Development of SBC based Machine-Vision System for PCB Board Assembly Automatic Optical Inspection

Faisal Ardhy[1], Farkhad Ihsan Hariadi[1,2]

[1]Microelectronic Center, Institut Teknologi Bandung, Bandung, Indonesia
[2]School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Bandung, Indonesia
(*Corresponding author: phone : +62-22-251-0801; e-mail: hariadi@ic-proc.paume.itb.ac.id)

Abstract – Visual Inspection is an essential part in the quality control in electronic manufacturing industry, especially in PCB assembly processes. With the advance of surface mount technology as a means to increase automation level in electronic assembly line, with production volume achieving thousands of board per day, manual visual inspection becomes increasingly prohitive. For this purpose, Automatic Optical Inspection (AOI) based on machine vision can be used. This method, however requires considerablce amount of investment that is not affordable by small companies. In this research work, a low-cost machine vision based on an SBC (Single Board Computer) is developed. Raspberry-Pi™ as a populer SBC offers capability of being interfaced with camera and of running Linux-based operating system such as Ubuntu™. These capabilities enable this board to be used to run sophisticated image-processing programs for machine vision.

AOI is generally accomplished by first capturing the image of the inspected board. The stored imaged is then processed using certain algorithm to detect the deviation of this image as compared to a reference image. Assembly processing defects such as component missing, placement position inaccuracies, and other defects could be detected using this manner. In this particular researh work, we make a prototyping system to detect it automatically in order to help the worker recognizing the PCB defects using a Raspberry-Pi board, mouse and keyboard, CNC X-Y mover, a display device, and a USB microscope.

We compare 2 usual methods those are Canny Edge Detection and Sobel Edge Detection with our proposed method that is Adaptive Gaussian Threshold.

Keywords: PCB Assembly, Automatic Optical Inspection (AOI), Single Board Computer (SBC), Adaptive Gaussian Threshold

I. INTRODUCTION

At present, as the increase of electronic equipments demand, the need of accurate PCB also increased. One of the testing stage is defect test of circuit and component.

II. LITERATURE STUDY

A. OpenCV

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code. [5]. There are a lot of programming language supported by OpenCV such as C++, Python, etc. OpenCV has a complet documentation site and big community because of its simplicity and robustness.

B. Gradient-based Edge Detection

Every edge in the image usually has very high difference with another especially color. By measuring the gradient of every point, we can recognize the edge. There are a lot of operators that commonly used such as Sobel and Canny operators. It consist of 3x3 horizontal and vertical masking matrix.

Sobel Edge Detection

Horizontal $\begin{vmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{vmatrix}$ Vertical $\begin{vmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{vmatrix}$

Canny Edge Detection

Horizontal $\begin{vmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{vmatrix}$ Vertical $\begin{vmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{vmatrix}$

The output image can be computed using these formulas:

Horizontal Gradient (HG) = Input * Horizontal Mask
Vertical Gradient (VG) = Input * Vertical Mask
Output = Square Root $((HG)^2 + ((VG)^2)$

C. Binary Threshold

Binary Thresholding is a method to make binarization (black and white) of the image base on a threshold level. There are two common methods:
1. Global Threshold, whole points will be thresholded by a threshold level.

```
   if >= Threshold then Black, else White
```

2. Adaptive Threshold, divide the image into little parts then thresholding every parts. There are 2 common methods:
a. Median, threshold value is the mean of neighbourhood area.
b. Gaussian, threshold value is the weighted sum of neighbourhood values where weights are a gaussian window.

