

Vision-Based Pipe Monitoring Robot for Crack Detection Using Canny Edge Detection Method as an Image Processing Technique

Nur Mutiara Syahrin¹, Pola Risma², Tresna Dewi³

^{1,2,3}Politeknik Negeri Sriwijaya Palembang/Electrical Engineering Department
nmsyahrin@gmail.com¹, pola_risma@polsri.ac.id², tresna_dewi@polsri.ac.id³

Abstract

Piping setup is very important to ensure the safety and eligibility of the piping system before applied in industry. One of the techniques to facilitate perfect piping setup is by employing pipe monitoring robot. Pipe monitoring robot is designed in this research to monitor cracks or any other defects occurring inside a pipe. This automatic monitoring is conducted by the application of image processing with Canny edge detection. Canny edge detection method detects the edges or lines of any cracks inside a pipe and processes them to create image separations. Therefore, only the cracks can be visualized and those cracks can afterwards be analyzed. Canny edge detection has 5 processing techniques namely smoothing, finding gradients, giving non-maximum suppression, executing double thresholding, and tracking edges by hysteresis. In this research, the experiment was conducted by utilizing a robot monitoring new pipes and detecting cracks. Two crack samples were taken and analyzed. The results show that the best value for smoothing is 10 and 5 for thresholding in getting not too blur or too sharp result.

Keywords: Image Processing, Canny Algorithm, Edge Detection, Vision-based Monitoring robot.

1. Introduction

Pipe is the main medium for distributing gas and fluid in an industry; therefore, ensuring pipes in good condition before installation becomes highly important. The inspection and motoring operations need to be conducted to ensure neither cracks nor defects occur since the occurrences of any cracks and defects will result in huge drawbacks in the industry such as explosion and fire due to the release of hazardous gas and fluid [1].

Research have been conducted to detect defects and cracks on a pipe, one of them was done by Anindya et al. (pipe tracking robot as a leakage detector). This method only works for installed pipe where the gas is already distributed inside the pipe. The novelty of this research is that the robot detects cracks and defects inside the pipe before installation, thus, dangerous conditions can be prevented. This method is safer and less costly since implementing detection before installation. If the monitoring is completed after pipes installed, the defected pipe needs to be replaced. It can cause more complicated cases compared to checking before installation. This study applies vision-based pipe monitoring robot as a pipe crack detector using Canny edge detection method for image processing technique.

Pipe monitoring robot is a mobile robot equipped with a camera as the robot vision functioning as the "eye" to monitor any cracks or defects inside pipes. The detection was made possible by applying image processing technique [2][3]. Image processing technique using Canny detection algorithm can detect edges or lines indicating cracks inside the pipe and processed the images separation to produce a different image by only displaying detected cracks; hence, the crack can be analyzed.

1.1 Image Processing

Image processing is a method to process raw images from camera mathematically using any signal processing methods. The input of raw image is processed to be another image or a certain characteristic/parameter related to the image. Digital image processing was implemented by applying computer application/algorithm to process digital raw image and by isolating any intended characteristic or parameter. Digital image processing allowed more complex detections and faster results [4].

1.2 Edge Detection

Edge is a variation of gray scale indicating the boundaries of objects inside an image as the result of difference brightness occurring within an image that changes either sharply or gradually. Edge detection method detects all the edges or lines showing objects within an image and makes them more visualized and clear to be observed by the viewer. One of the edge detection methods is Canny [5][6].

1.3 Canny

Canny edge detection is the development of basic edge detection method. This method was developed by John Canny in 1986, using multi-step algorithm to detect edges within an image [5][7][8][9][10]. There are steps conducted to execute Canny edge detection, smoothing, finding gradients, performing non-maximum suppression, performing double thresholding, and tracking edge by hysteresis.

1.4 Smoothing

Smoothing blurs the image to reduce noises. Noises naturally occur from the camera. Hence, to ensure edge detection instead of noise detection, certain numbers of noises have to be eliminated. The first step of smoothing was using a filter, Gaussian filter. The main idea of Gaussian filter employed a standard deviation, $\sigma = 1.4$ The equation is presented by Equation 1 below.

$$B = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} \quad (1)$$

Equation 1 is matrix filter used to convert matrix pixel within to convert raw image. Meanwhile, standard deviation is the multiplier of how many pixels within a raw image will be converted by the Gaussian filter. The matrix shows the pixel distribution referring to the tested edge.

