Automatic Extraction Of Isocontours From Historical Maps

Peter Bajcsy
National Center for Supercomputing Applications
605 East Springfield Avenue, Champaign, IL 61820, USA

ABSTRACT

This paper addresses the problem of automated extraction of constant valued elevation curves (isocontours) from 2-dimensional (2D) images of historical maps. We present a new isocontour extraction method that can operate in a semi automatic or fully automatic mode. The novelty of the work is in developing an isocontour extraction technique for historical map analysis that is unique in the GIS application domain and applicable in web, object recognition and semiconductor defect analysis applications. Three components of the presented extraction process are described including contour initialization, contour tracing and automatic extraction of all isocontours. An experimental comparison of the extraction results obtained using semi automatic and fully automatic modes of the algorithm is shown.

Keywords: Image Processing, Map Analysis, Isocontour Detection

1. INTRODUCTION

The problem presented in this paper is formulated as a computer assisted extraction of isocontours from 2D digital images of historical maps. The isocontours represent horizontal cross sections of 3D terrain at equal elevations [1] and can be classified as curvilinear image features [3] in the image feature space (see Figure 1, left). The overlay of time varying isocontours (see Figure 1, right) provides very valuable piece of information for understanding an elevation changes in time, for investigating an impact of building damns on local ecology, and for preparing any ecological restoration project. The GIS application for our work is the project of backwater restoration of the Illinois River [2].

Nonetheless, there are other applications in web information processing, computer vision and machine vision domains that can directly benefit from an algorithm for automatic extraction of curvilinear features. For example, (1) classification of web-collected images, such as, logos, caricatures or sketches, (2) object recognition for collision avoidance tasks and for navigation tasks of manipulators and robots in computer vision, or (3) defect analysis, such as, scratches on semiconductor wafers or chip packages, in machine vision applications. Although these applications are not the major drivers of our work, they are mentioned to demonstrate the wide applicability of the proposed algorithm.

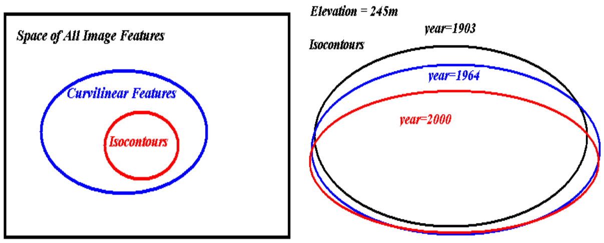


Figure 1: The relative relationship of isocontours and curvilinear features (left). An overlay of constant elevation curves over time (right).

The motivation for developing isocontour extraction methods from 2D images comes from the lack of currently available software tools. None of the commercially available remote sensing or GIS software packages, such as, ENVI [8] or ARC tools [7], contains a tool for automatic isocontour extraction from 2D images. The current solution to the presented problem would be a time consuming and laborious hand tracing of contours, for example, in Adobe Photoshop [9].

Other approach to the problem of curvilinear feature extraction would be to binarize all scanned maps by thresholding and then conduct automatic line following [6, pp. 413-420]. The pitfall of this approach is in the fact that (a) isocontours intersect many other curvilinear features and map symbols, (b) maps contain creases and folds causing isocontour gaps and (c) isocontours have varying local contrast and therefore no single threshold would provide clean separation of contours from the background. We eliminated this approach due to the high cost of required post-processing and fine-tuning for each map.

We propose to approach the problem of isocontour extraction by tracing curvilinear features based on local edge information. At any intersection of curvilinear features, the straight line with the highest contrast score and the smallest angular deviation from the contour tangential direction will be followed. While a contour initiation is manual in the case of semi automatic extraction, the problem of automatic contour initiation and the problem of separating isocontours from other curvilinear features (also called contours in this paper) are addressed in this paper for the case of fully automatic extraction.

Next, we describe the following issues related to automated isocontour extraction from historical maps; (1) finding an initial contour point and contour direction in Section 2, (2) tracing contours in Section 3 and (3) extracting all isocontours in large historical maps automatically in Section 4. The isocontour extraction process is illustrated in Figure 2. Section 5 describes an experimental comparison of isocontour extraction in semi automatic and fully automatic modes. We conclude in Section 6.

