ISSN 22773061

Noise Elimination for Image Subtraction in Printed Circuit Board Defect Detection Algorithm

Zuwairie Ibrahim[1], Ismail Ibrahim[1], Zulfakar Aspar[2], Kamal Khalil[2], Sophan Wahyudi Nawawi[2], Muhammad Arif Abdul Rahim[2], Wan Khairunizam Wan Ahmad[3]
1Faculty of Electrical and Electronics Engineering
Universiti Malaysia Pahang, Pahang Darul Makmur, Malaysia
zuwairie@ump.edu.my
2Faculty of Electrical Engineering
Universiti Teknologi Malaysia, Johor Darul Takzim, Malaysia
3Center for Postgraduate Studies
Universiti Malaysia Perlis, Perlis, Malaysia

ABSTRACT

Image subtraction operation has been frequently used for automated visual inspection of printed circuit board (PCB) defects. Even though the image subtraction operation able to detect all defects occurred on PCB, some unwanted noise could be detected as well. Hence, before the image subtraction operation can be applied to real images of PCB, image registration operation should be done to align a defective PCB image against a template PCB image. This study shows how the image registration operation is incorporated with a thresholding algorithm to eliminate unwanted noise. The results show that all defects occurred on real images of PCB can be correctly detected without interfere by any unwanted noise.

Indexing terms/Keywords

Printed Circuit Boards, Image Registration, Affine Transformation, Bi-cubic Interpolation, Defect Detection

Academic Discipline And Sub-Disciplines

Image Processing

SUBJECT CLASSIFICATION

Image Processing

Council for Innovative Research
Peer Review Research Publishing System
Journal: INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY
Vol 10, No 2
editor@cirworld.com
www.cirworld.com, member.cirworld.com

INTRODUCTION

Nowadays, there are numerous algorithms, techniques, and approaches available in the area of automated visual inspection of defects. Those algorithms can be divided into three main approaches: referential approaches, rule-based approaches, and hybrid approaches [1]. First, concerning the referential approaches, the image comparison technique is used to compare the defective printed circuit board (PCB) image against the template PCB image using simple XOR logic operator [2]. Second, rule-based approaches are techniques that match the pattern under inspection with a set of predefined models [3]. Last, the hybrid approaches combine the referential approaches and rule-based approaches to cover the shortcomings of each approach with the advantages of other approach. From these three approaches, PCB inspection based on referential approaches is applied because it can be easily employed in hardware, and subsequently high processing rates are possible to be obtained [1]. These approaches seem to be able to detect large defects such as missing conductor wires [4].

However, since the inspection process involves real images of PCB, misalignment issue is important to be solved. Thus, a software-based approach is proposed in this paper to align the template image and defective image [5]. These two images are then subjected to image subtraction operation to produce a positive image and a negative image. These two images, however, consist of all defects and some unwanted noise, which occurred from the presence of non-uniform illumination. Hence, the objective of this study is to introduce a noise elimination method to eliminate the unwanted noise, which is the residue of image subtraction operation. A few thresholding algorithms are investigated and the best algorithm is applied to the positive image and negative image. Finally, a suitable filter is implemented to further eliminate any remaining unwanted noise.

ARITHMETIC OPERATIONS

Usually, arithmetic or logic operations of images are performed on pixel-by-pixel basis between two or more images. As an example, subtraction of two images results in a new image whose pixel at particular coordinates is the difference between the pixels value in that same location in the two images being subtracted. Depending on the hardware and/or software being used, the actual mechanics of implementing arithmetic/logic operations can be done sequentially, one pixel at a time, or in parallel, where all operations are performed simultaneously.

Logic operations similarly operate on a pixel-by-pixel basis. The ability of implementation of AND, OR, and NOT logic operators need to be concerned because these three operators are functionally complete. In other words, any other logic operator can be implemented by using only these three basic functions. Logic operations also are used frequently in conjunction with morphological operations [6].

Logic operations comprise of image subtraction and image addition are employed in this study.

Image Subtraction Operation

The image subtraction operation is primarily used to reveal the differences between two images. In such a way, it removes all objects that do not change while preserves those that do change in pixel value, as executed by Wu et al. [2]. 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:

$$g_1(x,y) = f(x,y) - h(x,y) \quad (1)$$

$$g_2(x,y) = h(x,y) - f(x,y) \quad (2)$$

where g_1 and g_2 denote positive image and negative image, respectively. These two equations compute the difference between all pairs of corresponding pixels from image f and h. The concept of simple subtraction and the corresponding rules are shown in Table 1.

Figure 1 shows the image subtraction operation. The parts of the picture those are essentially unchanged in the two images cancel out and appear as a uniform medium gray except for minor variations due to the precision of digitization and changes in illumination. The defects that existed in the defective image are clearly shown. The black or dark objects show where the defects are. The black objects are yielded from the difference between the template image and defective image.

Table 1. Four rules for the image subtraction operation

Rule	Result
If 1 - 0	Positive pixel image
If 0 - 1	Negative pixel image
If 0 - 0	No change
If 1 - 1	No change

Image Addition Operation

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. Equation 3 illustrates this operation while Figure 2 illustrates the result of the image addition operation.

$$g(x, y) = f(x, y) + h(x, y) \quad (3)$$

Image Registration

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, while image resampling corresponds the transformed grid points of the output image with the grid points of the input image. The four geometric transformations used for the image registration process are rotation, scaling, skewing, and shifting. This image registration has been previously proposed in [7].