III. SPECIFICATION AND DESIGN

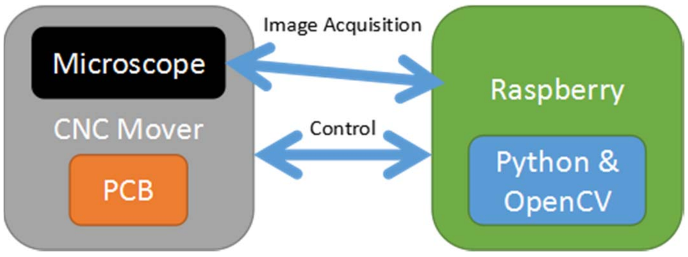

Fig. 1. The Proposed AOI Diagram Block

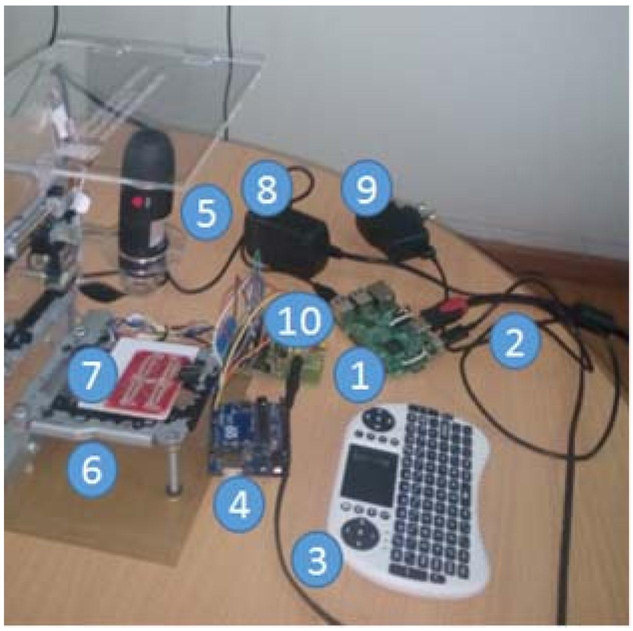

Fig. 2. The Proposed System

Devices:

1. Raspberry Pi 3 Model B with Ubuntu Mate^{TM} OS

2. HDMI Cable

3. Wireless keyboard with touchpad

4. Arduino Uno

5. Digital USB microscope

6. CNC X-Y

7. Board place

8. 3A 5V adapter for motor

9. 2A 5V adapter for Raspberry

10. Motor Driver

11. Monitor for display

A. CNC X-Y Mover

This module is originated from Hans Ega, Hans Kasan, and Revie Marthensa's microprocessor system class final project using 2 modified CD Player, an Arduino Uno Board, a motor driver module, and an adapter 3A 5V.[6] Their project then modified with some addition for microscope holding, cables line, and cover. The horizontal direction is for moving the microscope and the vertical direction is for moving the object. The X-Y position can be programmed and uploaded by using Arduino IDE that is installed in the Raspberry. It also can do serial communication with Raspberry through USB so the movement can be directly controlled.

B. Image Acquisition

Our proposed system use a 2 megapixels Digital USB Microscope with built-in 8 controllable LED for take the photograph of board. We choose it because of it use USB connection, low price, up to 500 times optical zoom, and easy to mount. The picture then saved in the Raspberry SDCard memory through our python program using OpenCV module.

C. Image Processing

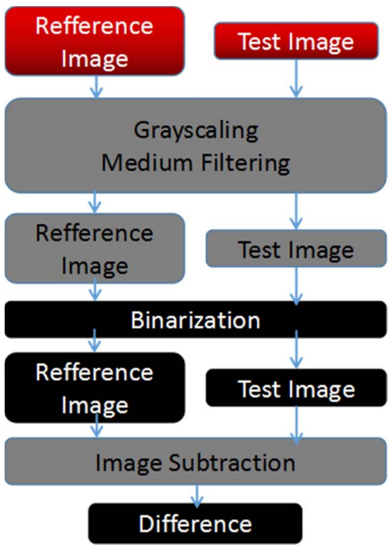

Fig. 3. The Proposed Raspberry-Pi Machine Vision

This module divided into several stages:

1. Grayscaling from RGB image

2. Median filter for optimal noise removal and less blur effect so the image detail doesn't decrease significantly

3. Binarization (black and white), the proposed method is using adaptive gaussian threshold but we compare it with Sobel and Canny edge detection

4. Image subtraction operation for getting the difference of refference and test image

IV. IMPLEMENTATION AND RESULT

In this research, we compare 2 common methods with our proposed method:

1. Canny Edge Detection

```
#refference image

cannyr = cv2.Canny(reff,100,200)

#test image

cannyt = cv2.Canny(test,100,200)

#subtraction test
```

We use the built-in canny edge detection in OpenCV that need 2 threshold values. 100 is the minVal and 200 is the maxVal. Any edges with intensity gradient more than maxVal are sure to be edges and those below minVal are sure to be non-edges, so discarded. Those who lie betwen these two thresholds are classified edges or non-edges based on their connectivity. If they ar connected to "sure-edge" pixels, they ar considered to be part of edges. Otherwise, they are also discarded.

