Research Article

Identification of Contour Lines from Average-Quality Scanned Topographic Maps

Bin Xu,^{1,2} Jianping Chen,^{1,2} and Meijuan Yao^{1,2}

¹*School of Earth Sciences and Resources, China University of Geosciences (Beijing), Beijing 100083, China*

²*Beijing Key Laboratory of Development and Research for Land Resources Information, Beijing 100083, China*

Correspondence should be addressed to Jianping Chen; 3s@cugb.edu.cn

Received 1 September 2015; Revised 11 December 2015; Accepted 24 December 2015

Academic Editor: Andrzej Swierniak

Copyright © 2016 Bin Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Contour line is the main linear feature on topographic maps. Extraction of contour lines is tedious and time-consuming process, but is still an interesting problem. This paper presents a novel method for extracting contour lines from average-quality scanned topographic maps. First, it uses spatial fuzzy *c*-means algorithm (sFCM) to solve color aliasing and false color problems by taking into consideration both color and spatial information of topographic maps during color segmentation. In order to improve the categorizing rate, upper and lower cut-sets are introduced into sFCM. Second, to deal with the problem of thick lines, node segments are removed before gaps are repaired. Third, different methods are used to repair contour lines gaps according to the causes, which improves the break points matching accuracy. The performance of the method is tested on several topographic maps comparing with other methods, and the results show that the method can avoid misleading results caused by distortion and wrong branches at intersecting regions when using thinning algorithms and have more accurate and higher quality extraction results.

1. Introduction

Topographic maps are carriers of spatial information. Contour lines, as special information signs on topographic maps, reflect regional landform features on topographic maps [1, 2]. It is of great significance to accurately identify contour lines from a topographic map to provide important data for the 3D landform reconstruction. A topographic map usually consists of points, linear and area features, and different features are printed in different colors [3]. Contour lines are brown, smooth, continuous curves [4], usually taking up more than 40% of an entire map, so manual extraction of contour lines is a long and arduous task. Therefore, how to automatically identify contour lines on topographic maps is an urgent problem.

Research on automatic identification and extraction of contour lines on maps has a long history and involves a variety of methods. These methods have produced good results for some high-quality topographic maps, but the results are not satisfactory for low-quality maps, mainly due to the following three problems [5]: (1) color aliasing and

false colors on scanned maps due to poor paper (such as paper turning yellow over time) or printing quality and the performance of the scanner (on a 96-dot-per-inch (DPI) resolution map, most contour lines are 2 to 4 pixels in width; thus, color deviation occupies a large portion in pixels of lines); (2) conglutination of adjacent lines to form thick lines in some areas of a scanned map where contour lines are densely distributed [6]; (3) a large number of contour lines gaps caused by intersecting and overlapping information on a topographic map after color segmentation.

Actually, the former two problems can be merged into one matter, namely, contour lines segmentation. The better the segmentation result is, the easier it can be solved. For the thick lines, the traditional solution is to break the intersecting points on the basis of thinning and then repair gaps. However, the existing thinning algorithms are likely to cause distortion and wrong branches at the intersecting regions, resulting in considerable errors and even mistakes in the extraction results. Therefore, this paper presents a new method for extracting contour lines from average-quality scanned topographic maps. Compared with other methods,

the proposed method has the following features: (1) upper and lower cut-sets are introduced to improve the categorizing rate when using spatial fuzzy c -means (sFCM) algorithm to solve color aliasing and false colors; (2) it deals with the problem of thick lines by removing node segments, with more accurate results; (3) according to the causes of gaps, different methods are used to repair gaps to obtain maps with continuous and complete contour lines.

2. Related Work

Automatic identification of contour lines on scanned topographic maps has been a hot issue, and a great number of related literatures have been published recently. These available literatures generally divided the automatic identification process into four principal steps [7]: (1) scanning of a paper map; (2) color segmentation; (3) thinning and pruning of the binary contour map; and (4) vectorization of contour lines. Steps (2) and (4) are the most crucial steps. Most researchers focused their attention on how to extract clear contour lines and solve the problem of contour lines gaps. Below is an analysis of the existing algorithms in the above two aspects.

2.1. Analysis of Color Segmentation Algorithms. Color features are key elements for the extraction of targets from color pictures. In order to make full use of the color information during color segmentation, Pezeshk and Tutwiler converted the RGB color spaces of topographic maps into CIELAB color spaces and quantified and equalized their brightness histograms to enhance the contrast and improve the color segmentation results [8]. Su et al. converted images into Munsell color spaces by means of nonlinear transformation and acquired the global characteristics through color study [9]. This method not only took into consideration the color features of maps but also used the Markov model to characterize the local features of topographic maps, thus improving color segmentation results and having higher image segmentation rates.

To solve the problem of color deviation and aliasing of contour lines caused in the printing and scanning processes, Khotanzad and Zink used the color key set technique to make up for color distortions on topographic maps so as to improve the image segmentation results. But this method applies only to high-resolution United States Geological Survey (USGS) topographic maps and it is not effective in other low-resolution topographic maps [5]. In order to minimize color key sets, Chen et al. used the eigenvector-fitting algorithm to create a typical color database [10, 11]. Chen et al. used Gaussian kernel functions to build color feature sets for the purpose of achieving a lifelike color distribution effect after the segmentation [12]. They also introduced topological information to characterize the spatial features of contour lines and further improved the color segmentation results by means of relaxation iteration. Test results indicated that these methods could still not fully solve the color deviation and aliasing problems of contour lines.

