TM03-2

Analysis of the Wavelet-based Image Difference Algorithm for PCB Inspection

Z. Ibrahim[1], S.A.R. Al-Attas[2], Z. Aspar[3]
Department of Microelectronics and Computer Engineering
Faculty of Electrical Engineering
Universiti Teknologi Malaysia
81310 Skudai, Johor, Malaysia
[1]zuwairie@fke.utm.my
[2]syed@fke.utm.my
[3]zulfakar@fke.utm.my

Abstract: The methodology and results regarding the use of wavelet transform and multiresolution analysis in automated visual printed circuit board (PCB) inspection provide the motivation for this research. In this paper, the wavelet-based image difference algorithm proposed is applied to a sample PCB image. The algorithm is applied by using Haar wavelet where several different numbers of levels are considered. One conclusion from this paper is that the second level Haar wavelet transform should be selected for the application of visual PCB inspection.

Keywords: Industrial inspection, printed circuit board, wavelet.

1. Introduction

Nowadays, there exist a lot of PCB defect detection approaches, techniques and algorithms. Instead of designing a new algorithm for PCB defect detection, this paper emphasizes on the algorithm development for computation time reduction since the inspection time is a major factor in industries. On the other hand, there has been an increasing number of applications for wavelets and multiresolution analysis including (but not limited to) image compression, image denoising and edge detection. Up till now there is still no clear advantages of wavelets in industrial inspection application, especially for PCB inspection. Hence, the wavelets and multiresolution analysis is explored to achieve real time inspection of PCBs.

The rest of this paper is organized as follows. Section 2 describes the reviews of the automated PCB inspection approaches and the research methodology chosen for this project. Section 3 and Section 4 present in detail the image difference operation and the simplified computation of wavelet transform respectively. Section 5 contains the experimental results of defect detection. The experimental computation time is gathered for comparison. The experimental results are discussed in Section 6. Finally, the conclusions of this paper are presented in Section 7.

2. Literature Reviews and Research Methodology

Visual PCB inspection approaches involved either one or both of these approaches: referential approaches and rule-based approaches. The referential approaches compare the test PCB images against the reference PCB images. There are two major techniques: image comparison and model-based technique. Image comparison, which is the simplest technique, consists of comparing both images pixel-by-pixel by XOR logic operator (1). The operation is also called as image difference operation. Model-based methods are techniques, which match the pattern under inspection with a set of predefined models.

Rule-based approaches test and determine whether each feature of PCB images fall within the required dimensions or not. These method typically use morphological technique where erosion and dilation as a basic operation (2 - 4). It does not require a reference model but it might miss large defects.

Ercal et al. (5 - 7) claimed that RLE-based technique could be used to reduce the computation time of a PCB inspection system. However, the conversion from Bitmap to RLE and vice versa is a time consuming process. Thus, the PCB images are segmented into basic patterns before the conversion process takes place. Instead of RLE-based technique, wavelet-based image difference algorithm is proposed where the PCB images are modeled via third level Haar wavelet transform as clearly discussed by Ibrahim and his colleagues [8 - 9]. The flow of the algorithm is shown in Fig. 1.