2. Sobel Edge Detection

```
ddepth = cv2.CV_64F #output image depth

ksize = 3 #3x3 common sobel masking matrix

#refference image

sobelrx = cv2.Sobel(reff,ddepth,1,0,ksize)
#hor grad

sobelry = cv2.Sobel(reff,ddepth,0,1,ksize)
#vert grad

sobelr = (sobelrx*sobelrx +
sobelry*sobelry)**0.5

cv2.imwrite('sobelreff.jpg',sobelr)

#test image

sobeltx = cv2.Sobel(test,ddepth,1,0,-1) #hor
grad
```

We use built-in sobel edge detection in OpenCV to get horizontal and vertical sobel gradient. We use 3x3 masking matrix as commonly used.S

3. Adaptive Gaussian Threshold (proposed method)

```
blocksize = 11 #thresholding each 11x11 area

#adaptive gaussian threshold refference image

threff =
cv2.adaptiveThreshold(reff,255,cv2.ADAPTIVE_THR
ESH_GAUSSIAN_C,\cv2.THRESH_BINARY,blocksize,2)

#adaptive gaussian threshold test image

thtest =
cv2.adaptiveThreshold(test,255,cv2.ADAPTIVE_THR
ESH_GAUSSIAN_C,\cv2.THRESH_BINARY,blocksize,2)
```

We use built-in adaptive gaussian threshold in OpenCV. Blocksize decides the size of neighbourhood area used as adaptive threshold.

We make two experiments. The first experiment is to recognize the component missing. And the second one is to recognize the short circuit. By using direct subtraction test for the refference and test image, we prove that the test image is another photo, not the manipulated of the refference picture. So the difficulty to know the defect is very high because there are a lot of differences such as color intencity, luminous, etc.

