Locating Lake and Enclosed Islands in SRTM DEM Data Using Boundaries Obtained from Landsat Imagery via Geodesic Active Contour Model

SHEN Chaomin
Joint Laboratory for Imaging Science & Technology, *and* Dept of Computer Science
East China Normal University
Shanghai 200062, P.R. China
cmshen@cs.ecnu.edu.cn

FAN Jinsong
School of Mathematics and Information Science, Wenzhou University,
Wenzhou, Zhejiang Province 325000, P. R. China
and
Dept of Mathematics, East China Normal University
Shanghai 200062, P.R. China
fjs@wzu.edu.cn

PI Ling
Dept of Mathematics, East China Normal University
Shanghai 200062, P.R. China
plingzh@eyou.com

Abstract: The Shuttle Radar Topography Mission (SRTM) Digital Elevation Model (DEM) data over a lake with more than 1000 islands in China are studied. DEM data may have missing data or be inaccurate over waterbodies because of mirror effect. Thus, locating the shorelines of the lake and enclosed islands from DEM data is non-trivial. We achieve this by overlaying the boundary derived from Landsat imagery of same area. In the Landsat image, the delineation of interior and exterior boundaries of the lake is conducted by the method of modified geodesic contour model. In the Landsat image, the boundary delineation procedure begins by studying the properties of pixels the lake area. This is done by choosing sample pixels visually from the lake area. Principal component analysis is then applied to these sample pixels and the interval estimation of pixel values of the lake is obtained. The delineation of shorelines is an evolution process and the evolution equation is derived according to the theory of active contour model as well as the properties of the lake pixels. We construct an initial contour inside lake and then start evolution according to the evolution equation. The initial contour expands according to the gradient and color information at the location of contour. The active contour procedure terminates when it meets the interior and exterior boundaries of the lake, returning the shorelines. The result is satisfactory. Then the shorelines are overlaid to DEM data according to latitude and longitude.

Keywords: SRTM, DEM, Landsat, geodesic active contour model.

1. Introduction

The Shuttle Radar Topography Mission (SRTM) is a joint project between the National Geospatial-Intelligence Agency (NGA) and the National Aeronautics and Space Administration (NASA) [1]. The SRTM Digital Elevation Model (DEM) data were calculated from the data acquired by the space shuttle Endeavour in February 2000. The DEM data studied here are in a resolution of 3 arc seconds (approximately 90 m).

DEM data have many useful applications. From DEM data, we can easily distinguish features such as mountain and plain areas by elevation values. In this paper, we want to locate shorelines from SRTM DEM data.

Locating shorelines would not be a difficult task if the DEM data were absolutely accurate. Since a lake surface is almost flat, the DEM data values of the lake surface can be assumed to be same. Thus, in the DEM data plane, an area of constant elevation should be easily found and the boundary of this area would be the shoreline. But in the real case, DEM data may have errors (according to NASA, the vertical accuracy is 16 m), and in lake areas, a lot of data are missing because of mirror effect and some DEM values would be higher than the values of real lake surface due to the contribution of higher elevation of islands in the lake. Therefore, an area of constant elevation corresponding to the lake areas does not exist. Thus, other method and information are needed to retrieve shorelines.

Fig.1. Location of study area (the red dot indicates the location of Qiandao Lake in China)

In this paper, we retrieve shorelines from a Landsat image of same area and overlay the Landsat-retrieved shorelines onto the DEM data. The SRTM DEM data are from U.S. Geological Survey (USGS) [1], and the Landsat data are from Global Land Cover Facility (GLCF), University of Maryland [2]. We will first delineate shorelines from the Landsat image using the method of modified geodesic contour model, and then overlay the Landsat-derived shorelines to DEM data according to latitude / longitude. Thus we can locate shorelines in DEM data. The description and implementation of the shoreline delineation algorithm is the main part of this paper.

The study area is Qiandao Lake in Zhejiang Province, China. (Figure 1)

Section 2 describes the general information of Qiandao Lake area. The corresponding Landsat image and SRTM DEM data are shown and some features of these data are discussed.