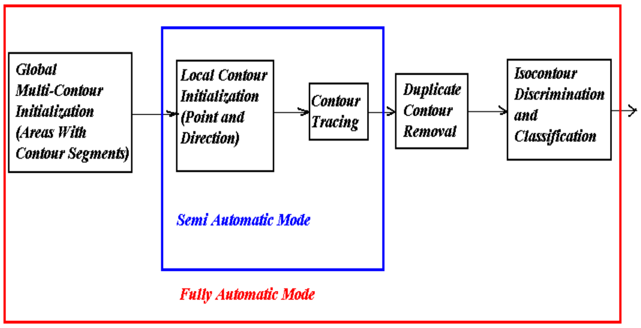

Global Multi-Contour Initialization (Areas With Contour Segments)
Local Contour Initialization (Point and Direction)
Contour Tracing
Duplicate Contour Removal
Isocontour Discrimination and Classification
Semi Automatic Mode
Fully Automatic Mode

Figure 2: The schema of isocontour extraction in two modes.

2. CONTOUR INITIALIZATION

Given a small sub-area of an image, initial contour point and contour direction have to be found. The sub-area of an image can be selected by a mouse click (semi automatic mode) or can be determined automatically as it will be described in Section 4 (fully automatic mode).

A contour initialization process starts with one-directional vertical edge detection defined in Eq. (1). The output of the edge detection is a score that corresponds to the normalized magnitude of the directional derivative. The row and column variables in Eq. (1) are defined with respect to the sub-area coordinate system having the origin in the left upper corner of the sub-area. The I variable represents image intensity.

$$Der = \frac{\partial \sum_{row} I(row, column)}{\partial column} \qquad (1)$$

In the second step, the edge detection is repeated after the sub-area has been incrementally rotated around the left upper corner of the sub-area (see Figure 3 and 4). The rotation angle spans 360 degrees in order to explore a circular image area around the left upper corner of the sub-area. The initial contour direction is selected along the angular direction with the highest edge score. The center of the detected edge with the highest score is the first point of an extracted contour. The contour will be traced in both directions unless the contour is closed, which is detected automatically.

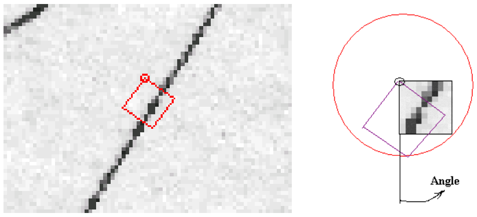


Figure 3: Given an initial point in the image displayed by an overlaid circle, the result of contour initialization is shown with a rectangle (left). The angular rotation of the rectangle is selected by rotating it around the initial point and finding the maximum edge score over the range of angles (right).

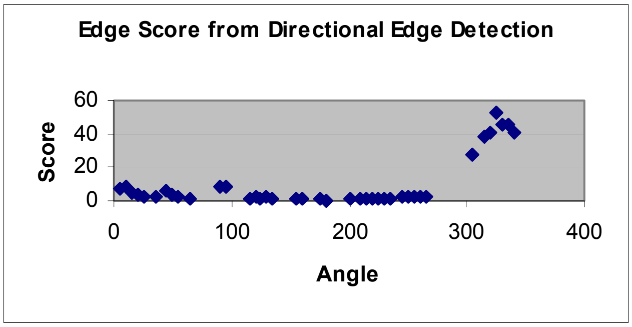


Figure 4: Angular scores from directional edge detection computed for the image sub-areas obtained by rotating a rectangle around an initial point (circle) shown in Figure 3.

3. TRACING CONTOURS

In general, an isocontour extraction algorithm has to be robust to (a) within contour class variations (curvature variations, contour gaps, edge intensity contrast variation or spatially dense isocontours), (b) across class similarities of isocontours with other curvilinear features (dashed lines in a river bed, railroad lines, latitude and longitude straight lines or edges of a geo-registered map) and (c) other background clutter conditions (touching text or other map symbols, intersecting contours with other lines or map artifacts, such as shading due to creases and folds). Some examples are shown in Figure 5 and Figure 6.

