Contour Line Tracing Algorithm for Digital Topographic Maps

Ratika Pradhan ratika_pradhan@yahoo.co.in
Department of CSE, SMIT, Rangpo, Sikkim, INDIA

Shikhar Kumar shikarkum@gmail.com
Department of CSE, SMIT, Rangpo, Sikkim, INDIA

Ruchika Agarwal ag.ruch@gmail.com
Department of CSE, SMIT, Rangpo, Sikkim, INDIA

Mohan P. Pradhan mohanp_pradhan25@yahoo.com.sg
Department of CSE, SMIT, Rangpo, Sikkim, INDIA

M. K. Ghose mkghose@smu.edu.in
Department of CSE, SMIT, Rangpo, Sikkim, INDIA

Abstract

Topographic maps contain information related to roads, contours, landmarks, land covers and rivers etc. For any Remote sensing and GIS based project, creating a database using digitization techniques is a tedious and time consuming process especially for contour tracing. Contour line is very important information that these maps provide. They are mainly used for determining slope of the landforms or rivers. These contour lines are also used for generating Digital Elevation Model (DEM) for 3D surface generation from any satellite imagery or aerial photographs. This paper suggests an algorithm that can be used for tracing contour lines automatically from contour maps extracted from the topographical sheets and creating a database. In our approach, we have proposed a modified Moore's Neighbor contour tracing algorithm to trace all contours in the given topographic maps. The proposed approach is tested on several topographic maps and provides satisfactory results and takes less time to trace the contour lines compared with other existing algorithms.

Keywords: Topographic map, Contour line, Tracing, Moore neighborhood, Digital Elevation Map (DEM)

1. INTRODUCTION

Topographic map is a type of map that provides detailed and graphical representation of natural features on the ground. Topographic maps conventionally show topography, or land contours, by means of contour lines. These maps usually show not only the contours, but also any significant streams, other water bodies, forest covers, built-up areas or individual buildings (depending on scale) and other features. These maps are taken as reference or base map for many Remote Sensing and GIS based application for generating thematic maps like drainage maps, slope maps, road maps, land cover maps etc. The important and distinct characteristic of these maps is that the earth's surface can be mapped using contour lines. Digitization or vectorization process for generating contour map for a state like Sikkim where there is large variation of slope takes tremendous amount of time and manpower. Many research works are currently being conducted in this field to automate the entire digitization process. Till today, a fully automated digitization process does not provide satisfactory result.

Contour lines are imaginary lines that join points of equal elevation on the earth's surface with reference to mean sea level or curves that connect contiguous points of the same altitude (isohypse). These lines are depicted brown in color in topographic maps, and are smooth and continuous curves with a width of three to four pixels. These lines runs almost parallel or they may be taken as nonintersecting lines except in steep cliffs. However, along with contour line, the topographic maps also contain text information overlaid on these lines. This makes the entire automation of extracting and tracing contour lines from the contour maps more complex and difficult.

Traditional method for vectorization of contour line involves mainly the following steps:

- Scanning paper topographic maps using high resolution scanner.
- Registration of one or more maps with reference to the nearest datum.
- Mosaicing or stitching various topographic maps.
- Vectorization of various contour lines manually using line tracing by rubber band method.
- Feeding depth information for each contour line.
- Generating digital elevation models (DEM) for 3D surface reconstruction.

Uses of computer and digital topographic maps have made the task simpler. Currently research is being carried out on automatic extraction of contour lines from topographic maps that involves following five main tasks.

- Registration of topographic map.
- Filtering for enhancing map.
- Color segmentation for extracting contour lines.
- Thinning and pruning the binary images.
- Raster to vector conversion.

The proposed work suggests a method that efficiently extracts contour lines, performs tracing of contour lines and prepares a database wherein user can feed the height value interactively. In this paper, we have proposed a modified Moore's Neighbor contour tracing algorithm to trace all contours in the given topographic maps. The content of the paper is organized as follows. In section II we have summarized the related work carried out in this area. In section III, we have discussed contour extraction and thinning algorithm. In section IV, we have discussed the original Moore's Neighbor contour tracing algorithm, followed by Modified Moore's Neighbor Algorithm in section V. Result and discussion in section VI provides detail result for study area and comparison of these two algorithms. Finally Conclusion and future scope is given in section VII.