Thresholding Algorithms

In gray-scale image, the difference is obvious between the grey levels of pixels belonging to the objects and background. To separate the objects from their background, a thresholding algorithm can be used. The basic idea of thresholding is to select an optimal gray-level threshold value for separating objects in an image from their background based on their gray-level distribution.

The output of the thresholding algorithm 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.

In fact, automatic thresholding has been widely used in machine vision industry for automated visual inspection of defects [8]. Thresholding techniques can be categorized into global thresholding and local thresholding. Global thresholding selects a single threshold value from the histogram of the entire image. Local thresholding uses localized gray-level information to choose multiple threshold values; each is optimized for a small region in the image. Global thresholding is simpler and easier to implement but its result relies on good (uniform) illumination. Local thresholding methods can deal with non-uniform illumination but they are complicated and slow. For automated visual inspection applications, where non-uniform illumination is usually not an issue due to controlled lighting conditions, global thresholding is commonly used for its simplicity and speed.

Regularly, a statistic of an image is used to distinguish the best foreground or object. 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 paper, ten histogram-shape-based thresholding algorithms have been applied one by one to eliminate noise occurred in the positive image and negative image. These methods are minimum, intermodes, concavity, median, mean, moments, entropy, intermeans, iterative intermeans, and minerror algorithms [9].

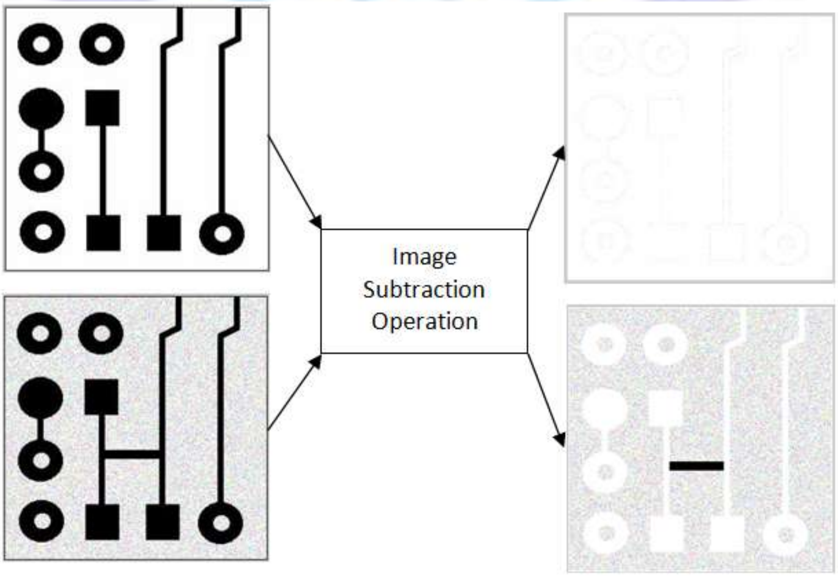


Fig 1: Image subtraction operation as the defect detection operation

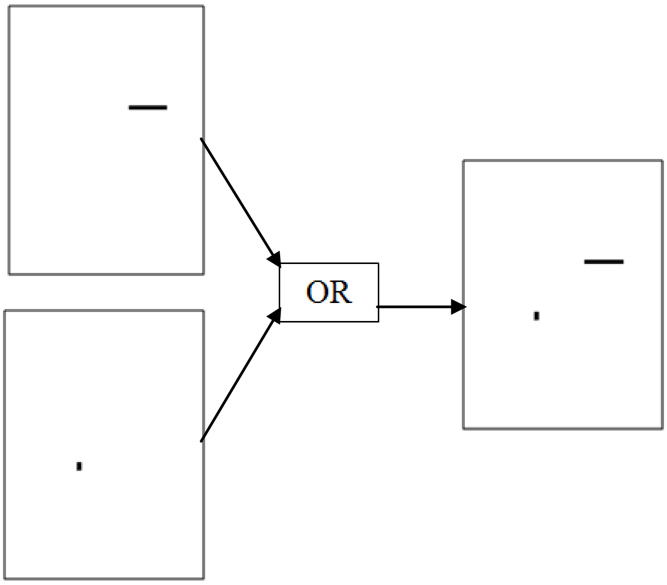


Fig 2: Image addition operation

Notation

A histogram, h_g, is denoted by h_0, h_1, … , h_G, where h_i is the number of pixels in the image with the gray-level i, and G is the maximum gray-level. For an 8-bit image, G is equal to 255. Probability mass function (pmf) of an image is denoted by $p(g)$, g = 0,...,G, which can be approximated by normalizing the image histogram, h_g. The cumulative probability function is defined as:

$$P(g) = \sum_{i=0}^{g} p(i) \quad (4)$$

In this paper, the foreground (object) is the set of pixels with luminance values less than the threshold value, T, while the background pixels have luminance value above this threshold. The foreground (object) and background pmf's will be expressed as $P_f(g),\ 0 \le g \le T$, and $P_b(g),\ T + 1 \le g \le G$, respectively, where T is the threshold value. The foreground and background cumulative functions are calculated as:

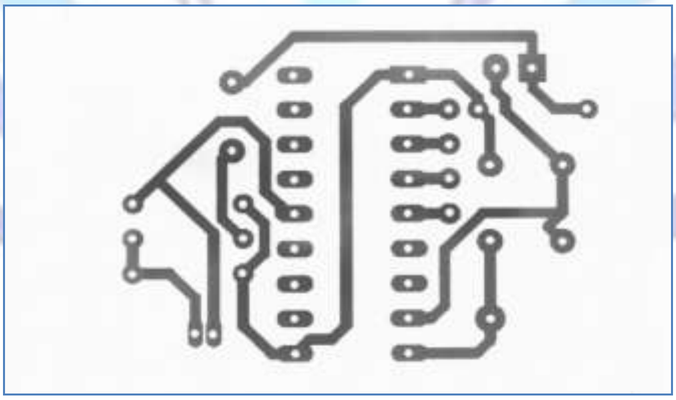


Fig 3: Example of image to be threshold

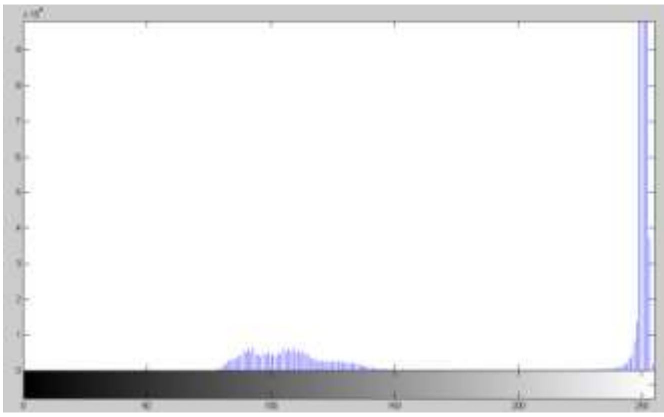


Fig 4: Brightness histogram of Fig 3

$$P_f(T) = P_f = \sum_{g=0}^{T} p(g) \quad (5)$$

$$P_b(T) = P_b = \sum_{g=T+1}^{G} p(g) \quad (6)$$

Figure 3 shows an example of the bare image of PCB used in this study. Its histogram is shown in Figure 4. By using different thresholding algorithms, the threshold value, T, for each algorithm will be calculated, respectively.

Minimum Algorithm

This algorithm was developed by Prewitt and Mendelsohn [10]. 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 $h_{T-1} < h_T \le h_{T+1}$. This algorithm is unsuitable for images that have a histogram with extremely unequal peaks or a broad and flat valley.

Intermodes Algorithm

This algorithm is an alternative of the Minimum algorithm [10]. It also makes the assumption of bimodality. T is chosen such that it is the average of the two gray levels corresponding to the two histogram maximum h(max$_1$) and h(max$_2$).

Concavity Algorithm

A well-known heuristic for segmenting an image into gray level subpopulations is to select thresholds at the bottom valleys on the image's histogram [11]. Both valleys and peaks correspond to concavities on the histogram. Hence, it should be possible to find good thresholds by analysing the histograms concavity structure.

In general, a typical gray level histogram consists of a number of significant peaks and valleys. The significant peaks and valleys are also called local maxima and local minima respectively. Significant peaks can be identified by constructing a convex hull of the histogram, which is defined as the smallest convex polygon, H, that contains the given histogram, h. The convex hull consists of straight-line segments joining the significant peaks in histogram. The histogram concavity at any gray level is defined as the vertical distance between the convex hull and the histogram that is [H-h]. When the convex hull of the histogram is calculated, the deepest concavity points become candidates for a threshold. In case of competing concavities, some object attribute feedback, such as low busyness of the threshold image is used.

$$T_{opt} = arg\,max\{[p(g) - H(g)]\} \quad (7)$$

Median Algorithm

This algorithm was developed by Doyle [12]. It determines a threshold such that a given percentage of pixels remain after thresholding. In order to use Median algorithm, the percentage of object pixels should be known in advance. In other word, this algorithm is an algorithm that assumes that the percentage of black pixels, p, is known, and a fraction of foreground pixels to be 0.5. T is then set to the highest gray-level which maps at least (100 − p) percents of the pixels into the object category. In this algorithm, p is set to be 50, or T is chosen so that A$_T$/A$_G$ is as close as possible to 0.5.

$$A_T = \sum_{g=0}^{T} h(g) \quad (8)$$

$$A_G = \sum_{g=0}^{G} h(g) \quad (9)$$

where A_T and A_G are the cumulative number of pixels from gray level 0 to and the cumulative number of pixels from gray level 0 to 255, respectively. In other words, a global threshold for a gray-scale image is found by assuming that half of the pixels belong to the background and half to the foreground. This algorithm is unsuitable to use if the object area is unknown.

Mean Algorithm

This algorithm was introduced by Glasbey [13]. This algorithm is operated by using the mean of gray levels as the threshold. It is also used by some other methods as a first guess threshold. T is set such that it is integer part of the mean of all pixel values. The partial sum notation for T is, T = B_G/A_G .

$$A_G = \sum_{g=0}^{G} h(g) \quad (10)$$

$$B_G = \sum_{g=0}^{G} gh(g) \quad (11)$$

where A_G and B_G are the cumulative number of pixels from gray level 0 to 255, and the cumulative number of pixels from gray level 0 to 255 multiply with each gray level.

