

# Edge Detection in Remote Sensing Image Based on the Advanced Snake Algorithm

Xueqin Yu<sup>1,2</sup>, Guanshi Li<sup>3</sup>, Tao Zhang<sup>2</sup>, Ling Jiang<sup>4</sup>

1. College of GeoExploration Science and Technology, Jilin University, Changchun, China

2. School of Geomatics and Prospecting Engineering, Jilin Jianzhu University, Changchun, China

3. Foundational Geography Information Center of Jiangsu Province, Nanjing, China

4. School of Resources & Environment, University of Electronic Science & Technology of China, Chengdu, China  
54974729@qq.com

**Abstract**—The paper takes an advanced snake algorithm to make research on the edge detection in order to improve the automatic analysis of the computer and select the objects interested in the remote sensing image. Edge detection by the algorithm is the process of image gradient characteristic, continuity, flatness, and shape information constrained into an energy function. We can achieve the goal of objective selection by calculation the minimum of the energy function. In the water quality monitoring of the Yalu River, the effectiveness of the proposed method and its value of practicality can be confirmed.

**Index Terms**—Advanced snake algorithm, gradient orientation, edge detection, the Yalu River.

## I. INTRODUCTION

Image edge detection is always the hot topic of image processing research. In the 50's, Deneen has detected the edge of uniform distribution by the spatial differential method. In the 60's, Roberts put forward a simple gradient operator to detect the edge. From then on, many people brought up all kinds of methods such as the Facet method which deal with filtering before detection firstly, local fitting intensity function by some model, further statistical analysis and derivation to detect the image edge. However, many methods have their own advantages and disadvantages [1] [2].

In the processing of the remote sensing imagery interpretation, the needs of selecting the objects interested become more and more important. Many Remote Sensing images are multi-wave-band. Therefore, using vector order Prewitt operator is more useful than conventional Prewitt operators in processing the imagery of Remote Sensing. The Snake algorithm is often used when we need a precise contour of an object in image processing field. According to above, this paper introduces an advanced snake algorithm that take the concept of vector order Prewitt operators into the snake algorithm. Furthermore, the method will be more reliable if we consider the shape factor among all the Snake points in the procedure of finding the minimum energy position.

## II. GRADIENT ORIENTATION CALCULATION

### A. Basic Principle

The basic principle is the usage of the vector Prewitt operators. We are already familiar with the conventional

Prewitt operators applied only to one-dimensional images such as gray images and monochrome images. Seeing that the disadvantage of the conventional Prewitt operators, J Scharcanski develop it into vector Prewitt operators. The idea of using vector order Prewitt operators to edge detectors can help us make full use of the color information. Recent studies show that using vector operators to color image processing and analysis can help us make full use of the color information. Another application field of vector Prewitt operator is Remote Sensing image analysis [3]. The light values of the same feature in different band images consist of a spectral characteristics vector  $X$  which is a multi-dimension random vector, so it is necessary to use vector Prewitt operators.

### B. The Value of Gradient Calculation

The vector Prewitt operator is defined in formula (1):

$$\Delta H = [H_- \ 0 \ H_+] \quad \Delta H = [H_- \ 0 \ H_+]^T \quad (1)$$

Vector elements such as  $H_+$ ,  $H_-$ ,  $V_+$ , and  $V_-$  are regarded as the average color vector value of image point  $(x_0, y_0)$  in the window of  $(2\omega+1) \times (2\omega-1)$ . Formula (2) expresses the vector elements.