A. First Experiment

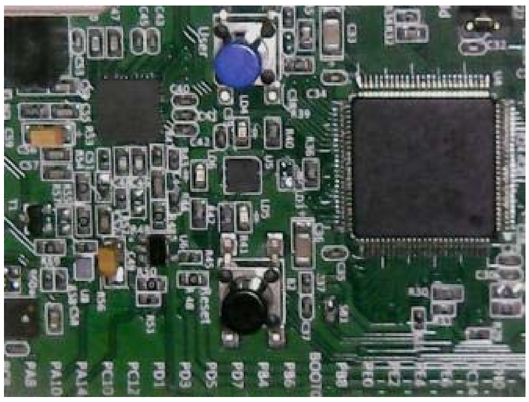

Fig. 4. First Experiment-Refference Image

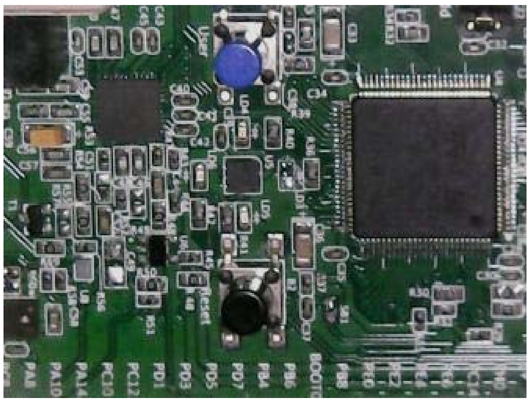

Fig. 5. First Experiment-Test Image

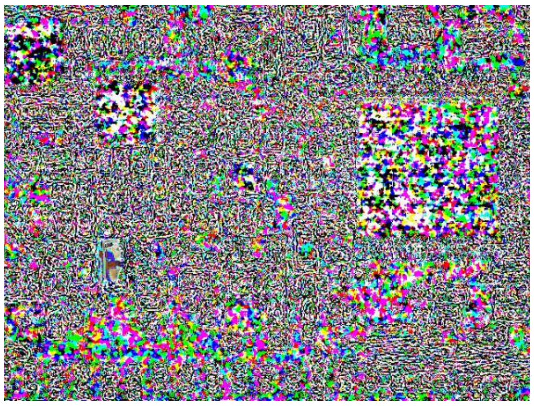

Fig. 6. First Experiment-subtraction Test

A.1 Canny

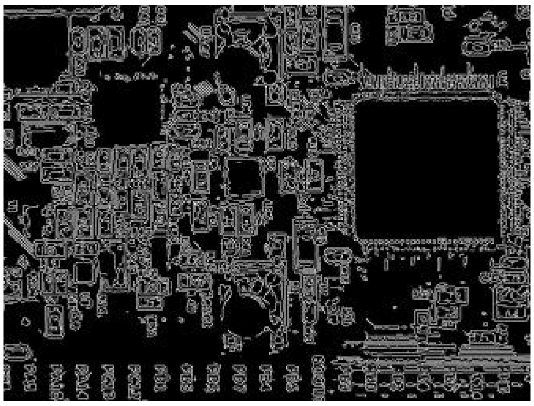

Fig. 7. First Experiment-Canny Edge Detection of Refference Image

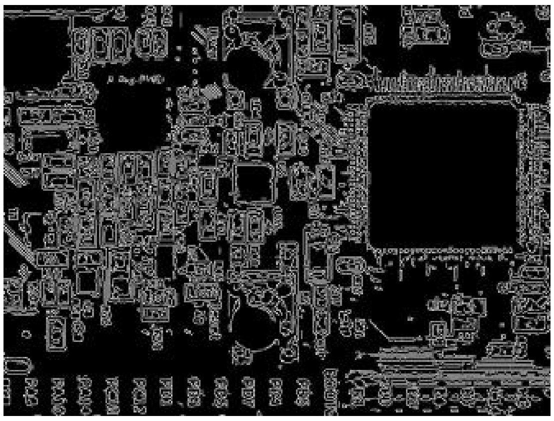

Fig. 8. First Experiment-Canny Edge Detection of Test Image

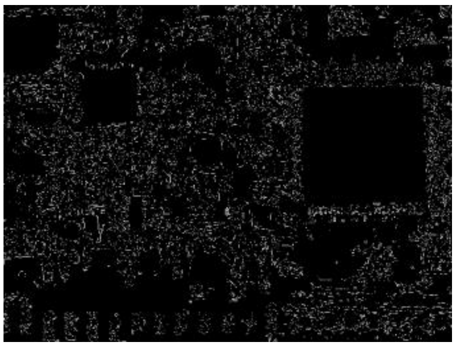

Fig. 9. First Experiment-subtraction Test of Canny Edge Detection

As we can see, the canny edge detection very poor to detect the details of component. It also very poor to detect the defect.