2. RELATED WORK

Many researchers have indulged themselves to come up with a technique to completely automate information extraction from topographic maps. Leberl and Olson [1] have suggested a method that involves the entire four tasks mentioned above for automatic vectorization of clean contour and drainage. Greenle [2] have made an attempt to extract elevation contour lines from topographic maps. Soille and Arrighi [3] have suggested image based approach using mathematical morphology operator to reconstruct contour lines. Most of these procedures fail at discontinuities. Frischknecht [4] have used hierarchical template matching algorithm for extracting text but fails to extract contour lines. Spinello [5] have used geometric properties to recognize the contour line that is based on global topology. It uses Delaunay triangulation to thin and vectorize contour line. Zhou and Zhen [6] have proposed deformable model and field flow orientation method for extracting contour lines. Dongjun et.al [7] has suggested a method based on Generalized Gradient Vector Flow (GGVF) snake model to extract contour lines. In this paper we have extended the work of Dongjun et.al [7] to trace the contour lines more efficiently and automatically using Modified Moore's Neighbor tracing algorithm. It also prepares databases of these contour lines to feed the elevation value interactively. Since the topology of contour lines are well defined i.e. a set of non-intersecting closed lines, it makes the tracing of contour lines simpler.

There exists many contour tracing algorithms - Square tracing, Moore neighbor, Radial sweep, Theo Pavlidis' tracing algorithms[8] etc. but each algorithm has its own pros and cons. Most of these algorithms fail to trace the contour of a large class of patterns due to their special kind of connectivity i.e. contour family of 8 connected patterns (that are not 4 connected). Disadvantage of these algorithms are that they do not trace holes present in the pattern. Hole searching algorithms are first used to extract holes and then tracing algorithms are applied to each hole in order to trace the complete contour. Another problem with this algorithm is defining the stopping criterion for terminating an algorithm.

3. CONTOUR EXTRACTION AND THINNING

Contours are depicted as brown colored line in topographic maps usually of width four to five pixel length. After removing noise in the input images, we have used color segmentation technique to extract all the information given in brown color. There are many color spaces widely used to view digital images but most commonly RGB color space is used for the satellite imagery as it possesses compatibility with the computer displays. Since this color space is not perceptually uniform, selecting range of values for brown color in all the three bands is difficult and does not give satisfactory end result, therefore we have first transformed the satellite imagery from RGB to HSV color space and then color segmentation was performed on HSV color space. The color segmentation algorithm is given below:

ALGORITHM *Color Segmentation on HSV color space*
Input: A square tessellation T containing a connected component P of pixels in HSV color space.

Output: A sequence B(b1, b2, …, bk) of brown colored pixels.

Begin

- Set B to be empty.
- From bottom to top and left to right scan the cells of T until a pixel, s, of P is found.
- Set the current pixel point, c, to s i.e. c = s.
- While c is not in B do
    - If hue_range of c between 0 to 0.11 and saturation_range of c between 0.2 to 0.7
        - Insert c in B.
    - End if
    - Advance c to the next pixel in P.
- End while

End

The segmented information includes contours and altitude information. The filtered or segmented image is then thinned using morphological thinning algorithm [9] given below.

- Divide the image into two distinct subfields in a checkerboard pattern.
- In the first sub-iteration, delete pixel p from the first subfield if and only if the conditions G1, G2, and G3 are all satisfied.
- In the second sub-iteration, delete pixel p from the second subfield if and only if the conditions G1, G2, and G3' are all satisfied.

Condition G1:

$$X_H(p) = 1 \quad (1)$$

where

$$X_H(p) = \sum_{i=1}^{4} b_i \quad (2)$$

$$b_i = \begin{Bmatrix} 1 \text{ if } x_{2i-1} = 0 \text{ and } x_{2i} = 1 \text{ or } x_{2i+1} = 1 \\ 0 \text{ otherwise} \end{Bmatrix} \quad (3)$$

x1, x2, ..., x8 are the values of the eight neighbors of p, starting with the east neighbor and numbered in counter-clockwise order.

Condition G2:

$$2 \leq \min\{n_1(p), n_2(p)\} \leq 3 \quad (4)$$

where

$$n_1(p) = \sum_{k=1}^{4} x_{2k-1} \vee x_{2k} \quad (5)$$

$$n_2(p) = \sum_{k=1}^{4} x_{2k} \vee x_{2k+1} \quad (6)$$

Condition G3:

$$(x_2 \vee x_{3 \vee x_8}) \wedge x_1 = 0 \quad (7)$$

Condition G3':

$$(x_6 \vee x_{7 \vee x_4}) \wedge x_5 = 0 \quad (8)$$

The processed image thus obtained contains broken contour lines, we have used broken contour lines reconnection algorithm [7] based on GGVF to connect the gaps in contour lines.

