A Counting Algorithm and Application of Image-Based Printed Circuit Boards

Ping-Huang Wu[1]* and Chin-Hwa Kuo[2]

[1]Department of Electrical Engineering, Tungnan University, Taipei, Taiwan 222, R.O.C.
[2]Department of Computer Science and Information Engineering, Tamkang University, Tamsui, Taiwan 251, R.O.C.

Abstract

In the production line of printed circuit boards (PCB), quality control requires constant counting and recording of the number of boards which, in some special cases, is done by workers in the factory. This form of counting consumes labor and time, not to mention the high error rate. In this paper, an automatic method to count PCBs is proposed. With the use of image processing techniques, the number of PCBs can be non-destructively calculated as they pass through the production line, which will effectively improve the production efficiency. The first step of the proposed system is to take a picture of a stack of PCBs using a digital camera. The captured image is transformed to a gray level picture and processed by a noise removal algorithm to improve the proposed mechanism's precision. The output of the counting algorithm includes PCB count and confidence level. In practical applications, the confidence level can help people to on-line diagnose erroneous counting and correct it when the visualization counting system fails to accurately differentiate the boards.

Key Words: Image Processing, Printed Circuit Board (PCB), Counting Algorithm

1. Introduction

The Printed circuit boards (or PCB) are mainly used to mechanically support parts assembly and they electrically connect electronic components using conductive pathways. The characteristic of PCBs varies according to user design needs, which gives rise to electronic products that can be light, thin, short, or small. The PCBs are extensively used in computers, automated offices, electronics, as well as in communication, medicine, and military. The PCBs have become inseparable from daily life.

Many PCB manufacturers [1,2] have pointed out that frequent checking the number of PCBs in the production line is an essential part of quality control, such as the number of input, output and defective rate. Nevertheless, the numerous types of PCBs with different thickness and with different production procedure have made it quite difficult for the industry to come up with an automatic and precise PCB calculation method. Therefore, some companies rely on cost- and time-consuming human labor. There are already numbers of calculation methods in the industry field. For example, the banknote counting machines in banks, and the flow control systems that limits the number of people in open exhibitions for safety reasons; however, none of these would apply to counting PCBs. In this paper, we proposed a novel method to automatically calculate the number of PCBs. A digital camera is set up at a location, and when a stack of PCBs with the same specifications is sent to the designated spot, an image is taken that will be analyzed by our designed algorithm. The number of PCBs can be calculated instantly, which allows manufacturers to lower their labor cost, and at the same time, improve efficiency.

Due to the fact that different PCBs have different thickness, colors, in addition to the color variations in the

*Corresponding author. E-mail: phwu@mail.tnu.edu.tw

multilayer PCBs and uncontrollable lighting conditions, image capturing is presented with great complexity. This paper aims to overcome these problems to produce accurate counting results.

The digital camera captures an image of a whole stack of PCBs with the same specifications on the moving conveyor belt, of which is sent to a computer to visualize the image and find the average thickness that will in turn produce the number of PCBs through our designed algorithm.

The problems in the past and related literature will be introduced in section 2. Section 3 will explain in detail about our designed algorithm. Section 4 and 5 are the result analysis and conclusion, respectively.

2. Literature Review

Traditionally, there are several ways to automatically count the number of PCBs, for example, measuring the weight or thickness of PCBs directly or detecting the added marks on the PCBs by sensors. The former requires PCB specification beforehand in some situations, and the latter requires unnecessary marks on the PCBs, thus increasing the risk of damage.

Making use of image processing technologies, the proposed method of counting the number of PCBs on a moving conveyor belt requires to overcome the following problems.

(a) Unpredictable lighting effect

Image quality is greatly affected by environmental lighting variations. In a dim lighting condition, the gray value of image pixels will be lower, making it difficult to clearly make out the features of the PCB boundary taken by the camera [3]. On the other hand, in an overexposed lighting condition, reflection of the false boundary occurs. As presented in Figure 1, the light source comes from the top, where the light is unevenly distributed between each piece of PCB, turns out increasing complexity to distinguish individual PCBs.

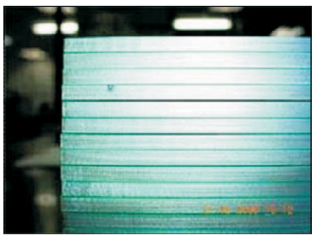


Figure 1. PCBs under the effects of a lighting condition.