A.2 Sobel

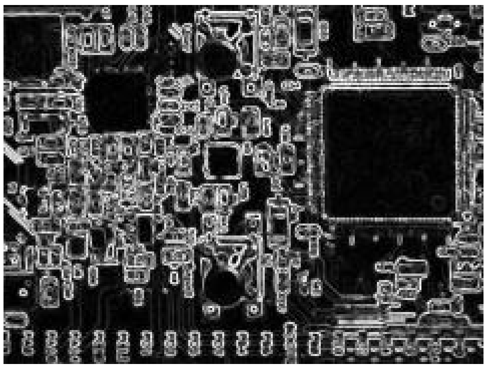

Fig. 10. First Experiment-Sobel Edge Detection of Refference Image

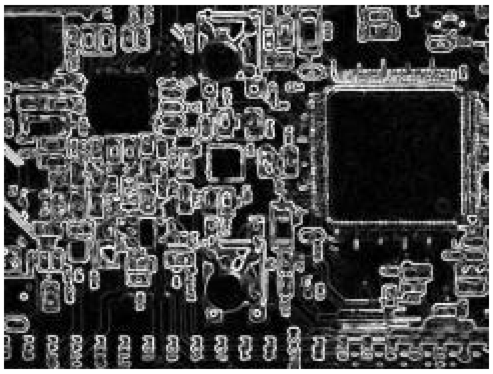

Fig. 11. First Experiment-Sobel Edge Detection of Test Image

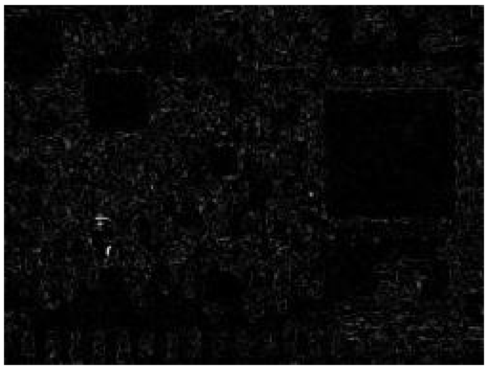

Fig. 12. First Experiment-subtraction Test of Sobel Edge Detection

As we can see, the sobel edge detection has better detection of component's details. But it is poor to detect the defect.

A.3 Adaptive Gaussian Threshold

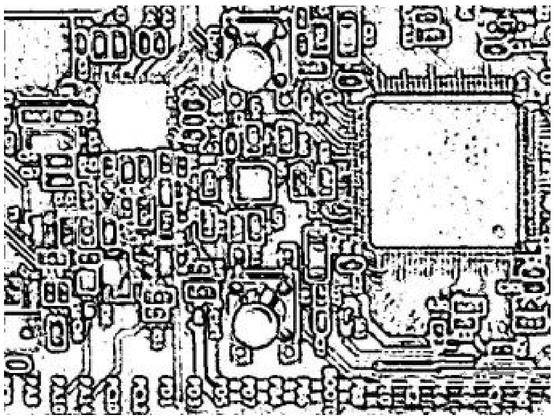

Fig. 13. First Experiment-Adaptive Gaussian Threshold of Refference Image

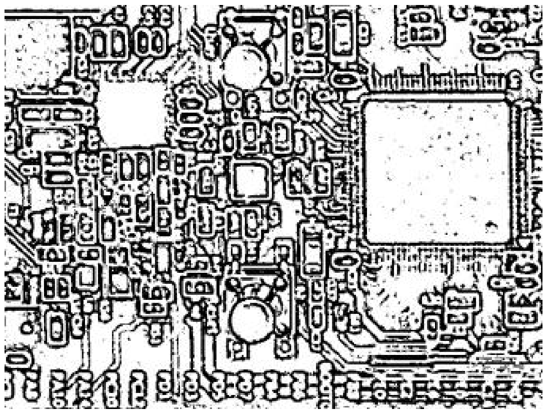

Fig. 14. First Experiment-Adaptive Gaussian Threshold of Test Image

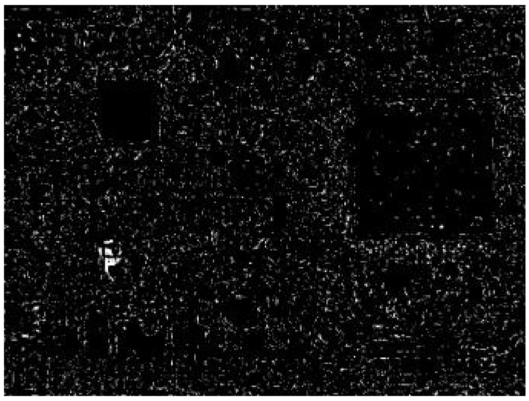

Fig. 15. First Experiment-subtraction Test of Adaptive Gaussian Threshold

As we can see, the Adaptive Gaussian Threshold is very good to detect the details of PCB and the defect too.

