

A performance analysis for real-time flood monitoring using image-based processing

Qianyu Zhang, Nattha Jindapetch, Rakkrit Duangsoithong, Dujdow Buranapanichkit

Department of Electrical Engineering, Prince of Songkla University, Thailand

Article Info

Article history:

Received Jun 1, 2019

Revised Aug 3, 2019

Accepted Aug 17, 2019

Keywords:

Edge detection

Image segmentation

Monitoring system

Normalized cuts

Region growing

ABSTRACT

Nowadays, various image-based methods have been used in the area of monitoring. Whereas the precision of detection objects and real-time processing are the key issues for many applications. Considering the limitation of the working environment, the higher correctness and faster operating time can guarantee the work efficiency. In this paper, the image-based methods have been studied to monitoring the state of the flood in the real-time system. The performance of each image processing technique has been evaluated based on accuracy and processing time. In the flood monitoring system, the variation of important parameters can cause the change of performance and the effect of the variable parameters has been demonstrated from the experiment results. After comparing to the other image-based techniques, canny edge detection presents the best one, which also has better repeatability with the source image from different locations. Consequently, the improved canny edge detection method has been proved that can work very well on the real hardware in the outdoor environment.

Copyright © 2020 Institute of Advanced Engineering and Science.

All rights reserved.

Corresponding Author:

Dujdow Buranapanichkit,
Department of Electrical Engineering,
Prince of Songkla University,
Hatyai, Songkhla 90112, Thailand.
Email: dujdow.b@psu.ac.th

1. INTRODUCTION

As we all know, floods have a huge influence on people's daily life. The flood often causes a loss of life. Additionally, it will destroy the buildings, pollute the drinking water, and damage the power transmission. It's vital to monitor the state of the flood for saving life and minimizing the losses. There are many methods can be used for the flood monitoring, but in the literature review from the reference [1-9], the image-based methods: region growing, canny edge detection and normalized cuts are better than the others based on the achievability and timeliness. These image-based methods will be made a simulation on MATLAB in the beginning and the method which has the best performance will be kept and improved. The improved method is our proposed method, which will be implemented in the real embedded system. In this paper, considering the fast speed of the flooding, the proposed method can detect the flood area with the image-based method in real time based on the embedded system.

In the reference, the related papers are [1, 5] and [7]. Paper [1] proposed a method can detect the flash flood with the fast processing speed based on the stationary surveillance cameras, which can be used to the urban areas. The process of detection includes the background subtraction, morphological closing, and canny edge detection, in which the canny edge detection has great effects on distinguishing the flood area with the others based on the edge density. Paper [5] can monitor and identify the flood automatically with the proposed method based on real-time video images. The images are captured from the digital camera. The proposed method includes mean-shift and region growing, which can be used in bad weather condition such as rainy. The region growing method can partition the pixels into meaningful parts based on the seed

points and grow rules. Paper [7] aimed to monitor the flood with a specific area automatically. The images are captured from the CCTV and infrared camera. The proposed minimal spanning tree method belongs to the graph-based segmentation method, which can be used to detect the flood area with scale and with no scale.

Through the study of the related papers, the Canny edge detection, region growing method, and graph-based method are suitable for the flood monitoring, which belong to the image-based methods. The edge detection methods usually find the places of the edges through the abrupt changes of the pixel intensity [10, 11], which has the potential to find the boundaries of the water area. While the region growing method can group the pixels into different parts according to the similarity of them, which is based on the seed point and grow rules. The graph-based segmentation is a popular trend in the image segmentation area for recent years, which can partition one graph into many sub-graphs that each graph represents the interested meaningful object [7]. In order to find the most suitable method for flood monitoring, the three image-based methods will be compared on the MATLAB. According to the comparison standard, the method which has the best performance will be selected and improved. The improved method is our proposed method, which has been implemented on the real hardware for testing. In this paper, the proposed method can achieve the goal of detecting flood based on the embedded system with the limit memory in the real outdoor environment.

The remainder of this paper is as follows. The image-based techniques have been introduced in the second part, which included three sub-sections about the region growing, canny edge detection and graph-based methods. In the third part, the experiment of the three methods will be made on MATLAB based on the comparison standard with the accuracy and processing time. The method that has the best performance from the third part will be improved and implemented on the real embedded system in the outdoor environment in the fourth part. Last but not least, the discussion which presented in the last part was summarized.

2. STUDY OF IMAGE PROCESSING TECHNIQUES

From the study of related work, the three image-based methods are interesting and meaningful, which are suitable for flood monitoring based on the feasibility and timeliness. The flood monitoring will be implemented on the embedded system in the real outdoor environment. In this study, the vital parameter of each method is also noticed for achieving better monitoring.

2.1. Region Growing

The region growing algorithm operates through grouping the pixels into the different regions, which is based on the predefined grow criterion [12-14]. The seed point is set initially at the place of the interesting object by ourselves. After placing the seed point, the growth will start from the seed point to its neighborhood for gathering the homogeneous regions. When applying the region growing with flood monitoring, the process is shown as follows.