(b) Irregular board colors

Since the PCBs are made with Multilayer of different materials [1], the side of the PCBs is therefore uneven in color in some cases. The PCBs can have colors with no regular patterns, such as the one shown in Figure 2.

Figure 2. Multilayer PCBs.

(c) PCB with inclination

As shown in Figure 3, lopsided image might introduce additional disturbance to the designed algorithm. For example, the convolution and the length of noise reduction cannot be too long, and multi-point calculation leads to failure of compensation.

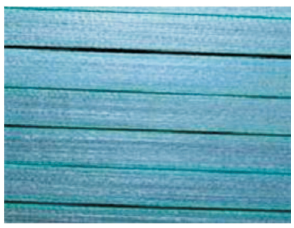


Figure 3. An image of lopsided PCBs.

This paper aims to design a visualization-based software in calculating the number of PCBs. Image processing techniques employed in the study such as noise reduction and contrast adjustment are presented in the following sections.

2.1 Image Noise Processing

Most camera noises can be divided into two types: high frequency noise and salt-and-pepper noise [4,5]. The Gaussian noise usually occurs at the time of image capture. It is possible to filter out the noise by employing features of the Gaussian distribution, and uses low pass filter or morphology operations to filter out regions with high frequencies. The salt-and-pepper noise usually occurs in the process of digitalization in the camera, for example, errors in the memory location [5]. This type of noise can be reduced using either the median filter or the peak-and-valley filter [4].

In this paper, we have to overcome the problems caused by these two types of noises. The filtering techniques mentioned above have provided us with effective ways to reduce the noise of captured images, which will help us to avoid the unwanted errors.

2.2 Image Contrast Adjustment

In addition to the noise reduction technology, an effective analysis of the input image can provide us with the possibility to improve the contrast of the PCB images when necessary.

The effect of histogram equalization is shown in Figure 4, where Figure 4(a) is the original image, and Figure 4(b) is the equalization result in HSI color space. In terms of the visual effect, the contrast is increased after the histogram equalization; the gray level histogram is equally expanded on the overall intensity level [6,7]. Regardless of the images under dim or overexposed lighting conditions, the histogram equalization enables the same images taken under these conditions to achieve a better distinguishing level.

The technique of histogram equalization is used in this paper to avoid the influence of lighting conditions, and when coupled with image processing in the color space, the accuracy of proposed method can be improved.

2.3 Related Literature on Counting Algorithm Using Image Processing

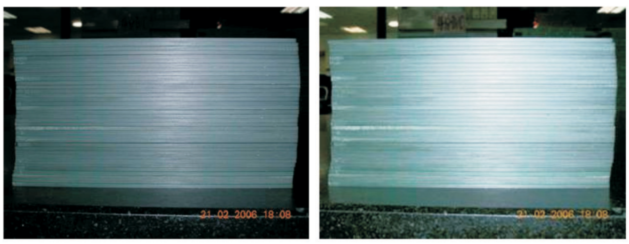
Prior studies, such as the pedestrian counting systems [8–10], have employed counting algorithm using image processing, but since the targets of these systems are moving objects, these algorithms would not be applicable to counting static objects like PCBs.

Another paper [11] used photo counting detectors to read out cross strips. Though the system is capable of counting the number of cross strips, the use of photon technology equipments requires delicate handling and is quite expensive, making this counting method unsuitable to PCB counting.

So far, there is no related study that uses similar image processing technique proposed in this study for the PCB counting.

3. PCB Counting Algorithm

Presented in Figure 5 is the flow chart of the proposed PCB counting visualization system. First, the image is input, followed by the noise reduction and the gray level transformation and then the images are vertically divided into several regions, with the individual horizontal intensity level signal of each region. The rough single PCB board thickness is estimated by a global processing method. The rough thickness is then used as an input to identify the PCB area recursively. During the process of PCB area identification, the confidence level of this cal-



(a) Original image (b) Equalization result in HSI color space

Figure 4. Histogram equalization.

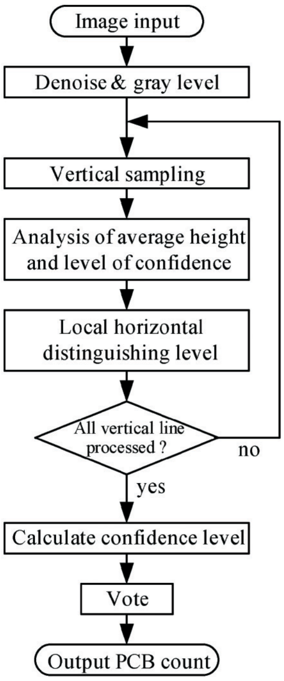


Figure 5. Flow chart of counting the number of PCBs.

culation is produced as well. When each vertical region is processed by this algorithm, the final PCB count is decided by a voting algorithm. Detailed descriptions of each step are presented in subsections 3.1 to 3.6.

