographic map, we propose to extract the contour lines without undergoing skeletonization.

Generalized gradient vector flow (GGVF) can control curve contour features through appropriate restriction applied to deformable curve. GGVF points to image boundary. Because it makes edge attraction flow extending smoothly to the image border, it expands decisive range of attraction flow. In the disconnected object boundary, there is not attraction flow. Attraction flow pointing toward visual concept edge is established in the disconnected object boundary. Simultaneous, GGVF an capture a snake from a long range. In far from concave edges, it establishes attraction flow pointing toward concave edge. The attraction flow can force deformable curve into concave regions. In the following section, we utilizing improved GGVF active contour model to extract contour lines.

5.1. GGVF Active Contour Model

Traditional active contour model, or snake model, is proposed by Kass[18] in 1987. The algorithm is widely applied to image processing and computer vision domains. For example, applying to image boundary detection, region segmentation and skeletonization. Its main idea is: finding out some initial pixels and connecting them into elastic closing lines. Through minimizing energy deformation movements, it achieves initial contours approaching aim contours. There are two considerable limitations about traditional active contour models. First, the initial contour must, in general, be close to the true boundary or else it will likely converge to the wrong result. Second, active contours have difficulty progressing into boundary concavities. To overcome these shortcomings, Xu[19][20] presents generalized gradient vector flow active contour model.

An energy function is firstly defined in Snake algorithm. This energy function mainly consists of internal potential energy and external energy function or image energy function. Let $v(s) = (x(s), y(s))$ is deformable space line nearby control points. Its target energy function is defined as:

$$E_{\text{snake}} = \int E_{\text{int}}(v(s)) + E_{\text{ext}}(v(s))ds \tag{4}$$

where internal potential energy is:

$$E_{\text{int}} = (\alpha\left|v'(s)\right|^2 + \beta\left|v''(s)\right|^2)/2 \tag{5}$$

Active contour algorithm is to discover a closed continuous curve through the movements of deformable space curves. This realizes through making energy E_{snake} reach to minimum $\min E_{\text{snake}}$. To achieve this goal, equation (4) must satisfy Euler equation:

$$\alpha v''(s) - \beta v''''(s) - \nabla E_{ext} = 0 \tag{6}$$

To increase the capture attraction range, replacing $-\nabla E_{ext}$ with $\bar{v}(x,y)$ in equation (6). Defining $\bar{v} = (u(x,y), v(x,y))$ as gradient vector flow (GVF). Then $\bar{v}(x,y)$ needs satisfy minimum energy function:

$$\varepsilon = \iint g(|\nabla f|)\nabla^2 \bar{v} - h(|\nabla f|)(\bar{v} - \nabla f)\,dxdy \tag{7}$$

where: $g(\cdot) + h(\cdot) = 1$.

$f(x, y)$ is edge map derived from the image $I(x, y)$.

On the right of equation (7), the first term is smoothing term that will produce a smoothly varying vector field. The second term is referred as the data term since it encourages the vector field $\bar{v}(x,y)$ to be close to ∇f computed from the data.

5.2. Extraction Algorithm of Contour Lines

To utilize GGVF snake model to extract contour lines, we must define energy functions firstly.

A. Internal Energy

In snake model, internal potential energy in equation (4) includes continuity constraint and smooth constraint. Value of continuity constraint is approximately the quadratic sum of adjacent two points' distance. Energy of continuity constraint decreases ceaselessly resulting in deformable curves moving toward desired edge. In this process, through minimizing energy $\left|v_i - v_{i-1}\right|^2$, attempting to shorten spacing of key points. In some case, some points on contour will be converged at a point resulting from minimum of energy $\left|v_i - v_{i-1}\right|^2$. To satisfy spacing of the points first order continuity without the effect of shrinking, this paper refers to fast Greedy algorithm replacing $\left|v_i - v_{i-1}\right|^2$ with $E_{\text{int}1} = \alpha\left(\bar{d} - \left|v_i - v_{i-1}\right|\right)$.