Moments Algorithm

This algorithm was proposed by Tsai [14] to preserve the moments of the original image in the thresholded result. T is chosen so that the binary image and its original have the same three first moments as the gray-level image. The gray-level moments, m_k, and binary image moments, b_k, are defined as:

$$m_k = \sum_{g=0}^{G} p(g) g^k \quad (12)$$

$$b_k = p_f m_f^k + p_b m_b^k \quad (13)$$

The threshold is given by:

$$T_{opt} = arg\,equal[m_1 = b_1(T), m_2 = b_2(T), m_3 = b_3(T)] \quad (14)$$

Entropy Algorithm

Entropy algorith is an example of maximum entropy algorithm which was established by Kapur et al. [15]. The algorithm works by dividing the histogram of the image into two probability distributions, one representing the objects and one representing the background. T is chosen such that the sum of the entropies of these probability distributions is maximized.

$$H_f(T) = -\sum_{g=0}^{T} \frac{p(g)}{P(T)} \log \frac{p(g)}{P(T)} \quad (15)$$

$$H_b(T) = -\sum_{g=T+1}^{G} \frac{p(g)}{P(T)} \log \frac{p(g)}{P(T)} \quad (16)$$

$$T_{opt} = argmax\left[H_f(T) + H_b(T)\right] \quad (17)$$

Intermeans Algorithm

The Intermeans algorithm, which was developed by Otsu [16], calculates a global threshold by accepting the existence of two classes, namely foreground and background. The mean and variance of the foreground and background as functions of the thresholding level, T, can be denoted as:

$$m_f(T) = \sum_{g=0}^{T} g\, p(g) \quad (18)$$

$$\sigma_f^2(T) = \sum_{g=0}^{T} \left(g - m_f(T)\right)^2 p(g) \quad (19)$$

$$m_b(T) = \sum_{g=T+1}^{G} g\, p(g) \quad (20)$$

$$\sigma_b^2(T) = \sum_{g=T+1}^{G} \left(g - m_b(T)\right)^2 p(g) \quad (21)$$

Threshold is chosen by minimizing the interclass variance of the thresholded black and white pixels. The choice of the optimum threshold can be formulated as:

$$T_{opt} = \arg max \left\{ \frac{P(T)[1-P(T)][m_f(T)-m_b(T)]^2}{P(T)\sigma_f^2(T)+[1-P(T)\sigma_b^2(T)} \right\} \quad (22)$$

The Otsu method gives satisfactory results when the numbers of pixels in each class are close to each other.

Iterative Intermeans Algorithm

This iterative algorithm which was introduced by Ridler and Calvard [17] gives similar results as the intermeans algorithm. This algorithm is one of the first iterative schemes based on two-class Gaussian mixture models. The advantage of this iterative algorithm is that less intensive computation is required than intermeans algorithm. At iteration n, a new threshold T_n is established by using the average of the foreground and background class means. In practice, however, iteration is terminated when the change $|T_n - T_{n-1}|$ becomes sufficiently small. The mean of the foreground and background as functions of the T_n can be denoted as:

$$m_f(T_n) = \sum_{g=0}^{T_n} g p(g) \quad (23)$$

$$m_b(T_n) = \sum_{g=T_n+1}^{G} g p(g) \quad (24)$$

The choice of the optimum threshold can be formulated as:

$$T_{opt} = \lim_{n\to\infty} \frac{m_f(T_n)+m_b(T_n)}{2} \quad (25)$$

Minerror Algorithm

This algorithm is introduced by Kittler and Illingworth [18]. Based on this algorithm, the foreground and background class conditional probability density functions are assumed to be Gaussian. The error expression can be interpreted also as a fitting error expression to be minimized such that:

$$T_{opt} = argmin[P(T)log_{\sigma f}\ (T) + (1 - P(T)\ )log_{\sigma b}\ (T) - \ P(T)logP(T)\log\left(1 - P(T)\ \right) \quad (26)$$

where $\sigma_f(T)$ and $\sigma_b(T)$ are, respectively, the foreground and background standard deviations for each choice of T.

ISSN 22773061

$$\sigma_f(T) = \sqrt{\sum_{g=0}^{T} \left(g - m_f(T)\right)^2 p(g)} \quad (27)$$

$$\sigma_b(T) = \sqrt{\sum_{g=T+1}^{G} \left(g - m_b(T)\right)^2 p(g)} \quad (28)$$

Misclassification error

Based on misclassification error (ME) calculated for each algorithm, the best algorithm is chosen. The measurement is able to compute the background pixels that wrongly assigned to objects, and vice versa [19]. The ME can be simply expressed as:

$$ME = 1 - \frac{|B_O \cap B_T| + |F_O \cap F_T|}{|B_O| + |F_O|} \quad (29)$$

where B_O and F_O represent the background and foreground of the ground-truth image, respectively, and B_T and F_T represent the background and foreground area pixels in the test image, respectively. $|\bullet|$ is the cardinality of the set •. The ME varies from 0 for a perfectly classified image to 1 for a totally wrongly classified image.