3.1 Image Input

The image of the stacks of PCBs is taken by a digital camera set up at a designated location with some distance to the moving conveyor belt. A sample image taken by the camera is shown in Figure 6.

To allow our designed algorithm to operate normally, the input image must comply with the following assumptions:

(1) The PCBs are positioned horizontally.
(2) The thickness of PCBs in the captured image should be greater than 10 pixels.
(3) The tilt angle of PCBs should be less than 10 pixels and not greater than 30% of the pixel value of the thickness.

3.2 Noise Reduction and Gray Level Transformation

In our study, a standard gray level transformation formula (1) is used to convert the RGB signal into the gray level signal intensity.

$$Y = [0.299,\ \ 0.587,\ \ 0.114] \cdot [B, G, R]^{-1} \tag{1}$$

At the present stage, a standard gray level transformation formula is used, but the resolution of green component will inevitably be greater than the other two components if we take the PCB specification into consideration. A better transformation can be studied in the future for the PCB counting application.

After the color signal has been transformed into single intensity level signal, a two-dimensional low-pass filter as formula (2) is used to reduce the noise interference.

$$\begin{bmatrix} 0.00 & 0.00 & 0.06 & 0.00 & 0.00 \\ 0.03 & 0.14 & 0.54 & 0.14 & 0.03 \\ 0.00 & 0.00 & 0.06 & 0.00 & 0.00 \end{bmatrix} \tag{2}$$

This filter is designed to reinforce the characteristics of the horizontal position layout of the PCBs, so horizontal sampling points are increased while not affecting the vertical signal characteristics. Please note that this filter takes account of the vertical edge feature and mainly deals with the horizontal noise; the horizontal direction signal could be smoothen to eliminate the noise, while the shorter vertical filter length could maintain the luminance changes.

3.3 Vertical Sampling

The image of the PCB stack takes up a large portion of the captured image, and to make use this characteristic, the designed algorithm takes several evenly spaced vertical samplings to calculate the number of PCBs. As illustrated in Figure 7, for every 100 pixels, 10 pixels are sampled. Each pixel is subject to apply noise reduction filter by formula (2), and an average from horizontal 10 pixels is computed to get the light intensity signal for each vertical sampling line. As presented in Figure 8, the y-axis is the light intensity of the gray level signal, where as the x-axis stands for the top-bottom displacement of one of the vertical lines in Figure 7.

When 10 pixels are averaged for each sampling, the tilt angle of the PCBs must be taken into account; if the partition line sampling of the tilt is greater than 10 pixels, our algorithm will not work out a reliable result.

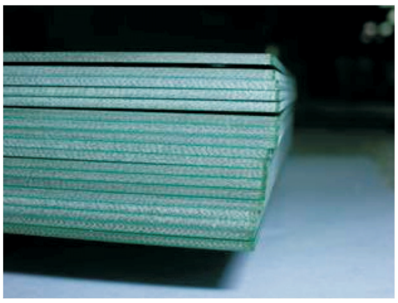


Figure 6. An input image from the PCB counting visualization system.

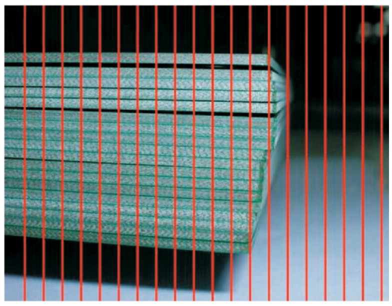


Figure 7. Vertical sampling.

3.4 Analysis of Average Height and Level of Confidence

To obtain the average distance between two wave troughs, the average value of the highest and lowest signal is set as the global horizontal distinguishing level, which is the initial level as shown in Figure 8.

Before analyzing the PCB board height, a pre-processing to identify possible PCB stack boundary can be applied to speed up the iterative process. Take Figure 8 as an example, the first significant signal change happens at around 1 to 100, and the last significant change happens at around 1300 to 1400; samples outside this range can be ignored. The initial values are used to expedite the processing; extra background area could be taken into calculation, but PCB board area is always in the valid range.

