International Journal of Innovative Computing, Information and Control
Volume **8**, Number **5(A)**, May **2012**

ICIC International ©2012 ISSN 1349-4198
pp. **3239–3250**

AN IMPROVED DEFECT CLASSIFICATION ALGORITHM FOR SIX PRINTING DEFECTS AND ITS IMPLEMENTATION ON REAL PRINTED CIRCUIT BOARD IMAGES

Ismail Ibrahim[1], Zuwairie Ibrahim[2], Kamal Khalil[1], Musa Mohd Mokji[1]
Syed Abdul Rahman Syed Abu Bakar[1], Norrima Mokhtar[3]
and Wan Khairunizam Wan Ahmad[4]

[1]Faculty of Electrical Engineering
Universiti Teknologi Malaysia
81310 UTM Skudai, Johor Darul Takzim, Malaysia
tsuri_inc@hotmail.com; { kamal; musa; syed }@fke.utm.my

[2]Faculty of Electrical and Electronic Engineering
Universiti Malaysia Pahang
26600 Pekan, Pahang, Malaysia
zuwairie@fke.utm.my

[3]Faculty of Engineering
University of Malaya
50603 Kuala Lumpur, Malaysia
norrimamokhtar@um.edu.my

[4]School of Mechatronic Engineering
Universiti Malaysia Perlis
Main Campus Ulu Pauh, 02600 Perlis, Malaysia
khairunizam@unimap.edu.my

Received November 2010; revised October 2011

Abstract. *Because decisions made by human inspectors often involve subjective judgment, in addition to being intensive and therefore costly, an automated approach for printed circuit board (PCB) inspection is preferred to eliminate subjective discrimination and thus provide fast, quantitative, and dimensional assessments. In this study, defect classification is essential to the identification of defect sources. Therefore, an algorithm for PCB defect classification is presented that consists of well-known conventional operations, including image difference, image subtraction, image addition, counted image comparator, flood-fill, and labeling for the classification of six different defects, namely, missing hole, pinhole, underetch, short-circuit, open-circuit, and mousebite. The defect classification algorithm is improved by incorporating proper image registration and thresholding techniques to solve the alignment and uneven illumination problem. The improved PCB defect classification algorithm has been applied to real PCB images to successfully classify all of the defects.*

Keywords: Printed circuit boards, Defect classification, Defect detection

1. **Introduction.** A bare printed circuit board (PCB) is a PCB that is used before the placement of components and the soldering process [1]. It is used along with other components to produce electronic goods. To reduce manufacturing costs associated with defected bare PCBs, the inspection of bare PCBs is required as the foremost step of the manufacturing process. Nevertheless, many important visual inspection systems are used in manufacturing processes. The processes start from inspection work, measurement, and

some assembly operations. One of these systems is the automatic visual inspection of printed circuit boards (PCB).

Moganti et al. [2] proposed three categories of PCB inspection algorithms: referential approaches, non-referential approaches, and hybrid approaches. Referential inspection is performed by making a comparison between the template PCB image and tested PCB images. Non-referential approaches are based on the verification of the general design rules that essentially verify the widths of conductors and insulators. Lastly, hybrid approaches involve a combination both of referential approaches and non-referential approaches. These approaches have the advantages of the both of referential approaches and non-referential approaches, but at the expense of being more complex.

These PCB inspection algorithms mainly focus on defect detection [2]. However, defect detection does not provide satisfactory information for repair and quality control work because the type of defects detected cannot be clearly identified. Based on this deficiency of defect detection, a defect classification operation is needed in PCB inspection. Therefore, an accurate defect classification procedure is essential especially for an on-line inspection system during the PCB production process [3].

However, only Wu et al. [4], Heriansyah et al. [5], and Rau et al. [6] have proposed defect classification algorithms for PCB inspection. Wu et al. [4] developed a PCB defects classification method based on pixel processing. The method is divided into two stages: defect detection and defect classification. Heriansyah et al. [5] developed an algorithm using the advantages of artificial neural networks to correctly classify defective PCB patterns. Rau et al. [6] also developed a PCB defect classification algorithm based on a hybrid approach. The development of this research can be divided into five stages: reference image rebuilding, inspection image normalisation, image subtraction, defects separation and defect classification.

In this paper, a PCB defect classification algorithm is proposed. The proposed algorithm is developed to detect and classify six different printing defects, namely, missing hole, pinhole, underetch, short-circuit, open-circuit, and mousebite, using a combination of a few image processing operations such as image difference, image subtraction, image addition, counter object comparator, flood-fill and labeling. Even though a similar algorithm has been previously proposed [7,8], the applicability of the defect classification algorithm has been demonstrated solely based on computer-generated images. Hence, the difficulties of solving the alignment and uneven illumination issues have been ignored, and apparently, this is a limitation of these previous papers. In this study, a software-based image registration and uneven illumination are taken into account, and the defect classification algorithm is implemented based on real PCB images. Also, the best thresholding algorithm aided with filtering algorithm has been investigated, and thus all unwanted noise interfered can be eliminated, ensuring that just real defects will be inspected.