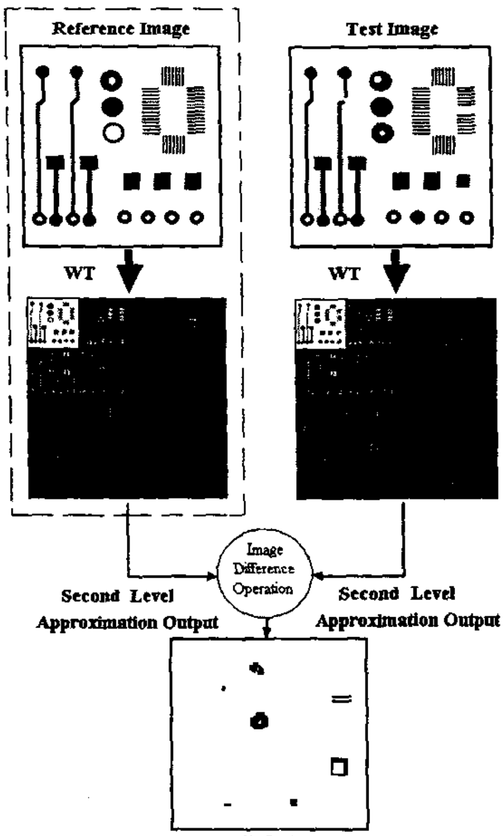


Fig. 1: Wavelet-based image difference algorithm

3. Image Difference Operation

There are several types of Boolean logical operator which include AND, NAND, NOT, OR, X-OR (exclusive OR) and NOR. Image difference operation can be likened as using the X-OR logic operator. For 2 inputs 1-bit data, the truth table of X-OR is given in Table 1.

The notation of XOR logic operator is:

$$Output = Bit1 \oplus Bit2 \tag{1}$$

where $\oplus$ represents the exclusive OR operation. For an image data, the operation is similar but the logical operation is applied at every pixel in an image. Consider two binary images named as *Img1* and *Img2* of size M x N where M is the length and N is the height of the image respectively. For each pixel at corresponding location (x,y) that is x from 1 to M, y from 1 to N where both x and y are integers, $Img1[x,y]$ and $Img2[x,y]$ has a value of either one or zero. In this case, it is assumed that one is identified as background pixel, whereas zero is identified as foreground pixel. Similarly, for every corresponding value of x and y in an image, the image difference operation is defined as:

$$Difference(x,y) = Img1(x,y) \oplus Img2(x,y) \tag{2}$$

Table 1: Truth table of X-OR logic operator

Bit 1	Bit 2	Output
0	0	0
0	1	1
1	0	1
1	1	0

In this project, image difference operation is applied onto the output of the second level wavelet transform and not on the binary image. The input data is no longer an integer value but rather a floating value. Therefore, it is important to do some modification in order to suite the image difference operation with the input data. The modification of image difference operation is given in the form of iteration, applied pixel-by-pixel. Therefore, for every pixel of *Img1* and *Img2*:

- For *Img1*, grab a pixel value at location *Img1(x,y)* and named it as *PixVal1*.
- For *Img2*, grab a pixel value at location *Img2(x,y)* and named it as *PixVal2*.
- If *PixVal1* is equal to *PixVal2*, assign the corresponding pixel at location *Difference(x,y)* as background pixel.
- If *PixVal1* is **not** equal to *PixVal2*, assign the corresponding pixel at location *Difference(x,y)* as foreground pixel.

4. Simplified Computation of Wavelet Transform

Basically, in order to retrieve the second level approximation of wavelet transform LL_2, the computation involves two iterations of separable convolution. Since for each level i, the detail components HL_i, LH_i and HH_i are not needed in this project, the convolution of image data by high pass filter to get those detail components can be neglected.

Thus, the convolution occupies mainly on low pass filter convolution only but the iteration still must be applied for two times.

There are many considerable parameters within the calculation of the wavelet transform. Speed and boundary treatment are some of them. In order to simplify and to speed up the computation of wavelet transform, Haar wavelet is chosen. The coefficients for the Haar wavelet is given in Table 2. Haar wavelet is chosen because, generally, long filters require more computing time than the short ones [10]. Furthermore, other wavelet filters excluding Haar require a solution to the boundary, and cause the computation to be more complex. In this subsection, a simplified version of two level approximation, LL_2 computation is given where the iteration is applied only once in order to increase the speed and to make the computation straightforward.

Table 2 : Haar wavelet filter

	First Coefficient	Second Coefficient
Low Pass Filter, {c0, c1}	½	½
High Pass Filter, {c1, -c0}	½	-½

As an example, lets take a look at a small portion of an image. Noted that, the calculation of every level of wavelet transform involved a separable successive convolution of row/column followed by column/row. For better understanding, observe the portion of an image represented by $p1_{(k)}, p2_{(k)}, p3_{(k)}$ and $p4_{(k)}$ within the 2 x 2 pixels, shown in Fig. 2.