To improve the color segmentation results, Leyk and Boesch proposed a method based on seeded region growing

(SRG) by making full use of the information of local images, frequency domains, and color spaces [3]. However, it still could not overcome the disadvantage in initial seed selection and the order dependencies of SRG. Wu et al. extracted lines from color maps by means of fuzzy clustering and supervised learning but failed to comprehend color aliasing and false colors inherent to topographic maps [13]. Xin et al. proposed a contour lines extraction method based on gradient directional field by using a c -means algorithm to obtain a set of initial seed nodes, then using these nodes to get the initial contour, and finally using a general gradient vector flow algorithm to extract nonthinned contour lines [14]. This is an effective method for contour lines extraction, but it involves high time complexity. Zheng et al. achieved automatic layer separation of topographic maps by the modified fuzzy c -means (FCM) algorithm that combined spatial and color information of map [15]. This method overcame the defect of most algorithms to consider only the color information during segmentation of topographic maps, with improved segmentation precision and antinoise capability.

2.2. Analysis of Contour Lines Reconstruction Algorithms. Currently, there are roughly three contour lines reconstruction approaches.

2.2.1. The Geometric-Based Approach. This approach features a change from connecting break points of contour lines to reconstructing curves. Spinello and Pascal connected most of break points through filtering the sides of vectorized triangular meshes according to local and global criteria [6]. But such greedy algorithm is unable to bring the best results under the strictest connection criteria. Du and Zhang obtained the spatial topological information of contour lines by means of a dilation algorithm based on mathematical morphology and used such information to match and connect gaps [16]. The said algorithm may produce encouraging results if the gaps are not too big. San et al. constructed an adjacency graph with the information of break points, namely, break points which are vertexes and cost functions based on certain criteria which are sides, and used an A^* algorithm to find the optimal ways to connect the contour lines [17]. This algorithm applies to narrow gaps. For wider gaps, it may result in misconnections.

2.2.2. The Image-Based Approach. This approach categorizes two different pixel points into one group and connects break points in strict accordance with the Gestalt principles of perceptual organization. Arrighi and Soille estimated the conditions for connection by calculating the Euclidean distance and direction between two end points [18]. This method is not applicable to the connection of larger contour lines gaps, and it is likely to cause misconnection of break points between parallel contour lines. Eikvil et al. used the linear tracking method to reconstruct contour lines by setting up sector areas at break points along the contour lines and searching for matchable break points in such areas [19]. But most closed algorithms based on perceptual principles are likely to cause misconnections of contour lines. Huang

FIGURE 1: Flowchart of the proposed algorithm.

et al. proposed an improved method in accordance with the principle of minimal distance or direction by painting contour lines into different colors first and then dividing the map into meshes, searching for break points according to the grid index, and finally matching up break points according to distance and color [20]. Chen et al. presented a local window segmentation approach to overcome the problems of gaps and thick lines by assuming that there was only one natural continuation from an end point while solving the gap problem [11]. The continuation can be found along the direction of the contour lines. However, the gap is crossed by searching from the end point within a sector in the current direction.

2.2.3. The Gradient Vector Flow Approach. This approach is characterized by matching and connection of break points within a gradient vector field constructed according to images of contour lines. Wu et al. used cartographic and geographic knowledge to remove interferences from other geographic layers and connected break points by using the gradient vector flow method [21]. This method works well for connecting break points but operates slowly. Zhou and Zheng used an improved snake algorithm to extract contour lines by filling gaps according to information from gradient vector fields of images and achieved satisfactory results, but this method also operates slowly [22]. Poudroux and Spinello proposed a nonparametric method for connecting broken contour lines by using information of unbroken contour lines after thinning to construct a gradient vector field of an entire map and then searching for break points that match with the gradient vector flow information for contour lines connection [23]. It is based on the global gradient flow information, with a low misconnection rate, but the algorithm is too complicated for practical application.

2.3. Existing Problems and Difficulties. In conclusion, although some progress has been made in the research on the automatic identification and extraction of contour lines on topographic maps, there are problems in the following two aspects:

- (1) Most researchers focused their attention on how to solve the problem of contour lines gaps but ignored the importance of the quality of the extraction

result after color segmentation. Color segmentation as a key step in the identification and extraction of contour lines is of prime importance [24]. An effective segmentation method can not only result in clear, continuous contour lines, but also facilitate the follow-up connection of break points.

- (2) A lack of consideration of the inherent complexity of topographic maps has simplified the identification and extraction of contour lines. A topographic map contains complicated geographic elements that appear in different colors and different forms, so there are problems such as quality degradation, discontinuity, and conglutination after scanning. Most of existing algorithms apply only to scanned high-quality maps; for average-quality scanned maps, these algorithms often result in problems such as misconnection and distortion of contour lines, particularly the problem of conglutination of contour lines.

3. Methodology

In this section, a new algorithm with detailed steps (Figure 1) is presented. The input of this method is a scanned topographic map. (1) The areal elements are removed using the method proposed by Miao et al. [25]. This paper presented a method that separated lines from complicated background in color scanned topographic maps based on energy density and the shear transform. Because linear features and background can be separated from each other based on the difference of the energy densities and shear transform can solve the problem of linear features loss in the process of separation due to the directional limitation (some lines can only be separated in one direction), this method can work well for the removal of areal elements. Thus, it was adopted to get the linear elements map. The method is not described in this paper. Please refer to literature [25] for details. (2) Clustering is made using the improved sFCM. At the same time, the whole features are transformed into a grey version. (3) Special runs are extracted and added into the contour lines layer. (4) Node segments are removed. (5) One-pixel thick lines are obtained by morphological filtering, thinning, and pruning. (6) Break points are connected according to the gray image and node segments.

3.1. Color Segmentation. FCM uses membership to determine the degree of each data point belonging to a certain cluster center for the purpose of automatic categorization [26]. Because the neighborhood information is not considered in the pixels, this algorithm has a relatively weak capability to process intense noises. Neighborhood pixels of an image have similar features [27, 28]. Due to their similarity, it is very possible that they are categorized into the same cluster, so the spatial information is important for the categorization of images with noise. Chuang et al. introduced spatial neighborhood information into FCM and modified the membership function with a new spatial function (sFCM) [29]. Considering that the introduction of spatial neighborhood information adds to the time complexity of the algorithm to some extent, upper and lower cut-sets are introduced to dynamically adjust the convergence rate of elements with varying membership degrees, which raises the convergence rate of those with higher membership degrees and reduces the impact of low membership degrees on cluster centers. Thus, it can improve the categorizing rate.