2. Defects on Bare Printed Circuit Board.

In PCB fabrication, there are several processes that must be followed: artwork masters, phototools, exposure and development of inner layers, etching of inner layers, laminating and drilling, plating through holes, exposure and development of outer layers, plating tin-lead and etch, and machine and solder mask [2]. The printing processes of artwork masters, phototools, and exposure and development of inner layers, which are performed before the etching process, constitute the source of two groups of defects. The first group of defects includes short-circuit and open-circuit. These defects fall into the fatal defects category. Meanwhile, other defects such as pinhole, underetch, mousebite, and missing hole fall into the potential defects category. It should be noted that fatal defects are those in which the PCB does not

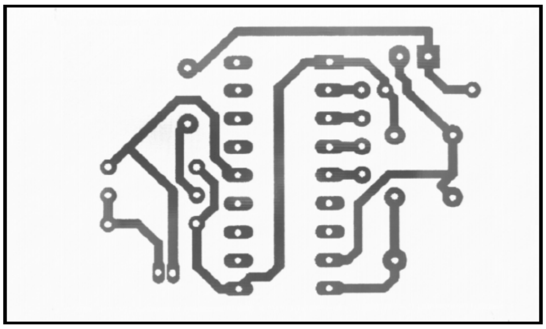

Figure 1. A template bare PCB image

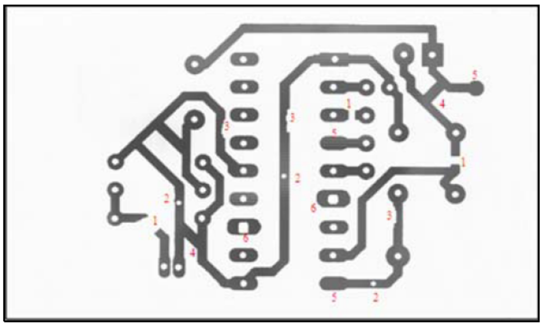

Figure 2. A defective bare PCB image

address the objective for which they were designed, while potential defects are those that compromise the PCB during utilization.

Figures 1 and 2 show real a PCB template image and defective image, respectively. In Figure 2, each defect has been indicated with a specific number, which open-circuit, pinhole, mousebite, short-circuit, missing hole, and underetch are represented by 1, 2, 3, 4, 5 and 6, respectively. Though each defect shown in Figure 2 is a representative example of a certain kind of defect, the shape and size of the defects may vary from one instance to another.

3. Image Processing Operations.

3.1. Image subtraction operation.

The image subtraction operation is primarily used to reveal the differences between images. Subtracting one image from another effectively removes all objects that do not change while preserving those that do change in pixel value. The way the input images are processed is similar to the image difference operation. The difference between two images $f(x, y)$ and $h(x, y)$ is expressed as:

$$g_1(x, y) = f(x, y) - h(x, y) \tag{1}$$

$$g_2(x, y) = h(x, y) - f(x, y) \tag{2}$$

where g_1 and g_2 denote positive and negative images, respectively. These formulations compute the difference between all pairs of corresponding pixels from image f and h.

3.2. Image difference operation.

The image difference operation is performed to obtain a difference image of two images, specifically the template image and the defective image. The method operates by comparing both images pixel by pixel using the XOR logical operator. This operation is similar to the image subtraction operation. The different between these two operations is positive and negative pixel image are combined together in an output image, and it is consider as a defective.

3.3. Image separation operation.

Image separation operation is used to compare the difference and similarity of objects between two input images and then separate the objects into two groups of output images. The first group of image output consists of objects that have difference pixels value and the second group of image output consists of objects that have similar pixels value.

3.4. **Image addition operation.** The image addition operation is a method for combining objects in two images into one image by using the OR logical operator. In other words, this operation is used to create double exposures. If two images $f(x, y)$ and $h(x, y)$ are combined, the expression of this operation can be defined as follows:

$$g_3(x, y) = f(x, y) + h(x, y) \tag{3}$$

where g_3 is the combined image.

3.5. **Flood-fill (filling hole) operator.** A hole can be defined as a background region that is surrounded by a connected border of foreground pixels. In this research, a flood fill operator which is formed from a combination of operations, namely dilation, complementation, and intersection operations have been employed for filling hole in an image [9].

3.6. **Labeling operation.** Labeling is a procedure for assigning a unique label to each object in an image. There are a number of different approaches to labeling connected components. The approaches could be grouped as one pass, two pass, and multiple pass algorithms. In this research, the two pass algorithm developed by Haralick and Shapiro [10] is chosen because this algorithm is theoretically optimal with respect to time complexity, as compared to the one pass algorithm and simpler than the multiple pass algorithm.

3.7. **Counted object comparator operation.** The counted object comparator operation is used to compare the total counted objects between two images. This operation is needed in order to determine if there is any change in the total counted objects when some objects are inserted into an image.

4. **The Improved PCB Inspection System.**

Figure 3 depicts the PCB inspection system developed in this research for detecting and classifying defects on PCB which includes five major stages. The stages are:

Stage 1: Defective image is registered according to the template image. First, the geometric transformation will align the defective image to the template image. This transformation includes rotation, scaling, skewing, and shifting (translating) operations. Second, bi-cubic interpolation is used to get the brightness value for each pixel in the transformed defective image.

Stage 2: An image subtraction process is used to detect all defects occurred. The defects are separated into two images; positive and negative images.

Stage 3: A thresholding algorithm is performed to each image. Minimum thresholding algorithm has been used to positive and negative images. This process is used to remove noise and it will also convert gray image to binary image.

Stage 4: The template of size 3 × 3 of median filter is employed to remove small noise in the both images.