Once the column convolution is calculated, the defined segment and also the whole image will be compressed in column direction. Depending on the application, the convolution is applied either by low pass filter or high pass filter resulting the low pass output coefficient or high pass output coefficient. Let L and H denote the output coefficient of the low pass filter and the high pass filter respectively. The calculation of L and H correspond to the pixel $p1_{(k)}$, $p2_{(k)}, p3_{(k)}$ and $p4_{(k)}$ at every segment, k can be written as:

$$L_{p1,p2\,(k)} = \left(\tfrac{1}{2}\right)p1_{(k)} + \left(\tfrac{1}{2}\right)p2_{(k)} = \left(\frac{p1_{(k)}}{2}\right) + \left(\frac{p2_{(k)}}{2}\right) \tag{3}$$

$$L_{p3,p4\,(k)} = \left(\tfrac{1}{2}\right)p3_{(k)} + \left(\tfrac{1}{2}\right)p4_{(k)} = \left(\frac{p3_{(k)}}{2}\right) + \left(\frac{p4_{(k)}}{2}\right) \tag{4}$$

$$H_{p1,p2\,(k)} = \left(\tfrac{1}{2}\right)p1_{(k)} + \left(-\tfrac{1}{2}\right)p2_{(k)} = \left(\frac{p1_{(k)}}{2}\right) - \left(\frac{p2_{(k)}}{2}\right) \tag{5}$$

$$H_{p3,p4\,(k)} = \left(\tfrac{1}{2}\right)p3_{(k)} + \left(-\tfrac{1}{2}\right)p4_{(k)} = \left(\frac{p3_{(k)}}{2}\right) - \left(\frac{p4_{(k)}}{2}\right) \tag{6}$$

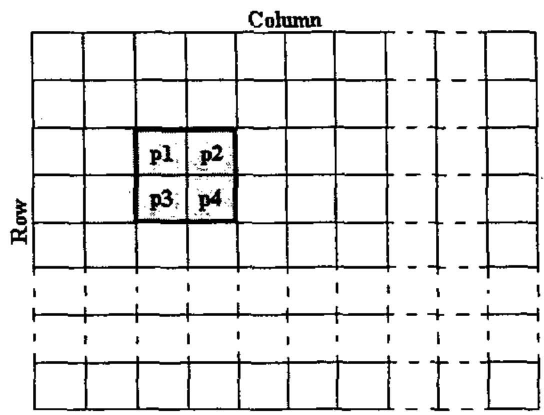


Fig. 2: A 2 x 2 pixels, a segment of an image

The row convolution is calculated next. Hence, for every segment, the coefficients for approximation, horizontal detail, vertical detail and diagonal detail, as named by $APPROXIMATION1_{(k)}$, $HORIZONTAL1_{(k)}$, $VERTICAL1_{(k)}$ and $DIAGONAL1_{(k)}$ respectively will be generated. As a consequence, the calculation of each $APPROXIMATION1_{(k)}$, $HORIZONTAL1_{(k)}$, $VERTICAL1_{(k)}$ and $DIAGONAL1_{(k)}$ corresponding to the pixel $p1_{(k)}, p2_{(k)}, p3_{(k)}$ and $p4_{(k)}$ at each segment can be represented by these equations:

$$APPROXIMATION1_{(k)} = \frac{\left(\frac{p1_{(k)}}{2}\right)+\left(\frac{p2_{(k)}}{2}\right)}{2} + \frac{\left(\frac{p3_{(k)}}{2}\right)+\left(\frac{p4_{(k)}}{2}\right)}{2}$$

$$= \frac{p1_{(k)} + p2_{(k)} + p3_{(k)} + p4_{(k)}}{4} \tag{7}$$

$HORIZONTAL1_{(k)}$ =

$$\frac{\left(\frac{p1_{(k)}}{2}\right) + \left(\frac{p2_{(k)}}{2}\right)}{2} - \frac{\left(\frac{p3_{(k)}}{2}\right) + \left(\frac{p4}{2}\right)}{2}$$