1.5 Finding Gradients

The idea of the Canny algorithm is to get the edges of the grayscale image by investigating the highest changing intensity. The highest intensity areas are decided by calculating the gradient of the raw image. The gradient of each pixel was processed using Sobel-operator. The first step was by approximating the x and y direction of the processed gradient. The x and y gradient is shown in Equation 2.

$$K_{GX} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (2)$$

$$K_{GY} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Raw and column numbers inside the matrix were calculated using image gradient (∇f) of the raw image and simplified using finite differences $f(x+1, y) - f(x, y)$ and $f(x, y+1) - f(x, y)$, where x and y were the position of the pixel within the raw image.

Gradient magnitude is defined as the distance from Euclidean measured by using Pythagoras law, presented in Equation 3. This measurement is usually simplified by applying Manhattan distance (calculating distance by measuring the distance from the initial point to final point/block to block), as shown in Equation 4. Manhattan distance is intended to reduce the computational complexity.

G_y and G_x are the gradients of x and y direction respectively in a unit pixel. The magnitude of G_y and G_x shows the clearest edge. However, sometimes the edges were wide; therefore, not showing the exact position of the edges. To overcome this condition, the edges direction had to be determined and memorized using Equation 5.

$$|G| = \sqrt{G_x^2 + G_y^2} \quad (3)$$

$$|G| = |G_x| + |G_y| \quad (4)$$

$$\theta = \arctan \left(\frac{|G_y|}{|G_x|} \right) \quad (5)$$

1.6 Performing Non-Maximum Suppression

After blurring the raw image in smoothing, the next step was performing non-maximum suppression intended to sharpen the blurred image. This was completed by letting only local maxima of existing gradient image and erasing others.

1.7 Performing Double Thresholding

The next step after non-maximum suppression was performing double thresholding conducted by letting the remaining edge pixels marked by using the strengthened pixel by pixel. Most of the pixels could be the real edge of the image. Nevertheless, some of the pixels might occur due to noises or color variation caused by the rough surfaces. To isolate the real edge of the image, double thresholding technique was utilized, so only the strongest edge was maintained. The edge pixels higher than thresholding value were marked and maintained while those edge pixels lower than thresholding value were erased. Pixels situated between higher thresholding and lower thresholding were considered weak.

1.8 Edge tracking by hysteresis

The strong edge was considered as the real edge and was included as the final edge detection for image processing. The weak edge was only considered if connected to the actual edge. It was based on the logic that noises and other colors variation cannot create a sharp strong edge. Therefore, the strong and sharp edge will create the real edge of the image. This process can be implemented by using Blob Analysis to the image. After this final process, the strongest edge will be presented, and Canny edge detection was finished.

1.9 Robot Design

The pipe monitoring robot was designed to be able to move smoothly inside a pipe by installing retractable feature making robot locked its "feet". This robot can retract up to 10-inch pipe. This limitation is just for prototype purpose. The robot can be designed bigger or larger for larger pipes, or smaller for smaller pipes, without any significant design changes. Robot design is illustrated in Figure 1.

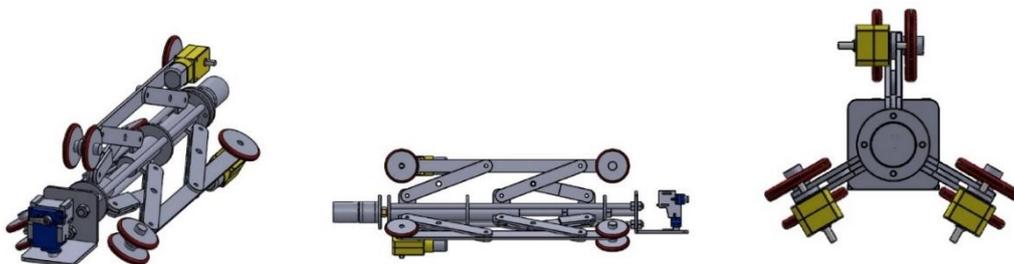


Figure 1. Robot Design

2.0 Block Diagram

Block diagram for robot design and control is illustrated in Figure 2. The detail explanation of the blocks in Figure 2 is presented below:

2.0.1 Ground Station

Ground station block is the monitoring station for pipe monitoring robot. This station includes monitoring devices and remote control. The ground station consists of:

1. PC or mobile devices to monitor the image captured by the camera producing raw image processed using Microsoft Visual Studio and OpenCV.
2. Video receiver block as a data receiver from the transmitter yielding data as an image captured by a camera attached to the pipe monitoring robot, utilizing a wireless camera of TS5823 TX.
3. Remote transmitter block as a transmitter device for sending data to control the robot movement inside the pipe.