$$H_+(x_0, y_0) = \frac{1}{\omega(2\omega+1)} \sum_{y=y_0-\omega}^{y=y_0+\omega} \sum_{x=x_0+1}^{x=x_0+\omega} c(x, y)$$

$$H_-(x_0, y_0) = \frac{1}{\omega(2\omega+1)} \sum_{y=y_0-\omega}^{y=y_0+\omega} \sum_{x=x_0-1}^{x=x_0+1} c(x, y)$$

$$V_+(x_0, y_0) = \frac{1}{\omega(2\omega+1)} \sum_{y=y_0+1}^{y=y_0+\omega} \sum_{x=x_0-\omega}^{x=x_0+\omega} c(x, y)$$

$$V_-(x_0, y_0) = \frac{1}{\omega(2\omega+1)} \sum_{y=y_0-1}^{y=y_0-\omega} \sum_{x=x_0-\omega}^{x=x_0+\omega} c(x, y) \quad (2)$$

$\omega$  in the above formula is a positive integer which represents the window size.  $c(x, y)$  represents the color value of point  $(x_0, y_0)$  which is made of  $(R, G, B)$ . Formula (3) and Formula (4) defines the color gradient value of point  $(x_0, y_0)$ :

$$B(x_0, y_0) = \sqrt{\|\Delta H(x_0, y_0)\|^2 + \|\Delta V(x_0, y_0)\|^2} \quad (3)$$

The work was supported by the National Natural Science Foundation for young. (No.41201388)

The work was also supported by Youth Science Technology Foundation of Jilin JianZhu University. (No. J20111007)

$$\begin{aligned}\Delta H(x_0, y_0) &= H_+(x_0, y_0) - H_-(x_0, y_0) \\ \Delta V(x_0, y_0) &= V_+(x_0, y_0) - V_-(x_0, y_0)\end{aligned}\quad (4)$$

### C. The Gradient Orientation Calculation

Formula (5) and Formula (6) defines the largest gradient orientation of color value of point  $(x_0, y_0)$ , which is named  $\theta$ . Among the formulas  $\|\cdot\|$  is Euclidean norm.

$$\theta = \tan^{-1} \frac{\Delta V'(x_0, y_0)}{\Delta H'(x_0, y_0)} + k\pi \quad (k \text{ is an integer}) \quad (5)$$

$$\Delta V'(x_0, y_0) = \begin{cases} \|\Delta V(x_0, y_0)\| & \text{when } \|\Delta V_+(x_0, y_0)\| \geq \|\Delta V_-(x_0, y_0)\| \\ -\|\Delta V(x_0, y_0)\| & \text{other} \end{cases}$$

$$\Delta H'(x_0, y_0) = \begin{cases} \|\Delta H(x_0, y_0)\| & \text{when } \|\Delta H_+(x_0, y_0)\| \geq \|\Delta H_-(x_0, y_0)\| \\ -\|\Delta H(x_0, y_0)\| & \text{other} \end{cases} \quad (6)$$

When  $\omega$  is equal to 1, the size of image window is  $3 \times 3$ . Formula (7) shows the vector elements such as  $H_+(x_0, y_0)$ ,  $H_-(x_0, y_0)$ ,  $V_+(x_0, y_0)$  and  $V_-(x_0, y_0)$ .

$$\begin{aligned}H_+(x_0, y_0) &= \frac{1}{3}[c(x_0+1, y_0-1) + c(x_0+1, y_0) + c(x_0+1, y_0+1)] \\ H_-(x_0, y_0) &= \frac{1}{3}[c(x_0-1, y_0-1) + c(x_0-1, y_0) + c(x_0-1, y_0+1)] \\ V_+(x_0, y_0) &= \frac{1}{3}[c(x_0-1, y_0+1) + c(x_0, y_0+1) + c(x_0+1, y_0+1)] \\ V_-(x_0, y_0) &= \frac{1}{3}[c(x_0-1, y_0-1) + c(x_0, y_0-1) + c(x_0+1, y_0-1)]\end{aligned}\quad (7)$$

When we solve the edge orientation of pixel point by formula (5), the value of  $\Delta H(x_0, y_0)$  can not be equal to 0. But in the practical processing, if color information of the pixel in neighborhood of a certain pixel point nearly has no difference,  $H_+(x_0, y_0)$  is equal to  $H_-(x_0, y_0)$ ,  $\Delta H(x_0, y_0)$  may be equal to 0. Therefore, the procedure can not continue running.