In Figure 1, the RGB image will be changed to the grayscale image in the beginning through sum the weight of the three color components. The formula is in (1). The seed point will be placed in the flood area in the image for getting the pixel intensity. According to the grow criterion, if the distance between the seed point and its neighborhood is lower than the maximum intensity distance (Mid), the neighborhood pixel will be group into the same region with the seed point. If not, the neighborhood pixel will not be contained. The Mid is the variable of the region growing method, which defines the pixel similarity. The changes in the variable will affect the result of the region growing method. In the end, the flood area will be the segmented region with the seed point. The growing process is iterative that all the pixels in the image will be calculated.

$$Y=0.2989R+0.5870G+0.1140B \quad (1)$$

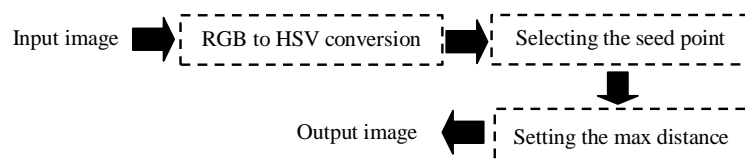


Figure 1. Process steps of the region growing method

2.2. Canny Edge Detection

The edge detection is a commonly used method for image segmentation, which operates through finding the sudden changes between the pixel intensity. Among the different edge detection methods [10-16], canny edge detection has advantages in the aspects of low error rate, well positioning and single edge response, which can be used to find the edge of the ware area in this research. The process of canny detection method is shown in Figure 2.

From Figure 2, the input image will be transformed into the grayscale image. The Gaussian filter is used for removing the noise and smoothing the image. The gradient represents the differences between the pixel intensity and the edge can be found by searching the maximal gradient of the image. The direction and magnitude of the pixel can be got by calculating the finite-difference of the first-order partial derivatives. Then the non-maxima suppression will be applied on the gradient amplitude for finding the correct location of edges. The variable in the Canny edge detection method is the threshold, which will affect the final result. In order to remain the meaningful edges, the double thresholds are used. There is a threshold with higher value and another is with lower value, but both of them are in the range from 0 to 1. According to Canny’s suggestion, the ratio between the two thresholds is about 3:1 and 2:1. Therefore, the default ratio is 0.4 in this paper. When the intensity of the edge is higher than the high threshold, the edge will be kept. If not, the edge will be discarded when its intensity is smaller than the low threshold. If the edge value is between the double threshold, the edge will only remain when it connected to a remained high-intensity edge.

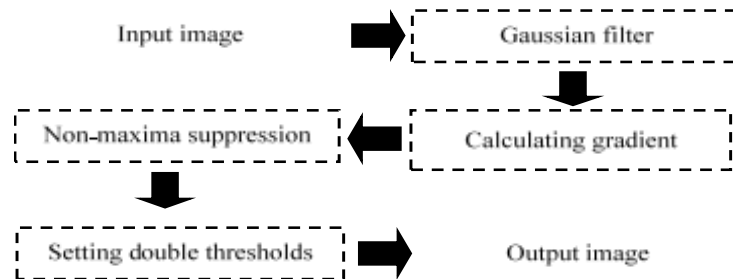


Figure 2. Process steps of the canny edge detection method

2.3. Normalized Cuts

The normalized cuts method is developed from the minimum cut, which belongs to the graph-based method in the field of image processing [17-19]. Usually, the graph is represented as $G = (V, E)$, where V refers to the set of vertices while E is defined as the set of edges that connects the vertices. In the graph, each pixel is regarded as a vertex. The image segmentation operates by suitable cutting edges in the graph for dividing the data into disjoint subregions. The image segmentation is associated with the cut criterion in formula (2), which will minimize the graph cut $cut(X, Y)$. But the minimal cut tends to produce the small and isolated components. In order to solve this problem, the normalized cuts will operate by fixing the bias of min cut through normalizing the size of segments. The formula of normalized cuts is in (3), where $vol(X)$ represents all the connections of the vertices in the set X . $vol(Y)$ has the same meaning with the set Y . When the normalized cuts method operates, the generalized eigenvalues can be used as the approximate solutions. The process is demonstrated in Figure 3. In the beginning, the image will be regarded as the weighted graph G . Then define w as the adjacency matrix of the graph and let D be the diagonal matrix with diagonal entries in formula (4). The cost of normalized cuts can be written in (5). y_i is the indicator vector, where $y_i = 1$ if the node i is in set X and otherwise, $y_i = -1$.