Section 3 has two parts. The first part describes how to delineate shorelines in Landsat image. Section 3.1 discusses general active models and why they are not applicable to our case. Section 3.2 describes our improved geodesic active contour model. The second part describes how to overlay the derived contour to DEM data.

Section 4 shows the result of the delineated contour from the Landsat image and the DEM data with contour overlaid.

Section 5 concludes the paper and discusses the limitations as well as future work.

2. General information of study area and corresponding Landsat and DEM images

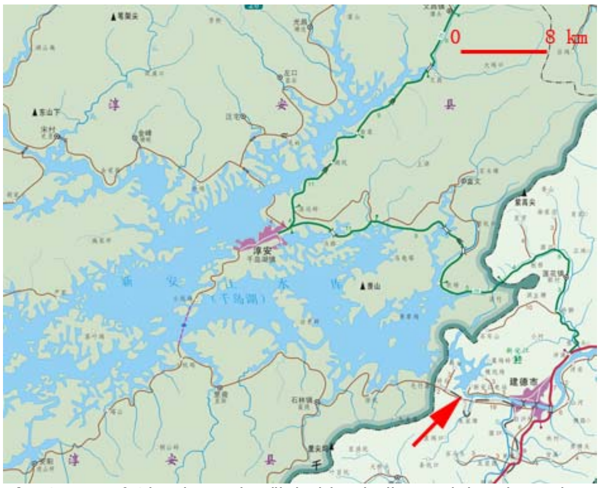

Fig. 2 Map of Qiandao Lake (light blue indicates lake; the red arrow indicates the location of dam)

Our study area is Qiandao Lake. In Chinese, Qiandao Lake means “a lake with 1000 islands”. Actually this is a reservoir built in the 1960s. The area of the reservoir is around 580 km^2, the volume is around 17.84 billion m^3, and the average depth is 34 m if the water level is 108 m. There are 1078 islands (each is bigger than 500 m^2) inside the reservoir [3]. The average water level of the reservoir is around 98 m. The map of this reservoir is shown in Figure 2. The reservoir is shown in light blue in the map, and the red arrow indicates the location of dam. The dam top is at 115 m above sea level and the length of the dam is 465 m. Below the dam is a river called Xin’an River. Now this reservoir is usually called Qiandao Lake. Figure 3 shows the scenery of Qiandao Lake. Figure 3a shows the lake and islands. The exposed soil (yellow in the figure) is caused by to the rising and falling of the water level. Thus there is no vegetation in this area. Figure 3b shows the dam.

Fig. 3a Scenery of Qiandao Lake

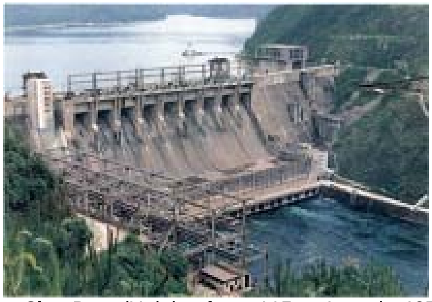

Fig. 3b Dam (Height of top: 115 m; Length: 465 m)

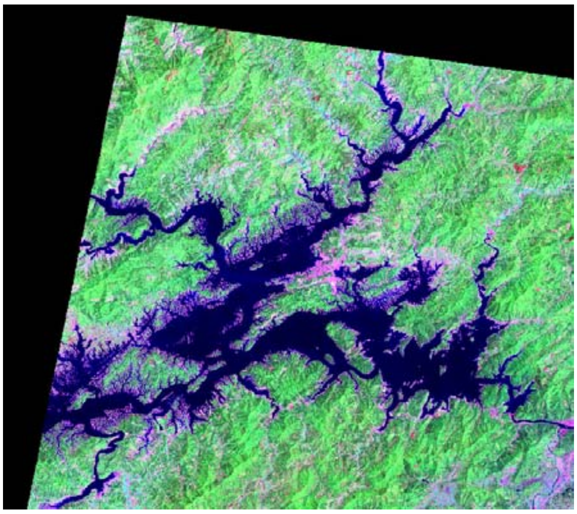

Fig. 4 Landsat image of Qiandao Lake area (2000-05-04)

Figure 4 shows the corresponding Landsat image (Orbit number: path 119, row 040; Date: 2000-05-04; Band combination: 742, Courtesy of USGS). Figure 5 shows the DEM data of same area (Courtesy of USGS).