In fact, only in the processing of thinning, we calculate the  $\theta$ , the maximal orientation of a pixel's change of gradient, compare the gradient with the two neighboring pixels, reject the pixel (not the biggest) to make the image thinning. In a window of  $3 \times 3$  size, the relationship between the center point and the other eight neighborhood points only has eight directions. In view of the two neighborhood points need to be compared, which must in a same direction, so the relationship between the points may be further simplified to four directions.

So, we can judge the maximal orientation of a pixel's change of gradient belong to which direction of the four directions above-mentioned without computing the exact degree of the gradient of every points. How to mark the points, we can execute from following several aspects:

- If a pixel's change of gradient in vertical and horizontal direction is both equal to 0: It illustrates the color value of the neighborhood points is very similar. Consequently, we can think the point not is edge point, so the marker of the point is set to 0 (the darkest).
- If change of gradient in vertical direction is equal to 0 ( $\Delta V(x_0, y_0) = 0$ ), not equal to 0 in horizontal direction ( $\Delta H(x_0, y_0) \neq 0$ ): It means the point may be the edge in

horizontal direction, so the marker of the point is set to 1. If a pixel's change of gradient in vertical direction is not equal to 0 ( $\Delta V(x_0, y_0) \neq 0$ ), is equal to 0 in horizontal direction ( $\Delta H(x_0, y_0) = 0$ ), that means the point may be the edge in vertical direction, so the marker of the point is set to 2.

- If change of gradient in vertical and horizontal direction is not both equal to 0 ( $\Delta H(x_0, y_0) \cdot \Delta V(x_0, y_0) \neq 0$ ): We still need distinguish into the two basic types:  $\Delta H(x_0, y_0) \cdot \Delta V(x_0, y_0) > 0$  and  $\Delta H(x_0, y_0) \cdot \Delta V(x_0, y_0) < 0$ .

In view of  $\Delta H(x_0, y_0) \cdot \Delta V(x_0, y_0) > 0$ , assume  $K = \frac{\Delta V(x_0, y_0)}{\Delta H(x_0, y_0)}$ , now,  $K$  is grater than 0. If  $K \leq \frac{1}{3}$  is

true, the marker of the point is set to 1. If is true  $\frac{1}{3} < K \leq 3$ , the marker of the point is set to 3. If  $K > 3$  is true, the marker of the point is set to 2.

In view of  $\Delta H(x_0, y_0) \cdot \Delta V(x_0, y_0) < 0$ , now,  $K$  is smaller than 0. If  $K < -3$  is true, the marker of the point is set to 2.

If  $-3 \leq K \leq -\frac{1}{3}$  is true, the marker of the point is set to 4.

If  $K \geq -\frac{1}{3}$  is true, the marker of the point is set to 1.

With regard to the points with marker 1, compare  $f(i, j)$  with  $f(i+1, j)$  and  $f(i-1, j)$ . With regard to the points with marker 2, compare  $f(i, j)$  with  $f(i, j-1)$  and  $f(i, j+1)$ . With regard to the points with marker 3, compare  $f(i, j)$  with  $f(i+1, j-1)$  and  $f(i-1, j+1)$ . With regard to the points with marker 4, compare  $f(i, j)$  with  $f(i+1, j+1)$  and  $f(i-1, j-1)$ .  $f(i, j)$  refers to the gradient of point  $(i, j)$ .

We conserve the point with maximal value after the above comparison. All the points of the image should be handled repeatedly. Although some other methods also can calculate the orientation of gradient, it needs the arctangent calculation and judges the region of the calculated angle. After that, we also compare and select the points. The method discussed above can avoid the arctangent calculation and achieve the goal of calculation simplification.