Stage 5: The proposed defect classification algorithms are used to classify all defects occurred in both images. The types of defects that occur in positive image are short-circuit, missing hole, and underetch positive. Meanwhile, the types of defects that occur in negative image are open-circuit, pin hole, mousebite and underetch negative.

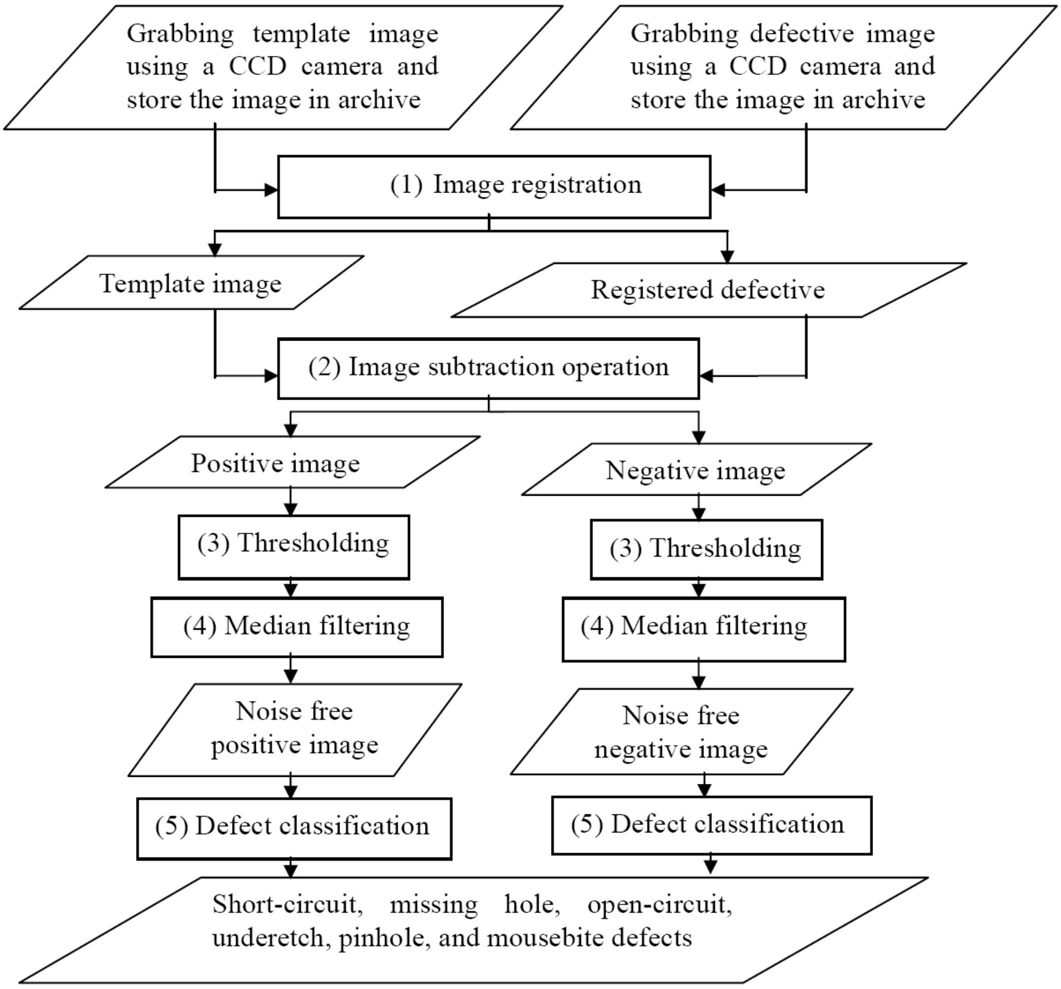
Grabbing template image using a CCD camera and store the image in archive
Grabbing defective image using a CCD camera and store the image in archive
(1) Image registration
Template image
Registered defective
(2) Image subtraction operation
Positive image
Negative image
(3) Thresholding
(3) Thresholding
(4) Median filtering
(4) Median filtering
Noise free positive image
Noise free negative image
(5) Defect classification
(5) Defect classification
Short-circuit, missing hole, open-circuit, underetch, pinhole, and mousebite defects

FIGURE 3. Flow chart of the improved PCB inspection system

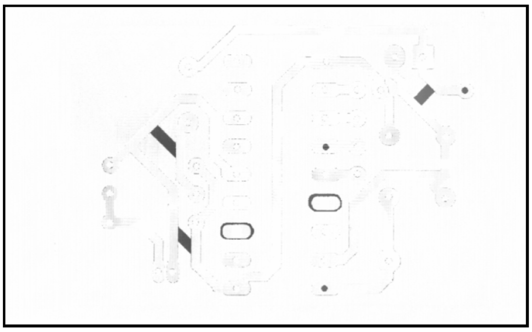

FIGURE 4. Positive image

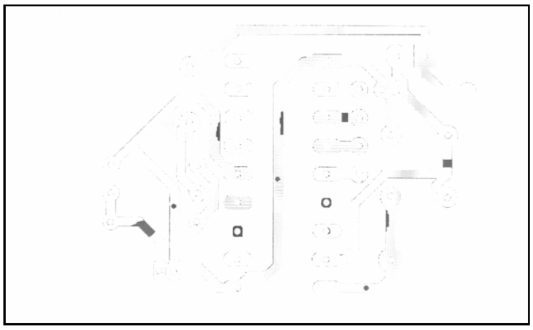

FIGURE 5. Negative image

4.1. **Image registration process.** The image registration process is an important stage in inspecting real PCB images. Image registration or matching can be broadly defined as the process of finding a transformation that aligns one image to another. In this research, geometric transformation is used to find a transformation that aligns a template image and a defective image. The four geometric transformations used for the image registration process are rotation, scaling, skewing, and shifting. Their operation can be checked in reference [10] for details.