3.1.1. The sFCM Algorithm. In order to introduce spatial information, a spatial function is defined as follows:

$$w_{ij} = \sum_{k \in \Omega(x_j)} u_{ik}, \quad (1)$$

where $i = 1, 2, \dots, c$, $j = 1, 2, \dots, n$, and $\Omega(x_j)$ is the neighborhood window with x_j being the central pixel. u_{ik} is the membership of pixel x_k to class i ; the value of w_{ij} shows the degree of membership of pixel x_k to class i . If most neighborhood pixels belong to the same class, we will have greater values of w_{ij} at the point.

The sFCM makes use of the spatial features of pixels to modify the membership function of FCM. Then, we obtain a new membership iteration formula as follows:

$$u'_{ij} = \frac{(u_{ij})^p (w_{ij})^q}{\sum_{k=1}^c (u_{kj})^p (w_{kj})^q}, \quad (2)$$

where p and q are used to control the relative importance between the original membership and the spatial function. The iteration function of the new cluster center and objective function are as follows:

$$v_i = \frac{\sum_{k=1}^n (u'_{ik})^m x_k}{\sum_{k=1}^n (u'_{ik})^m}, \quad (3)$$

$$J_m = \sum_{i=1}^c \sum_{k=1}^n (u_{ik})^m \|x_k - v_i\|^2. \quad (4)$$

3.1.2. Upper and Lower Cut-Sets. To improve the categorizing rate, upper and lower cut-sets are introduced; that is, some fuzzier elements in the fuzzy membership matrix are retained and other elements are defuzzified, to allow the sample categorization matrix to have some certainty while the fuzziness in the spatial distribution of the samples is retained for the purpose of improving the categorizing rate

and accuracy. During categorization, when the membership degree μ_{ik} of the pixel at a sample point x_k to a subclass is far greater than the membership degree to other subclasses, it can be taken that sample x_k belongs to class i , and calculations will be simplified for the pixel point when next iteration is performed. Iteration optimization is needed only when the membership degrees to subclasses do not vary significantly and categorization becomes difficult. In practical operation, given the upper cut-set threshold tu and the lower cut-set threshold td , if fuzzy membership $\mu_{ik} > tu$, then let $\mu_{ik} = 1$; if $\mu_{ik} < td$, let $\mu_{ik} = 0$. Elements satisfying $td \leq \mu_{ik} \leq tu$ in the fuzzy membership matrix remain unchanged for further iterative categorization. The set of elements in the membership matrix to be categorized by the upper cut-set parameter tu are called upper cut-set, while the set of elements to be categorized by the lower cut-set parameter td are called lower cut-set.

3.1.3. The Steps of Color Segmentation. Based on the above, the steps of color segmentation are as follows.

Step 1. Initialize the number of clustering classes c and the cluster center $V^{(0)}$; determine the upper cut-set tu and the lower cut-set td ; set the fuzzy weighting exponent m , the end iteration error ϵ , the initial number of iterations ($t = 0$), and the maximum number of iterations $t_{\max}$.

Step 2. Obtain the categorization matrix U^0 using FCM to process the grey map.

Step 3. Perform calculations according to formula (2) for the new membership matrix, and treat it with the upper and lower cut-sets.

Step 4. Perform calculations for the new cluster center according to formula (3).

Step 5. If $\|V^{(t+1)} - V^{(t)}\| < \epsilon$, then the operation of the algorithm ends with an output of the categorization matrix U and the cluster center V ; otherwise, let $t = t + 1$; go to Step 2.

Step 6. Determine the classes of pixels by using the maximum membership conversion method after convergence of the algorithm.

3.2. Extraction of Special Runs

3.2.1. Construction of Runs. In bitmaps, a horizontal (or vertical) line formed by connected pixels of an identical color in a row (or column) is called a run. In grey scale images, a run along the horizontal direction is defined as follows:

$$\text{Run}_{y, x_0, x_1} = \{(x, y) \mid x_0 \leq x \leq x_1, p(x, y) \neq p(x_0 - 1, y), p(x, y) \neq p(x_1 + 1, y)\}, \quad (5)$$

where y is the row in which the run is located; $p(x, y)$ is the pixel value at (x, y) ; let (x_0, y) be the start point, let (x_1, y) be the end point, and the run width is $\text{RunWidth} = x_1 - x_0 + 1$.

FIGURE 2: (a) Connected runs. (b) Adjacent runs.

FIGURE 3: The types of runs.

Assume that a run in row y is Run_{y,x_1,x_2} and another run is Run_{y,x_3,x_4} , if

$$x_2 + 1 = x_3. \quad (6)$$

Then, Run_{y,x_1,x_2} and Run_{y,x_3,x_4} are connected runs (Figure 2(a)). Assume that a run in row y is Run_{y,a_1,a_2} and a run in row $y+1$ is Run_{y+1,b_1,b_2} , if

$$(a_1 \leq b_2) \cap (a_2 \geq b_1). \quad (7)$$

Then, Run_{y,a_1,a_2} and Run_{y+1,b_1,b_2} are adjacent runs, and Run_{y,a_1,a_2} is the predecessor run of Run_{y+1,b_1,b_2} and Run_{y+1,b_1,b_2} is the successor run of Run_{y,a_1,a_2} (Figure 2(b)).