### D. Edge Detection

The processing of edge detection is as follows:

- Select  $\sigma_1$  and  $\sigma_2$  as two scales of Gaussian filter. One is small, another is big.
- Make convolution between Gaussian filter with the small scale and the original image. We can get a high-resolution image.
- Make convolution between Gaussian filter with the big scale and the original image. We can get a low-resolution image.
- Make convolution in high-resolution and low-resolution image using Prewitt operator. We can get two color gradient values and edge orientation,

combine multi-scale, and select the bigger gradient value as the final result.

- Thinning to assure that the edge width is only one pixel. Select the appropriate threshold, the brightness of point those gradient value greater than the threshold is set to 255, others is set 0. At last, result of the edge detection outputs in binary image format.

Much image edge we select after thinning is not the real edge or the edge considered. We need set a threshold now. When the gradient of the pixel point is grater than the threshold, the point is set to light; otherwise, set to dark. Effect of threshold selection will be different in different time.

This paper carried out two methods and compared the result of theirs. In the first, we thin the image before threshold selection, which is filtering, vector Prewitt operator convolution, multi-scale combination, thinning, and threshold selection at last. Secondly, we select the threshold before thin the image, which is filtering, vector Prewitt operator convolution, threshold selection, multi-scale combination, and thinning at last. From the point of the results of this experiment, the latter is better. Figure 1 shows the different results. Figure 1(a) is the original image; Figure 1(b) shows the result of image thinning before threshold selection. Figure 1(c) shows the result of threshold selection before image thinning.

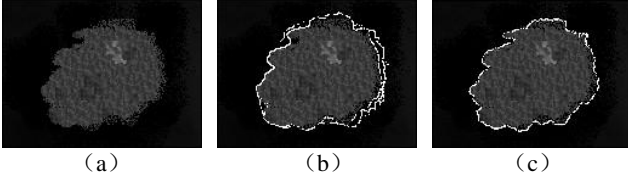


Fig. 1. The results comparison between the different order thresholds

### III. EDGE DETECTION BASED ON THE ADVANCED SNAKE ALGORITHM

#### A. The Conventional Snake Algorithm

As one special case of the variable model, it applied to the fields of computer vision successfully such as medical image processing, target identification, 3D object reconstruction, optical flow computation, and geometric rectification and registration of imagery etc [4]. Objective selection by Snake model is the process of all kinds of image characteristic, continuity, and flatness constraint into an energy function. We can achieve the goal of objective selection by calculation the minimum of the energy function. Contrast to other methods in characteristic selection, Snake algorithm always reduces the self-energy constantly during the processing of motion to select characteristic.

Snake is a closed or unclosed elastic curve that is a data set formed by several control point [5]. When finding the designated objective outline, Snake algorithm closes to the characteristic location from the original location in the image by searching for the minimal energy during the processing of the elastic curve deformation and motion.

The potential energy function of Snake is defined as:

$$E_{snake} = \sum_{i=1}^N (E_{int}(i) + E_{ext}(i))$$

Among the above formula,  $N$  refers to the number of enveloping point;  $E_{int}$  refers to the inside energy function which considers the continuity of the envelope itself and the size of curvature;  $E_{ext}$  refers to the external energy function which mainly relates to some specific situation such as the gradient of image gray change etc. In particular,

$E_{int} = \alpha_i \|V_i - V_{i-1}\|^2 + \beta_i \|V_{i-1} - 2V_i + V_{i+1}\|^2$ .  $V_i = (x_i, y_i)$  refers to the coordinate of the  $i$  enveloping point. The function of  $\alpha_i \|V_i - V_{i-1}\|^2$  is making Snake resist to stretching when it sustains force.  $\alpha_i$  is a continuity restraint coefficient that is used in the two neighborhood points on the envelope, which adjusts the elasticity of the Snake. The function of  $\beta_i \|V_{i-1} - 2V_i + V_{i+1}\|^2$  is making Snake resist to winding when it sustains force.  $\beta_i$  is a curvature restraint coefficient that is decided by the three neighborhood points on the envelope, which adjusts the inflexibility of the Snake.  $E_{ext}$  usually is an image gradient function during edge detection.

