Lake Edge Detection Using Canny Algorithm and Otsu Thresholding

Budi Darma Setiawan
Faculty Of Computer Science
Universitas Brawijaya Malang, Indonesia
s.budidarma@ub.ac.id

Alfi Nur Rusydi
Faculty Of Computer Science
Universitas Brawijaya Malang, Indonesia
alfi.nurrusydi@ub.ac.id

Koko Pradityo
Faculty Of Computer Science
Universitas Brawijaya Malang, Indonesia
kk.pradityo@gmail.com

***Abstract*— Change that is happened to the lake can have implication to its surrounding. Monitoring the change can be done remotely by using remote sensing images. To do it automatically, we first need to detect the lake region in the images, then gain the edges between water and land area. This research aims to implement the Canny edge detection method, combining with Otsu thresholding to detect the edges. Otsu thresholding is used to gain threshold value for Canny Method. In result, some edges are well detected, but some others are missed. Some of the false detected edges are gained from thresholding process, where shadow or dark pixel area which have nearly same color as water, are also detected as water.**

***Keywords*—Lake; Canny; Otsu; Edge Detection**

I. INTRODUCTION

Water is a very important resource in life. Water can be found in sources like the sea, river, and lake. Every year, the volume of the water in some lake can change due to small weather cycle, climate change, or may from human activity near the lake that affects the environment. The change of the water volume can be detected by the change of the water surface and the shape of the lake. That is why some research has been done to detect the change of the lake like found in [1] which detected the lake using SVM classification and using contour tracing algorithm to identify the edges. In [2], they create rules using Decision Tree Classifier for extracting wetland from images, and then study the change.

Before detecting the change, it needs to detect the boundary of the lake. From remote sensing images, we can detect the boundary of the lake by using edge detection method. One of the methods for edge detection is the Canny algorithm that first proposed by John Canny[3], and implemented in some other research like in [4]. It also improved in [5] and [6]. In [6], the threshold in Canny is gained using Otsu Thresholding.

This research will see the result of the implementation of the method that proposed in[6] for detecting the edges of the lake. Canny's method is chosen because it produces single pixel thick, continuous edges[7].

II. DATA

We gain lake images from Landsat L8 OLI/TIRS C1 Level-1 of USGS earth explorer. For research dataset, we used 2 lake images. One image is Buyan lake which is acquired on June 2017, and another one is Mahalona lake which is acquired in September 2016. Those lake images can be seen in figure 1.

(a)

(b)

Fig. 1. (a) Buyan Lake; (b) Mahalona Lake

III. CANNY EDGE DETECTION

John Canny proposed this method on 1986 and broadly used for detecting the edge. His method then simply and well explained in some other research like in [4] and [5].

Canny's Edge detection methods are:

a. Smoothing using Gaussian filter

This Gaussian filter is used for removing noise that presents in images, or smoothing the images. Gaussian filter is generated using (1).

$$H_{(i,j)} = \frac{1}{2\pi\sigma^2} e^{-\frac{(i)^2+(j)^2}{2\sigma^2}} \quad (1)$$

Where:

i, j: are filter coordinate in x, and y respectively, and they have zero of mean. For example if kernel size is 4, then I and j are in range [-4,4].

k: is the size of Gaussian kernel

σ: is standard deviation, or the spread of the bell curve which is can acquired from experiment

After the Gaussian filter is generated, we applied the filter using convolution method.

b. Computing Gradient magnitude and direction

Gradient magnitude is computed using (2)

$$G = \sqrt{{G_x}^2 + {G_y}^2} \quad (2)$$

And direction (θ) is computed using (3)

$$\theta = tan^{-1}(\frac{G_x}{G_y}) \quad (3)$$

Gx is 1st derivative for horizontal component and *Gy* is 1st derivative for vertical component. *Gx* and *Gy* are computed from grayscale images using mask in figure 2.