B. Second Experiment

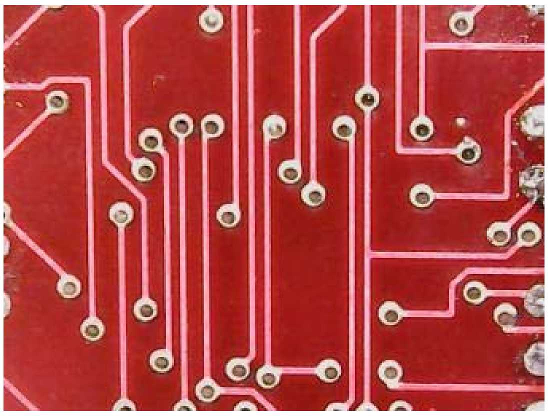

Fig. 16. Second Experiment-Refference Image

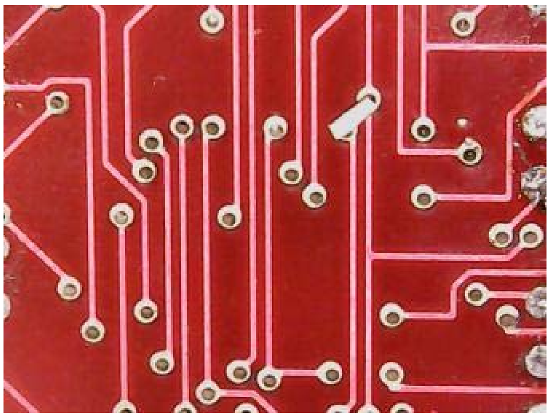

Fig. 17. Second Experiment-Test Image

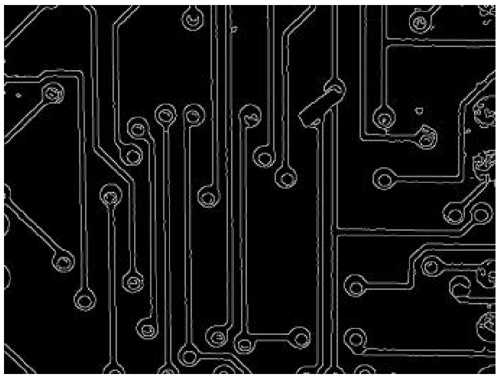

Fig. 19. Second Experiment-Canny Edge Detection of Test Image

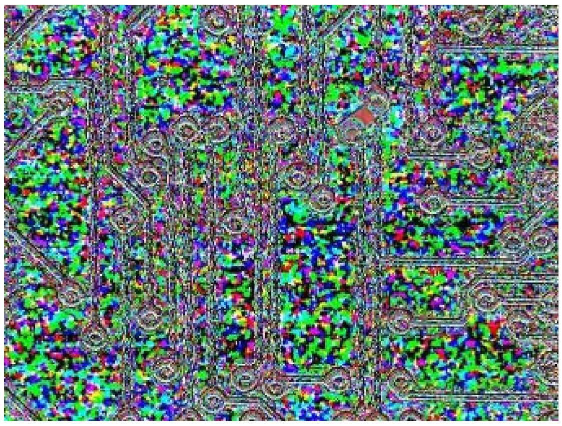

Fig. 19. Second Experiment-subtraction Test

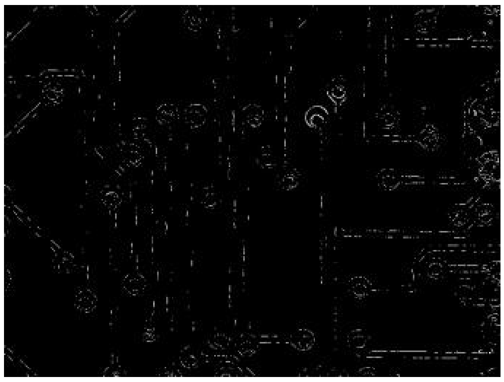

Fig. 21. Second Experiment-subtraction Test of Canny Edge Detection

B. 1 Canny

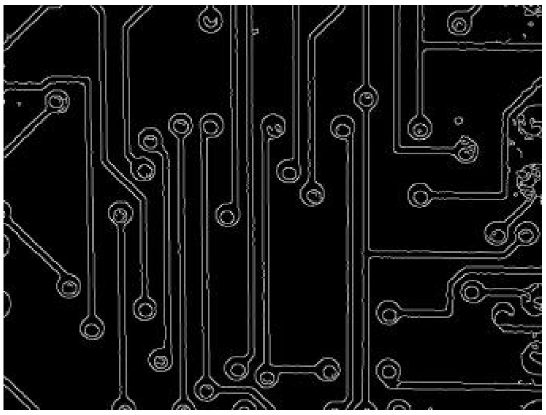

Fig. 20. Second Experiment-Canny Edge Detection of Refference Image