4.2. **The subtraction of images.** The image subtraction operation is one of the reference-based inspection methods. This method is a powerful tool because it is simple, quick and effective in finding defects. Thus, this operation is used for defect detection in this study. However, this method suffers from inspection errors when noise occurs. This kind of noise can be introduced by the environment or the information transformation process used. Image subtraction can easily reduce the noise problem if the defective and template image can be aligned closely. By performing this method, two resultant images are produced: a positive and negative image. The positive image consists of open-circuit, pinhole, mousebite, and underetch positive defects. On the other hand, the negative image consists of missing hole, short-circuit and underetch negative defects. The positive image and negative image are shown in Figure 4 and Figure 5, respectively.

4.3. **Thresholding.** In many applications of image processing, the grey levels of pixels belonging to the object are substantially different from the grey levels of the pixels belonging to the background. The principal idea is that the intensity values of object pixels and the background pixels differ such that the object and background can be separated by selecting an appropriate threshold.

The output of the thresholding operation is a binary image whose one state will indicate the foreground objects and whose complementary state will correspond to the background. Depending on the application, the foreground can be represented by grey-level 0, or black, and the background, which is 255 in 8 bit images; conversely, the foreground can be represented by white and the background by black.

To separate the foreground and background in an image, the correct threshold value must be found. Furthermore, it is better to find the correct threshold value automatically by using a special universal algorithm. Regularly, a statistic of an image is used to distinguish the best foreground. Most researchers use a histogram to analyze the statistics of an image. Histogram-shape-based methods are used to analyze the peaks, valleys and curvatures of the smoothed histogram.

In this research, a nonparametric histogram-shape-based thresholding algorithm was used to eliminate noise from both the positive and negative images. First, trial-and-error threshold values are experimentally obtained for both positive and negative images. Then, minimum algorithm is applied to both images, respectively [12]. This is done because these algorithms give values that are closer to the trial-and-error threshold values.

4.4. **Median filtering.** Median filtering is particularly effective in the presence of salt and pepper noise because of its appearance as white and black dots overlaid on an image. Median filtering calculates the median of the neighborhood of the pixel under consideration and assigns this value to the same position in the output image. Noise free pixels should remain unchanged during the filtering process [13]. The median filter can be formulated as in Equation (4) [14]:

$$\hat{f}(x,y) = median\{g(s,t)\}, \quad (s,t) \in S_{xy} \tag{4}$$

where S_{xy} represents the set of coordinates in a rectangular subimage window of size $m \times n$, centered at point (x, y). Meanwhile, $G(x, y)$ is the corrupted image in the area defines by S_{xy}, and the value of the restored image f at any point (x, y) is simply the median computed using the pixels in the region defined by S_{xy}.

4.5. **The classification of defects.** The proposed algorithms consist of two stages: defect detection and defect classification. These two stages are employed to detect and classify the defects, including missing hole, open-circuit, mousebite, short-circuit, pinhole, and underetch defects. To perform the proposed algorithms, two images are needed, a template image and a defective image. In this study, these algorithms used Figure 1 as the template image and Figure 2 as the defective image.

At first, both images are subjected to the image subtraction operation to produce a negative image and a positive image. This operation detects all types of defects that occur in PCB image. In this research, the template image was subtracted from the defective image (defective-template) to obtain the positive image. To obtain the negative image, the defective image was subtracted from the template image (template-defective). Then, the defects were separated, thresholded, and filtered using the appropriate methods. The algorithms were then applied to classify all defects and produce the results.

5. **The Proposed Defect Classification Algorithms.** Initially, a defective image is aligned and registered to a template image. Then, the registered defective image and template image are subjected to image subtraction operation to produce negative image, Y_{neg} and positive image, Y_{pos}. Equations (5) and (6) illustrate these two processes.

$$Y_{pos}(x, y) = Y_{temp}(x, y) - Y_{regdef}(x, y) \tag{5}$$

$$Y_{neg}(x, y) = Y_{regdef}(x, y) - Y_{temp}(x, y) \tag{6}$$

where x, y, $Y_{temp}(x, y)$ and $Y_{regdef}(x, y)$ are respectively, column, row, template image and registered defective image.

The negative and positive images produced are in gray level image. Hence, these two images are thresholded in order to convert them to binary images, and at the same time to reduce noises occurred in both images.

5.1. **Missing hole defect classification algorithm.** First, the image difference operation is applied between the positive image Y_{pos} and the template image that has gone through the flood-fill operation, Y_{ff} can be described using Equation (7). This operation detects a particular defect, namely, the missing hole defect Y_{mh}.

$$Y_{mh}(x, y) = Y_{pos}(x, y) - Y_{ff}(x, y) \tag{7}$$

5.2. **Short-circuit defect classification algorithm.** To detect the short-circuit defect, firstly, the image difference operation is executed between positive image Y_{pos} and missing hole defect Y_{mh} obtained previously. This operation produces a group of defect images that consists of an underetch positive and short-circuit defect image Y_{unp+sc}. Next, the labeling operation is performed onto Y_{unp+sc} to yield an output image $Y_{lbl(unp+sc)}$. By using this operation, the total number of counted labelled objects can be known.