According to the number of predecessor and successor runs [30], there are seven types of runs (Figure 3): (1) singular runs (with no predecessor and successor), (2) beginning runs (with no predecessor and one successor), (3) end runs (with one predecessor and no successor), (4) regular runs (with one predecessor and one successor), (5) merging runs (with more than one predecessor and one successor at most), (6) branching runs (with one predecessor at most and more than one successor), and (7) cross runs (with more than one predecessor and successor, resp.).

3.2.2. Extraction of Special Runs. After color segmentation, contour lines usually are disconnected at intersecting regions with kilometer-scale grids, water systems, roads, and signs. If these intersecting portions are extracted and added into the contour map layer for connection, it would certainly simplify the follow-up connection of break points. Analyzing the topographic maps' features, contour lines gaps caused by intersections with kilometer-scale grids must be treated before treating gaps caused by intersection with other map elements.

A kilometer-scale grid is formed by two sets of parallel lines (vertical and horizontal lines) that run parallel with projection axes. On a topographic map, it has the following features: (1) equal interval distribution in either horizontal

or vertical direction (namely, the line (or column) spacing of adjacent runs is fixed); (2) long coordinate lines, generally disconnected only at labeling or residential areas. The length of the runs is almost the length (or width) of the map. Therefore, horizontal and vertical runs are built, respectively, on a topographic map after color segmentation for easy identification of kilometer-scale grids according to the features of the runs.

After kilometer-scale grids are identified, kilometer-scale grid-related special runs are extracted. Vertical runs formed by horizontal coordinate lines, if one or two connected runs are of the target color (i.e., the color of the contour lines), are regarded as special runs and are extracted. Similarly, special runs formed by vertical coordinate lines are also extracted. Then, special runs related to other elements (water systems, signs, etc.) are extracted. In our opinion, any run with two connected runs of the target color is a special run and should be extracted. All extracted special runs are added into the contour map layer; then, some of the contour lines gaps can be repaired automatically (Figures 8(g), 9(g), and 10(g)).

3.3. Removal of Node Segments

3.3.1. Definition and Categorization of Segments. A segment is a collection of one-to-one simply connected runs. To ensure the univocity of a segment, the following constraints are given:

- (1) Runs are adjacent to each other.
- (2) To ensure the correct establishment of cross, merging, and branching domains, the beginning run is not a branching run or cross run and the end run is not a merging run or cross run.
- (3) There is no abrupt change in the width (i.e., length) of a run.
- (4) Singular runs and cross runs constitute segments independently.

It needs to be noted that if two runs differ greatly in width, consideration should be given to constructing separate segments for them when segments are constructed, despite the one-to-one simple connectivity between them. Assume that the width of the current run is RunWidth_y and the segment to be constructed is composed of $\{\text{Run}_{y_i}, x_1, x_2\}$ and $i = 1, 2, \dots, n$; then the width of the segment is as follows:

$$\text{SegWidth} = \max \{ \text{RunWidth}_{y_i} \}, \quad i = 1, 2, \dots, n. \quad (8)$$

If the width of a run satisfies formula (9), it means that there is a significant change in the width of the run; a new segment should be constructed:

$$\frac{|\text{SegWidth} - \text{RunWidth}|}{\min(\text{SegWidth}, \text{RunWidth})} > \begin{cases} 1 & \min(\text{SegWidth}, \text{RunWidth}) \leq 8, \\ 0.5 & \min(\text{SegWidth}, \text{RunWidth}) > 8. \end{cases} \quad (9)$$

FIGURE 4: Schematic diagrams of a segment. (a) Original. (b) Notational.

FIGURE 5: Relations between segments.

The schematic diagrams of a segment are shown in Figure 4.

A segment is composed of a number of runs, and many segments constitute a topographic map. The adjacency relations between segments are defined as follows: For segments *c* and *d*, if the end run of *c* is adjacent to the beginning run of *d*, then *c* is known as the upper adjacent segment of *d*, and *d* is the lower adjacent segment of *c*, and *c* and *d* have a parent-child relationship. If *c* is an upper adjacent segment of *d* and *c* is also an upper adjacent segment of *e*, then *d* and *e* have a brother-brother relationship (Figure 5).

According to the upper and lower adjacency relations, segments are divided into two types: (1) node segments (having a number of upper or lower adjacent segments), such as segment *c* in Figure 5, and (2) linear segments (having one upper and one lower adjacent segment at most), such as segments *a*, *b*, *d*, and *e* in Figure 5.

3.3.2. Extraction of Node Segments. Node segments correspond to thick lines on the binary map layer. To solve the conglutination problem, node segments should be deleted. According to the constraints for the generation of a segment and the definition of node segment, the steps for extracting

node segments on a binary map (Figure 6) are defined as follows.

Step 1. Find out all runs with more than 1 predecessor run (or successor runs), and traverse the recordset.

Step 2. Obtain the current run.

Step 3. Check if the segment ID of the current run is 0. If not, it means that the current run has been processed; skip to next record and go to Step 2; otherwise, construct a new node segment.

Step 4. Obtain information of the run. If the number of predecessor runs is more than 1, search downwards (or search upwards if the number of successor runs is more than 1) for adjacent runs according to formula (7). If there is no successor (or predecessor) run, go to Step 2.

Step 5. Check for significant changes in the width of the run according to formula (9). If there is a significant change, stop the search and go to Step 2.

Step 6. Check the number of successor (or predecessor) runs. If it is more than 1 or equal to 0, stop the search and go to Step 2.

Step 7. Add the current run into the node segment, modify its segment ID, set the successor (or predecessor) run as the current run, and go to Step 4.

Step 8. This is the end of the node segment extraction process.

In node segments, there is a kind of node segment that has only two adjacent segments. It is formed not by intersecting geographic elements (but probably as a result of larger curvature of the contour lines). This kind of node segment is known as a pseudo-node segment (Figure 7). Since it is not caused by contour lines conglutination, it cannot be deleted.