4. MOORE NEIGHBOR CONTOUR TRACING ALGORITHM

Moore Neighborhood of a pixel, P, is the set of 8 pixels which share a vertex or an edge with that pixel. The basic idea is: - When the current pixel p is black, the Moore neighborhood of p is examined in clockwise direction starting with the pixel from which p was entered and advancing pixel by pixel until a new black pixel in P is encountered. The algorithm terminates when the start pixel is visited for second time. The black pixel walked over will be the contour of the pattern.

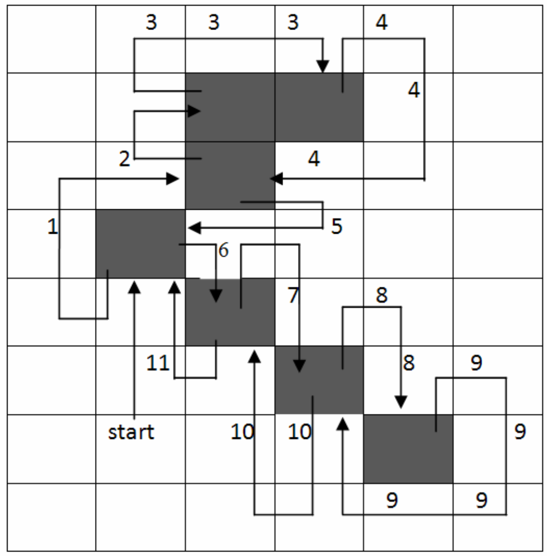

FIGURE 1: Working of Moore's Neighbor tracing algorithm.

The main weakness of Moore Neighbor tracing lies in the choice of stopping criteria i.e. visiting the start pixel for second time. If the algorithm depends on this criterion all the time it fails to trace contour of large family of patterns. Mostly it uses Jacob's stopping criterion i.e.

i. Stop after visiting the start pixel n times, where n is at least 2, or

ii. Stop after visiting the start pixel second time.

Figure 1 demonstrates the working of Moore Neighbor contour tracing algorithm for an input pattern. In figure, line number indicates the iteration number of traversal. For the input pattern, start pixel is encountered three times when the algorithm ends.

5. MODIFIED MOORE NEIGHBOR CONTOUR TRACING ALGORITHM

The original Moore Neighbor tracing algorithm is defined for contours of multiple pixel width. It requires either visiting start pixel 2 times or use Jacob's stopping criteria to terminate the algorithm. In our algorithm the basic idea is: - When the current pixel is black, the Moore neighborhood of P is examined in clockwise direction till no more black pixels are encountered. Then, we move to the start pixel and the Moore Neighborhood of P is examined in an anti-clockwise direction until no new black pixels are left. The algorithm for the Modified Moore's Neighbor tracing is given below:

ALGORITHM *Modified Moore's neighbor algorithm*
Input: A square tessellation T containing a connected component P of black cells.

Output: A sequence B(b1, b2, …, bk) of boundary pixels i.e. the contour line. We define M(p) to be the Moore neighborhood of pixel p, c denotes the current pixel under consideration i.e. c is in M(p).

Begin

- Set B to be empty.
- From bottom to top and left to right scan the cells of T until a black pixel, s, of P is found.
- Insert s in B.
- Set the current boundary point, p, to s i.e. p = s.
- Set c to be the next clockwise pixel in M(p).

- While c is not in B do
  - If c is black
    - o Insert c in B.
    - o Set p=c.
  - End if
  - Advance c to the next clockwise pixel in M(p).
- End while
- Set B to be empty.
- Insert s in B.
- Set p=s.
- Set c to the next anticlockwise pixel in M(p).
- While c is not in B do
  - If c is black
    - o Insert c in B.
    - o Set p=c.
  - End if
  - Advance c to the next anticlockwise pixel in M(p).
- End while

End

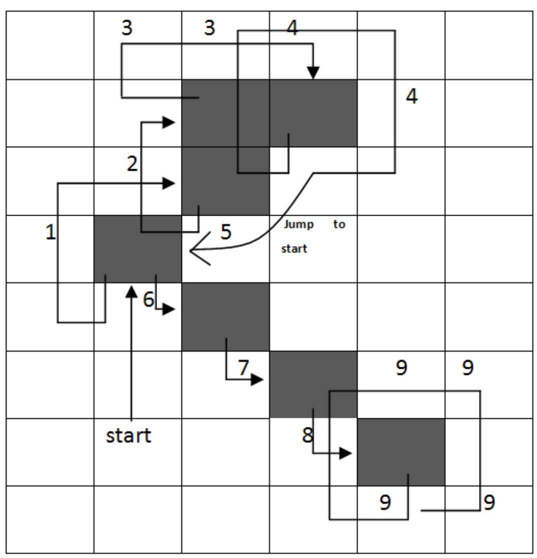

FIGURE 2: Working of Modified Moore's Neighbor tracing algorithm.