Figure 5a shows the original DEM data. Compared with Figure 4, it can be observed that a lot of data are missing in lake area. The arrow in Figure 5a indicates the location of the dam. The DEM values of the Xin'an River below the dam are significantly lower than the lake area. The reason is that this is the river area below the dam. Since Figure 5a contains many missing data, various interpolation methods are used to fill those gaps. Figure 5b shows such a result. This was implemented by remote sensing software ENVI.

The DEM values in the lake area are not constant in both Figure 5a and Figure 5b, and in some places those values are obviously wrong. Thus it is hard to get shorelines directly from this DEM data.

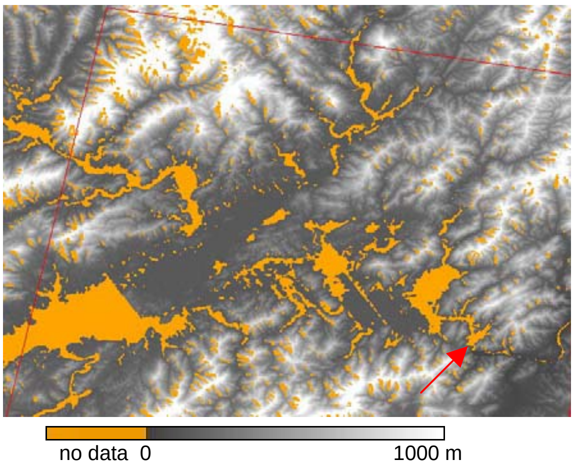

Fig. 5a DEM data with missing values
(The red arrow indicates the location of the dam)

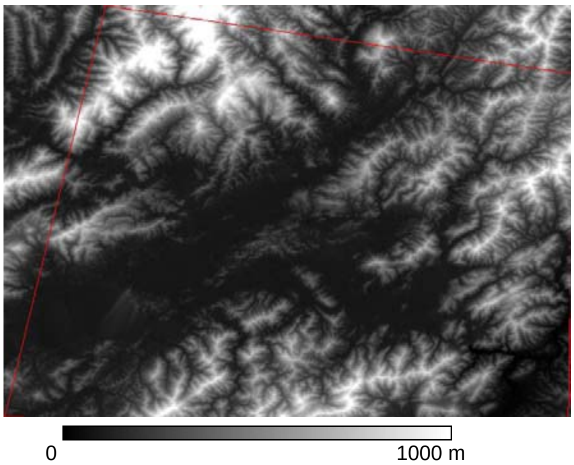

Fig. 5b DEM data after interpolation (processed by ENVI)

So our task is: with the help of Figure 4, locating interior and exterior boundary of Lake in Figure 5b.

3. Description on how to delineate interior and exterior shorelines

We first discuss the mathematical problem concerned in delineating interior and exterior shorelines in Figure 4. Mathematically, our problem is to find a solution on how to delineate the boundaries of Objects Of Interest (OOI).

Roughly speaking, there are two categories of methods for delineating boundaries of objects: edge detection methods and active contour methods. The first category is not suitable for our case because edge detection methods have two disadvantages: 1) the detected edges may not be continuous, and 2) the detected edges may contain many un-wanted points. Therefore, we adapt a method from the second category. Our method is a geodesic active contour method developed from the snake algorithm.

3.1 From snake algorithm to the level set method

The snake algorithm is an active contour method to delineate the boundary of objects, proposed by Kass et al. [4]. The main idea is as follows: Given an initial contour in an image, a corresponding energy can be defined. If the contour changes, the energy will change accordingly. When the energy reaches its minimum, the corresponding contour is regarded as the boundary of object(s).