3.4. Connection of Break Points. Before connecting the two kinds of break points, the binary map is morphologically filtered [18], thinned, and pruned to get one-pixel thick lines [31–33]. The connection of two broken curves depends largely on the curve trend and the distance between the break points. If the curve trend is represented by the angle between the tangent lines at two break points θ , the distance between two break points is d , and the probability of connecting the broken curve is P ; then the functional relationship between P , θ , and d is as follows:

$$P = \lambda_1 |\cos \theta| + \frac{\lambda_2}{d}, \quad (10)$$

where λ_1 and λ_2 are proportional factors. The probability of connecting the broken curve P is in inverse proportion to the distance between the break points and the angle between the tangent lines at the break points θ . The greater the value of P , the greater the probability of connecting the break points. It

FIGURE 6: Flowchart for extraction of node segments.

seems that, in the above two factors, the curve trend is more important, so it is assumed that $\lambda_1 = 0.6$ and $\lambda_2 = 0.4$.

Generally speaking, contour lines gaps are mainly caused by the following: (1) the removal of node segments of thick lines (as shown in red circles 4 and 5 in Figure 9 and 7 and 8 in Figure 10); (2) nonuniform colors of contour lines due to color aliasing and false colors (as shown in red circle 6 in

Figure 9); (3) intersecting or overlapping of contour lines and other map elements (as shown in red circles 1–3 in Figure 8). For contour lines gaps caused by the removal of node segments as described in (1), the break points can be easily found by searching for removed node segments. Searching for matchable break points in the neighboring regions of node segments can improve the processing efficiency. For gaps

FIGURE 7: A pseudo-node segment.

within the rectangular box of a node segment, the probability of connecting any pair of break points is calculated. When the maximum total probability appears, it will be taken as the best scheme for gap connection in the rectangular box of a node segment. For the second one, because contour lines with pairs of end points in the segmented map are continuous in the original map, gaps can be repaired according to the grey map. For the third problem, some gaps will be automatically connected after special runs are added into the contour map layer, and the remainder may be repaired by using the abovementioned methods.

4. Results and Discussions

In order to validate the effectiveness of the proposed method, we used it to extract contour lines from real topographic maps and made qualitative and quantitative evaluations of the extraction results. The results for three maps are shown in Figures 8, 9, and 10.

Figures 8(a), 9(a), and 10(a) are parts of three different topographical maps scanned at a resolution of 96 DPI. They are used to test the overall performance of the proposed algorithm in the treatment of thick lines and gaps, including gaps caused by interesting or overlapping of contour lines and other map elements (as shown in red circles 1–3 in Figure 8), conglutinations caused by densely arranged contour lines (as shown in red circles 4 and 5 in Figure 9 and red circles 7 and 8 in Figure 10), and gaps caused by nonuniform colors of contour lines (as shown in red circle 6 in Figure 9). Figures 8(b), 9(b), and 10(b) are linear maps after removing areal elements using the method proposed by Miao et al. [25], and their grey versions are shown in Figures 8(c), 9(c), and 10(c), respectively. The color segmentation results by the improved sFCM are shown in Figures 8(d), 9(d), and 10(d).

Before color segmentation, a series of parameters must be initialized. We assume that the upper cut-set $tu = 0.8$, lower cut-set $td = 0.2$, and fuzzy weighting exponent $m = 2$. The number of clustering types c and the cluster center $V^{(0)}$ will directly affect the clustering results and convergence rate. If the initial cluster center is approximate to the final convergence result, the convergence rate will be increased substantially and the number of iterations will be reduced

significantly. At the same time, the possibility of being trapped in local optimum is also reduced. Many scholars have done research on the initialization of the number of clustering types c and the cluster center $V^{(0)}$, but unfortunately, there is hardly any simple but effective method proposed so far.

Considering that sFCM can correct the FCM categorization results, in other words, it has less dependence on the initial cluster center $V^{(0)}$, and there are a small number of clusters on the topographic maps after the areal elements are removed, we set the cluster center $V^{(0)}$ by the method of image enhancement [1, 2]. More specifically, if pixels are little different in R, G, and B values, they are either black or white depending upon their values (e.g., all three values are greater than 180; the pixel is regarded as white; otherwise it is black). Except for white and black, pixels with the maximum G and B in their R, G, and B values are regarded as green and blue, respectively. If two values are the same and maximal, the pixel is mapped into green or blue depending on its neighborhood. The others are brown. According to the characters of color, the color topographic maps can be divided into five layers (white, black, green, blue, and brown), and $V^{(0)}$ can be initialized with the average R, G, and B values of each layer. For p and q , the sFCM algorithm with a higher q parameter shows a better smoothing effect according to the research results of Chuang et al. [29]. So, we set $p = 0$ and $q = 2$ and used a 3×3 neighborhood window for the central pixel to ensure smooth images after segmentation. Furthermore, partition coefficient V_{pc} , partition entropy V_{pe} , and compactness and separation V_{xb} are used to evaluate the performance of clustering [29]:

$$\begin{aligned} V_{pc} &= \frac{\sum_j^n \sum_i^c \mu_{ij}^2}{n}, \\ V_{pe} &= \frac{-\sum_j^n \sum_i^c (\mu_{ij} \log \mu_{ij})}{n}, \\ V_{xb} &= \frac{-\sum_j^n \sum_i^c \mu_{ij} \|x_j - v_i\|^2}{n \left(\min_{i \neq k} \{ \|v_k - v_i\|^2 \} \right)}. \end{aligned} \quad (11)$$

The idea of V_{pc} and V_{pe} is that the partition with less fuzziness means better performance. The best clustering is achieved when V_{pc} is maximal or V_{pe} is minimal. V_{xb} measures the featuring property. A good clustering result generates samples that are compacted within one cluster and samples that are separated between different clusters. So, minimizing V_{xb} is expected to lead to a good clustering. The effect and efficiency of the improved sFCM were verified by implementing it in MATLAB and applying it to five different scanned topographic maps. The evaluation results are shown in Table 1. The test environment was an Intel(R) Xeon(R) CPU E5630 @ 2.53 GHz, 12 GB RAM, with a Microsoft Windows 7 Ultimate 64-Bit Operating System.