Then, all foreground objects pixels in $Y_{lbl(unp+sc)}$ are set to 1 for each object. This operation produces $Y_{updlbl(unp+sc)}$. The flood-fill operation is then executed onto $Y_{updlbl(unp+sc)}$ to produce $Y_{updlblff(unp+sc)}$. Then, the image separation operation in the form of a rule is applied to the $Y_{updlblff(unp+sc)}$ image. The rule states that each object in $Y_{updlbl(unp+sc)}$ that shares the same attributes with $Y_{updlblff(unp+sc)}$ can be defined as an entity or object in the short-circuit image Y_{sc}. Otherwise, the entities or objects produced belong to the underetch positive defect Y_{unp}. The rule that separates Y_{unp+sc} image into image Y_{unp} and image Y_{sc} is described by Equation (8).

$$Y_{defect}(x, y) = \begin{cases} Y_{sc}, & \text{if pixels value in } Y_{updlblff(unp+sc)}(x, y) \\ & \text{are same with } Y_{updlbl(unp+sc)}(x, y) \\ Y_{unp}, & \text{otherwise} \end{cases} \tag{8}$$

5.3. **Pinhole defect classification algorithm.** Initially, a negative image Y_{neg} is produced by executing image subtraction operation on a template image Y_{temp} and a registered defective image Y_{regdef}. Then, the defective image Y_{regdef} undergoes flood-fill operation. This process produces an output image Y_{ff}. Next, the image difference operation is applied to Y_{ff} and Y_{neg}. This operation produces an image containing pinhole and underetch negative defects Y_{ph+unn}. Then, labeling operation is performed to yield $Y_{lbl(ph+unn)}$. Then, foreground objects pixels in $Y_{lbl(ph+unn)}$ are set to 1. As a result, $Y_{updlbl(ph+unn)}$ is formed. Next, the flood-fill operator is used to fill all holes in $Y_{updlbl(ph+unn)}$, which yields $Y_{updlblff(ph+unn)}$. Then, the image separation operation in the form of a rule is applied to the $Y_{updlblff(ph+unn)}$ image.

The rule states that each object in $Y_{updlbl(ph+unn)}$ that has same pixels value as $Y_{updlblff(ph+unn)}$ can be defined as an entity or object in the pinhole image Y_{ph}. Otherwise, the entities or objects produced belong to the underetch negative image Y_{unn}. The rule that separates the Y_{ph+unn} image into image Y_{unn} and image Y_{ph} is described by Equation (9).

$$Y_{defect}(x,y) = \begin{cases} Y_{ph}, & \text{if pixels value in } Y_{updlblff(ph+unn)}(x,y) \\ & \text{are same with } Y_{updlbl(ph+unn)}(x,y) \\ Y_{unn}, & \text{otherwise} \end{cases} \tag{9}$$

5.4. **Underetch defect classification algorithm.** Underetch defect can be obtained by using image addition operation to underetch positive image and underetch negative image obtained previously. The image addition operation is denoted by Equation (10).

$$Y_{und}(x,y) = Y_{unp}(x,y) + Y_{unn}(x,y) \tag{10}$$

5.5. **Mousebite and open-circuit defect classification algorithm.** To acquire an image of open-circuit and mousebite defects, the image difference operation is first performed between the negative image Y_{neg} and the image of pinhole and underetch negative defects Y_{ph+unn}. This operation produces an image of open-circuit and mousebite defects Y_{oc+mb}. Then, the labeling operation is applied to Y_{oc+mb}, which produces $Y_{lbl(oc+mb)}$. Foreground object pixels in $Y_{lbl(oc+mb)}$ are set to 1. As a result, $Y_{updlbl(oc+mb)}$ is formed.

The image addition operation is then executed between Y_{ff} and $Y_{updlbl(oc+mb)}$. As a consequence, Y_{add} is created. Then, the labeling operation is also applied to $Y_{updlbl(oc+mb)}$ to produce Y_{addlbl}. Next, the total number of counted objects and images Y_{addlbl} and Y_{ff} are compared. Mousebite defect image Y_{mb} is produced if the total number of counted objects in Y_{addlbl} image is same as in image Y_{ff}. Otherwise, open-circuit image Y_{oc} is produced. The condition to get mousebite and open-circuit defect is presented in Equation (11).

$$Y_{defect}(x,y) = \begin{cases} Y_{mb}, & \text{if total objects in } Y_{updlblff(ph+unn)}(x,y) \\ & \text{is same as in } Y_{updlbl(ph+unn)}(x,y) \\ Y_{oc}, & \text{otherwise} \end{cases} \tag{11}$$

6. **Implementation.** In this study, real PCB images are captured using a 1620 × 1236 pixels monochrome charged coupled device camera. A PC2-Vision frame grabber has been used to digitize and store the images into computer. Two bars of LED are used in the illumination part. Detection and classification algorithms are developed and tested in MATLAB 7.7.0 environment, on Windows Vista platform, with Pentium Intel® Core $^{\text{TM}}$ 2 personal computer, 1.86 GHz and 2 GB RAM (Random Access Memory).