1	0	-1
1	0	-1
1	0	-1

(a)

1	1	1
0	0	0
-1	-1	-1

(b)

Fig. 2. Derivative mask (a) for *Gx*, (b) for *Gy*

c. Non-maxima Suppression

This step is used for thinning the edges. This is done by comparing the pixel's gradient with it neighbor's. If it is smaller than it neighbor's, then the pixel removed from edges. The neighbor is an adjoin pixel within the direction (θ), which is divided into 4 region:

a. Horizontal, when direction in between [-22.5°,22.5°] or between [-157.5°,157.5°]

b. Vertical, when direction in between [-112.5°,-67.5°] or between [67.5°,112.5°]

c. +45, when direction in between [-67.5°,-22.5°] or between [112.5°,157.5°]

d. -45, when direction in between [-157.5°,112.5°] or between [22.5°,67.5°]

The illustration of those angle range can be seen in figure 3 [9].

d. Double Thresholding and edge linking

Two thresholds T1 and T2 are used. T2 is the upper threshold and T1 is the lower threshold. If pixel's gradient value is greater than T2, then the pixel is assigned to a strong edge. If pixel's gradient value is between T1 and T2, the pixel is assigned to a weak edge.

The further process is doing edge linking by change pixel's status from weak edge to strong edge if that weak edge is adjoined with the strong edge.

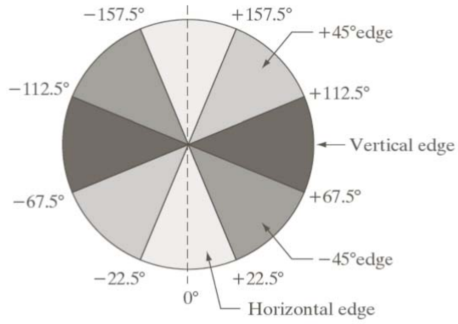


Fig. 3. Angle range for non-maxima suppression[9]

IV. OTSU THRESHOLDING

Otsu method is a method proposed by Nobuyuki Otsu on 1979. It is based on grayscale images histogram[8]. The main idea is to maximize the between-class variance in the histogram. Or on the other hand, it needs to minimize within-class variance. The within-class variance is computed using (4).

$$\sigma^2(t) = \omega_0(t)\sigma_0^2(t) + \omega_1(t)\sigma_1^2(t) \quad (4)$$

t is the threshold dividing the class. $\sigma_0^2(t)$and $\sigma_1^2(t)$ are the variance of class 1 and class 2 respectively.

$\omega_0(t)$ and $\omega_1(t)$ are the probability of class 1 and class 2, respectively. These probabilities are computed using (5) and (6).

$$\omega_0(t) = \sum_{i=0}^{t-1} p(i) \quad (5)$$

$$\omega_1(t) = \sum_{i=t}^{L-1} p(i) \quad (6)$$

Where L-1 is maximum value of pixel intensity. It commonly 255 (8-bit). *P(i)* probability of occurrence of ith intensity in the class.

The main task is finding t between 0 to 255, so the $\sigma^2(t)$is minimum.

V. APPLYING OTSU ON CANNY EDGE DETECTION

There is a problem when applying Canny edge detection. We have to set the double threshold T1 and T2 manually. If T2 is too high, many edges are disappearing from the result. But, if

T2 is too small, many noises will also detect as the edge. To choose T1 and T2, Otsu threshold can be used [6]. T2 is directly obtained by computing t using Otsu thresholding. T1 may be set half of T2[6].

$$T2 = t \text{ and}$$
$$T1 = 0.5T2 \tag{7}$$

By using Otsu method, we get the best t that maximize the lake and background variance.

VI. Lake Edge Detection

Overall process we used to get the lake edge is shown in the flowchart in figure 4 and figure 5.