Two types of curves, upward or downward, are derived based on the intersections between the global horizontal distinguishing level and the signal waves. The starting and ending positions of each interval, as well as the length, are recorded and can be classified as either upward or downward. Figure 9 is an enlargement of offset from 500 to 700 in Figure 8, where the intersections have been segmented and are classified as black or gray intervals.

One of the two classified sets represents the PCB area. After each set is sorted and filtered, lengths that are too long or too short will not be taken into consideration. The confidence level of the black line segments ($Confid_B$) is calculated by formula (3), as shown in Figure 9, n represents the number of black lines, and L_{Bs} stands for the length of the sth black line. A value of 0~100 is obtained, in which 100 means that all the lines are of average length, and 0 means that there is a large variation in lengths. Since PCBs should have consistent thickness, a confidence level of 100 indicates that this set of lines is very likely to represent a stack of PCBs. After $Confid_B$ is calculated, the same formula is used to find the confidence level ($Confid_G$) of the gray line segments. $Confid_B$ and $Confid_G$ are compared, and whichever is greater will represent the confidence level of that particular horizontal line.

For this set of signal, the global horizontal distinguishing level is moved 1-pixel up, and then the result is recorded to obtain the confidence level in each case.

$$
\begin{aligned}
&Confid_B = \left(1 - \frac{\sum_{s=1}^{n} f_s}{n}\right) \times 100 \\
&f_s = \begin{cases} f'_s & \\ 1 \quad ,if \quad f'_s > 1 \end{cases} \\
&f'_s = \left| 1 - \frac{L_{Bs}}{Average_B} \right| \\
&Average_B = \frac{\sum_{s=1}^{n} L_{Bs}}{n}
\end{aligned}
\tag{3}
$$

After the confidence levels of all the horizontal lines have been computed, the one with the highest confidence level will be used to represent the fundamental horizontal level for this particular signal, and at the same time, the average thickness of PCBs can be obtained. Intervals with lengths that are close to the average length of PCBs are immediately marked to represent PCBs.

3.5 Local Horizontal Distinguishing Level

From section 3.4, an initial result of the correct seg-

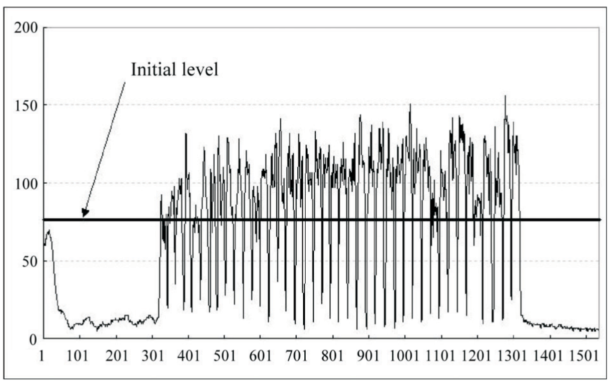


Figure 8. Global horizontal distinguishing level.

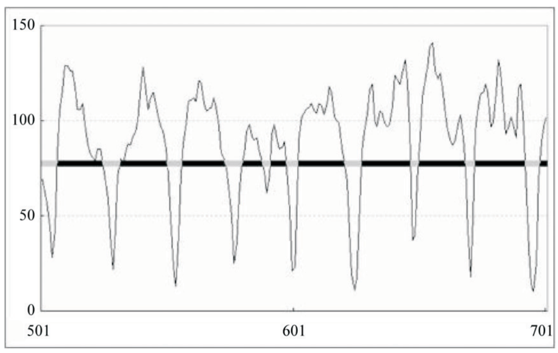


Figure 9. Classification of intersecting lines based on intersection attributes.

ments representing the PCBs can be found, but in the regions where indistinct signals still exist, the global horizontal distinguishing level will not be able to identify these segments. The horizontal distinguishing level should be adjusted locally to find possible PCB in a given segment.

A recursive algorithm is used to repeatedly search for valid intervals with the out-of-proportion lengths, i.e. intervals with longer length that cannot be classified. When an expected interval length appears after adjusting the horizontal line, this means that a new piece of PCB has been found. The overall interval distribution would have been changed, and the search is repeated until no new interval is generated. As presented in Figure 10, a new segment representing PCBs which cannot be processed using the global horizontal distinguishing level is found using the above-mentioned search method.