Figure 2 demonstrate the working of Modified Moore's Neighbor tracing algorithm. Line number in the figure indicates the pixels from where they are traced from. The algorithm terminates when no more black pixel in an input pattern is left. Unlike original Moore's Neighbor tracing algorithm back tracking is not used here and is not dependent on the stopping criterion used by original Moore algorithm or Jacob stopping criterion. The start pixel is encountered only twice for terminating the algorithm for every pattern.

6. RESULTS AND DISCUSSION

The study area taken into consideration is in and around Majitar, East Sikkim, situated between $27^o09'00"$ and $28^o13'48"$ north latitudes and $88^o29'24"$ and $88^o36'00"$ east longitude. The topographic map for the study area is on scale of 1:250000. Figure 3(a) is the topographic map of the study area. Figure 3(b) is the result of applying color segmentation algorithm. Figure 3(c) is the result of applying broken contour lines reconnection algorithm based on GGVF followed by thinning. 3(d) is the result of Moore Neighbor tracing using Jacob stopping criterion, 3(e) is the result of Modified Moore Neighbor tracing algorithm. Table 1 is the database prepared for the contour map traced using proposed method.

The efficiency of any algorithm entirely depends on the choice of stopping criterion. Original Moore Neighbor tracing algorithm using Jacob stopping criterion that needs N + (n-1) * (N-1) pixels to be traversed, where n is the number of times that the start pixel is visited and N is the number of black pixels that forms a contour line. The choice of scanning anticlockwise after we move to the start pixel in our algorithm is to avoid detection of black pixels already encountered in the clockwise scanning. Since we do not use backtracking, for every detection of black pixel, there is a maximum overhead of checking 6 pixel locations (worst case) before finding a black pixel. Using the Moore-neighbor algorithm, since the algorithm has to retrace the start pixel, there is an overhead of redetection of each and every already traced pixel.

In Modified Moore Neighbor algorithm we have removed the dependency of reaching the start pixel in order to stop the algorithm i.e. start pixel is no longer required as a landmark to indicate the end of algorithm. The proposed algorithm does not require hole searching algorithm to detect holes in the input pattern. The drawback of this algorithm however is consistent checking of every pixel encountered in the Moore Neighbor to decide whether it has been encountered before or not. For very large size images, checking pixels every time could be time consuming and costly. Another disadvantage of the algorithm is that it works only on contour lines of single pixel width. Hence the extracted contour map has to undergo thinning.