From Table 1, it can be seen that the classification efficiency of the improved algorithm is higher than that of sFCM. Meanwhile, the segmentation effects of the improved algorithm and sFCM are almost the same. Additionally, compared with the segmentation results using FCM (Figures

FIGURE 8: The extraction result from the first scanned topographic map. (a) Original scanned map. (b) The linear elements map. (c) The gray map. (d) Color segmentation result by the improved sFCM. (e) Color segmentation result by FCM. (f) The contour lines layer. (g) Contour lines layer after merged special runs. (h) Contour lines layer after node segments removal. (i) The resulting thinned contour lines. (j) The final repaired result. (k) Result of Chen et al. (l) Result of Samet et al.

TABLE 1: Evaluation results of different clustering algorithms.

Images sizes	FCM				sFCM				The improved sFCM			
	V_{pc}	V_{pe}	$V_{xb} (\times 10^{-3})$	Time (s)	V_{pc}	V_{pe}	$V_{xb} (\times 10^{-3})$	Time (s)	V_{pc}	V_{pe}	$V_{xb} (\times 10^{-3})$	Time (s)
305 * 196 (Figure 8(a))	0.867	0.274	1.218	1.7	0.907	0.198	1.519	1.8	0.906	0.198	1.521	1.7
378 * 252 (Figure 9(a))	0.886	0.241	0.982	2.5	0.916	0.176	1.082	2.7	0.914	0.177	1.086	2.4
1036 * 719 (Figure 10(a))	0.801	0.360	2.453	31.0	0.857	0.291	2.924	39.5	0.856	0.292	2.925	27.6
2918 * 2251	0.749	0.452	2.834	314.6	0.805	0.357	3.505	375.9	0.802	0.359	3.510	286.2
4241 * 3104	0.819	0.337	2.019	689.2	0.879	0.273	2.708	810.7	0.876	0.275	2.713	574.3

FIGURE 9: The extraction result from the second scanned topographic map. (a) Original scanned map. (b) The linear elements map. (c) The gray map. (d) Color segmentation result by the improved sFCM. (e) Color segmentation result by FCM. (f) The contour lines layer. (g) Contour lines layer after merged special runs. (h) Contour lines layer after node segments removal. (i) The resulting thinned contour lines. (j) The final repaired result. (k) Result of Chen et al. (l) Result of Samet et al.

8(e), 9(e), and 10(e)), the improved sFCM can produce segmented images that are smoother and have better antinoise capabilities because the latter takes into account the relevance of the neighborhood information and provides compensation in the noise region.

As shown in the segmented contour lines layer (Figures 8(f), 9(f), and 10(f)), large amounts of contour lines have gaps. According to analysis, some of the gaps are located in the

intersecting or overlapping regions of contour lines and other map elements. So, we constructed runs on topographic maps after color segmentation, extracted special runs, and added them into the segmented contour lines layer. As shown in Figures 8(g), 9(g), and 10(g), some gaps at the intersecting or overlapping regions of contour lines and other map elements are repaired. Although there are some gaps that are not fully repaired after special runs are added, they are significantly

FIGURE 10: The extraction result from the third scanned topographic map. (a) Original scanned map. (b) The linear elements map. (c) The gray map. (d) Color segmentation result by the improved sFCM. (e) Color segmentation result by FCM. (f) The contour lines layer. (g) Contour lines layer after merged special runs. (h) Contour lines layer after node segments removal. (i) The resulting thinned contour lines. (j) The final repaired result. (k) Result of Chen et al. (l) Result of Samet et al.

narrowed to facilitate the follow-up connection of break points. However, the addition of special runs may lead to new thick lines. So special runs must be added before thick lines are handled.

Densely arranged contour lines are difficult to separate from the background elements and thus are easy to cause thick lines after color segmentation. In such case, if the

intersecting points of contour lines are determined after thinning, this would cause difficulty in the follow-up connection of contour lines and significant errors in the extraction results because existing thinning algorithms are likely to cause distortion and wrong branches at the intersecting regions in the thinning process. Therefore, we solved the thick lines problem by removing the node segments of contour

lines. First, the contour lines layer was converted into a binary map after special runs were added; then the node segments were extracted and deleted (Figures 8(h), 9(h), and 10(h)). The location of the node segments was recorded before being deleted. To improve the processing efficiency, matchable break points were searched within a larger region (e.g., 3 pixels larger than the bounding rectangle of a node segment). To solve the problem of direction dependence of run-length codes, the intersecting points were determined by means of horizontal and vertical scanning in the process of node segment construction. In the meantime, if the adjacent segments of a node segment are singular runs with widths of less than 3, then the segment was deemed to be caused by burs and was excluded from the statistics of adjacent segments. Before connecting the break points, one-pixel thick lines in the contour lines (Figures 8(i), 9(i), and 10(i)) were obtained by morphological filtering, thinning, and pruning.

Besides the above two kinds of gaps, there is another kind of gap caused by nonuniform colors in contour lines

due to aliasing and false colors. Such gaps can be repaired easily according to the grey image. The final results using the proposed method are shown in Figures 8(j), 9(j), and 10(j). As indicated in these figures, the proposed algorithm can better solve the problems of gaps and thick lines and produce maps with complete contour lines.

Figures 8(k), 9(k), and 10(k) and Figures 8(l), 9(l), and 10(l) show the test results using the method proposed by Chen et al. [11] and Samet et al. [7], respectively. The said methods exhibited satisfactory results in Figure 8. However, in Figures 9 and 10, there were some mistakes (as shown in red circle 5 in Figure 9 and red circles 7 and 8 in Figure 10). The reason is that the contour lines were close together and the background region between the contour lines was fuzzified into a brown color, which was difficult to separate, resulting in thick lines and distortion of the contour lines after thinning, hence the misconnections.