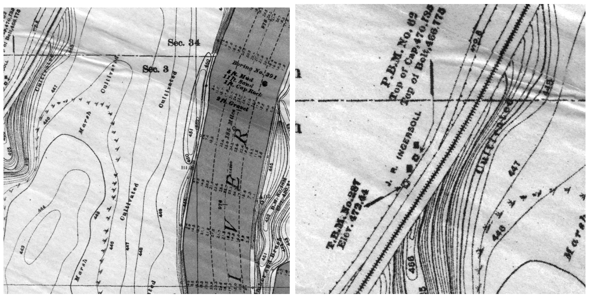


Figure 5: Examples of map regions illustrating the robustness challenges for an isocontour extraction algorithm.

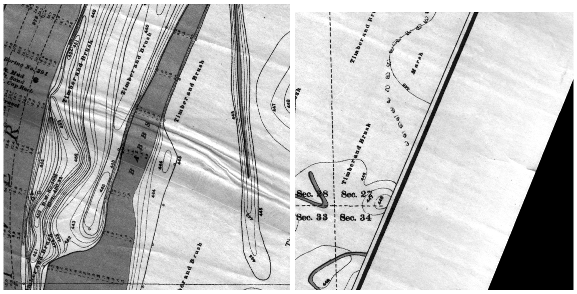


Figure 6: Examples of (a) large curvature isocontours (left), spatially high density of contours (left), edges of a geo-registered map (right), and dashed lines (right) to be eliminated.

3.1. Curvilinear Feature Extraction Algorithm

We have researched an algorithm for tracing isocontours in the presence of all aforementioned conditions given an initial contour point and direction. The underlying approach to tracing isocontours is based on a piece-wise linear approximation of curvilinear contours. This approach uses only local contour information for tracing and therefore it introduces an inherent tradeoff between contour curvature and statistical robustness of the extraction algorithm with respect to the described isocontour variations. The tradeoff comes from the fact that while high-order contour approximations, e.g., quadratic or qubic approximations, give more accurate curvature description, they are notoriously known to be very noise sensitive in digital imagery.

Once a local piece-wise linear approximation has been selected, one has to choose a robust linear approximation method. In this work, we selected a statistical edge detection method used also for the contour initialization and described in Section 2. This edge detection method has proven to be more robust to most of the background conditions than a least square approximation method.

A mathematical model of a contour $C(\vec{v}(t))$ is represented by a set of vectors $\vec{v}$ (linear approximations) that is a function of the contour parameter t, where t represents a contour location.

$$C(\vec{v}(t)); \ \vec{v}(t) = (v(t)*\cos\varphi(t), v(t)*\sin\varphi(t)) \tag{2}$$

We have constrained the contour tracing by imposing a directional continuity constraint according to Eq. 3.

$$\frac{\partial C(\vec{v}(t))}{\partial \varphi(t)} < \varphi_{thresh}; \forall t \tag{3}$$

For now, we assumed that $\vec{v}$ is not a function of sub-area's width or height.

$$\frac{\partial C(\vec{v}(t))}{\partial v(t)} = 0 \tag{4}$$

The implementation of the directional continuity constraint in a discrete image domain takes the form expressed in Eq. (5).

$$\left\| \frac{1}{n}\sum_{i=1}^{n}\varphi(t_i) - \varphi(t_{n+1}) \right\| < \varphi_{thresh}\frac{1}{n} \tag{5}$$

The value φ_{thresh} was set to 90 degrees for the neighboring vectors (n=1) to avoid large directional changes between consecutive line segments. If the sub-area size would change then the value φ_{thresh} would be a function of the sub-area height. This constraint improves robustness of the method with respect to background clutter and intersecting features.

Figure 7 shows a curvature approximation error due to the piece-wise linear model and it also illustrates the robustness of the extraction method with respect to background map annotation ("Timber and Brush" text). In this case, the sub-area size (height and width) is fixed during the tracing.