Regarding our case, the snake algorithm has one problem: it can not handle changes in topology. For example, if the initial contour is a circle, no matter how to evolve, it can not evolve to two circles. Thus, the snake algorithm is only applicable to detecting the edges of objects with known topology. The topology of the lake is unknown before processing, thus we can not assume the topology of the lake is the same as that of the initial contour. Besides the problem of topology, for a given initial contour, say, the boundary of the image, it shrinks during the evolution. When the contour reaches the exterior shoreline, it will terminate evolution. Thus, the snake algorithm can not the detect interior shorelines. For these two reasons, the snake algorithm is not applicable to our case.

The level set method is an improved contour evolution method due to J. Sethian and S. Osher [5]. Better than the snake algorithm, it can handle the topology change in evolution. The main idea of the level set method is to find a family of surfaces $z = \phi(x, y, t)$, where x, y are image coordinates, t is time, such that the intersections of these surfaces and $z = 0$ are the evolving contours. For a given time t, $\phi(x, y, t) = 0$ is the contour in the (x, y) plane. Thus, $\phi(x, y, 0) = 0$ represents the initial contour and $\phi(x, y, t \to \infty) = 0$ tends to the desired contour of objects. The evolving level set function $z = \phi(x, y, t)$ can be obtained by solving its corresponding partial differential equations.

But even the level set method cannot solve our problem directly. The evolved contour may stop at some non-shoreline points. The reason is as follows. In level set theory, the points where the evolved curve should stop marching on is controlled by a stopping function, a function defined at every image point. The stopping function indicates the speed of evolution at that point. If the value of the stopping function is very big, it means the curve will pass that point with that speed along a certain direction. If the speed is reaching 0, it means the curve will stop at that point. So a good stopping function is crucial in the evolution process. But a usual stopping function may not be good enough in our case. Usually, the stopping function is defined as the decreasing function of the gradient and only depends on gradient, i.e., at the location where the gradient is big, the stopping function is small, which means the evolution tends to stop at that location. So the evolution curve of level set method may stop at the place where the gradient is big, but that place may not be OOI at all.

So what we want is: the curve should only stop at the boundary of OOI. At other places, no matter how big the gradient is, the curve should not stop. Thus the color information of the OOI should be considered, i.e., the properties of the OOI pixels should be studied. In Section 3.2, we will discuss how to determine the properties of OOI pixels.

There is another problem in the level set method. It is similar to the problem mentioned for the snake algorithm: when the evolved curve meets the exterior boundary, it will terminate evolution. Assume that the initial curve is the image boundary, after several evolutions, it successfully arrives at the exterior boundary of the lake. We want it shrink again and also find the interior shoreline. But since the stopping function reaches 0 at the exterior shoreline, the curve will not march on. Thus, the interior boundary can not be detected. The Chan-Vese model proposed by Chan and Vese can be used to solve this problem [6]. But in our case, we can avoid this by giving an initial contour inside the lake, then let it expand. During the expansion, when the contour reaches the interior boundaries (islands), it will delineate the islands and goes on, until it reaches the exterior boundary.

3.2 Our algorithm of delineating boundaries of OOI

Our algorithm is presented in Section 3.2. In order to delineating interior and exterior shorelines in the Landsat image, following problems should be solved.

1. determine the properties of lake pixels and give out a criterion whether a pixel is in lake area;
2. give the stopping function at each point in the image. The stopping function should be small if and only if the pixel is in lake area and the gradient is big;
3. give the evolution equation;
4. give the numerical solution of the evolution equation.

In what follows, assume the image $I(x_1, x_2)$ is a color image unless stated. If $x = (x_1, x_2)$, the image also can be expressed by $I(x)$, where $I(x)$ is a vector denoted by $I(x) = (I_1(x), I_2(x), I_3(x)) = (R, G, B)$.

3.2.1 Properties of the lake pixels and interval estimation to the lake pixels

As stated in Section 3.1, we will choose the "expansion" method, i.e., the initial contour is inside the lake and will expand until reaching the shorelines. Thus when the curve evolves to a new point, the speed and direction at that point must be given. Inside the lake area, the speed should be big so that the algorithm can be efficient, and at the boundary the speed should be reaching zero. So we need to give out a criterion to distinguish whether a point is in lake area or not. To achieve this, the properties of the lake pixels should be studied. This is done by choosing some sample points from the lake, and then the properties of lake population are estimated from these samples.