As we can see, the canny edge detection can detect the edge. It can detect the defect but another undefect components are also detected as defect. So it is very poor to detect the defect.

B.2 Sobel

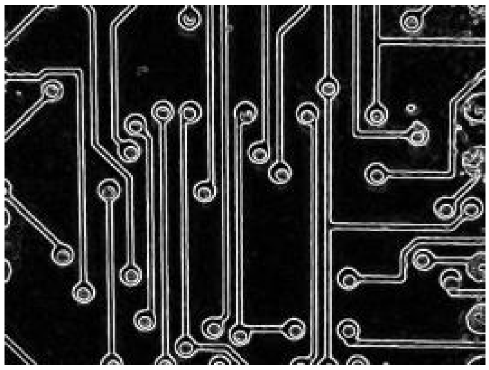

Fig. 22. Second Experiment-Sobel Edge Detection of Refference Image

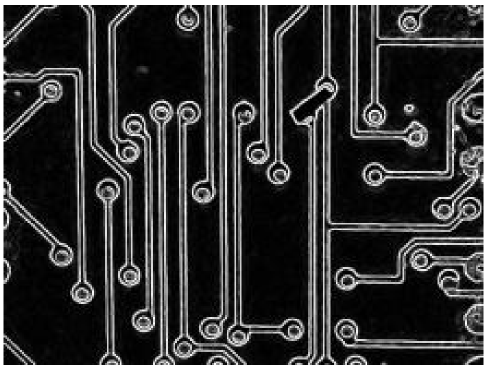
Fig. 23. Second Experiment-Sobel Edge Detection of Test Image

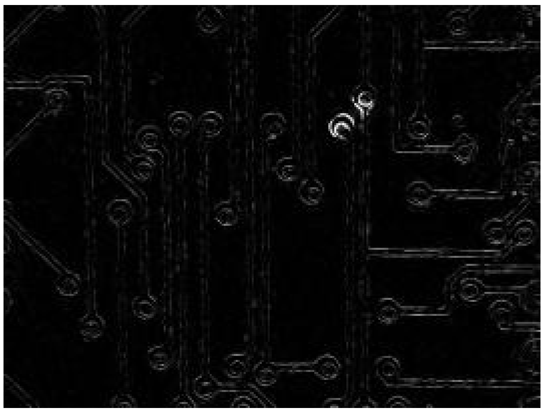
Fig. 24. Second Experiment-subtraction Test of Sobel Edge Detection

As we can see, the sobel edge detection can detect the edge very nice and also the defect.

B.3 Adaptive Gaussian Threshold

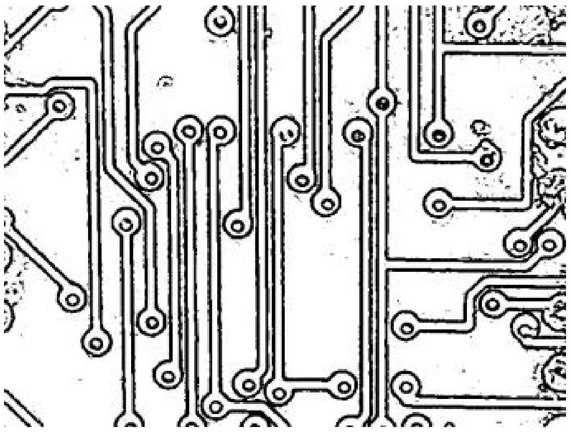
Fig. 25. Second Experiment-Adaptive Gaussian Threshold of Refference Image