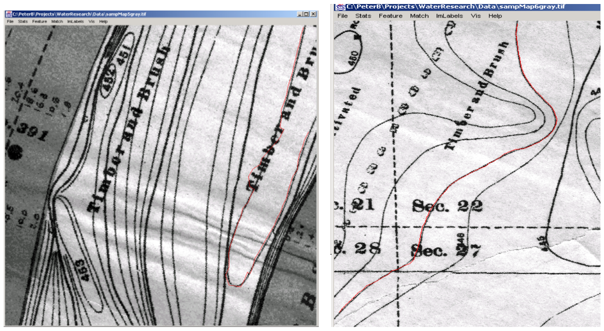


Figure 7: An illustration of the trade-off between accuracy of contour curvature description and robustness with respect to background map annotation (left). An example of robustness improvement of the tracing method by imposing a directional continuity constraint (right).

3.2. Self-Correcting Capability of Extraction Algorithm

Another aspect of the proposed isocontour extraction algorithm is its self-recovery in the case of losing the traced isocontour. The reason for losing a traced contour can be a large contour gap, intersecting line or a high-contrast background clutter. We have identified two ways how to detect that a traced contour was lost. First, the maximum edge score over all possible directions at a contour location t+n dropped below a minimum score value. Second, the sequence of contour vectors (or contour tangential lines) changes directions abruptly over a small contour segment; for example, the sequence is detecting outer edges of a map symbol.

It is desirable to recover from the incorrect tracing direction at a location t+n and search for other acceptable contour directions at a previous contour location t<t+n. Other acceptable contour directions can be found at locations [t, t+n] by exploring edge distribution inside of a sub-area or by varying the sub-area size. In the self-correcting process, we applied the analysis based on a variable sub-area size after performing the analysis based on edge distribution because enlarging a sub-area height significantly reduces the maximum curvature of a traced isocontour. Short and high-contrast spurious edges inside of a sub-area, for instance, due to background clutter, can mislead the local edge detection by undesirable increase of score value, while small gaps in isocontours due to map folds and creases will significantly decrease score values (see Figure 8). To deal with both issues, we analyzed the score distribution as a function of a sub-area rotation angle and as a function of horizontal sub-area strips. If the score distribution over the range of rotation angles is multi-modal then the analysis of horizontal sub-area strips will reveal edge discontinuities based on the score values and edge locations. The mechanism for forming edge distributions (score and location) from horizontal sub-area strips is shown in Figure 9.

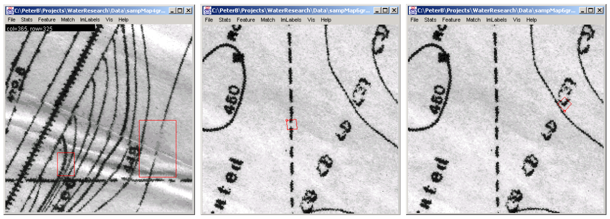


Figure 8: Examples of contour gaps and parallel lines in proximity (left), dashed line (middle) and touching symbol (right) delineated with a rectangular.

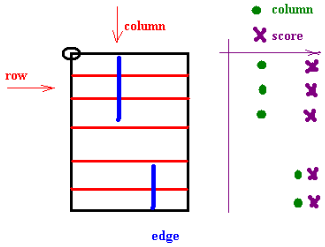


Figure 9: The mechanism for forming edge distributions, such as score and location (column) as a function of a sub-area strip (row).

In order to correct traced directions, we studied edge distributions by considering all edge scenarios inside of a sub-area. There are four basic scenarios of multiple contour segments (and their linear approximations) inside of a sub-area that we studied in detail; (1) co-linear segments, (2) parallel segments, (3) misaligned disconnected segments and (4) misaligned but intersecting segments. The scenarios with a single solid contour segment and no contour segment (or no intensity contrast) are clearly distinguishable and do not create confusion during isocontour extraction. All scenarios are illustrated in Figure 10. The analysis can be summarized in Table 1. It focuses on the number of peaks in the score distribution as a function of a sub-area rotational angle (called angular distribution) and the description of score and location dependency as a function of horizontal sub-area strips (called edge distribution). Table 1 does not contain the case of parallel co-linear segments because we have not found such a case in a dataset of historical maps.

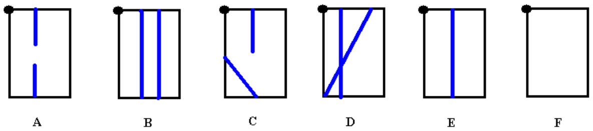


Figure 10: All possible scenarios of contour segments in a sub-area under investigation.