7. **Experimental Results and Discussion.** The result of the previous PCB inspection system [7,8] is compared to the improved PCB inspection system. Both systems have been executed using real PCB images. Unlike the improved PCB inspection system, the previous PCB inspection system has had not incorporated with image registration

operation because it was built using computer-generated PCB images. The results of the previous PCB inspection systems are shown in Figures 6-11.

The result of the previous PCB inspection system has indicated that without a proper image registration operation, the PCB inspection system is likely to totally fail. Hence, the improved PCB inspection system incorporating with a reliable image registration operation has been developed to classify the six printing defects which their shapes are identical with those in practical. The result for this system is shown in Figures 12-17.

The proposed defect classification algorithms use the image subtraction operation to locate the defects, followed by a process to classify the defects. Two gray-scale images are needed for this operation, a template image and a defective image as shown in Figures 1 and 2. These two images must be aligned. However, the subtracted image may still feature interference by unwanted noise due to slight misalignment and uneven binarization. To effectively eliminate the noise, the subtracted image is first divided into positive and negative image, as shown by Figures 18 and 19. This operation is important to detect all defects that occur in the defective image. It should be noted that these two images are in gray-scale.

Second, the threshold operation is applied to both images to remove noise. Actual defects still remain in the positive and negative images. The threshold operation is also used to convert both images to binary images. For the positive image and negative image, minimum threshold algorithm developed by Prewitt and Mendelsohn [12] has been applied. It assumes that the histogram to be a bimodal shape. In a bimodal histogram, the distribution of the pixels has two different modes and two distinct peaks. Threshold value, T is chosen such that $h_{T-1} < h_T < h_{T+1}$, where h_T is the number of pixels in the image with the gray-level T. For implementation, the threshold values T for the positive and negative images yielded from the minimum thresholding algorithm are 150 and 127,