B. External Energy

Because linear and area features always overlap or crossover on topographic map, there are some gaps (broken contour line) on contour lines. If there isn't other contour line near the gap, GGVF algorithm work correctly on contour line with gaps. However, if there is another contour line near this gap, the curves will move out of the gap. This isn't our desirable result. To this circumstance, Yu brings forward Normalized Gradient Flow Diffusion[21]. In addition, GGVF algorithm only makes use of gradient flow information, the boundary information isn't used directly. So we modify the energy item to overcome noise influence on edges.

$$\varepsilon = \iint g(|\nabla f|)\nabla(\frac{\nabla \bar{v}}{\sqrt{u^2 + v^2}}) - h(|\nabla f|)(\bar{v} - \nabla f)\,I dxdy \tag{8}$$

In equation (8), treating $\bar{v}(x,y)$ as function of time *t*, then equation (8)'s Euler equation is:

$$\begin{cases} \dfrac{\partial u}{\partial t} = g(|\nabla f|)\nabla(\nabla u / \sqrt{u^2 + v^2}) - h(|\nabla f|)(u - f_x)I \\ \dfrac{\partial v}{\partial t} = g(|\nabla f|)\nabla(\nabla v / \sqrt{u^2 + v^2}) - h(|\nabla f|)(v - f_y)I \end{cases} \tag{9}$$

Through solving equation (9), we achieve values of *u* and *v*. then achieve solution of $\bar{v}(x,y)$ as minimum of external energy function. Because contour line takes on linear feature, we select $f = G_\sigma(x,y) * I(x,y)$

$$g(|\nabla f|) = \exp\left(-k|\nabla f|^2\right) \tag{10}$$

C. Contour line extraction algorithm C

Algorithm 3: Contour line extraction algorithm

Step1. Let coordinate of centroid is $(x_i, y_i), i = 1,2,...c$. Count $Sum_i = x_i + y_i$, then ordering Sum_i from small to big and reordering labels of contour line pixels. Taken ordered Sum_i as initial seed points.

Step2. Setting threshold, initializing all parameters.

Step3. Search seed points that locate on the same contour line. According to adjacency of pixels of the same contour line and slope of adjacent points of contour line to judge adjacency of seed points.

Step4. Take adjacent seed points as initial contour points inputting active contour model and iterating.

Step5. When no adjacent seed point is found, judging whether meets broken point. If meeting broken point, recording it.

Step6. After finishing search for the whole topographic map, algorithm executes reconnection program to repair gaps.

5.3. Contour Line Reconnection

Our algorithm can span certain gaps, however, it is inevitable that there exists some gaps on resulting contour lines. Firstly, we find out proper broken point pairs based on analysis of the directional fields of contour lines. Then, utilizing GGVF algorithm to reconnect gaps.

Algorithm 4: Broken contour line reconnection algorithm

Step1. Looking for broken point pairs according to sequence of the recorded broken points. Constructing a square, broken point *b* as its center, *r* as border length. Number m of points of intersection with contour lines is figured out. If *m* is an odd number, we alter *r* magnitude until *m* becomes even. In box region, the directional field of contour lines is calculated[22]. Recording direction as *θ*.

Let ϕ is the average gradient direction. $\phi, \theta \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$.

Define:

$$\phi = \frac{1}{2}\angle\left(G_{xx} - G_{yy}, 2G_{xy}\right) \tag{11}$$

$\angle$ is given by:

$$\angle(x,y) = \begin{cases} \tan^{-1}(y/x) & for \quad x \geq 0 \\ \tan^{-1}(y/x) + \pi & x < 0 \cap y \geq 0 \\ \tan^{-1}(y/x) - \pi & x < 0 \cap y < 0 \end{cases} \tag{12}$$

θ is given by:

$$\theta = \begin{cases} \phi + \pi/2 & for \quad \phi \leq 0 \\ \phi - \pi/2 & \phi > 0 \end{cases} \tag{13}$$

Step2. Record every point of intersection. If *m*=2, let angle between two broken points equate β . If $|\beta - \theta| > threshold$, to enlarge r until $m \geq 4$.

Step3. Calculate angle degrees[23] between *b* point and each point of intersection $a_i \in [0, 2\pi)$, then, calculate:

$$\eta_i = \begin{cases} |\theta - a_i| & \pi/2 > |\theta - a_i| \geq 0 \\ \pi - |\theta - a_i| & for \quad \pi > |\theta - a_i| \geq \pi/2 \\ |\theta - a_i| - \pi & 3\pi/2 > |\theta - a_i| \geq \pi \end{cases} \quad i = 1,2,\cdots,m \tag{14}$$

and

$$\gamma = \min \sum_{n=1}^{m/2} |\eta_i + \eta_j| \quad (i \neq j) \tag{15}$$

In reference[23], when looking for broken point pairs, if space between contour lines is too small, but gaps is relatively large, a wrong judgement will take place. Points on different contour lines are mistaken as on the same contour line. To overcome this error, the required measure is provide in Step 2.