$$cut(X, Y) = \sum_{i \in X, j \in Y} w(i, j) \tag{2}$$

$$Ncut(X, Y) = \frac{cut(X, Y)}{vol(X)} + \frac{cut(X, Y)}{vol(Y)} \tag{3}$$

$$D(i, i) = \sum_j w(i, j) \tag{4}$$

$$Ncut = \frac{y_i^T (D - W) y_i}{y_i^T D y_i} \tag{5}$$

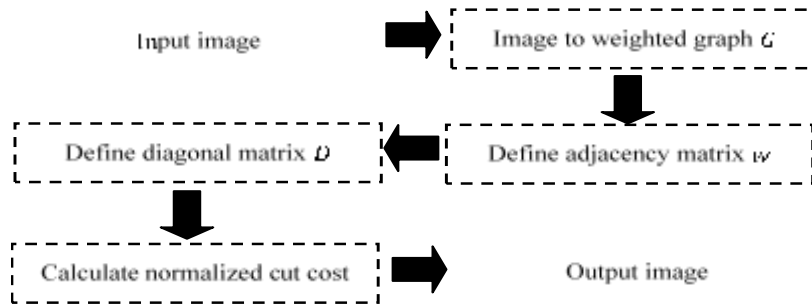


Figure 3. Algorithm of normalized cuts method

3. EXPERIMENTAL RESULT

Real-time monitoring system using image-based processing operates with many methods for example background subtraction, edge detection, image segmentation, etc. The systems using the hardware platform for multimedia processing require the high performance for the processor and the proper resolution for the camera. We investigated on the computer with processor, Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz, RAM 8GB. Whereas the image resolution is 600x1024 pixels for all images in the experiment. The captured images have been studied from the same area but the different views of the real outside environment. In order to study the suitable image-based techniques, we consider the comparison for entire methods in MATLAB.

3.1. Performance Metrics

The study on the image processing methods has been implemented using a computer with a software tool. The important things to evaluate the efficiency of the image-based processing applications are both the accuracy and the processing time. We believed these two keys would be affected the real system in order to make it easier for implementation on the real hardware such as the embedded system. They have been considered for the real outdoors environment for the future of work.

Our experimental results have been made a comparison with the ground truth image in order to evaluate the accuracy of the predicted water area for each method. Based on the confusion matrix as shown in Figure 4 has been used for the accuracy's calculation [20, 21]. Whereas the ground truth image can be gained from the expert by observed the original image. A standard method to evaluate the accuracy is the confusion matrix which consists of the matrix of rows represented the number of the actual data which has both positive and negative and the matrix of columns represented the number of the predicted data which also have two categories as same the horizontal matrix.

| | Predicted: Positive | Predicted: Negative |
|------------------|---------------------|---------------------|
| Actual: Positive | TP | FP |
| Actual: Negative | FN | TN |

Figure 4. A matrix for calculating the accuracy

In these experiments, we identify the confusion matrix into four types. The result has been predicted correctly for the water area represented in the first matrix which is true positive while the prediction of that area shows positive but the real data is the other object represented the false positive. The others mention that the actual information is negative. FN indicates the prediction is wrong because this data is actually the water area. Finally, the prediction is correct because this area is not the interesting object represented the true negative. The calculation of the accuracy has been obtained using (6).

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN} \times 100\% \quad (6)$$

Another factor is also an important thing that we have considered in performance evaluation for image processing in the real-time flooding monitoring system. The processing time has been calculated with the common function ‘tic-toc’ in a standard tool. The ‘tic’ function starts the timer to measure the operation time for a program. The internal time has been counted when the command was executed. Then, the ‘toc’ function stops the record and shows the final time in a unit of second.

From now on, we demonstrated the methodology for study the experiment in each image processing techniques such as region growing, canny edge detection and the normalized cut respectively. The first method has shown in Figure 5 after capturing the input image then the seed point has been selected and the Mid has been set up. The output image after performing the region growing is represented in Figure 5(c). As seen in Figure 5(d), we converted this result to black and white color using the thresholding. After that, the accuracy has been calculated by comparing the ground truth image in Figure 5(e).

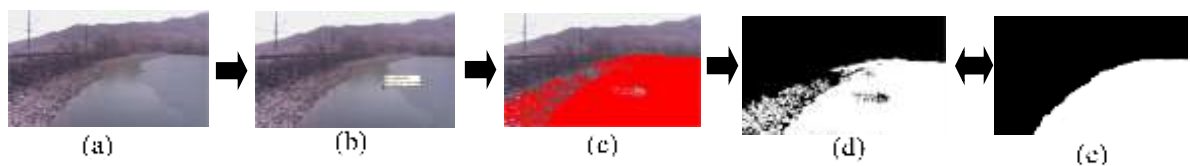


Figure 5. An example of the experimental process for performance evaluation of the region growing technique when Mid is 0.1

The experimental result as shown in Figure 6(b) has been represented the image-based processing of the canny edge detection while the normalized cut method has presented the output image as demonstrated in Figure 7(b). For the canny edge detection, the threshold is an important parameter to study in the next subsection. After processing of the closing morphology, the output image has been obtained in Figure 6(c), then the comparison with the same ground truth image in Figure 5(e) has been evaluated the accuracy of this image.

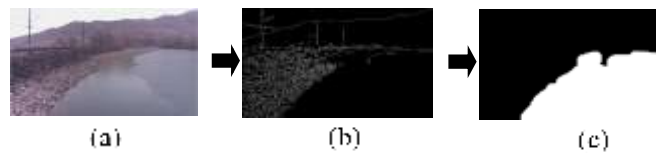


Figure 6. An example of the analytic process for performance evaluation of the canny edge detection when threshold is 0.2.