$$= \frac{p1_{(k)} + p2_{(k)} - p3_{(k)} - p4_{(k)}}{4} \tag{8}$$

$VERTICAL1_{(k)}$ =

$$\frac{\left(\frac{p1_{(k)}}{2}\right) - \left(\frac{p2_{(k)}}{2}\right)}{2} + \frac{\left(\frac{p3_{(k)}}{2}\right) - \left(\frac{p4_{(k)}}{2}\right)}{2}$$

$$= \frac{p1_{(k)} - p2_{(k)} + p3_{(k)} - p4_{(k)}}{4} \tag{9}$$

$DIAGONAL1_{(k)}$ =

$$\frac{\left(\frac{p1_{(k)}}{2}\right) - \left(\frac{p2_{(k)}}{2}\right)}{2} - \frac{\left(\frac{p3_{(k)}}{2}\right) - \left(\frac{p4_{(k)}}{2}\right)}{2}$$

$$= \frac{p1_{(k)} - p2_{(k)} - p3_{(k)} + p4_{(k)}}{4} \tag{10}$$

Equations 3 through 10 represent the calculation involve in the defined segment shown in Fig. 2. In the same way, these equations is applied also for each 2 x 2 segment of the image. However, as stated before, only the approximation part of each level is needed instead of the horizontal detail, vertical detail and the diagonal detail. Thus, only the calculation of the approximation part is considered.

Referring to equation 7, for the first level of wavelet transform, it can be said that each coefficient of the approximation can be achieved by simply averaging the four pixels $p1_{(k)}$, $p2_{(k)}$, $p3_{(k)}$ and $p4_{(k)}$. Based on this argument, it is possible to calculate the second level approximation directly from the original image.

As opposed to the first level wavelet transform, in which the calculation is applied on every 4 pixels within every 2 x 2 pixels segment, for the second level wavelet transform, the calculation involved every 16 pixels within every 4 x 4 pixels segment of the original image. Figure 3 demonstrates a 4 x 4 pixels segment of an original image and the way these pixels are operated in order to get the second level approximation coefficient at each segment, $APPROXIMATION2_{(k)}$ as shown by equation 11.

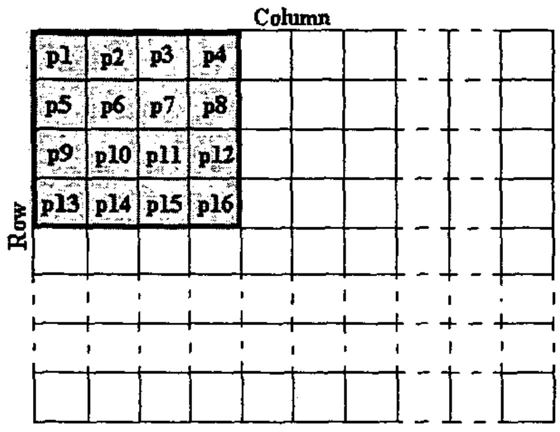


Fig. 3: A 4 x 4 pixels, segment of interest of an image

$$APPROXIMATION2_{(k)} = \frac{1}{16} \sum_{i=1}^{16} p_i^k \tag{11}$$

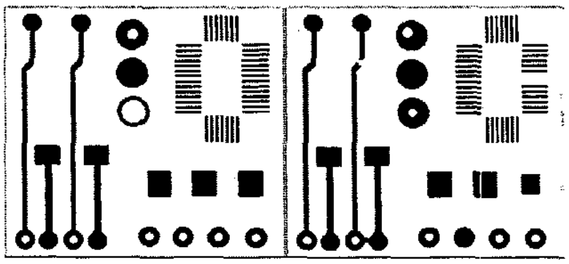
In the above equation, p_i is already defined in Fig. 3 and k denotes a respective segment in the image.