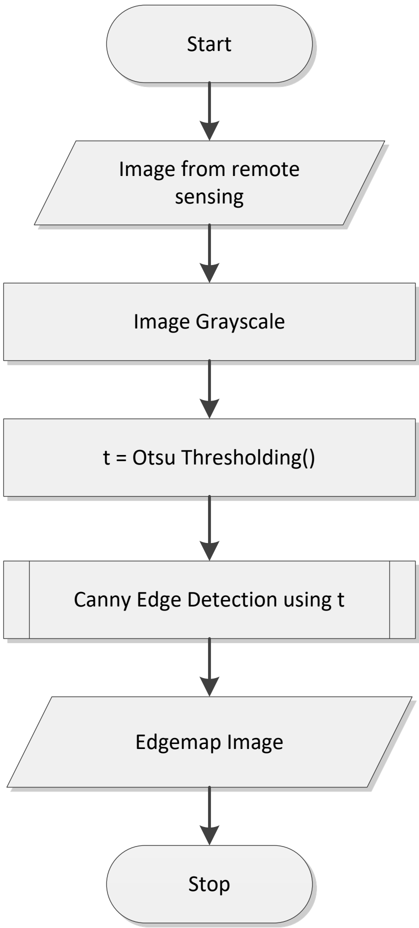


Fig. 4. Method flowchart

Image from remote sensing as an input are in Red, Green, and Blue (RGB) format. It then turned into grayscale format using (8).

$$I = 0.3R + 0.59G + 0.11B \tag{8}$$

Where,

I: grayscale pixel intensity

R, G, and B: pixel color intensity as Red, Green, and Blue respectively

That grayscale images then computed with Otsu thresholding to get *t*. This *t* value is used as the upper threshold in Canny edge detection.

Figure 5 shows some result example for each process, including all process in Canny edge detection. In figure 5, at last steps, we do an evaluation of the edge detected. We count the pixel that detected as an edge, shown as a white pixel in last step image. We also count the false pixel edge that detected, shown as a red pixel.

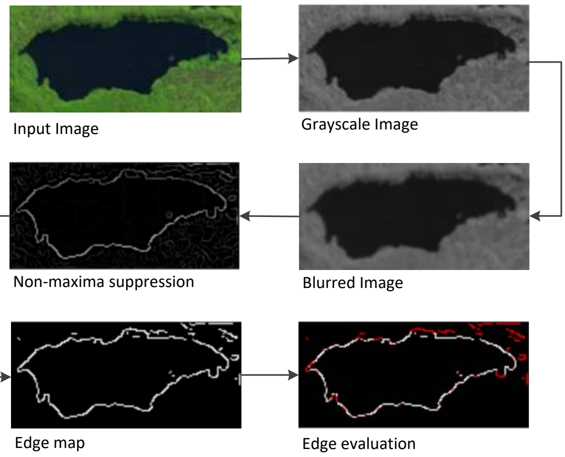


Fig. 5. Edge detection steps

VII. Result And Discussion

In edge detection process, we try some σ value and kernel size to get the best result. The result for 2 lake image we used, shown in table 1, and table 2.

The edge detected is all pixels that detected as an edge by the system. A large number of the edge detected does not mean good result because some of those pixels are the false detected edge. Otherwise, too small number of the edge detected, also not good, because many edge pixels are not detected as the edge. Some result comparisons are shown in figure 6. Also, the best combination of σ value and kernel size is highlighted in table 1 and table 2.

Best results are decided from smallest percentage of the false edge pixel. In figure 6, white pixels are the correct edges detected, and red pixels are false edges detected. We decide to choose figure 1(c) as the best result (figure from table 1) because it has less of the false edges detected, even though it also has less of the correct edges detected. And once again, it has smallest ratio value of error edges detected.

From figure 6 we can see that the greater σ value and kernel size, then the less detail we got. It is because the image becomes more blurred so that the detail of an unimportant edge can be removed. But, if these values are too large, the important edge is also gone, like what happened in Figure 6(c) at the right top of the lake. At that region, the lake edges are gone.

Some of the false edges are derived from the texture of the vegetation around the lake. Some of that vegetation or their shadows have the same color as the lake or even darker. For example in figure 6, those edges appear on the upper left corner of the images. We can see red texture edges which appear in every image.