Step4. Judging whether γ reaches to minimum. When γ reaches to minimum, a_i corresponding to b point is regarded as the object point. Treating broken point pairs as seed points, utilizing improved GGVF snake model to reconnect gaps.

6. Experiments and Analysis

Contour lines as linear feature are relatively complex on topographic maps. The efficiency of our method has been tested on 1:50000 topographic maps. Its scanned resolution is 300dpi. Fig.1 is a part of the topographic map. Fig.2 is a binary image resulting from color segmentation. There are many holes on contour lines shown in Fig.3. Fig.4 is an image resulting from mathematical morphological filtering.

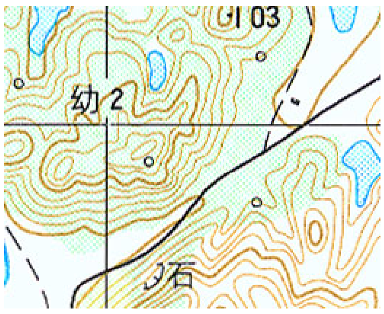


Fig .1: Original map

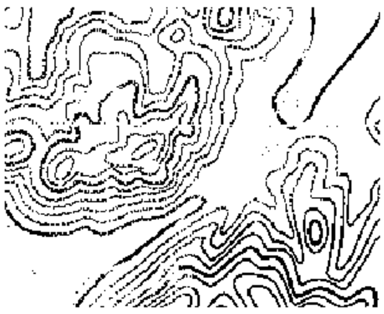


Fig.2: contour line after segmenting

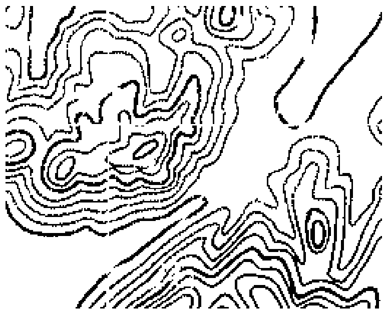


Fig. 3: After morphological filtering

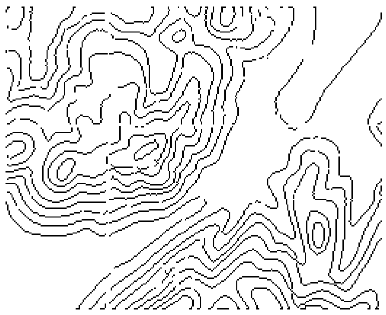


Fig. 4: Thinning result

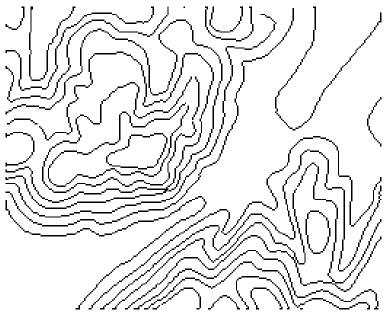


Fig.7: Result of reference[23]

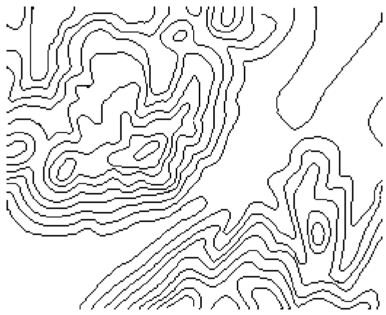


Fig.8: Result of this paper

In Fig.4, most holes on contour lines have been removed. Fig.8 is a resulting image that results from reference [23]. Seeing from Fig.7. Fig.8 is a resulting image that results from proposed algorithm in this paper.

In Fig.7, we discover contour lines which perimeters are relatively small can’t be extracted correctly. Our algorithm gives correct result in Fig.8. Theoretic analysis and test results indicate that the algorithm presented in this paper has a good performance.

7. Conclusion

In this paper, we present a novel algorithm to extract contour lines from scanned color topographic maps. Because strictly automated processing is very difficult in paper-based topographic maps, our approach is interactive in this paper. If original topographic map is poor, it will be difficult to find a proper classification algorithm. For color classification, we utilize the more tractable HSV color space to decrease computational complexity. Mathematical morphological filtering is able to wipe off noise and fill in holes on topographic map. Thinning algorithm decrease computation in c-means algorithm. Improved GGVF active contour model is used to extract contour lines on binary map. In the last, broken contour lines are repaired using GGVF active contour model combining with the directional field of contour lines. The performance manifests our approach validity. Future work is in progress to decrease computation.