Thus, we first discuss the properties of the pixels in lake area. Since the image is displayed in Red, Green, Blue (RGB for short) combination, we want to set up a criterion that describes whether a point is in lake area using a discrimination function of R, G, and B. For example, one simple criterion might be: "a point is regarded as in lake area if and only if 1) $R \in [a_1, b_1]$, 2) $G \in [a_2, b_2]$ and 3) $B \in [a_3, b_3]$, where a_i, b_i, i=1, 2, 3 are integers". The criterion should be as precise as possible, i.e., "type I error" (omission) and "type II error" (commission) should be avoided.

It is quite intuitive that the above mentioned rectangular parallelepiped may not best describe the lake area. For example, points in the lake area may concentrate along certain line in RGB co-ordinates. Thus, using a rectangular parallelepiped criterion in RGB co-ordinates will definitely cause type II error, which could be avoided by using rectangular parallelepiped criterion in a new co-ordinate system with the concentration line as the new *X*-axis. So if there exists a co-ordinate system better than RGB co-ordinate system, so that in that new system the discrimination function is simple and precise, that co-ordinate system should be used. Principal Component Analysis (PCA) is a way to find out this kind of new system. For the detail of PCA, please refer to [7].

The main procedures of PCA are as follows. In $I(x_1, x_2)$, we choose n sample pixels $(x_{11}, x_{12}, x_{13}), \cdots, (x_{n1}, x_{n2}, x_{n3})$ from the lake. Therefore, $X = (x_{ij})$ is an $(n \times 3)$ matrix. Let $X^* = (x^*_{ij})_{n\times 3}$ be the normalized matrix of X. If the correlation coefficient matrix of X^* is denoted by $\rho = (\rho_{ij})_{3\times 3}$, then the corresponding eigenvectors of ρ are axes of the new co-ordinate system. Assume the eigenvalues are λ_1, λ_2 and λ_3 ($\lambda_1 \geq \lambda_2 \geq \lambda_3$). Corresponding eigenvectors are $\omega_1, \omega_2, \omega_3$, where ω_i ($i = 1,2,3$) is a column. Then a point (x_{i1}, x_{i2}, x_{i3}) in the RGB co-ordinate system can be written as (p_{i1}, p_{i2}, p_{i3}) in the new co-ordinate system, i.e., if let $P = (p_{ij})_{n\times 3}$, then $P = X(\omega_1, \omega_2, \omega_3)$. If a vector is denoted by $a = (a_1, a_2, a_3)$ in the new co-ordinate system, then a_i is called the *i-th* principal component of a ($i = 1,2,3$). The percentage of contribution of the *i*-th principal component to the total sample variance is $e_i = \dfrac{\lambda_i}{\lambda_1 + \lambda_2 + \lambda_3}$, $i = 1, 2, 3$, which measures the importance of the *i*-th principal component. If the percentage of contribution of the first principal component e_1 is big enough, the information from R, G, B can be mostly explained by the first principal component. So we could only consider the first principal component. In what follows, we construct an interval in ω_1 axis by the method of interval estimation. The interval should cover almost all n samples by interval estimation.

Letting $\overline{P}_1 = \frac{1}{n}\sum_{i=1}^{n} p_{i1}, S_{P_1} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(p_{i1} - \overline{P}_1)^2}$, we can use a student's *t*-distribution with $n-1$ degrees of freedom to construct a confidence interval $\left[\overline{P}_1 - \frac{S_{P_1}}{\sqrt{n}} \cdot t_{\frac{\alpha}{2}}, \overline{P}_1 + \frac{S_{P_1}}{\sqrt{n}} \cdot t_{\frac{\alpha}{2}}\right]$ with degree of confidence $1-\alpha$, where $t_{\frac{\alpha}{2}}$ is the *t*-value with $n-1$ degrees of freedom and $0 < \alpha < 1$.