#### B. The Advanced Snake Algorithm

In this paper, the author introduces the concept of vector order Prewitt operators into the Snake algorithm with the shape factor, modifies the energy function to select the edge of image. From the formula of  $\|V_i - V_{i-1}\|^2$  we can see that the continuity restraint of the Snake  $E_{int}$  is the sum of square of the distance between two points in fact. During the solving processing, the total energy of the Snake needs to reduce constantly. In some cases, the all points in the Snake may be constraint to one point due to the restraint.

##### 1) Define the Continuity Restraint $E_{con}$ :

In this paper, the author put forward the continuity restrain  $E_{con}^i = \frac{D_i}{D_{i\max}}$ . In the formula,  $D_i = |d_i - aver|$ ,  $d_i$  refers the distance between of the  $i$  point and the point  $i-1$ ;  $aver$  refers the average distance between of the neighborhood points in the envelope;  $D_{i\max}$  refers the maximal value in the select given region contained the point  $i$ .

##### 2) Define the Curvature Restraint $E_{curv}$ :

We defined the curvature restraint  $E_{curv}^i = \frac{C_i}{C_{i\max}}$ . In the formula,  $C_i = (x_{i-1} - 2x_i + x_{i+1})^2 + (y_{i-1} - 2y_i + y_{i+1})^2$

##### 3) Define the Image Characteristic Restraint $E_{img}$ :

$E_{img}^i = \frac{G_{i\min} - G_i}{G_{i\max} - G_{i\min}}$  refers the image characteristic restraint. In the above formula,  $G_i$  refers the gradient,  $G_{i\min}$  and  $G_{i\max}$  refers the minimal and maximal value in the select given region contained the  $i$  point respectively.

##### 4) Define the Shape Restraint $E_{shape}$ :

Shape restraint use a curve named  $C: y = f(x)$  (maybe curve of second order or Section curve) to fit the given contour shape according to the selected envelope points firstly. The closer the point on the envelope to the curve  $C$  the smaller its energy is. But in order not to deviate the original given points so far, we also need establish a selection window to restrain the selection region, namely we only find the points most closing the curve  $C$  from the window contained the envelope points.

The reference drew a straight line  $S$  intersects the curve  $C$  at point  $(x_0, y_0)$  from the envelope point  $(x_i, y_i)$ ; moreover, the straight line  $S$  is perpendicular to the line tangent to the curve  $C$  at the point  $(x_0, y_0)$ . Consequently, the point  $(x_0, y_0)$  must satisfy the following two equations:

$$\begin{cases} f_1(x, y) = y - f(x) \\ f_2(x, y) = x + f'(x)y - x_i - f'(x)y_i \end{cases} \Rightarrow \begin{cases} f_1(x, y) = 0 \\ f_2(x, y) = 0 \end{cases}$$

Define the shape restraint as  $E_{shape}^i = \frac{Q_i - Q_{i\min}}{Q_{i\max} - Q_{i\min}}$ , in the formula,  $Q_i = \sqrt{|f_1(x, y) + \omega|f_2(x, y)|}$   $\omega \in [0, 1]$  is weighting coefficient, and  $(x, y)$  is a random point in the select window contained the  $i$  point.  $Q_{i\min}$  and  $Q_{i\max}$  refers the minimal and maximal value respectively. It is proved that the shape restraint is monotone function.

#### 5) Construct Energy Function

It Energy function consists of continuity restraint, curvature restraint, image characteristic restraint and shape restraint. Snake energy of the  $i$  point on the envelope will be defined as  $E_{snake}^i = A_i E_{con}^i + B_i E_{curv}^i + C_i E_{img}^i + D_i E_{shape}^i$ , among them,  $A_i, B_i, C_i$  and  $D_i$  represents the weights respectively.