Table 1: The summary of distribution analysis

Scenario	Number of Peaks in Angular Distribution	Edge Distribution	
	Distribution	Description of Score Distribution	Description of Location Distribution
Co-linear segments	1	Piece-Wise Constant	Constant
Parallel solid segments	1	Constant	Piece-Wise Constant
Misaligned disconnected segments	>1	Partially Constant	Partially Constant
Misaligned intersecting segments	>1	Partially Constant	Constant
Single segment	1	Constant	Constant
No discontinuity	0	N/A	N/A

To visualize score and location characteristics in the edge distribution, we selected examples of a single contour segment (see graphs in Figure 4 and Figure 11), co-linear segments (see image in Figure 8, middle, and corresponding graphs in Figure 12) and misaligned disconnected segments (see image in Figure 8, right, and corresponding graphs in Figure 13) and plotted the measured characteristics. The plots in Figure 11-13 support the entries in Table 1.

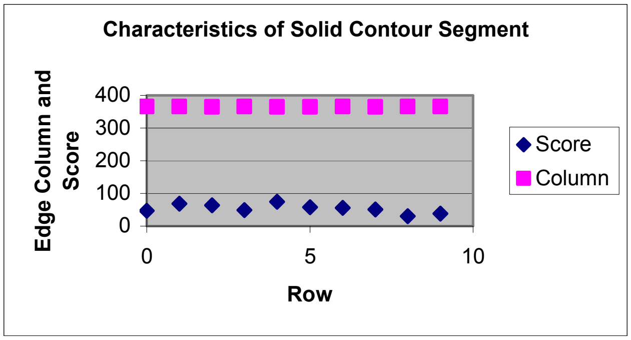
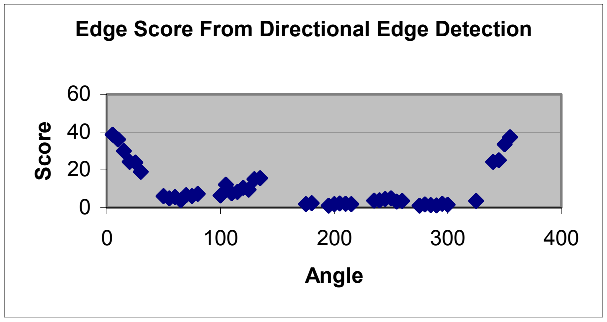


Figure 11: Edge Location and score distributions for a solid contour segment with the corresponding angular scores in Figure 4.



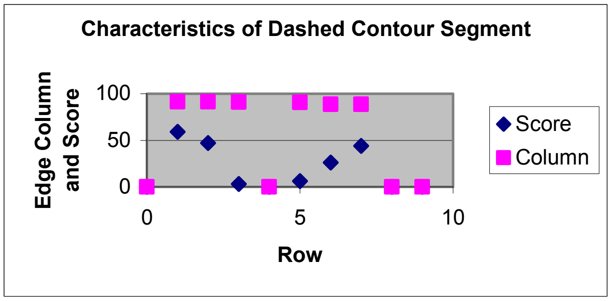


Figure 12: Angular score (top) and edge location with score (bottom) characteristics of a dashed line shown in Figure 8 (middle).

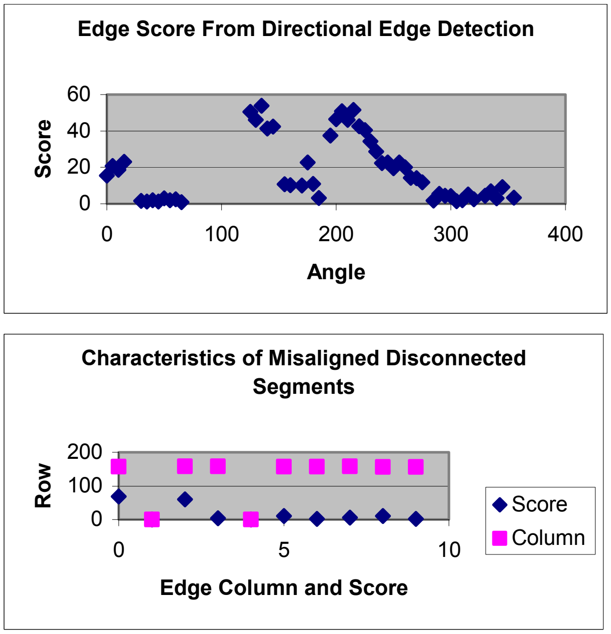


Figure 13: Angular score (top) and edge location with score (bottom) characteristics of a sub-area with touching symbol shown in Figure 8 (right).