When the recursion structure stops searching, meaning the intervals are stable, the confidence level for all the intervals is re-calculated.

The completion of this step denotes that one of the vertical lines in Figure 7 has been processed. The number of PCBs and confidence level for this vertical line signal are recorded.

The pseudo-code for this process is depicted as follows.

```
Function SearchBoundary(direction, start, end, avgheight,
level)
{
  Search possible area between start and end, tagged start' and
end'.
  length = end' – start'.
  if(region between start' and end' exists) return.
    if(length in PCB height range) return new PCB found.
  if( length / avgheight > ratio of 2 PCB board)
    {
      SearchBoundary(direction, start', end'-length/2, avgheight,
level+direction).
    SearchBoundary(direction, start'+length/2, end', avgheight,
level+direction).
    }
    if(length < ratio of one PCB board)
    {
      SearchBoundary(direction*-1, start'-length/2,
end'+length/2, level-direction).
    }
    return no PCB board in between start and end.
}
```

3.6 Multiple Sampling and Result Determination

Steps mentioned in section 3.3 through 3.5 are repeated until all the vertical line signals have been processed. The final result is determined by voting, in which the number of PCBs from each vertical line is taken into account, and the one with the most votes in addition to the highest level of confidence from this set of signals will be taken as the final result.

4. Implementation Result and Analysis

4.1 Correct Results

The proposed algorithm is able to produce correct results from all sorts of standard input images, as presented in Figure 11 which shows the execution results from 5 different input images. For example, in Figure 11(a), the position of the PCB stack is calculated every 100 pixels starting from the left, and each blue line segment represents one piece of PCB. Even though 1/3 of the right side of the captured image is not occupied by the image of the PCB stack, a correct result can still be drawn through the process of voting.

As for PCB with inclination, we have devised a way to find a correct result, as shown in Figure 11(b). In this figure, the PCB stack is not only tilted, but the thickness of the second PCB is evidently thinner than the rest as it is not classified as a piece of PCB, while the others are all identified correctly as PCBs. The first halves of the descriptions under Figure 11(a) to 11(e) stand for the thickness of each piece of PCB, where mil is the unit of length measurement (1 mil = 1/1000 inch); and the parts in parentheses represent the correct number of PCBs in each stack. For example, the description 9 mil (13p) in Figure 11(a) signifies that the thickness of the PCBs is 9 mils

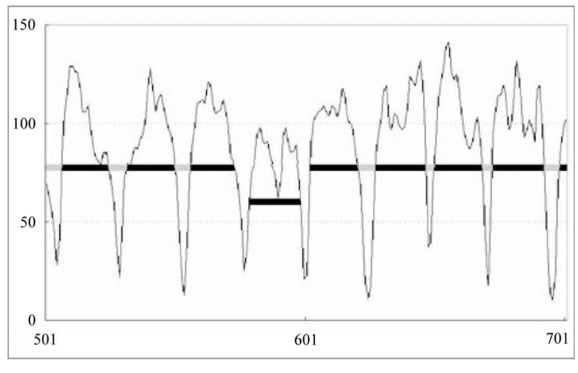


Figure 10. Result of regional horizontal line recursion search.

and there are 13 pieces in this stack.

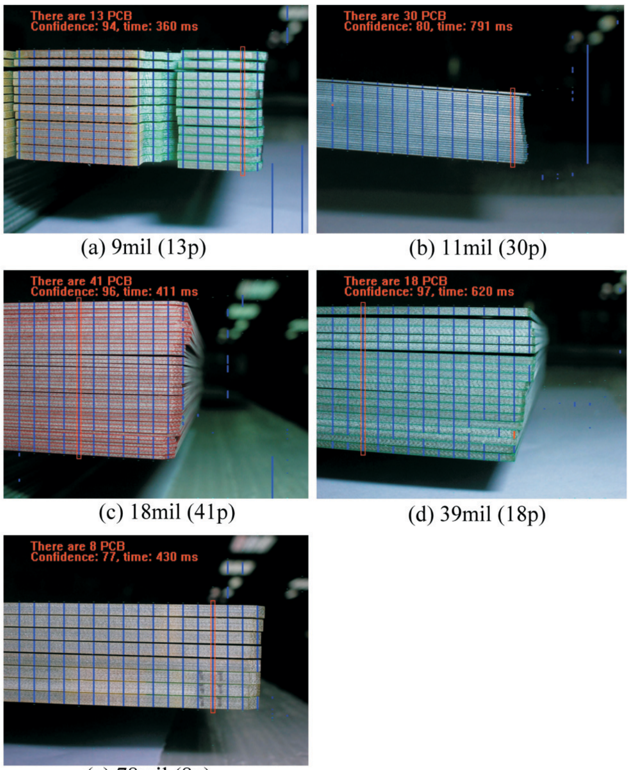
Figure 12 shows images taken from different angles of a PCB stack of the same number of boards (18 pieces) and of the same thickness (11 mils); the images are as follows: (a) the mid-section; (b) from the left; (c) from the right; (d) tilt to the left (5 degrees); (e) tilt to the left (10 degrees); (f) tilt to the left (15 degrees); (g) the entire front and (h) from the top.