2.0.2 Robot

Robot block diagram presents the installed devices for the robot:

1. Transmitter video block as a transmitter data device to transmit the captured image by using video TX RC305.
2. Camera block as the “eye” for the robot, utilizing FTV camera.
3. Remote receiver block as a data receiver sent by the remote transmitter, both remote transmitter and receiver used utilizing Turnigy 5X.
4. Microcontroller installed as an Arduino Nano.
5. Motor Driver Block as a controller for DC motor, employed a driver utilizing L293D module installed in this robot.
6. DC motor block as an actuator of the robot being able to adjust according to the size of the monitored pipe.
7. Servo motor block as an actuator for the camera.

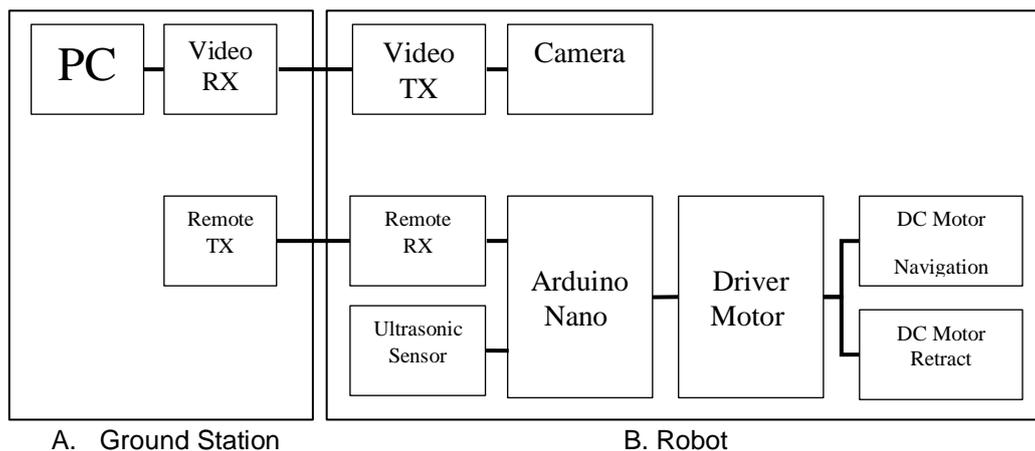


Figure 2. Robot Block Diagram

2. Research Method

The research methodology used in this research was by analyzing the changing/alterd values in smoothing and thresholding process during Canny edge detection image processing. Two cracks samples were taken from two different 6-inch pipes. The samples were taken by moving the robot inside the pipes. Cracks images taken by the camera installed on the robot were then processed by changing the smoothing and thresholding value for 11 times. The process was conducted 11 times since the values of smoothing kept on changing in short ranges as shown in Figure 3 [11][12].



Figure 3. Data Image Processing

The system analysis stage is a very important step since any mistakes occurring in this step can create a domino effect of errors to the next step depicted by Figure 4. The analysis

process stage in developing the system was the procedure to check the problem and arrange the problem solution to create a new system specification.

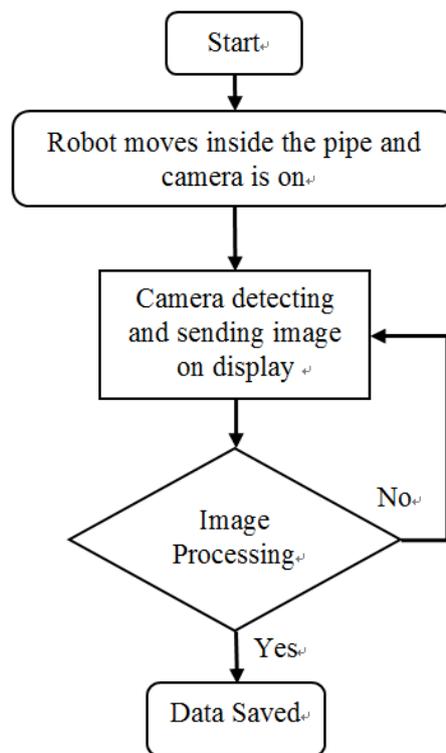


Figure 4. Flowchart

3. Results and Discussion

Canny edge detection experiment was conducted in 5 processes before getting the detected cracks inside the pipe. The processes are presented in Figure 5 [13][14].