Filtering operation

Filtering is an important part of any signal processing system, which involves an estimation of signal degradation and then restores the signal acceptably with its features that preserved together. Up to date, a number of filtering techniques have been reported and used if they were suitable for a variety of applications. Nonlinear filtering techniques are chosen if an image has been degraded by impulse noise. Furthermore, these nonlinear filtering techniques consider the nonlinear nature of the human visual system.

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 neighbourhood 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 [20]. The median filter can be formulated as [21]:

$$f(x,y) = median[g(s,t)],\ (s,t) \in S_{xy} \quad (30)$$

where S_{xy} represent the set of coordinates in a rectangular subimage window of size m x n, centered at point (x, y). 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}.

Generally, median filtering uses templates of size 3 x 3. The process for each pixel under consideration is as follows:

1. Consider the neighbourhood values.
2. Sort them in ascending or descending order.
3. Estimate the median value by using the following criteria:
   i. If there are an odd number of pixels in the template, the median is the centre of the sorted values.
   ii. If there are an even number of pixels in the template, the median is the average of the two centre pixels of the sorted values.
4. Assign this median value to the pixel in the same location in the output image.

For instance, in 3 x 3 and 5 x 5 neighborhoods, the median are the fifth and thirteen largest values, respectively, and so on. Once several values in a neighborhood are same, all equal values are grouped. As an example, a 3 x 3 neighborhood is supposed to has values (10, 20, 20, 15, 20, 20, 25, and 100). These values are then sorted as (10, 15, 20, 20, 20, 20, 20, 25, 100), which results in a median of 20. Thus, it can be concluded that the primary function of median filters is to force points with distinct gray levels to be more like their neighbors.

THE PROPOSED NOISE ELIMINATION PROCEDURE

Figure 5 depicts the PCB inspection system which includes image acquisition, image registration, defect detection, noise elimination procedure, and image addition. The elimination procedure consists of thresholding and filtering. After misalignment issue has been solved by implementing the image registration operation, the both images are then brought to defect detection using image subtraction operation, yielded positive image and negative image. These two images are then thresholded to eliminate noise occurred. In order to find out the thresholding algorithm that best fit to the noise elimination procedure, the misclassified pixels for each thresholding algorithm proposed have been computed using ME.

RESULT AND DISCUSSION

In order to obtain a well-aligned defective image, an image registration operation that consists of an affine transform and a bi-cubic interpolation were applied [7]. Figure 6 shows the defective image that subjected to be registered against the template image shown in Figure 3. The types of defects are also shown.

The results of the image subtraction operation on the PCB inspection system with and without registration's facility are illustrated in Figure 7 and Figure 8, and Figure 9 and Figure 10, respectively.

The results indicated that without a proper image registration operation, the PCB inspection system is likely to totally fail. Hence, the PCB inspection system incorporating with a reliable image registration operation has been used before thresholding algorithm takes place.

Tables 2 and Table 3 list the results in terms of threshold values, number of misclassified pixels, and ME. For evaluation of threshold algorithms on positive image, among these ten thresholding algorithms, the worst result was provided by the median algorithm, while the best result was provided by the minimum algorithm. In this case, for the median algorithm, it is equal to 253, corresponding to 315444 misclassified pixels and its ME is equal to 0.2177, while the minimum algorithm is equal to 139, corresponding to 19838 misclassified pixels and its ME is equal to 0.0137.

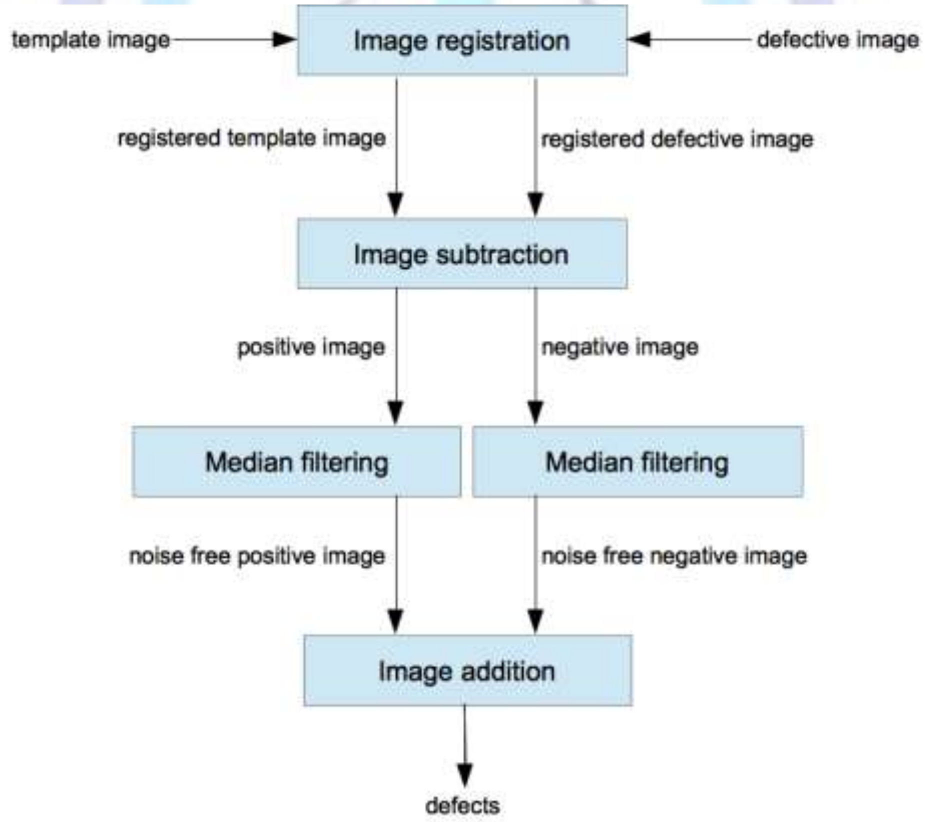
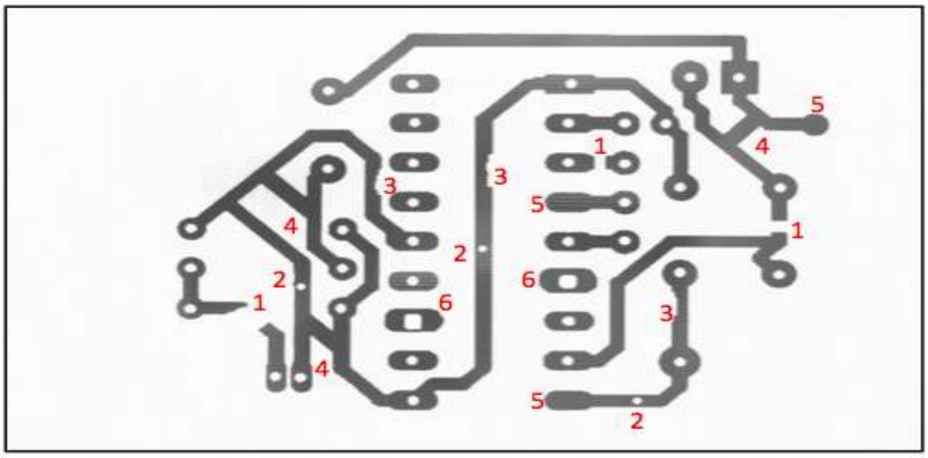