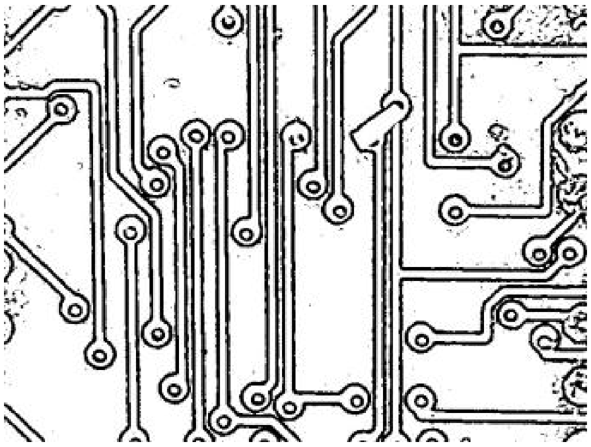
Fig. 26. Second Experiment-Adaptive Gaussian Threshold of Test Image

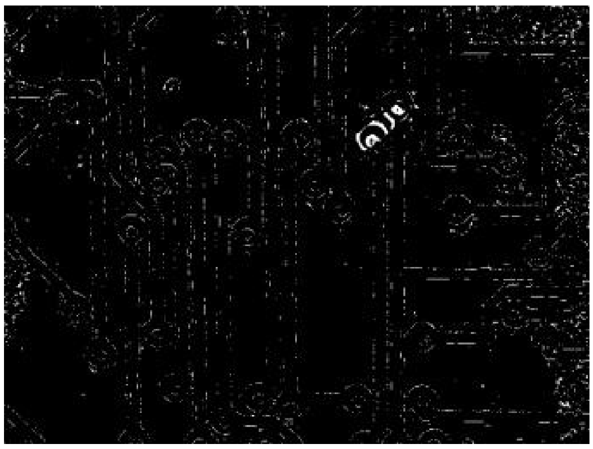
Fig. 27. Second Experiment-subtraction Test of Adaptive Gaussian Threshold

As we can see, the adaptive gaussian threshold can detect the edge very nice and also the defect.

The difficulty level of the first experiment is very high because there are a lot of components in the PCB Board. The canny edge detection fails to detect some complex edges. It is also very poor to detect the missing component. The sobel edge detection is better to detect some complex edges than canny but it also poor to detect the missing component. The Adaptive Gaussian Threshold success to detect complex edges and also the missing component.

The second experiment is medium level. All 3 methods success to detect the edge and also the short circuit. Based on it, the Adaptive Gaussian Threshold is the best method to detect the edges and the defect too. Because it can threshold the unideal picture that usually contain non-uniform illumination.

IV. CONCLUSION

In this work, we show that Adaptive Gaussian Threshold is the most suitable method to detect the PCB defect. It can solve the non-uniform illumination problem of the picture so it can increase the edge and defect detection accuracy.

The proposed method still have some restrictions such as highly sensitive to the miss allignment of test and refference image, manual defect tracking position and recognize the defect type. We hope the next work will improve it gradually.

REFERENCES

[1] Gonzales, Rafael C. and Woods, Richard E., *Digital Image Processing 2nd Edition*, USA, 2002. Prentice Hall.

[2] Moganti, Madhav, et. all, *Automatic PCB Inspection Algorithms: A Survey,* Journal of Computer Vision and Image Understanding. 1996

[3] Sezgin, Mehmet and Sankur, Bulent, *Survey Over Image Thresholding Techniques and Quantitative Performance Evaluation,* Journal of Electroning Imaging. 2014

[4] Davies, E. R. , *Computer and Machine Vision, Theory, Algorithms, Pacticalities 4th Edition*, USA, 2012. Elsevier

[5] OpenCV http://opencv.org

[6] https://sismik.stei.itb.ac.id/2016/05/25/two-dimensional-x-y-cnc-plotter/