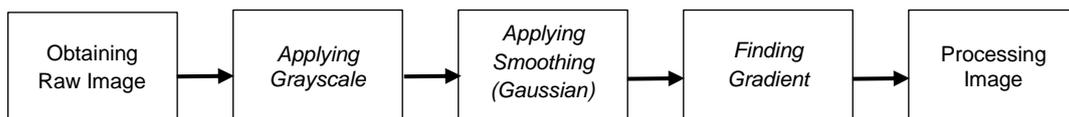


Figure 5. Canny Edge Detection Image Processing

Grayscale is the colors of pixels within the range from black gradating to white. By converting the raw image to grayscale, it helped to simplify the edge detection method. Smoothing process was conducted to reduce noises in the detecting process since errors might occur in clearly visible noises leading to misinterpretation as the edge. Subsequently, the smoothing process utilized a Gaussian filter to blur the image. The result from the Gaussian filter was used for the next step to define the image gradient potential.

A gradient is the closest operator to define an edge. In the process of calculating image gradient potential, there are two required information, edge strength or magnitude and edge direction or orientation. Sobel operator uses two edge templates in horizontal and vertical direction and calculates edges direction by using the arctangent of both values. Figure 6 presents the results of every process.

In this study, the images from two crack samples were processed in 11 times, to get the most suitable value of smoothing and thresholding, and the cracks will clearly visible. Data of the raw image and processed image of the crack is presented in Table 1.

Data resulted from sampling in Table 1 shows that if thresholding value is less than 5, the edge of crack cannot be detected. As mention in the introduction, the higher Gaussian blur magnitude used results in more deviation. The pixels are split and blurriness occurs. The

difference between two samples is the first sample is a thick crack, and the second one is a thin crack. These two samples are taken to see the detection discrepancies, the obvious cracks and defects and the less obvious ones.

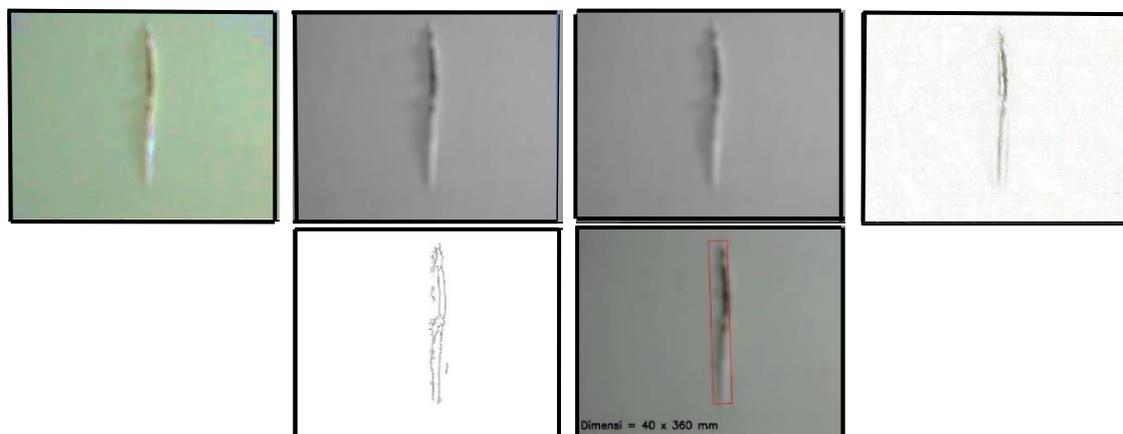


Figure 6. Canny Edge Detection Processes

Thresholding range in this study was from 3 to 7. In the second experiment, the thresholding value was increased to 5, and the edge was detected; however, the noises were still detected as edges due to the small value of smoothing making noises be visible. Measurement range of smoothing in this research was from 1 to 100. When the value of smoothing was increased to 3, noises reduced, but still lots of noises were considered as edges. The best thresholding value was 5, therefore, adjusting the smoothing value was necessary.

When smoothing value was increased to 5, the crack was visible with less noise. In the row no. 5 on Table 1, smoothing value increased to 7, and the edge was well detected with almost no noise to both sample. When smoothing value was increased to 10, there was no noise anymore and the edges were well detected.

After the detection was well achieved, the next experiment was by increasing smoothing value to 30 in the row no. 7. The noise was not detected anymore but the detected edge was quite different from the real edge, the line from the edge was not going to one direction. The crack to the second sample was slowly disappearing.

When smoothing value was increased to 50 in a row no. 8, the edge in the first sample was disappearing, and the noise was not visible anymore. On the other hand, the edge was getting smaller in the second sample. Smoothing was increased again to 80. The edge was difficult to detect leading to blurry image. Consequently, the crack on the second sample was not detected.