#### C. Edge Detection Based on the Advanced Snake Algorithm

The processing of edge detection based on the advanced snake algorithm is as follows:

- Setting the initial position.
- Selecting the category of the approximation curve.
- Curve fitting. Computing the respective curve coefficient with least square method.
- Setting a bigger initial value for the minimal total energy  $E_{\min}^i$  of the point  $i$ .
- Computing the  $E_{con}^i, E_{curv}^i, E_{img}^i, E_{shape}^i$  and  $E_{snake}^i$  for the every point in the selection window of the point  $i$ .
- If  $E_{snake}^j < E_{\min}^i$ , then order  $E_{\min}^i = E_{snake}^j$ , record the position of the point  $j$ , and order  $j_{\min} = j$ .
- Carry out the 5 and 6 steps repeatedly until finishing searching the region of the point  $i$  and move the position of point  $i$  to the point  $j_{\min}$ .
- Carry out the 7step in light of the next point until finishing the total envelope points.
- Connect the moved envelope points in proper order to form edge.

The result of the experiment was shown in the figure 2. Figure2 (a) was the remote image represented the grassland fire. Figure 2(b) showed the image edge enhanced. Figure 2(c) showed the selected envelope points. Figure 2(d) is the result of edge selection.

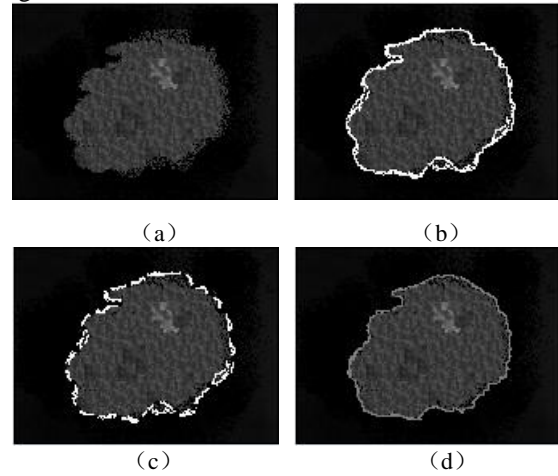


Fig. 2. Production of objective region with the shape factor

#### IV. CONCLUSION

The paper proposed the edge detection based on the advanced Snake algorithm that introduces the vector Prewitt operator with filter and multi-scale into the conventional snake algorithm. The significance of the Prewitt operator was discussed according to its form. During the processing of implementation, the paper improved the gradient direction algorithm that judged the pixel gradient direction without computing the exact degree of the gradient. Consequently this helps to reduce the amount of computation. The idea of using vector order Prewitt operators to edge detectors can help us make full use of the color information. This method will be more reliable in the procedure of finding the minimum energy position. The proposed method plays an important role in the water quality monitoring of the Yalu River.

#### REFERENCES

- [1] Fu jingguang, Xu gang, and Wang yuguo, "Removing Noises from Binary Image by Model Analyzing," Computer Engineering and Applications, vol. 39, pp. 88–90, 2003.
- [2] Guo jian, Liu bin, and Lan haiyan, "Edge Algorithm and Application Based on Binary Image," Infrared Technology, vol. 27, pp. 416–418, 2005.
- [3] Wang rongben, Yu tianhong, and Jia yang, "Study of stone detection method based on edge extraction and track," Journal of Computer Applications, vol. 32, pp. 195–197, 2006.
- [4] Meng fan, Fang shenghui, "Quasi-automatic Extraction of Zonal Roads from Remote Sensing Images Using Template Matching and B-Snake Model," Geomatics and Information Science of Wuhan University, vol. 37, 2012.
- [5] Qiu shubo, Wang huayang, and Liangzhiwei, "racking Object Contour Using a Novel B-Snake Algorithm," Journal of Image and Graphics, vol. 10, pp. 585–588, 2005.