As a conclusion, the simple and easy derivations in this few pages confirm that, by choosing Haar wavelet as a basis of wavelet transform, the calculation of wavelet transform can be shorten by just taking it as a moving average operation segment-by-segment of a given image. In contrast with the standard decomposition of wavelet transform, which is done iteration-by-iteration, it is possible that the second level coefficient of wavelet transform can be achieved from the original image. This circumstance is only valid for Haar wavelet. Similarly, the direct computation of the third level wavelet transform can also be calculated by the same way.

$$APPROXIMATION3_{(k)} = \frac{1}{64} \sum_{i=1}^{64} p_i^k \tag{12}$$

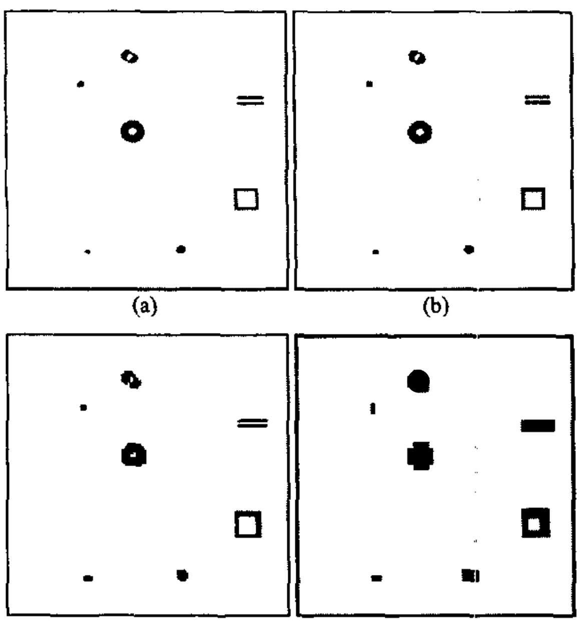
5. Experimental Results

A number of different level of Haar wavelet-based techniques are tested by the use of MATLAB as a platform powered by an 800 MHz Pentium III Microcomputer with 256 MB memory. The good quality and defective sample PCB image is shown in Fig. 4. The size of each of the image is 400 x 400 pixels. The results of detection along with the computation time are shown in Fig. 5 and Table 3 respectively.



(a) (b)

Fig. 4: A 400 x 400 sample PCB image (a) Good quality PCB image (b) Defective PCB image



(a) (b) (c) (d)

Fig. 5: Haar wavelet-based differenced image (a) Level 0 (no wavelets involved) (b) Level 1 (c) Level 2 (d) Level 3

6. Analysis and Discussions

According to the Table 3, the data associated to the computation time is plotted as shown in Fig. 6 in order to simplify the analysis. Note that the zero level wavelet transform means that no wavelet transform involves in the image difference operation and the image difference operation is done on the original full size of input images.

Table 3: Computation time of the Haar wavelet-based image difference

	Computation time (s)			
Operation	**Level 0**	**Level 1**	**Level 2**	**Level 3**
Wavelet transform	No need	1.582	1.162	1.092
Image difference operation	7.251	0.781	0.200	0.050
Total	7.251	2.363	1.362	1.142

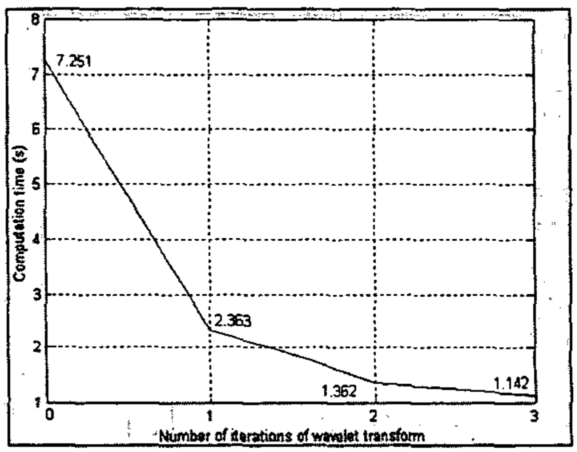


Fig. 6: Computation time reduction

Due to the graph in Fig. 6, the computation time reductions by using the first level Haar wavelet is about 67.41%. Meaning that more than half of the computation time acquired by image difference operation is saved.