Fig 5: Flow chart of the proposed noise elimination procedure for real images of PCB



1. Open-circuit
2. Pin hole
3. Mouse bite
4. Short-circuit
5. Missing hole
6. Underetch

Fig 6: The defective image

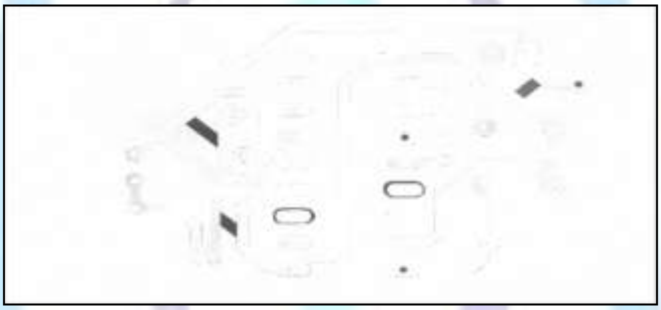


Fig 7: Positive image in gray-scale format with image registration

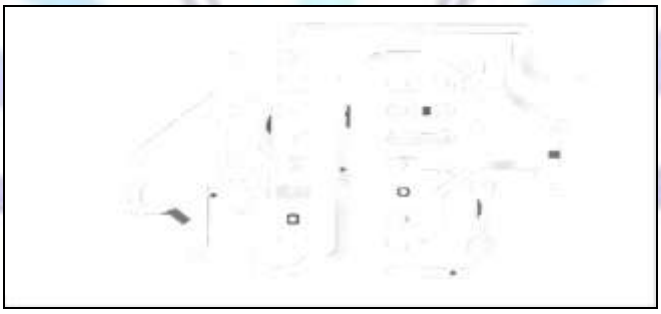


Fig 8: Negative image in gray-scale format with image registration

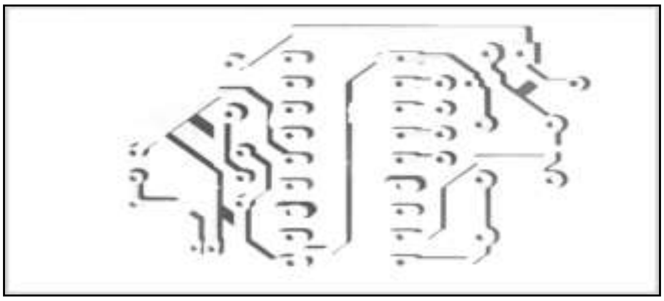


Fig 9: Positive image in gray-scale format without image registration

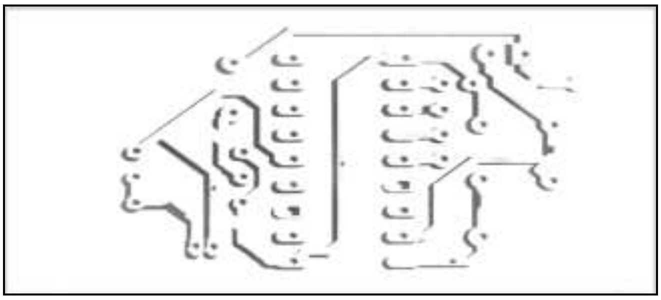


Fig 10: Negative image in gray-scale format without image registration

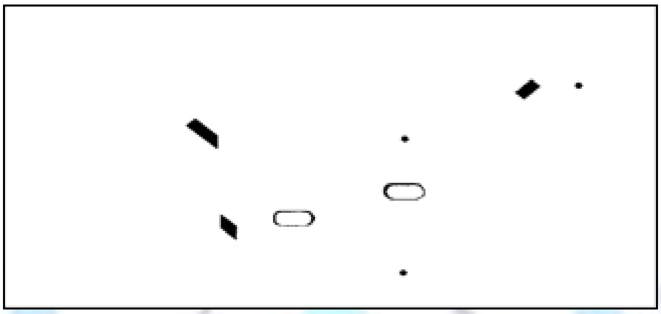


Fig 11: Thresholded positive image in binary format

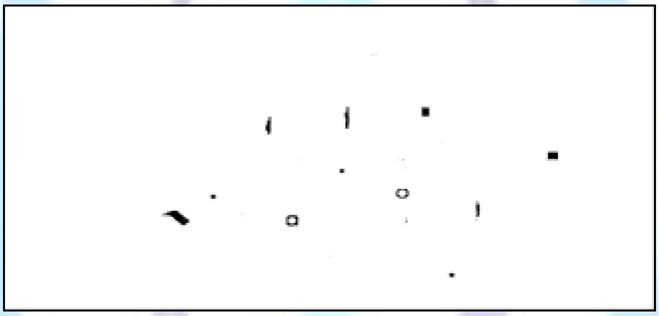


Fig 12: Thresholded negative image in binary format

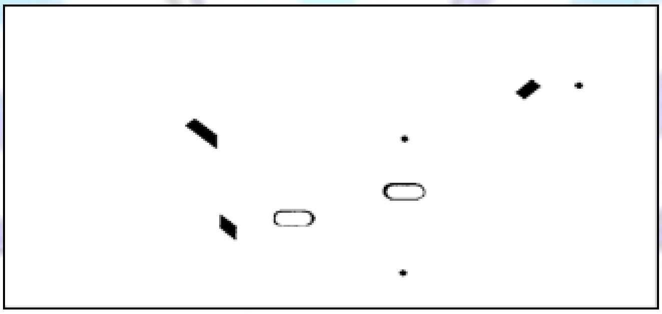


Fig 13: Filtered and thresholded positive image in binary format

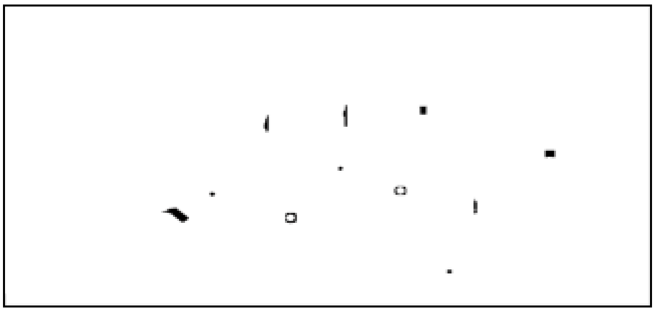


Fig 14: Filtered and thresholded negative image in binary format

Table 2. Performance evaluation of each thresholding algorithm on positive image

Thresholding algorithms	Threshold value, T	Misclassified pixels	Misclassification error (ME)
Minimum	139	19838	0.0137
Concavity	245	55555	0.0383
Entropy	241	43998	0.0304
Intermeans	144	19997	0.0138
Iterative intermeans	144	19997	0.0138
Intermodes	165	20753	0.0143
Mean	248	74451	0.0514
Median	253	315444	0.2177
Moments	124	18350	0.0127
Minerror	248	74451	0.0514

Table 3. Performance evaluation of each thresholding algorithm on negative image

Thresholding algorithms	Threshold value, T	Misclassified pixels	Misclassification error (ME)
Minimum	139	5242	0.0036
Concavity	252	65233	0.0450
Entropy	224	14897	0.0100