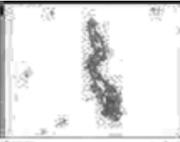
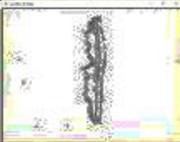
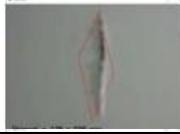
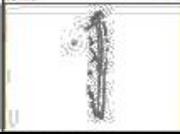
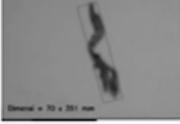
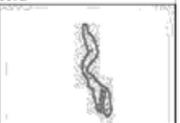
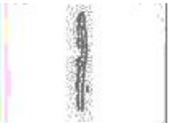
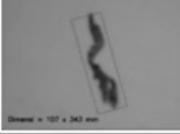
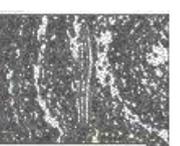
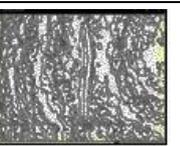
After the increased smoothing value could not detect the edge anymore. The experiment was conducted by increasing thresholding value to 7 and smoothing value was returned back to 5. This condition created noises leading to too bright image due to high value of thresholding. When smoothing value was increased to 7, the edge detection was still too noisy, so the processed image was not clearly displaying cracks. After comparing all results, the best value for smoothing is 10, and the best value for thresholding is 5. All cracks are well detected, and the processed image is not too bright nor too sharp.

4. Conclusion

Based on the experiment result and analysis, this study can be concluded as follows:

1. Edge detection method with Canny algorithm has 5 technical steps, smoothing, finding gradients, performing non-maximum suppression, performing double thresholding and tracking edge by hysteresis.
2. Processing the sample crack images can be performed inside pipes.
3. The data results show that the best value for smoothing is 10, and 5 for thresholding resulting in not too blurry nor too sharp processed image.
4. Smaller smoothing value results in less blur image but very noisy. However, the smaller thresholding value creates a less detected edge on the image.

Table 1. Data for Smoothing and Thresholding Process Within Canny Edge Detection

No	Smooth	Threshold	Raw (Left) and Processed Image (Right)			
			1 st Sample Image		2 nd Sample Image	
1	1	3				
2	1	5				
3	3	5				
4	5	5				
5	7	5				
6	10	5				
7	30	5				
8	50	5				
9	80	5				
10	5	7				
11	7	7				

References

- [1] C. Anindya, S. Fuada, S. Firmansyah, and D. Lestari, "*Design of Pipe Tracking Robot Using Electric Nose technology as a Leakage Detector*," Student Creativity Program - Karsa Cipta, Department of Electrical Engineering Education FT UNM, 2011.
- [2] T. H. Dinh, Q. P. Ha, and H. M. La, "*Computer Vision-Based Method for Concrete Crack Detection*," in 2016 14th International Conference on Control, Automation, Robotics & Vision (ICARCV), Thailand., 2016.
- [3] A. Mohan and S. Poobal, "*Crack Detection Using Image Processing: A Critical Review and Analysis*," Alexandria Engineering Journal, Feb. 2017.
- [4] T. Sutoyo, E. Mulyanto, V. Suhartono, and Wijanarto, "*Theory of Digital Image Processing*," Yogyakarta, 2009.
- [5] C. Danil, "*Edge Detection using Canny Algorithm*," STMIK IBBI, 2011.
- [6] A. Y. Zunaidi, "*Design of Empty Parking Space Based on Digital Image*," Pp. 5-17, 2011.
- [7] S. Riyanto, "*Image Processing Practice*," Politeknik Elektronika Negeri Surabaya, 2006.
- [8] F. Astuti, "*Digital Image Processing: Concept and Theory*," Yogyakarta: ANDI Publisher, 2013.
- [9] R. Munir, "*Digital Image Processing using Algorithm Approach*," Informatika Bandung, 2004.
- [10] U. Ahmad, "*Image Processing and Its Programming Techniques*," Yogyakarta: Graha Ilmu, 2005.
- [11] A. Basuki, F. Palandi, Jozua, and Fatchurochman, "*Digital Image Processing Using Visual Basic*," Yogyakarta: Graha Ilmu, 2005.
- [12] R. C. Gonzalez and R. E. Richard E. Woods, "*Digital Image Processing*," New Jersey: Prentice-Hall, 2009.
- [13] R. Sigit, D. Pramadihanto, and R. Sulaiman, "*Facial Recognition System in Real Time*," Industrial Electronic Seminar (IES) Proceedings, 2005.
- [14] L. B. Ethes, P. Menezes, F. Lerasle, and J. Hayet, "*Face Tracking and Hand Gesture Recognition For Human-Robot Interaction*."