Exploring angular and edge distributions enabled correcting a contour direction for the cases of co-linear segments (follow the co-linear direction and overcome the gap), parallel segments (match the direction o the previous segment) and misaligned disconnected segments (move to the end point of the closest discontinuity and search again). Other cases, for instance, large gaps or two intersecting lines with a small directional discrepancy, are treated by doubling the height extend of the sub-area and re-running the analysis.

4. AUTOMATIC EXTRACTION OF ALL ISOCONTOURS

It is assumed that in a semi automatic mode all initial sub-areas (or contour points) are valid and contain a contour segment. Furthermore, in contrast to fully automatic extraction, an extracted contour in semi automatic mode can be evaluated by a user and added or rejected from a set of already found isocontours. However, while the semi automatic mode is suitable for extracting a few isocontours of interest, a fully automatic mode for extraction is needed when several large historical maps with thousands of isocontours must be processed. Besides the large number of isocontours to be extracted, there are also other issues that motivated our development of fully automatic extraction, such as, a map size (it is possible to view only a sub-sampled image) and a fusion issue of several extracted contours from multiple geographically overlapping maps.

An automatic extraction of all isocontours poses challenges on (1) finding all initial contour points (sub-areas of image with contour segments), (2) avoiding extractions of duplicate contours, and (3) discriminating isocontours from other curvilinear features. The developed method resolves the first problem by partitioning an image into a set of small sub-areas and analyzing the distribution of high contrast edges. High-pass filtering of a sub-area and thresholding it by the sample mean plus standard deviation of the filtered sub-area detects high-contrast edges. Thus, the thresholded image contains "high-contrast" edges defined according to Eq. (6), where FT stands for Fourier Transform.

$$
\begin{aligned}
&H(row,col) = FT(I(row,col))\\
&\mu = \frac{1}{N}\sum_{row,col} H(row,col) \qquad (6)\\
&\sigma = \frac{1}{N-1}\sum_{row,col}\left(H(row,col)-\mu\right)^2\\
&HighContrast(row,col): if \ \left\|H(row,col)\right\| > \mu + k*\sigma
\end{aligned}
$$

A spatial distribution of high-contrast edges is evaluated by computing central moments and selecting those sub-areas of an image that contain highly elongated spatial distribution of high-contrast edges. This type of analysis is less computationally intensive than one-dimensional edge detection with an angular sweep of 180 (or 360) degrees. An example of a set of contour initial points that were automatically found is shown in Figure 14.

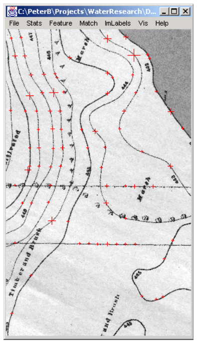


Figure 14: The result of global multi-contour initialization. Crosses denote locations of automatically found initial contour points.

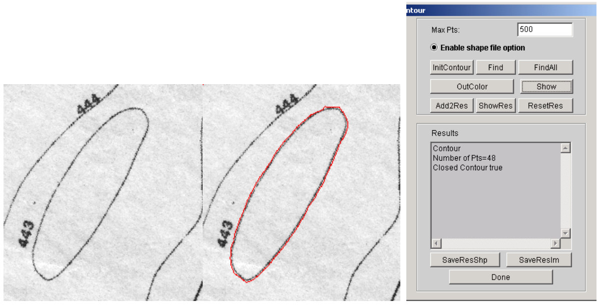
The second problem related to avoiding duplicate contours is addressed by eliminating any sub-area from the set of initial sub-areas that contains a point from already found contours. To minimize required computation for elimination, the maximum number of initial points can be set by a user and initial contour points are searched uniformly over the entire image. It is apparent that one should choose the maximum number of initial points larger than the number of existing contours.