The above procedures can be summarized as: Every pixel in the image is transformed from RGB co-ordinates to a new co-ordinates via PCA. The first component in the new co-ordinates is called first principal component and contains most important information. If the value of first principal component is within the confidence interval, the pixel is regarded in the lake area with a confidence of $1-\alpha$.

Thus for a given pixel x , we could define a function $\gamma(I(x))$, which describes whether this point is in the area or not. Supposing that the confidence interval obtained by PCA is $[a, b]$, and the value of the first principal component is $\tilde{u}(I(x))$, then $\gamma(I(x))$ is defined by

$$\gamma(I(x)) = \begin{cases} 1 & a \le \tilde{u}(I(x)) \le b, \\ 0 & \text{others.} \end{cases}$$

This information is used in the construction of stopping function below.

3.2.2 Stopping function

We shall define our stopping function now.

A stopping function is a decreasing function of gradient. Usually, it only depends on gradient, regardless of other image information. For example, for a grey image $I(x)$,

$$g(x) = \frac{1}{1 + \frac{|\nabla(G_\sigma * I)|^2}{K^2}}$$

is a typical stopping function, where $K > 0$ is a contrast factor, $G_\sigma(x) = \sigma^{-1/2} \frac{1}{2} \exp\left(-\frac{|x|^2}{4\sigma}\right)$ is a Gaussian filter with a parameter σ , ∇ is the gradient operator, and * is the convolution operator.

This kind of stopping function has disadvantages mentioned in 3.1 and it is only for grey images. We want to define a new stopping function for color images. A good stopping function should be small if and only if two conditions are satisfied: 1) the point is in lake area, and 2) that point has a big gradient.

In color images, the concept of gradient is replaced by the largest eigenvalue Λ of the following matrix

$$\begin{pmatrix} 1 + R_{x_1}^2 + G_{x_1}^2 + B_{x_1}^2 & R_{x_1}R_{x_2} + G_{x_1}G_{x_2} + B_{x_1}B_{x_2} \\ R_{x_1}R_{x_2} + G_{x_1}G_{x_2} + B_{x_1}B_{x_2} & 1 + R_{x_2}^2 + G_{x_2}^2 + B_{x_2}^2 \end{pmatrix},$$

where R, G and B represents the pixel’s value of red, green and blue after Gaussian convolution respectively; $R_{x_i} = \partial_{x_i} R$, etc. For the detailed definition of Λ , please refer to [8, 9].

Our stopping function is

$$g(x) = \frac{1}{1 + \gamma(I(x)) \cdot \Lambda^2} . \tag{1}$$

It can be verified that (1) satisfies the two conditions mentioned.

3.2.3 Evolution equation

Let $\vec{C}_t$ be the evolving curve at time t, i.e., $\phi(x,t)=0$ be the implicit representation of $\vec{C}_t$. For our model in its level set formulation, the partial differential equation of $\phi(x,t)$ should satisfy:

$$\begin{cases} \dfrac{\partial \phi}{\partial t} = |\nabla\phi| \operatorname{div}\left(g(x)\dfrac{\nabla\phi}{|\nabla\phi|}\right) + \nu \cdot g(x)|\nabla\phi|, \\ \phi(x,0) = \phi_0(x), \end{cases} \tag{2}$$

where $g(x)$ is defined in (1), ν is a constant called balloon force, and $\phi_0(x)$ is the initial curve (in our case, the initial curve is inside the lake). The balloon force indicates whether the contour will shrink or expand. When the balloon force is less than 0, the contour will expand.

3.2.4 Numerical solution of the evolution equation

In order to solve (2) numerically, the Additive Operator Splitting (AOS) numerical method is used. The AOS scheme is an unconditionally stable numerical scheme for nonlinear diffusion in image processing (see [10] for detail). The iterated formula of (2) is

$$\phi^{k+1} = \frac{1}{2}\sum_{l=\{x_1,x_2\}} \left[E - 2\tau \cdot A_l(\phi^k)\right]^{-1}\left(\phi^k + \tau\nu g(x)\right) \tag{3}$$

where τ is the time step, E is an identity matrix, and $A_l(\phi^k)$ is a matrix corresponding to derivatives along the *l*-th coordinate axis (for the detailed definition of $A_l(\phi^k)$ please refer to [10]).