Intermeans	198	9919	0.0068
Iterative intermeans	198	9919	0.0068
Intermodes	186	8612	0.0059
Mean	253	76845	0.0530
Median	254	96525	0.0666
Moments	204	10738	0.0074
Minerror	253	76845	0.0530

For evaluation of threshold algorithms on negative image, among these ten thresholding algorithms, the worst result was provided by the median algorithm, while the best result was provided by the minimum algorithm. In this case, for the median algorithm, it is equal to 254, corresponding to 96525 misclassified pixels and its ME is equal to 0.0666, while the minimum algorithm is equal to 139, corresponding to 5242 misclassified pixels and its ME is equal to 0.0036.

As shown in Figure 7 to Figure 8, and Figure 11 to Figure 14, these images are obtained from the implementation of the proposed noise elimination procedure. Figure 7 and Figure 8 are the resultant images of defect detection. Using these two images, the minimum thresholding algorithm that is the best fit for the system has been implemented to the both images, produced thresholded positive image and negative image, as portrayed in Figure 11 and Figure 12.

According to these two evaluations, minimum algorithm is used to eliminate noise in positive and negative images. This algorithm has been chosen as its misclassification error value is the smallest compared to others. From an observation, the thresholding algorithm was able to eliminate most of noise occurred in the both images, however, many tiny noise were still there. To solve this critical problem, median filtering has been executed to the images of Figure 11 and Figure 12. Finally, two images consist of just defects and without any noise have been produced, as represented in Figure 13 and Figure 14. All the defects have been successfully detected.