The last problem of discriminating isocontours from other curvilinear features requires (a) identification of unique isocontour features and (b) discrimination and classification of isocontours using the methods described in [3], [4]. This work is still in progress and is not discussed in this paper.

5. EXPERIMENTAL RESULTS

The presented algorithm has been tested in both semi automatic and fully automatic modes. We have developed a user-friendly interface [5] that allows extraction of isocontours one by one with a mouse click initialization or automatically without any human intervention. The results of isocontour extraction are added to a repository of results that can be viewed and saved. Individual isocontours can be color coded manually according to their elevation value. Example results are presented in Figure 15 and Figure 16.

A thorough performance robustness analysis will be conducted in future with synthetic data. Synthetic data will be used to demonstrate the extraction robustness with respect to (a) contour curvature, (b) contour contrast variations and (c) the edge scenarios described in Figure 10.



Max Pts: 500

● Enable shape file option

Results

Contour
Number of Pts=48
Closed Contour true

Figure 15: An example of a closed contour extraction. Original image (left), extracted contour (middle) and the interface demonstrate the input, output and control of the extraction process.

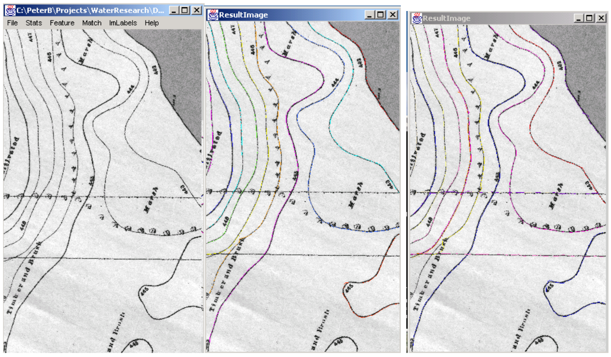


Figure 16: A comparison of the results obtained from the algorithm running in a semi-automatic (middle) or fully automatic (right) extraction mode on a map sub-area (left).

6. SUMMARY

In this paper we have presented a new isocontour extraction method operating in semi automatic and fully automatic modes. The method has been applied to historical map analysis for the GIS project of backwater restoration of the Illinois River. To our knowledge, there is no other commercially available software that would provide similar capabilities. We presented three basic components of the extraction process, described some of the processing challenges and showed experimental results for the two modes of the developed algorithm.

In future, we intend to complete a thorough performance study with synthetic data. Furthermore, we plan to finish our georeferencing capabilities of the developed tool in order to save out contour points in a latitude/longitude coordinate system (or UTM coordinate system) rather than in a pixel coordinate system.

7. REFERENCES

[1] B. Smith and D. M. Mark, "Ontology and Geographic Kinds,“ **Proceedings of International Symposium on Spatial Data Handling (SDH'98)**, Vancouver, Canada, 12-15 July 1998.

[2] J. C. Marlin, "Backwater Restoration Opportunities: Illinois River," **The Governor's Conference on the Management of the Illinois River**, October 4, 2001, http://www.heartlandwaterresources.org/page7.html

[3] R. Duda, P. Hart and D. Stork, **Pattern Classification**,"Second Edition, Wiley-Interscience, 2001.

[4] J. Han and M. Kamber, **Data Mining: Concepts and Techniques**, Morgan Kaufmann Publishers, 2001.

[5] P. Bajcsy, "Image To Knowledge (I2K)," **Software Overview and Documentation**, http://www.ncsa.uiuc.edu/Divisions/DMV/ALG/activities/projects/i2k/documentation/index.html**.**

[6] J. C. Russ, **The Image Processing Handbook**, Third Edition, the CRC Press LLC, 1999.

[7] ESRI, "Arc Tools Software Package for GIS Community," http://www.esri.com/software/index.html

[8] Research Systems, "ENVI Software Package for Remote Sensing," http://www.rsinc.com/envi/

[9] Adobe Systems Incorporated, "Adobe Photoshop Software," http://www.adobe.com/main.html.