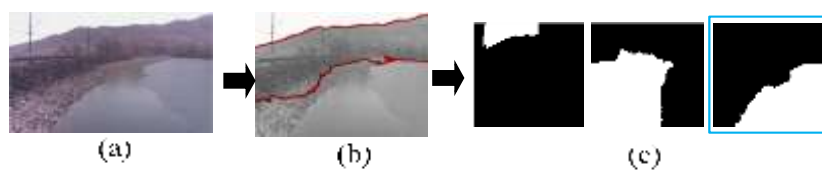


Figure 7. An example of the verification process for efficiency assessment of the normalized cut when the segment number is 3.

For the normalized cut, the binary image of this method has been separated into 3 parts as shown in Figure 7(c). Therefore, the suitable part in the square box has been selected for further comparison. The segment number for this method can be set up without limitation.

3.2. Performance Evaluation

In our experiment, we have been studied for the three categories of environment especially only the daytime, not at night, for example the rainy image, sunny and heavy rain respectively. We believe these captured images will cover every condition in the real situation. The Rainy Image represents the noise of the

image while the Sunny Image shows the problem about the shadow from the mountain. Whereas the Heavy Rain Image presents the blurred image because of heavy rainfall with some fog.

We begin with the experimental result of the region growing method. This technique is based on the similarity of the intensity of the pixels for making the segmentation in term of the graph. The Mid is the key variable to study in the region growing. When the maximum distance of intensity has been changed, the result of the predicted water area has also fluctuated. However, we found that the accuracy is significantly stable when the Mid has been set up until some value such as 0.5 for all test. Whereas the processing time is very high for the whole experiment especially the case of the Sunny Image.

The report of the accuracy and the processing for the canny edge detection has been demonstrated in Figure 9. The important variable to study for this technique is the threshold value. It can determine the remained edge in the image. The computational complexity describes the amount of time when the algorithm runs. It can be estimated by calculating the basic operations of the algorithm. For canny edge detection, the operating steps are all implemented based on the convolutions of the image with a fixed size kernel. Therefore, the whole time for the complexity of this method is $O(m \times n \log [m \times n])$. From the results, the processing time is very fast when compared with the case of region growing.

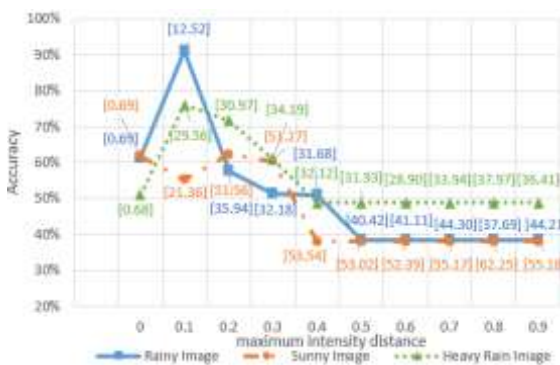


Figure 8. Accuracy [and processing time] under the region growing technique with the difference maximum intensity distance for three images

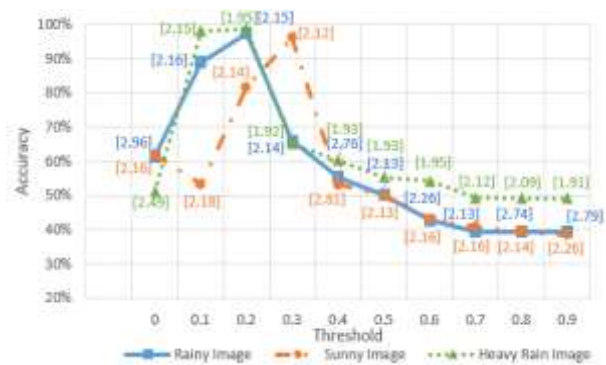


Figure 9. Accuracy [and processing time] of the canny edge detection with varied threshold for diverse environmental images

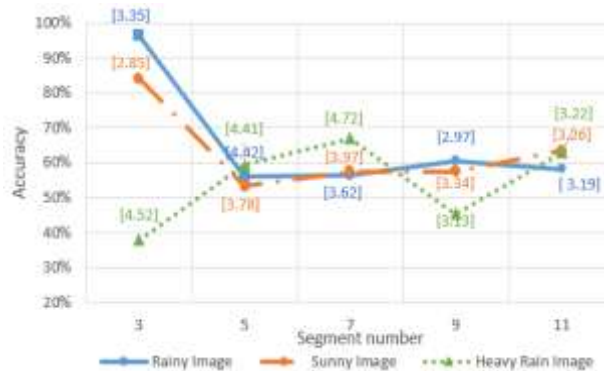


Figure 10. Accuracy [and processing time] of the normalized cut with different segment number for various environmental images

The experimental result of the normalized cut method is represented in Figure 10. The significant variable is the segment number has been studied the variation. This variable presents the desired number of the segment part of the image. The accuracy has oscillated for all test. We found that the accuracy changes dramatically when the segment number is the low value from the outcome. Nevertheless, the accuracy seems constant for the higher value of the segment number.