Up one level ahead, the percentage of reduction from the first level Haar wavelet-based to the second level Haar wavelet-based is 42.36%. However, the percentage of the overall inspection time reduction becomes poorer and unacceptable via the use of the third level Haar wavelet. The percentage is about just 16.15%. The result is unacceptable in the sense that it is not worthy to utilize the third level wavelet transform in order to reach just a little bit improvement in the computation time. In reality, wavelet transform is an expensive task when the transformation is going to be implemented on hardware such as DSP chip or FPGA. Based on the argument, second level wavelet transform is considered for hardware implementation.

7. Conclusions

As a conclusion, it can be said that the wavelet-based image difference for PCB defect detection is better than the existing PCB defect detection algorithms, which rely on image difference operation. Also, the second level Haar wavelet transform should be selected in order to minimized the computation time reasonably. Hence, the objective of this project has been fulfilled, which is to minimize the computation time for the automated defect detection of PCBs. The proposed algorithm could be used for an automated visual PCB inspection system to achieve real time inspection of PCBs.

8. References

[1] W. Wen-Yen, J. W. Mao-Jiun, and L. Chih-Ming, "Automated Inspection of Printed Circuit Boards Through Machine Vision", *Computers in Industry*, Vol. 28, Issue 2, 103 – 111, 1996.

[2] H. O. Seyfullah, and L. Onural, "An Automated System for Design-Rule-Based Visual Inspection of Printed Circuit Boards", *Proceedings of the IEEE International Conference on Robotics and Automation*, 2696 – 2701, 1991.

[3] A. M. Darwish, and A. K. Jain, "A Rule Based Approach for Visual Pattern Inspection", *IEEE Transaction of Pattern Analysis and Machine Intelligence*, Vol. PAMI-10, No.1, 56 – 58, 1988.

[4] Y. Qin-Zhong, and P. E. Danielson, "Inspection of Printed Circuit Boards by Connectivity Preserving Shrinking", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. PAMI 10, No. 5, 737 – 742, 1998.

[5] F. Ercal, F. Bunyak, H. Feng, and L. Zheng, "A Fast Modular RLE-based Inspection Scheme for PCBs", *Proceedings of SPIE – Architectures, Networks and Intelligent Systems for Manufacturing Integration*, Vol. 3203, 45 – 59, 1997.

[6] F. Ercal, F. Bunyak, and H. Feng, "Context-sensitive Filtering in RLE for PCB Inspection", *Proceedings of SPIE – Intelligent Systems and Advanced Manufacturing*, Vol. 3517, 1998.

[7] F. Ercal, M. Allen, and H. Feng, "Proof of Correctness and Performance Analysis of a Systolic Image Difference Algorithm for RLE-Compressed Images", *IEEE Transaction on Parallel and Distributed Computing Systems*, Vol. 11, No. 5, 433 – 443, 2000.

[8] Z. Ibrahim, S.A.R. Al-Attas, and Z. Aspar, "Wavelet Based Algorithm for Image Comparison Technique in Application of Coarse-To-Fine Inspection of Printed Circuit Board (PCB)", *Malaysian Science and Technology Congress (MSTC)*, Malaysia, October 2001, (Presented).

[9] Z. Ibrahim, S.A.R. Al-Attas, Z. Aspar, and R. Ghazali, "Performance Evaluation of Wavelet-Based Algorithm for Printed Circuit Board (PCB) Inspection", *Jurnal Teknologi D: Universiti Teknologi Malaysia*, No. 35(D), 39 – 54, 2001.

[10] C. Schremmer, C. Kuhmunch, and C. Esser, "Wavelet Filter Evaluation For Image Coding", *Technical Report TR-01-006*, Department for Mathematics and Computer Science, University of Mannheim.