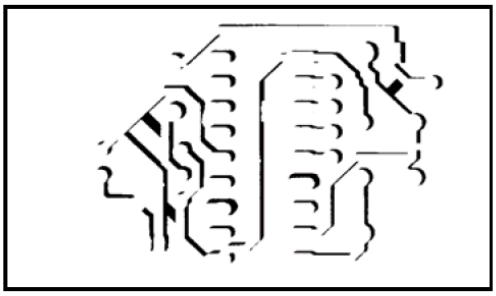

FIGURE 6. Image of short-circuit defect

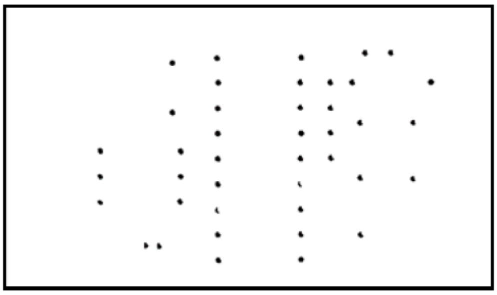

FIGURE 7. Image of missing hole defect

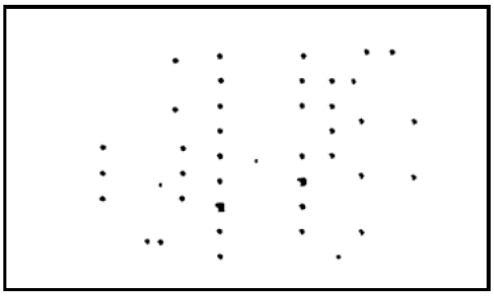

FIGURE 8. Image of pinhole defect

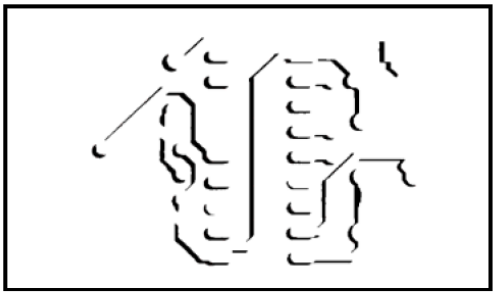

FIGURE 9. Image of mousebite defect

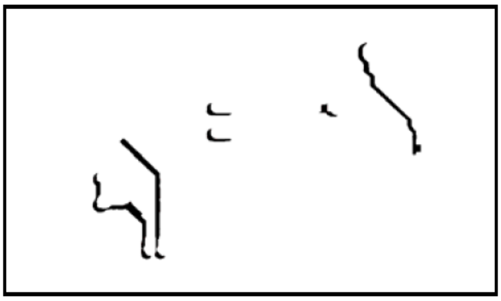

Figure 10. Image of open-circuit defect

Figure 11. Image of underetch defect

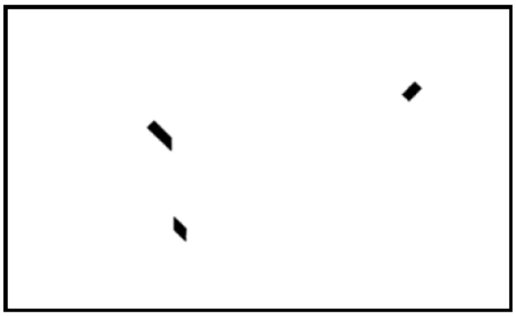

Figure 12. Image of short-circuit defect

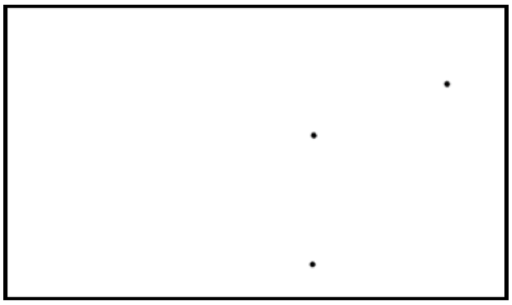

Figure 13. Image of missing hole defect

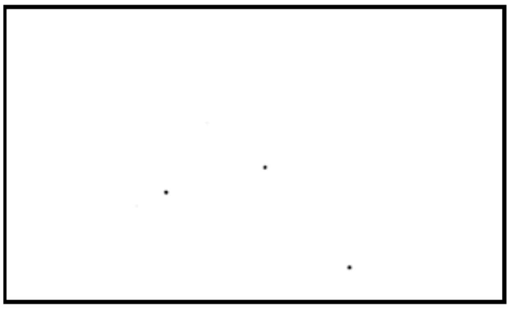

Figure 14. Image of pinhole defect

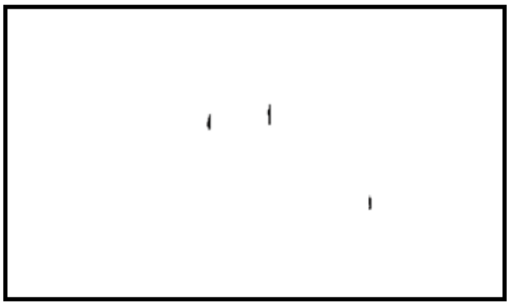

Figure 15. Image of mousebite defect

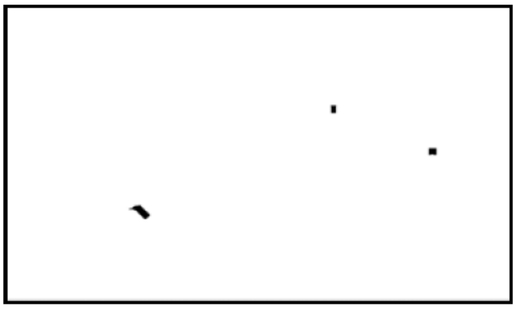

Figure 16. Image of open-circuit defect

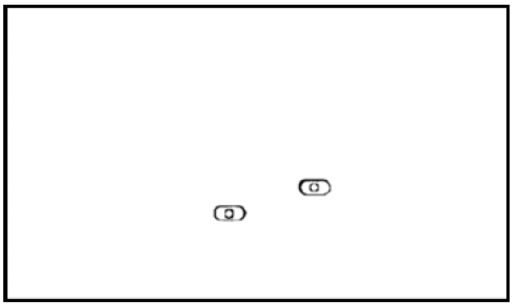

Figure 17. Image of underetch defect

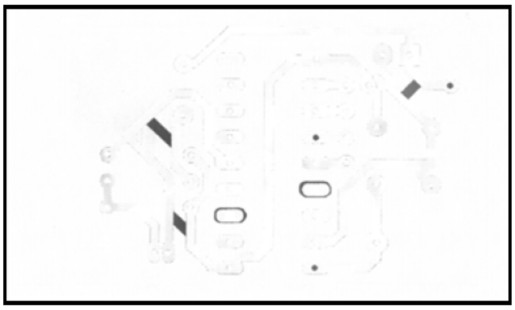

FIGURE 18. Positive image

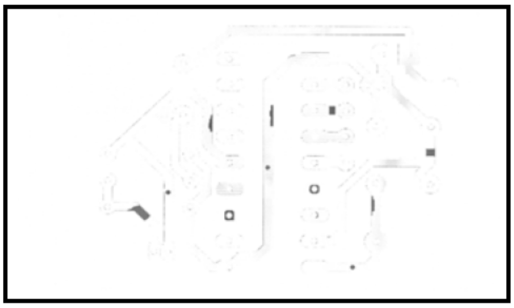

FIGURE 19. Negative image

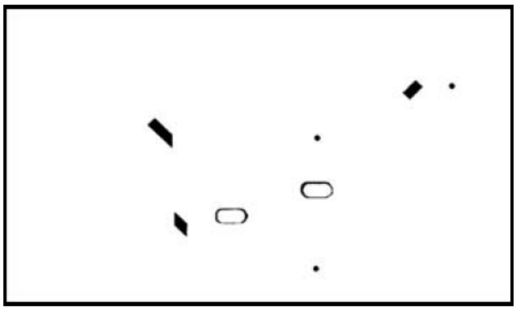

FIGURE 20. Noise-free positive image

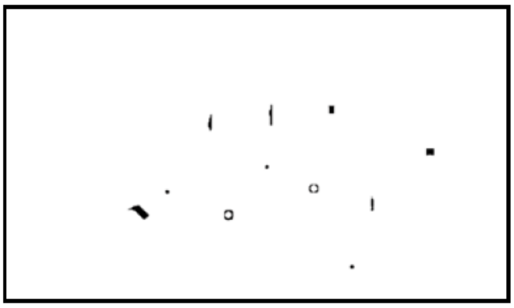

FIGURE 21. Noise-free negative image

respectively. As a consequence, the well-thresholded positive image and negative image are produced.