Our system desires to achieve high accuracy and low processing time in order to be able to implement in the real hardware such as the embedded system. The comparison of two metrics which are

accuracy and processing time have been studied in the deviation techniques and the adaptability environment images as shown in Table 1.

Table 1. Results for Each Technique in Diverse Environmental Images under Considering the Best Accuracy and Processing Time

| Accuracy (%) [Processing Time (s)] | Rainy Image | Sunny Image | HeavyRain Image |
|---------------------------------------|---------------|---------------|-----------------|
| Region growing | 91.17 [12.52] | 62.34 [51.56] | 76.05 [29.56] |
| Canny edge detection | 97.40 [2.15] | 96.05 [2.12] | 98.77 [1.95] |
| Normalized Cut | 96.43 [3.35] | 84.08 [2.85] | 66.97 [4.72] |

As demonstrated the summary of the experimental result in Table 1, the canny edge detection can gain the high accuracy and the low operating time when comparing to the other techniques. Nevertheless, the threshold parameter for this method are all different value such as 0.2, 0.3 and 0.2 for the Rainy Image, Sunny Image and HeavyRain Image respectively. Furthermore, the results appear obviously the processing time of the Canny edge detection is the best one for all schemes. it's explicitly shown that the precision of Canny edge detection and region growing keeps steady when the threshold and the maximum intensity distance have been changed more than limited value. Whereas the correctness of normalized cut are unstable with the change of the segment number parameter. In addition, the accuracy of Canny method is more than 95% and also higher than others. Besides, the processing time of Canny manner is the fastest time and not excess more than 2.2 seconds for all the times.

Considering the repeatability of the canny method, we still test our all methods with the other six images. In the six images, R1 to R4 have been obtained from the same place but with different locations and L1 to L2 have been captured from another lake. The six original images are shown as in Figure 11.

After the experiments have been repeated for all the other images, we have acquired the results as illustrated in Table 2 that canny edge detection is the best one comparing to the other method for the majority of images. In order to detection the flood level in the real environment, we will test this method on real hardware. For the case study, we use the embedded system, Raspberry Pi. It has been explained in the next section. Nevertheless, canny method has still been the best technique for the real-time flood monitoring system in term of the stability of the processing time. The experimental result has been measured for the reliability of the best selection.

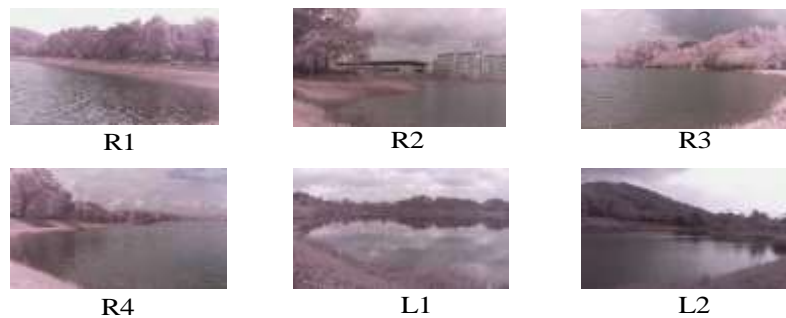


Figure 11. The different images for testing the repeatability of canny edge detection method

Table 2. Results under Considering the Highest Accuracy with Processing Time in the Bracklet for each Image Processing method in more Different Images

| Accuracy (%) [Processing Time (s)] | Region growing | Canny edge detection | Normalized Cut |
|---------------------------------------|----------------|----------------------|----------------|
| Picture R1 | 76.11 [9.96] | 56.05 [3.84] | 72.64 [4.80] |
| Picture R2 | 72.90 [1.65] | 90.64 [2.31] | 76.56 [4.42] |
| Picture R3 | 88.65 [23.36] | 96.72 [2.19] | 71.27 [4.01] |
| Picture R4 | 80.86 [47.09] | 94.92 [2.31] | 90.30 [3.97] |
| Picture L1 | 62.49 [45.97] | 76.15 [2.21] | 70.22 [4.56] |
| Picture L2 | 64.08 [1.65] | 64.08 [2.33] | 73.95 [3.69] |

4. IMPLEMENTATION

In our experiment, the input images still are the same with the source image in previous test, which is captured by the pi camera with the 600x1024 pixels resolution. The process of detecting the flood area will be completed in the Raspberry Pi, whose type is third-generation Raspberry Pi with model B and the processor is Quad Core 1.2GHz Broadcom BCM2837 64bit CPU, RAM 1GB.

However, some functions in python on Raspberry Pi are different with the experiment in MATLAB on the computer which has been changed and improved the process step from Figure 2 to detect the flood area in the real world implementation, which is shown as in Figure 12.