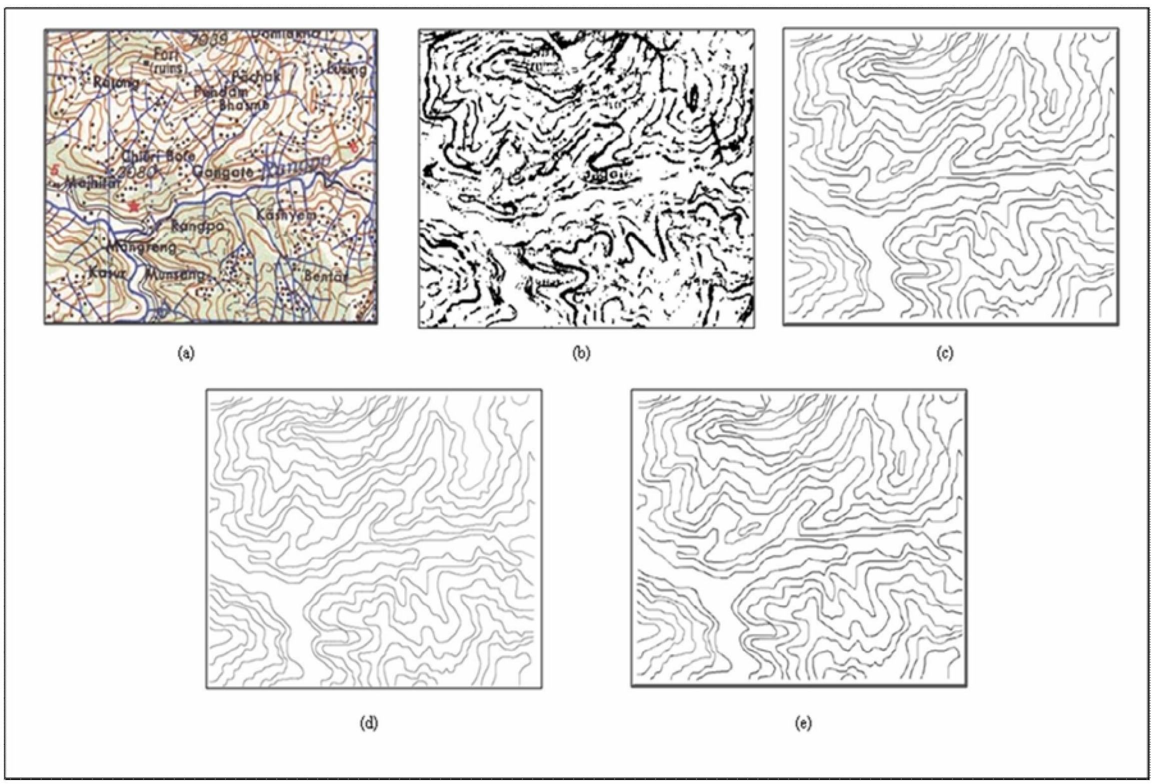

Figure3 a) Topographic map of the study area b) Contour Extraction using Color Segmentation c) Contour reconstructed using broken contour lines reconnection algorithm [7] based on GGVF d) Result obtained using Original Moore's Neighbor tracing algorithms where holes are not detected e) Results obtained using Modified Moore's Neighbor tracing algorithms with detected holes.

No. of contours:	42				
Serial No:	Starting Point		End Point		Elevation
	x	y	X	Y	
1	15	635	16	471	4000
2	15	598	16	494	3600
3	15	562	16	515	3200
4	15	446	52	644	2800
5	15	433	108	646	2400
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.

TABLE 1: Database generated for the result obtained.

7. CONCLUSION AND FUTURE WORK

The Modified Moore Neighbor algorithm works on pre-thinned contour lines (single pixel width). Its efficiency over the original Moore Neighbor algorithm lies in the stopping criterion as the complexity is greatly reduced and hole searching algorithm is not required which further reduces the time complexity. In order to overcome the disadvantage of rechecking black pixels in proposed algorithm, we can check whether the contour line on which the pixel exists has been traced or not rather than checking the pixel. This work can be refined further by automatically extracting altitude value from the topographic sheet by using and automated OCR method.

8. ACKNOWLEDGMENT

We would like to thank All India Council for Technical Education (AICTE) for funding the project title “Contour Mapping and 3D Surface Modeling of State Sikkim” fully sponsored by All India Council of Technical Education, Govt. of India vide order no- 8023/BOR/RID/RPS-44/2008-09. We also like to thank Dr. A. Jeyaram, Head, Regional Remote Sensing Service Centre (RRSSC), IIT campus, Kharagpur for his valuable comments and support.

9. REFERENCES

[1] F. Leberl, D. Olson, “*Raster scanning for operatioal digitizing of graphical data*”, Photogrammetric Engineering and Remote Sensing, 48(4), pp. 615-627,1982.

[2] D. Greenle, “*Raster and Vector Processing for Scanned line work*”, Photogrammetric and Remote Sensing, 53(10), pp. 1383-1387, 1987.

[3] P. Soille, P Arrighi, “*From Scanned Topographic Maps to Digital Elevation Models*”, Proc. of Geovision, International Symposium on Imaging Appications in Geology, pp.1-4,1999.

[4] S. Frischknecht, E. Kanani, “*Automatic Interpretation of Scanned Topographic Maps: A Raster – Based Approach*”, Proc.Second International Workshop, GREC, pp.207-220, 1997.

[5] S. Salvatore, P. Guitton, “Contour Lines Recognition from Scanned Topographic Maps”, Journal of WSCG, pp. 1-3, 2004.

[6] X. Z. Zhou, H. L. Zhen, “*Automatic vectorization of comtour lines based on Deformable model and Field Flow Orirntation*”, Chiense Journal of Computers,vol 8, pp. 1056-1063, 2004.

[7] Dongjum Xin, X. Z. Zhou, H.L.Zhen, “*Contour Line Extraction from Paper- based Topographic Maps*”.

[8] G. Toussaint, Course Notes: Grids, connectivity and contour Tracing <http://jeff.cs.mcgill.ca/~godfried/teaching/pr-notes/contour.ps>.

[9] Lam, L., Seong-Whan Lee, and Ching Y. Suen, "Thinning Methodologies-A Comprehensive Survey," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 14, No. 9, September 1992, page 879.