The exact solutions $\{(x_1,x_2)\}$ of $\phi(x_1,x_2)=0$ should be real numbers. But, because of the discreteness of grid points, we need to find the integer solutions of $\phi(x_1,x_2)=0$. Finding the integer solutions of $\phi(x_1,x_2)=0$ can be done by comparing every pixel (x_1,x_2) with its 8-neighborhood pixels. If we define $\operatorname{sign}(x_1,x_2)=\begin{cases}1 & \text{if} \quad \phi(x_1,x_2)\ge 0 \\ -1 & \text{else.}\end{cases}$, then (x_1,x_2) is regarded as the solution of $\phi(x_1,x_2)=0$ if and only if

$$8\cdot \operatorname{sign}(x_1,x_2) \quad - \sum_{(y_1,y_2)\in \text{ 8-neighbor of } (x_1,x_2)} \operatorname{sign}(y_1,y_2) \quad \ge \quad 0\,,$$

i.e., the sign of $\phi(x_1,x_2)$ changes in the 8-neighborhood of (x_1,x_2)

Thus, the contour $\phi(x_1,x_2)=0$ is found.

For the detailed description for our algorithm of modified geodesic active contour model, please refer to [11].

3.3 Overlaying delineated shorelines to DEM data

Section 3.2 describes the method of delineating shorelines in the Landsat image. In this section the delineated shorelines are overlaid to DEM data.

Since the Landsat image has geographical information, the latitude/longitude of every pixel in the delineated shorelines is known. Therefore the delineated shorelines can be overlaid to DEM data according to latitude/ longitude.

4. Results

Figure 5 shows the delineation result in the Landsat image.

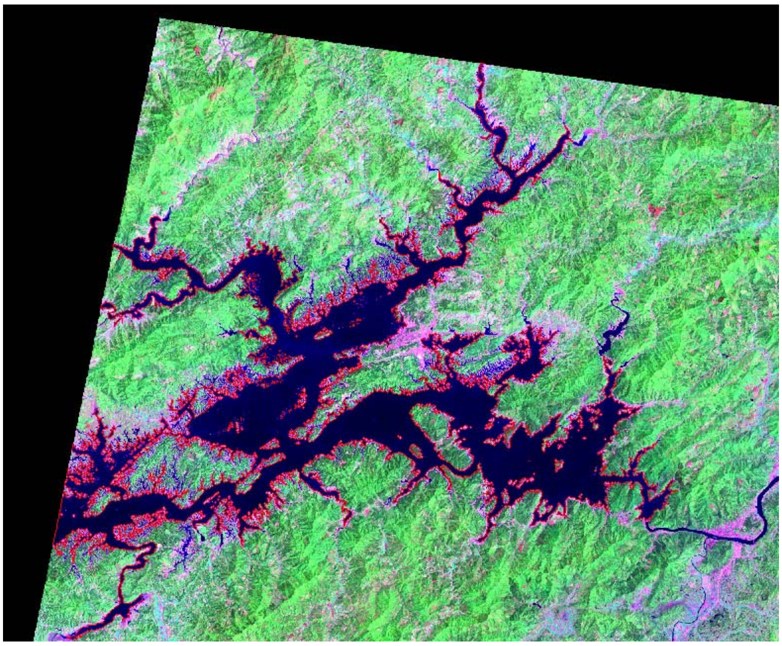

Fig. 5 Delineated Interior and exterior boundaries (shown in red)

Figure 6 shows the delineation result overlaid to the SRTM DEM data.