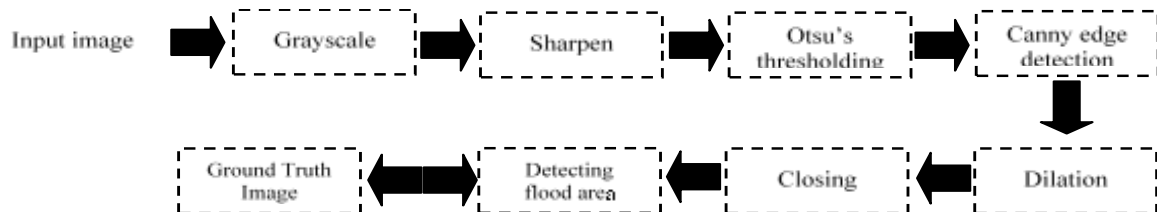


Figure 12. The improved process of canny edge detection on Raspberry pi

In Figure 12, comparing with the previous process in the experiment, we added the step of sharpening and Otsu's thresholding before canny edge detection, then added morphology dilation before closing. In the step of detecting the flood area, we change the selection of largest part as the water area to the detection based on seed points. In this process, sharpening is used to show more details of the image, which increase the contrast of edges. Otsu's thresholding is a method can detect the threshold of each image automatically [22, 23]. It is assumed that there are two classes in the image: foreground and background [24, 25]. Foreground represents the area of the flood, while the background stands for the other objects like mountain, plants, and riverbank. We can maximize the variance between the two classes or minimum the variance in the class to find the suitable threshold for segmenting the image [26-30]. Then the canny edge detection will detect the edges in the image. Previously, we just implement the morphology closing for filling the holes between the edges to separate the image into several blocks. Here we improved the method as adding the morphology dilation before morphology closing to make the edge of the image thicker for establishing the line completely. Later, the closing has filled the gaps between the edges. In the step of detecting the flood area, we detect the flood area based on the seed points. The positions of the seed points have been chosen based on the central of foreground and background in the ground truth image [29]. The area who contains the foreground seed points will present white color while the area with the background seed points will present black color. In the output image, the places show white is the detected flood area.

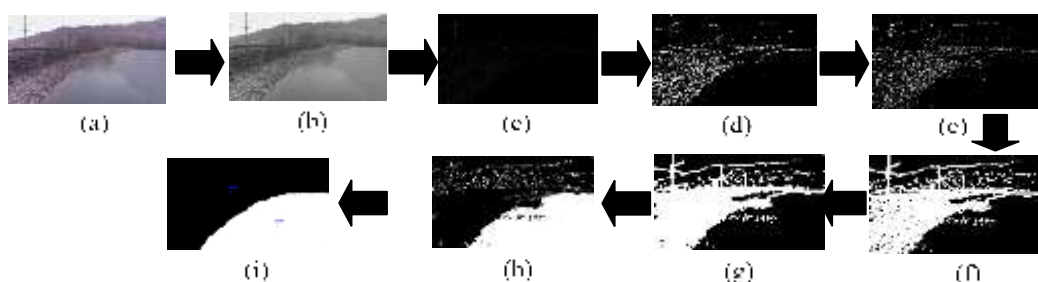

















Figure 13. The example of rainy image with improved canny edge detection on embedded system

Figure 13 is an example of the improved process with the rainy image as the input one. The image from (a) to (i) corresponded to each step in Figure 12. Figure 13 (a) is the input image and (b) is the grayscale image of (a). Figure 13 (c) shows the result of sharpening, while (d) is the result after Otsu's thresholding, where the white place is the foreground and the black area belongs to the background. Figure 13 (e) illustrates the result after canny edge detection, which can detect the edge with the automatic threshold which

has been obtained from Otsu’s thresholding. Figure 13 (f) makes the edges thicker through the morphology dilation, which will also connect the separate short edges into lines. Figure 13 (g) demonstrates the blocks after the morphology closing. Figure 13 (h) is the result of the detected water area based on the seed points which have been selected the position from the ground truth image in Figure 13 (i). The seed points in Figure 13 (i) are the center of the foreground (water area) and background (not water area).

We implemented the best image-based technique in the real hardware which is the embedded system using python. The experiment has been made to measure the metrics which are the accuracy and processing time in the real world implementation. Therefore, we can compare the results with the test using MATLAB in the previous section in term of these two metrics for the different images as shown in Table 3.

Table 3. Results under Considering the Highest Accuracy with Processing Time between Experiment and Implementation Results

| Image | Accuracy (%) | | Ground Truth Image | | |
|------------|------------------|---|--------------------|--|---|
| | MATLAB | [Processing Time (s)] | | | |
| Rainy | 97.40% [2.15] | Output Image | python | Output Image | Ground Truth Image |
| Sunny | 96.05% [2.12] |  | 92.60% [1.30] |  |  |
| Heavy Rain | 98.77% [1.95] |  | 79.25% [2.50] |  |  |
| R2 | 90.64% [2.31] |  | 94.34% [1.87] |  |  |
| R4 | 94.92% [2.31] |  | 78.86% [1.52] |  |  |
| L1 | 76.15% [2.21] |  | 81.30% [2.90] |  |  |

From the result, we can find that the processing time of MATLAB demonstration and Raspberry Pi implementation are similar. As all the processing time in implementation on python are not more than 3 seconds, which is much faster than the speed of flooding, we can consider that the process of monitoring is in real-time. For accuracy, the changes in programming language and process step make the difference results between MATLAB and python. The biggest difference of accuracy is 17.75% from image R4, while the minimum difference is 4.43% from image heavy rainy. However, the trend of the accuracy of MATLAB and python are the same. In this paper, the improved flood monitoring system can be implemented on the real embedded system. This system doesn’t like the other papers that only make simulation on the computer, which will ensure the feasibility of the flood monitoring work.