CONCLUSION

Implementation of image subtraction operation to for PCB defect detection on real images of PCB produces the unwanted noise due to non-uniform illumination problem. For this concern, in this study, a combination of image registration, image thresholding and filtering is presented to eliminating unwanted noise. The minimum thresholding algorithm has found to be the best thresholding algorithm. The remaining noise is filtered out by median filtering. As a result, two noise-free defect images have been produced.

ACKNOWLEDGEMENT

This work is financially supported by Ministry of Higher Education Malaysia through the Universiti Teknologi Malaysia GUP Research Funds (Vote Q.J130000.7123.01J35) and Prototype Research Grant Scheme (PRGS 9013-00004).

REFERENCES

[1] M. Moganti, F. Ercal, C. H. Dagli and S. Tsunekawa. 1996. Automatic PCB Inspection Algorithms: A Survey, Computer Vision and Image Understanding, vol.63, no.2.

[2] W.Y. Wu, M.J. Wang and C.M. Liu. 1996. Automated Inspection of Printed Circuit Boards Through Machine Vision, Computers in Industry, vol.28, no.2, pp.103-111.

[3] J.H. Koo and S.I. Yoo. 1998. A Structural Matching for Two-dimensional Visual Pattern Inspection, IEEE International Conference on Systems, Man, and Cybernetics, vol.5, pp.4429-4434.

[4] C.R. Charette, S. Park, R. Williams, B. Benhabi and K.C. Smith. 1988. Development and Integration of a Micro-computer based Image Analysis System for Automatic PCB Inspection, Proceedings of International Conference on Computer Integrated Manufacturing, pp.129-135.

[5] L. Brown. 1992. A Survey of Image Registration Techniques, ACM Computing Surveys, vol.24, no.4, pp.325-376.

[6] R.M. Haralick, S.R. Sternberg, and X. Zhuang. 1987. Image Analysis using Mathematical Morphology, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.9, no.4, pp. 532-550, 1987.

[7] Ismail Ibrahim, Syed Abdul Rahman Syed Abu Bakar, Musa Mohd Mokji, Kamal Khalil, Zulkifli Md Yusof, Jameel Abdulla Ahmed Mukred, Mohd Saberi Mohamed, Zuwairie Ibrahim, An Image Registration Technique to Enhance PCB Inspection Algorithm with Real Images, ICIC Express Letters, vol.6, no.3, pp. 717-721, 2012.

[8] M. Sezgin and B. Sankur, Survey over image thresholding techniques and quantitative performance evaluation, Journal of Electronic Imaging, vol.13, no.1, pp. 146-156, 2004.

[9] I. Ibrahim, S.A.R Abu-Bakar, M.M. Mokji, K. Khalil, Z.M. Yusof, and Z. Ibrahim. 2011. Performance measurement of thresholding algorithms in printed circuit board inspection system. 3rd International Conference on Computational Intelligence, Modelling and Simulation.

[10] J.M.S. Prewitt and M.L. Mendelsohn. 1966. The analysis of cell images, Annals of the New York Academy of Science, vol. 128, no. 3, pp. 1035-1053.

[11] J. Weszka and A. Rosenfeld. 1979. Histogram modification for threshold selection, IEEE Transaction on Systems, Man and Cybernetics, vol. 9, no. 1, pp. 38–52.

[12] W. Doyle. 1962. Operation useful for similarity-invariant pattern recognition, Journal of the Association for Computing Machinery, vol.9, no.2, pp. 259-267.

[13] C.A. Glasbey. 1993. An analysis of histogram-based thresholding algorithms, Graphical Models and Image Processing (CVGIP 1993), vol. 55, no. 6, pp. 532-537.

[14] W. Tsai. 1985. Moment-preserving thresholding: a new approach, Computer Vision, Graphics, and Image Processing, vol.29, no.3, pp. 377-393.

[15] J.N. Kapur, P.K. Sahoo and A.K.C. Wong. 1985. A new method for gray-level picture thresholding using the entropy of the histogram, Graphical Models and Image Processing, vol.29, no.3, pp. 273-285.

[16] N. Otsu. 1979. A threshold selection method from gray-level histogram, IEEE Transactions on Systems, Man, and Cybernetics, vol.9, no.2, pp. 62-66.

[17] T. Ridler and S. Calvard. 1978. Picture thresholding using an iterative selection method, IEEE Transactions on Systems, Man, and Cybernetics, vol. 8, no. 8, pp. 630-632.

[18] J. Kittler and J. Illingworth. 1986. Minimum error thresholding, Pattern Recognition, vol. 19, no. 1, pp. 41-47.

[19] Y. Bazi, L. Bruzzone and F. Melgani. 2007. Image thresholding based on the EM algorithm and generalized gaussian distribution, Pattern Recognition, vol. 40, no. 2, pp. 619-634.

[20] A. Hussain, M.A. Jaffar and A.M. Mirza. 2011. Random-Valued Impulse Noise Removal Using Fuzzy Logic, International Journal of Innovative Computing, Information and Control, vol. 6, no. 10, pp. 4273-4288.

[21] R.C Gonzalez and P. Wintz. 1987. Digital image processing, Addison-Wesley Publishing Company, Inc.