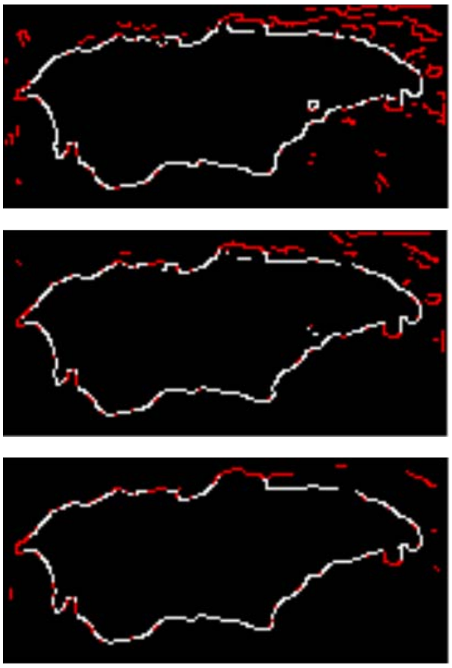
TABLE I. RESULT FOR BUYAN LAKE

σ	Kernel Size	Edge Detected	False Edge	False Edge Percentage
0.5	3	745	378	51%
	5	706	339	48%
	7	647	298	46%
0.75	3	598	242	40%
	5	543	193	36%
	7	528	179	34%
1	3	554	218	39%
	5	462	147	32%
	7	451	139	31%
1.25	3	531	200	38%
	5	425	147	35%
	7	417	140	34%
1.5	3	531	200	38%
	5	414	147	36%
	7	377	150	40%

TABLE II.

TABLE III. RESULT FOR MAHALONA LAKE

σ	Kernel Size	Edge Detected	False Edge	False Edge Percentage
0.5	3	1380	546	40%
	5	1362	526	39%
	7	1313	545	42%
0.75	3	1252	415	33%
	5	1191	387	32%
	7	1174	366	31%
1	3	1197	413	35%
	5	1122	378	34%
	7	1095	355	32%
1.25	3	1185	410	35%
	5	1052	369	35%
	7	1004	326	32%
1.5	3	1185	410	35%
	5	1017	373	37%
	7	920	333	36%



(a)

(b)

(c)

Fig. 6. Edge of Buyan Lake; (a)$\sigma = 0.5$, kernel size 3; (b) $\sigma = 1$, kernel size 3; (c) $\sigma = 1$, kernel size 7

VIII. CONCLUSSION

The Otsu method is used here to compute a threshold for double thresholding process on Canny edge detection. The result is not perfect. Some lake edges can be detected well, but some object or vegetation that have the same color as lake also detected as the lake. False detection can be minimized by adjusting the σ value and kernel size that affect the Gaussian filter for noise removal. But this also not perfect. If the Gaussian filter too large, the edges are missing, but if it too small, the other detail is increasing.

REFERENCES

[1] J. Li and R. M. Narayanan, "A Shape-Based Approach to Change Detection of Lakes Using Time Series Remote Sensing Images," *IEEE Transactions on Geoscience and Remote Sensing, vol. 41, no. 11,* pp. 2466-2477, 2003.

[2] R. Ruan and L. Ren, "Change Detection of Hongze Lake Wetland Using Rule-Based Inferring," in *2007 IEEE International Geoscience and Remote Sensing Symposium*, Barcelona, 2007.

[3] J. Canny, "A Computational Approach to Edge Detection," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE,,* Vols. PAMI-8, no. 6, pp. 679-698, 1986.

[4] A. Al-Nahid, Y. Kong and M. Hasan, "Performance Analysis of Canny's Edge Detection Method for Modified Threshold Algorithms," in *ist International Conference on Electrical & Electronic Engineering (ICEEE)*, Rajshahi, 2015.

[5] R. K. Sidhu, "Improved canny edge detector in various color spaces," in *3rd International Conference on Reliability, Infocom Technologies and Optimization*, Nodia, 2014.

[6] M. Fang, G. X. Yue and Q. C. Yu, "The Study on An Application of Otsu Method in Canny Operator," in *International Symposium on Information Processing (ISIP'09)*, Huangshan, 2009.

[7] E. Nadernejad, S. Sharifzadeh and H. Hassanpour, "Edge Detection Techniques:Evaluations and Comparisons," *Applied Mathematical Sciences,* vol. 2, no. 31, pp. 1507 - 1520, 2008.

[8] N. Otsu, "A Tlreshold Selection Method from Gray-Level Histograms," *IEEE Transacnons On Systems, Man, And Cybernencs,* Vols. SMC-9, no. 1, pp. 62-66, 1979.

[9] R. C. Gonzalez and R. E. Woods, Digital Image Processing, New Jersey: Pearson Prentice Hall, 2008.

978-1-5386-1683-3/17/$31.00 ©2017 IEEE