5. CONCLUSION

From three image processing techniques have been studied and compared in terms of the accuracy and processing time with different parameters settings. For the implementation of the best method on the actual application in the different environment of the monitoring system, the canny edge detection has both high precision and low operating time. Furthermore, diverse images with different places have also been tested, which proves that the canny edge detection is the optimum solution comparing to the other methods. On the real hardware, this method has been improved procedure for the image-based processing is able to detect the flood area automatically. These two metrics have been demonstrated for both experiment and

implement. It stated that the results have been shown the same trend. Therefore this enhanced method can be used in outdoor environment for the real-time flood monitoring.

ACKNOWLEDGEMENTS

The research was supported by the Electrical Engineering Department in Faculty of Engineering, Prince of Songkla University. The author would like to thank all the colleagues from this department who assisted the research.

REFERENCES

- [1] A. Filonenko, D. C. Hernández, D. Seo, and K.-H. Jo, "Real-time flood detection for video surveillance," in IECON 2015-41st Annual Conference of the IEEE Industrial Electronics Society, pp. 004082-004085, 2015.
- [2] S. Zafar, H. S. Azhar, and A. Tahir, "A GIS Based Hydrological Model for River Water Level Detection & Flood Prediction featuring morphological operations," in 2018 International Conference on Artificial Life and Robotics (ICAROB2018), 2018.
- [3] A. L. Sumalan, D. A. N. Popescu, and L. Ichim, "Flooded areas detection based on LBP from UAV images," *Recent Advances on Systems, Signals, Control, Communications and Computers; WSEAS Press: Budapest, Hungary*, pp. 186-191, 2015.
- [4] P. Matgen, R. Hostache, G. Schumann, L. Pfister, L. Hoffmann, and H. H. G. Savenije, "Towards an automated SAR-based flood monitoring system: Lessons learned from two case studies," *Physics and Chemistry of the Earth, Parts A/B/C*, vol. 36, no. 7-8, pp. 241-252, 2011.
- [5] W. Jyh-Horng, T. Chien-Hao, C. Lun-Chi, L. Shi-Wei, and L. Fang-Pang, "Automated image identification method for flood disaster monitoring in riverine environments: a case study in Taiwan," in AASRI International Conference on Industrial Electronics and Applications (IEA 2015), 2015.
- [6] S.-W. Lo, J.-H. Wu, F.-P. Lin, and C.-H. Hsu, "Cyber surveillance for flood disasters," *Sensors*, vol. 15, no. 2, pp. 2369-2387, 2015.
- [7] S.-W. Lo, J.-H. Wu, F.-P. Lin, and C.-H. Hsu, "Visual sensing for urban flood monitoring," *Sensors*, vol. 15, no. 8, pp. 20006-20029, 2015.
- [8] K. Hiroi and N. Kawaguchi, "FloodEye: Real-time flash flood prediction system for urban complex water flow," in 2016 IEEE SENSORS, pp. 1-3, 2016.
- [9] K. P. Menon and L. Kala, "Video surveillance system for realtime flood detection and mobile app for flood alert," in 2017 International Conference on Computing Methodologies and Communication (ICCMC), pp. 515-519, 2017.
- [10] M. Saha and C. Panda, "A Review on Various Image Segmentation Techniques for Brain Tumor Detection," *Scientific Research in 2018 IJSRCSEIT*, vol. 3, 2018.
- [11] N. Mahmood, A. Shah, A. Waqas, A. Abubakar, S. Kamran, and S. B. Zaidi, "Image segmentation methods and edge detection: An application to knee joint articular cartilage edge detection," *Journal of Theoretical and Applied Information Technology*, vol. 71, no. 1, pp. 87-96, 2015.
- [12] V. R. Borges, M. C. F. de Oliveira, T. G. Silva, A. A. H. Vieira, and B. Hamann, "Region growing for segmenting green microalgae images," *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, vol. 15, no. 1, pp. 257-270, 2018.
- [13] K. Somasundaram, J. H. Mercina, S. Magesh, and T. Kalaiselvi, "Brain Portion Extraction Scheme using Region Growing and Morphological Operation from MRI of Human Head Scans," 2018.
- [14] Y. Lei, L. Zheng, and Z. Lyu, "Lung Tumor Segmentation and 3D Reconstruction Based on Region Growing and Correlation," in *Journal of Physics: Conference Series*, vol. 1168, p. 062018, 2019.
- [15] N. Mahmood, A. Shah, A. Waqas, A. Abubakar, S. Kamran, and S. B. Zaidi, "Image segmentation methods and edge detection: An application to knee joint articular cartilage edge detection," *Journal of Theoretical and Applied Information Technology*, vol. 71, no. 1, pp. 87-96, 2015.
- [16] N. Manasa, G. Mounica, and B. D. Tejaswi, "Brain tumor detection based on canny edge detection algorithm and it's area calculation," *Brain*, vol. 5, pp. 10-13, 2016.
- [17] X Wang, "Graph based approaches for image segmentation and object tracking," *Doctoral dissertation, Ecully, Ecole centrale de Lyon*. 2015.
- [18] S. Lee, G. Lee, Y. Hong, and J. Kim, "A study on the improved normalized cut algorithm using a bilateral filter for efficient object extraction from image," *Wireless Personal Communications*, vol. 86, no. 1, pp. 77-90, 2016.
- [19] A. Dutta, J. Engels, and M. Hahn, "Segmentation of Laser Point Clouds in Urban Areas by a Modified Normalized Cut Method," *IEEE transactions on pattern analysis and machine intelligence*, 2018.
- [20] X. Deng, Q. Liu, Y. Deng, and S. Mahadevan, "An improved method to construct basic probability assignment based on the confusion matrix for classification problem," *Information Sciences*, vol. 340, pp. 250-261, 2016.
- [21] J. García-Balboa, M. Alba-Fernández, F. Ariza-López, and J. Rodríguez-Avi, "Analysis of Thematic Similarity Using Confusion Matrices," *ISPRS International Journal of Geo-Information*, vol. 7, no. 6, p. 233, 2018.
- [22] Q. Zhang, N. Jindapetch, R. Duangsoithong, and D. Buranapanichkit, "Investigation of Image Processing based Real-time Flood Monitoring," in 2018 IEEE 5th International Conference on Smart Instrumentation, Measurement and Application (ICSIMA), pp. 1-4, 2018.