Finally, we evaluated the results in completeness, accuracy, quality, and Root Mean Square (RMS) difference according to the following criteria [34]:

$$\begin{aligned}
 \text{completeness} &= \frac{\text{length of matched reference lines}}{\text{length of reference lines}}, \\
 \text{correctness} &= \frac{\text{length of matched extraction lines}}{\text{length of extraction lines}}, \\
 \text{quality} &= \frac{\text{length of matched extraciton lines}}{\text{length of extraciton lines} + \text{length of unmatched reference lines}}, \\
 \text{RMS} &= \sqrt{\frac{\sum_{i=1}^l (d(\text{extr}_i; \text{ref})^2)}{l}},
 \end{aligned} \tag{12}$$

where l is the number of pieces of matched extraction; $d(\text{extr}_i; \text{ref})$ is the shortest distance between the i th piece of the matched extraction and reference lines. The optimum values for completeness, accuracy, and quality all are 1, and the optimum value for RMS is 0. The manually plotted contour lines on axis are regarded as reference lines. Considering most contour lines are 2 to 4 pixels in width on a 96 DPI resolution map, thus, we set the buffer width to be 4 pixels. The evaluation results are shown in Table 2.

As shown in Table 2, there is no much difference in completeness for the extraction results of the three methods, but compared with the results of the method proposed by Chen et al. and Samet et al., the results of the proposed method have higher accuracy. The reason is that their methods solve thick lines problem based on thinning, where distortion and wrong branches are often caused at the intersecting regions in the course of thinning; this will reduce the accuracy and quality of the extraction results. In summary, as shown in the figures and the tables, the proposed method achieves a better performance in contour lines extraction.

In real topographic maps, besides the gaps mentioned above, labels and annotations may also cause contour lines gaps (Figure 11). In such case, the method proposed by Oka

et al. produced encouraging results [35], which will not be discussed in this paper.

5. Conclusions

In this paper, a novel method is proposed for extracting contour lines from average-quality scanned topographic maps based on color segmentation and connection of contour lines gaps. During color segmentation, the improved sFCM is used to solve color aliasing and false colors by taking into consideration both color and spatial information of topographic maps, and upper and lower cut-sets are introduced to improve the categorizing rate. To deal with the problem of thick lines, node segments are removed before gaps are repaired. In the process of contour lines gaps connection, different methods are used to repair contour lines gaps according to the causes in order to improve the break points matching accuracy. Compared with the thinning-based methods, the proposed method has less misconnections and more accurate results. In a word, the proposed method can effectively identify and extract contour lines from average-quality scanned topographic maps.

TABLE 2: Evaluation results of different contour lines extraction methods.

Method	Image label	Completeness	Accuracy	Quality	RMS	Gaps	Misconnections
The proposed method	Figure 8(j)	0.911	0.924	0.854	0.110	28	0
	Figure 9(j)	0.907	0.905	0.837	0.112	24	0
	Figure 10(j)	0.856	0.873	0.768	0.118	192	0
Chen's method	Figure 8(k)	0.920	0.893	0.841	0.112	57	0
	Figure 9(k)	0.878	0.828	0.759	0.120	44	2
	Figure 10(k)	0.837	0.807	0.708	0.127	219	4
Samet's method	Figure 8(l)	0.908	0.912	0.846	0.111	58	0
	Figure 9(l)	0.894	0.881	0.791	0.116	46	0
	Figure 10(l)	0.829	0.815	0.714	0.125	242	6

FIGURE 11: A gap caused by an elevation marker.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work was supported by the special funds project for scientific research of public welfare industry from the Ministry of Land and Resources of the People's Republic of China under Grant 201511079-02.

References

- [1] N. G. Ganpatrao and J. K. Ghosh, "Information extraction from topographic map using colour and shape analysis," *Sadhana*, vol. 39, no. 5, pp. 1095–1117, 2014.
- [2] D. B. Dhar and B. Chanda, "Extraction and recognition of geographical features from paper maps," *International Journal on Document Analysis and Recognition*, vol. 8, no. 4, pp. 232–245, 2006.
- [3] S. Leyk and R. Boesch, "Colors of the past: color image segmentation in historical topographic maps based on homogeneity," *GeoInformatica*, vol. 14, no. 1, pp. 1–21, 2010.
- [4] R. Pradhan, S. Kumar, R. Agarwal, M. P. Pradhan, and M. K. Ghose, "Contour line tracing algorithm for digital topographic maps," *International Journal of Image Processing*, vol. 4, no. 2, pp. 156–163, 2010.
- [5] A. Khotanzad and E. Zink, "Contour line and geographic feature extraction from USGS color topographical paper maps," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 1, pp. 18–31, 2003.
- [6] S. Spinello and G. Pascal, "Contour line recognition from scanned topographic maps," *Journal of Winter School of Computer Graphics*, vol. 12, no. 1–3, pp. 419–426, 2004.
- [7] R. Samet, I. Askerzade, and C. Varol, "An implementation of automatic contour line extraction from scanned digital topographic maps," *Applied Mathematics and Computation*, vol. 9, no. 1, pp. 116–127, 2010.
- [8] A. Pezeshk and R. L. Tutwiler, "Contour line recognition & extraction from scanned colour maps using dual quantization of the intensity image," in *Proceedings of the IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI '08)*, pp. 173–176, IEEE, Santa Fe, NM, USA, March 2008.
- [9] H. H. Su, R. B. Wang, and Z. Q. Zhuang, "A color segmentation algorithm applied in maps with its implementation," *Journal of Computer Aided Design & Computer Graphics*, vol. 11, no. 5, pp. 395–398, 1999.
- [10] Y. Chen and R. S. Wang, "Contour lines recognition from scanned topographic maps," *Computer Aided Engineering*, vol. 14, no. 3, pp. 1–9, 2005.
- [11] Y. Chen, R. Wang, and J. Qian, "Extracting contour lines from common-conditioned topographic maps," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 4, pp. 1048–1057, 2006.
- [12] H. Chen, X.-A. Tang, Y.-M. Yang, and M.-Y. Sun, "Extracting contour lines from scanned military topographical maps," *Journal of System Simulation*, vol. 21, no. 7, pp. 1954–1958, 2009.
- [13] J. Wu, H. Yan, and A. Chalmers, "Color image segmentation using fuzzy clustering and supervised learning," *Electronic Imaging*, vol. 3, no. 4, pp. 397–405, 1994.
- [14] D. J. Xin, X. Z. Zhou, and H. L. Zheng, "Contour line extraction from paper-based topographic maps," *Journal of Information and Computing Science*, vol. 1, no. 5, pp. 275–283, 2006.
- [15] H. L. Zheng, X. Z. Zhou, and J. Y. Wang, "Automatic color segmentation of topographic maps based on the combination of spatial relation information and color information," *Journal of Image and Graphics*, vol. 8, no. 3, pp. 334–340, 2003.
- [16] J. Y. Du and Y. M. Zhang, "Automatic extraction of contour lines from scanned topographic map," in *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS '04)*, vol. 5, pp. 2886–2888, Anchorage, Alaska, USA, September 2004.
- [17] L. M. San, S. M. Yatim, N. A. M. Sheriff, and N. Isrozaiddi, "Extracting contour lines from scanned topographic maps," in *Proceedings of the International Conference on Computer Graphics, Imaging and Visualization (CGIV '04)*, pp. 187–192, IEEE, Penang, Malaysia, July 2004.