Third, a median filtering is applied to both thresholded images. In fact, median filtering is a nonlinear spatial filtering whose responses is based on ranking of the pixels contained in the image area covered by the filter, and then replacing the value of a pixel by the median of the gray level in the neighborhood of that pixel. This operation is useful in removing any small noise that may still be present in an image. This operation produces noise-free positive and negative images as depicted in Figures 20 and 21, respectively. Defect classification then can be performed.

To prevent difficulties in detecting and classifying actual defects, the medium of the image subtraction operation must be in grey-scale. If the template and defective images are first converted to binary images, the noise may likely appear as actual defects. This scenario will provide false defects and seriously affect the performance of the proposed defect classification algorithms. As the improved PCB inspection is just capable to classify those six types of defects, in future, additional defects will be considered to be classified whether before or after etching.

Illumination is very important in capturing images. Poorly designed illumination fixtures can result in unintelligible images; indeed, under- and over-illuminated regions are poorly resolved. To obtain a picture with the desired appearance, one must search through the space of possible lighting specifications. Non-uniform illumination caused by irregular illumination is the main problem in PCB inspection. The vision system employed requires uniform illumination to recognize an object. Thus, this study adopted a system that consists of lighting devices and several camera lenses to capture good-quality images. The results obtained from processing were displayed on a monitor to the users.

Other parameters of analysis are the processing time for the image registration, defect detection, defect classification, and displaying defects processes. By using the PCB images as the experiment subjects, the average processing time for these four processes were 10.54, 0.04, 17.95 and 0.25 seconds respectively. The resolution of the PCB images is 1580×917 pixels. The image registration process consumed more processing time because the real PCB images used are bigger, or more pixels values were computed. In future, image registration process could be optimized, and FPGA software could be used instead of MATLAB, so the processing time can be speed up.

8. **Conclusions.** This study explores an automated visual inspection algorithm for the classification of defects on bare PCBs. The defect classification algorithms are capable of detecting six types of defects: missing hole, wrong size, open-circuit, short-circuit, pinhole, underetch and mousebite defects. Samples of real PCB images have been tested using the proposed algorithms. The proposed method proved to be an alternative way to efficiently detect and classify defects.

Acknowledgment. This work is supported by the Universiti Teknologi Malaysia GUP Research Fund (Vote Q.J130000.7123.01J35) and partly supported by the Fundamental Research Grant Scheme (Vote 4F021) and the Prototype Research Grant Scheme (PRGS 9013-00004), which have been awarded by the Ministry of Higher Education (MOHE) to Universiti Teknologi Malaysia and Universiti Malaysia Perlis, respectively.

REFERENCES

[1] J. Hong, K. Park and K. Kim, Parallel processing machine vision system for bare PCB inspection, *Proc. of the 24th Annual Conference of the IEEE*, pp.1346-1350, 1998.

[2] M. Moganti, F. Ercal, C. H. Dagli and S. Tsunekawa, Automatic PCB inspection algorithms: A survey, *Computer Vision and Image Understanding*, vol.63, no.2, 1996.

[3] R. S. Guh and J. D. T. Tannock, A neutral network approach to characterize pattern parameters in process control charts, *Journal of Intelligent Manufacturing*, vol.10, no.5, pp.449-462, 1996.

[4] W. Y. Wu, M. J. Wang and C. M. Liu, Automated inspection of printed circuit boards through machine vision, *Computers in Industry*, vol.28, no.2, pp.103-111, 1996.

[5] R. Heriansyah and S. A. R. Abu-Bakar, Defects classification on bare PCB using multiple learning vector quantization neural network paradigm, *Proc. of International Conference on Computer Graphics, Imaging, and Visualization*, Penang, Malaysia, 2004.

[6] H. Rau and C.-H. Wu, Automatic optical inspection for detecting defects on printed circuit board inner layers, *International Journal of Advanced Manufacturing Technology*, vol.25, no.9-10, pp.940-946, 2005.

[7] I. Ibrahim, Z. Ibrahim, M. S. Z. Abidin, M. M. Mokji, S. A. R. S. Abu-Bakar and S. Sudin, An algorithm for classification of six types of defect on bare printed circuit board, *Proc. of the 33rd International Electronics Manufacturing Technology Conference*, Penang, Malaysia, 2008.

[8] I. Ibrahim, Z. Ibrahim, M. S. Z. Abidin, M. M. Mokji and S. A. R. S. Abu-Bakar, Classification algorithm for six different defects on bare printed circuit board, *Proc. of the 6th Student Conference on Research and Development*, Johor Bahru, Malaysia, 2008.

[9] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd Edition, Prentice-Hall, 2002.

[10] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*, Addison-Wesley, 1992.

[11] M. Sonka, V. Hlavac and R. Boyle, *Image Processing, Analysis, and Machine Vision*, 2nd Edition, Brooks/Cole, CA, 1998.

[12] J. M. S. Prewitt and M. L. Mendelsohn, The analysis of cell images, *Ann. New York Acad. Sci.*, vol.128, no.3, pp.1035-1053, 1966.

[13] A. Hussain, M. A. Jaffar and A. M. Mirza, Random-valued impulse noise removal using fuzzy logic, *International Journal of Innovative Computing, Information and Control*, vol.6, no.10, pp.4273-4288, 2010.

[14] R. C. Gonzalez and P. Wintz, *Digital Image Processing*, Addison-Wesley Publishing Company, 1987.