- [23] K S Praveen, G Hamarnath, K Babu, M Sreenivasulu, K Sudhakar, "Implementation Of Image Sharpening And Smoothing Using Filters," *International Journal of Scientific Engineering and Applied Science*.vol. 2(1): 7-14, 2016.
- [24] G. Jie and L. Ning, "An improved adaptive threshold canny edge detection algorithm," in 2012 International Conference on Computer Science and Electronics Engineering, vol. 1, pp. 164-168, 2012.
- [25] J. Cao, L. Chen, M. Wang, and Y. Tian, "Implementing a parallel image edge detection algorithm based on the Otsu-canny operator on the Hadoop platform," *Computational intelligence and neuroscience*, vol. 2018, 2018.
- [26] V.Prema, M. Sivasubramanian, S Meenakshi, "Brain cancer feature extraction using Otsu's thresholding segmentation," *International Journal of Computer Application*, pp. 21-29, 2016.
- [27] P. P. Vijay and N. C. Patil, "Gray Scale Image Segmentation using OTSU Thresholding Optimal Approach," *Journal for Research*, vol. 2, no. 05, 2016.
- [28] G. Jie and L. Ning, "An improved adaptive threshold canny edge detection algorithm," in 2012 International Conference on Computer Science and Electronics Engineering, pp. 164-168, 2012.
- [29] S. Kansal and P. Jain, "Automatic seed selection algorithm for image segmentation using region growing," *International Journal of Advances in Engineering & Technology*, vol. 8, no. 3, p. 362, 2015.
- [30] T.S. Gunawan and M. Kartiwi, "On the use of edge features and exponential decaying number of nodes in the hidden layers for handwritten signature recognition," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 12(2). pp. 722-728, 2018.

BIOGRAPHIES OF AUTHORS



Qianyu Zhang. M.Eng student in the department of Electrical Engineering of Prince of Songkla University, Thailand. She received the B.Eng. degree in Electrical Engineering from Anhui Polytechnic University, China, in 2016. Her research interests are in embedded systems and image-based processing.



Nattha Jindapetch. Received the B.Eng. degree in Electrical Engineering from PSU, Thailand, in 1993 and the Ph.D. degree in Information Technology from the University of Tokyo, Japan, in 2004. She now is an Associate Professor in the Department of Electrical Engineering, PSU. Her research interests are FPGAs, embedded systems, and sensor networks.



Rakkrit Duangsoithong. Received the B. Eng degree in Electrical Engineering from Chiang Mai, Thailand followed by M.Eng. in Electrical Engineering from Prince of Songkla University, Thailand. After he finished PhD degree from University of Surrey, U.K in 2013, he has started to work as a lecturer in Electrical Engineering Department, Faculty of Engineering, Prince of Songkla University, Thailand. His research areas are Machine Learning, Computer Vision, Signal Processing and Data Analysis



Dujdow Buranapanichkit. Received the B.Eng. degree in the department of Electrical Engineering from Prince of Songkla University, Thailand, in 1999 and the Ph.D. degree in the Department of Electronic and Electrical Engineering from University College London, UK in 2013. She now is an Assistant Professor in the Department of Electrical Engineering, PSU. Her research interests are in wireless sensor networks and distributed synchronization mechanisms and protocol design.