- [18] P. Arrighi and P. Soille, "From scanned topographic maps to digital elevation models," in *Proceedings of the International Symposium on Imaging Applications in Geology (GeoVision '99)*, pp. 1–4, Liège, Belgium, May 1999.
- [19] L. Eikvil, K. Aas, and H. Koren, "Tools for interactive map conversion and vectorization," in *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, vol. 2, pp. 927–930, IEEE, Montreal, Canada, August 1995.
- [20] X. L. Huang, P. Hu, and Y. D. Bai, "An improved method for connecting broken contour," *Science of Surveying and Mapping*, vol. 31, no. 1, pp. 111–114, 2006.
- [21] R. Q. Wu, X. R. Cheng, and C. J. Yang, "Extracting contour lines from topographic maps based on cartography and graphics knowledge," *Journal of Computer Science & Technology*, vol. 9, no. 2, pp. 58–64, 2009.
- [22] X.-Z. Zhou and H.-L. Zheng, "Automatic vectorization of contour lines based on deformable model and field flow orientation," *Chinese Journal of Computers*, vol. 27, no. 8, pp. 1056–1063, 2004.
- [23] J. Pouderoux and S. Spinello, "Global contour lines reconstruction in topographic maps," in *Proceedings of the 9th International Conference on Document Analysis and Recognition (ICDAR '07)*, vol. 2, pp. 779–783, Parana, Brazil, September 2007.
- [24] S. Leyk, R. Boesch, and R. Weibel, "Saliency and semantic processing: extracting forest cover from historical topographic maps," *Pattern Recognition*, vol. 39, no. 5, pp. 953–968, 2006.
- [25] Q. G. Miao, P. F. Xu, T. G. Liu, Y. Yang, J. Y. Zhang, and W. S. Li, "Linear feature separation from topographic maps using energy density and the shear transform," *IEEE Transactions on Image Processing*, vol. 22, no. 4, pp. 1548–1558, 2013.
- [26] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, NY, USA, 1981.
- [27] S. C. Chen and D. Q. Zhang, "Robust image segmentation using FCM with spatial constraints based on new kernel-induced distance measure," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 34, no. 4, pp. 1907–1916, 2004.
- [28] P. Wang and H. L. Wang, "A modified FCM algorithm for MRI brain image segmentation," in *Proceedings of the International Seminar on Future BioMedical Information Engineering (FBIE '08)*, pp. 26–29, Wuhan, China, December 2008.
- [29] K.-S. Chuang, H.-L. Tzeng, S. Chen, J. Wu, and T.-J. Chen, "Fuzzy c-means clustering with spatial information for image segmentation," *Computerized Medical Imaging and Graphics*, vol. 30, no. 1, pp. 9–15, 2006.
- [30] K.-Z. Chen, X.-W. Zhang, Z.-Y. Ou, and X.-A. Feng, "Recognition of digital curves scanned from paper drawings using genetic algorithms," *Pattern Recognition*, vol. 36, no. 1, pp. 123–130, 2003.
- [31] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Communications of the ACM*, vol. 27, no. 3, pp. 236–239, 1984.
- [32] E. R. Davies and A. P. N. Plummer, "Thinning algorithms: a critique and a new methodology," *Pattern Recognition*, vol. 14, no. 1–6, pp. 53–63, 1981.
- [33] T. G. Liu, Q. G. Miao, P. F. Xu, J. F. Song, and Y. N. Quan, "Color topographical map segmentation algorithm based on linear element features," *Multimedia Tools and Applications*, pp. 1–22, 2015.
- [34] C. Heipke, H. Mayer, C. Wiedemann, and O. Jamet, "Evaluation of automatic road extraction," in *Proceedings of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences "3D Reconstruction and Modeling of Topographic Objects"*, vol. 32, pp. 47–56, Stuttgart, Germany, September 1997.
- [35] S. Oka, A. Garg, and K. Varghese, "Vectorization of contour lines from scanned topographic maps," *Automation in Construction*, vol. 22, pp. 192–202, 2012.