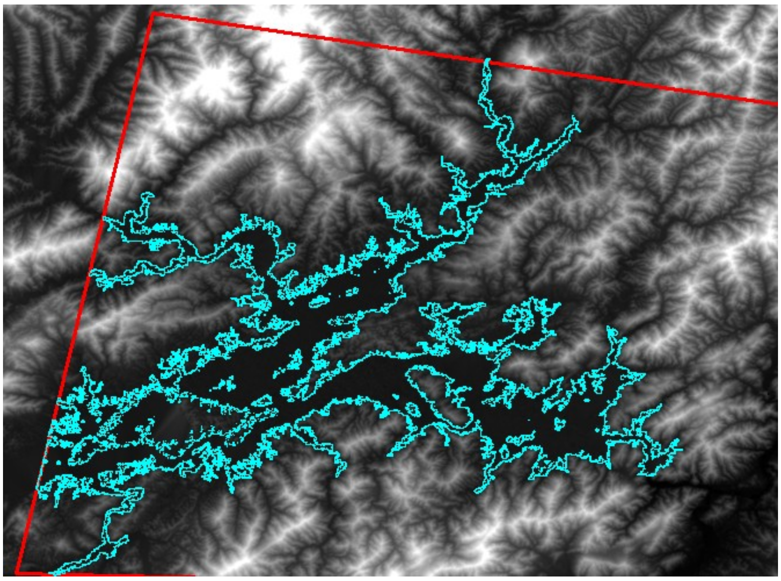

Fig. 6 DEM data with Landsat-delineated shorelines overlaid

The result is accurate. The contour expands and stops exactly at the location of real shorelines. This could be verified by the fact that the contour is stopped at the location of dam.

5. Conclusion

The Shuttle Radar Topography Mission (SRTM) Digital Elevation Model (DEM) data over a Qiandao Lake area in China are studied. Our aim is to locate shorelines from the DEM data. In order to achieve this, firstly the interior and exterior shorelines of the lake are delineated from Landsat data of the same area via the method of modified geodesic contour model. Then the shorelines are overlaid to DEM data according to latitude/longitude.

The main part of this paper describes the algorithm of delineating interior and exterior boundaries of the lake. The properties of lake pixels are studied and a criterion whether a pixel is in lake area is given; Our stopping function for the evolution curve at each point in the image is constructed. The evolution equation and the numerical solution of the equation are given. The delineation result is satisfactory. Then the shorelines are overlaid to DEM data according to latitude and longitude.

Acknowledgement

The authors would like to thank U.S. Geological Survey (USGS) for providing the SRTM DEM data and Global Land Cover Facility (GLCF), University of Maryland for providing the Landsat imagery. The authors also would like to thank Christian Melsheimer for discussion.

References

[1] **URL:** Shuttle Radar Topography Mission Homepage. Available at: http://srtm.usgs.gov/
[2] **URL:** Global Land Cover Facility Homepage. Available at: http://glcf.umiacs.umd.edu/index.shtml
[3] **URL:** Information of Qiandao Lake. Available at: http://www.nettvl.com/zj/xaj.htm
[4] Kass, M., Witkin, A., and Terzopoulos, D., 1988. Snakes: Active contour models, *International Journal of Computer Vision*, vol. 1, 321-331**.**
[5] Osher, S., Sethian, J.A., 1988. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton—Jacobi formulations. *Journal of Computational Physics*, 79, 12-49.
[6] Chan, T., Vese, L., 2001. Active contour without edges for vector-valued image, *Journal of Visual Communication and Image Representation*, Vol. 11, 130-141.
[7] Anderson, T. W., 1984. *An introduction to multivariate statistical analysis (2nd Edition)*, Wiley, New York.
[8] Sochen, N., Kimmel, R., and Malladi, R., 1988. A general framework for low level vision, *IEEE Trans. Image Processing*, vol. 7, 310–318.
[9] Goldenberg, R., Kimmel, R., Rivlin, E., and Rudzsky, M., 2001. Fast geodesic active contours, *IEEE Trans. Image Processing*, vol. 10, 1467-1475.
[10] Kühne, G., Weickert, J., Beier, M., and Effelsberg, W., 2002. Fast implicit active contour models, DAGM 2002, LNCS 2449, 133-140.
[11] Pi, L., Fan, J.S., and Shen, C.M., 2005. Segmentation for objects of interest with modified geodesic active contour, *preprint.*