8. Acknowledgement

Thanks are given to Dr Guo Ling, Dr Shi Yingchun and other peoples for their help and suggestions.

9. Reference

[1] F. Leberl, D. Olson. Raster scanning for operational digitizing of graphical data. *Photogrammetric Engineering and Remote Sensing*, 1982, **4**:615-627.

[2] D. Greenlee. Raster and Vector Processing for Scanned line work. *Photogrammetric Engineering and Remote Sensing*, 1987, **10**: 1383-1387.

[3] T. Amin, R. Kasturi. Map Data Processing: Recognition of Lines and Symbols. *Optical Engineering*, 1987, **4**: 54-358.

[4] P. Soille, M. Ansoult. Automated Basin Delineation from Digital Elevation Models using Mathematical Morphology [J]. *Signal Processing*, 1990, **20**: 171-182.

[5] L. M. Loh, S. M. Yatim. Extracting Contour Lines from Scanned Topographic Maps. *Proc. of the International Conference on Computer Graphics, Imaging and Visualization*, 2004, 187-192.

[6] F. Dupont, M. Deseilligny and M. Gondran. Terrain Reconstruction from Scanned Topographic Maps. *Proc. Third Int'l Workshop Graphics Recognition*, 1999, 53-60.

[7] P. Gamba, A. Mecocci. Perceptual Grouping for Symbol Chain Tracking in Digitized Topographic Maps, *Pattern Recognition Letters*, 1999, **4**: 355-365.

[8] H. Yamada, K. Yamamoto and K. Hosokawa. Directional Mathematical Morphology and Reformalized Hough Transformation for the Analysis of Topographic Maps. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1993, **15**: 380-387.

[9] H. Yamada, K. Yamamoto and T. Saito, et al. Recognition of Elevation Value in Topographic Maps by Multi-Angled Parallelism. *Pattern Recognition and Artificial Intelligence*, 1994, **8**: 1149-1169.

[10] J. Wu, H. Yan and A. Chalmers. Color Image Segmentation Using Fuzzy Clustering and Supervised Learning. *Electronic Imaging*, 1994, **3**: 397-405.

[11] M. Hedley, H. Yan. Segmentation of Color Images Using Spatial and Color Space Information. *Electronic Imaging*, 1992, **1**: 374-380.

[12] P. Arrighi, P. Soilees. From Scanned Topographic Maps to Digital Elevation Models. *Proc. of Geovision'99, Int. Symposium on Imaging Applications in Geology*, 1999, 1-4.

[13] L. Eikvil, K. Aas and K. Koren. Tools for Interactive Map Conversion and Vectorization. *Proc. of International Conference on Document Analysis and Recognition*, 1995, 14-16.

[14] S. Frischknecht, E. Kanani. Automatic Interpretation of Scanned Topographic Maps: A Raster-Based Approach. *Proc. Second Int'l Workshop, GREC*, 1997, 207-220.

[15] S. Salvatore, P. Guitton. Contour Lines Recognition from Scanned Topographic Maps. *Journal of WSCG 2004*, 2004, 1-3.

[16] J. X. Sun. *Image processing*, Beijing science press, 2004.

[17] T. Y. Zhang, C. Y. Suen. A fast parallel algorithm for thinning digital patterns. *Comm ACM*, 1984, **27**: 236-239

[18] M. Kass, A.Witkin. Snakes: active contour model. *Proc. Of First International Conference on Computer Vision*, 1987, 259-268.

[19] C. Y. Xu, J. L. Prince. Snakes, Shapes and Gradient Vector Flow. *IEEE transactions on Image Processing*, 1998, **3**: 359-369.

[20] C. Y. Xu, J. L. Prince. Generalized Gradient Vector Flow External Forces for Active Contours. *Signal Processing*, 1998, **71**, 131-139.

[21] Z. Y. Yu, C. Bajaj. Vector Normalized Gradient Vector Diffusion and Image Segmentation. *Proc. of 7th European Conference on Computer Vision*, 2002, 517-530.

[22] M. Asker, Bazen and H. Sabih. Systematic Methods for the Computation of the Directional Fields and Singular Points of Fingerprints. *IEEE TPAMI*, 2002, **7**: 905-919.

[23] X. Z. Zhou, H. L. Zhen. Automatic vectorization of Contour lines based on Deformable Model and Field Flow Orientation. *Chinese Journal of Computers*, 2004, **8**: 1056-1063.