Figure 13 shows the respective success rate of the calculation for images taken from different angles as presented in Figure 12. Success rate was 100% for images (a) to (e); 80% for image (f) due to its greater tilt at 15 degrees; 50% for image (g) because the image of the PCB stack was too small to be processed correctly; and 0% for image (h) since the top-down image captured the components on the PCB board in addition to the side view of the PCB stack. The results show that our algorithm can be applied successfully to conditions in images (a) to (e), i.e. under normal image capturing conditions, while conditions in images (f) to (h) may result in undesirable success rate.

4.2 False Analysis

In the database of the sampled images, a common problem is the reflection effect. Shown in Figure 14 where the reflection of PCBs appears in the sampled image, the global horizontal distinguishing level easily falls into the local minimum, which turns out to be a false result.

Our algorithm is designed to distinguish PCBs based on the consistent partition feature between the stacked PCBs, but there are still some samples that fail to meet this assumption. The right parts of Figure 15(a) are the section enlargements of the vertical sampling line of the image capture on the left. In Figure 15(b) (same axes as Figure 8), the black lines stand for gray level signals and the horizontal black line segments represent PCBs where the thickness of these line segments is the horizontal line intensity. A close inspection of Figure 15 tells us that

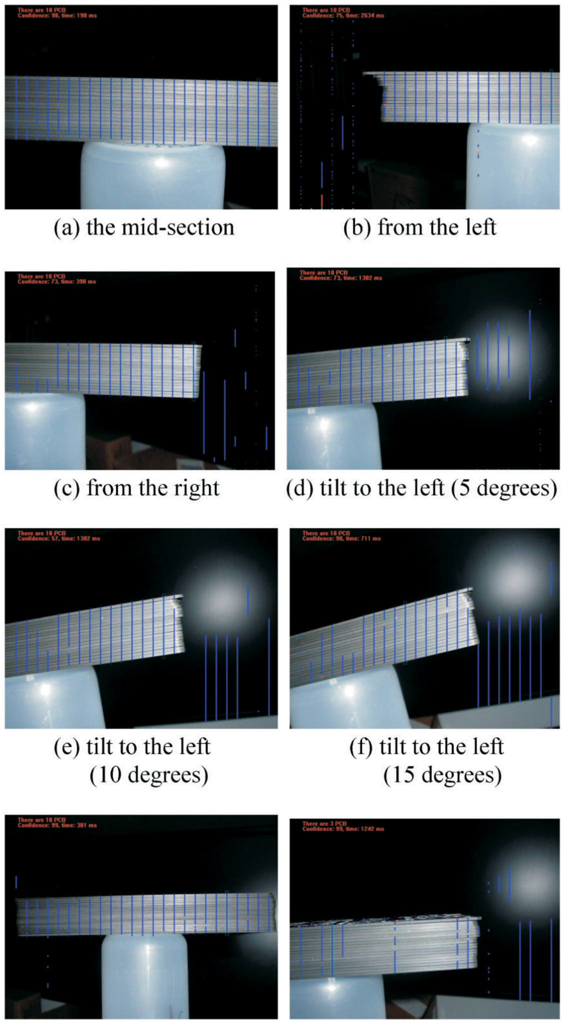


(a) 9mil (13p) (b) 11mil (30p)

(c) 18mil (41p) (d) 39mil (18p)

(e) 78mil (8p)

Figure 11. Results from normal processing.



(a) the mid-section (b) from the left

(c) from the right (d) tilt to the left (5 degrees)

(e) tilt to the left (10 degrees) (f) tilt to the left (15 degrees)

(g) the entire front (h) from the top

Figure 12. The images of the same sampled stack taken from different angles.

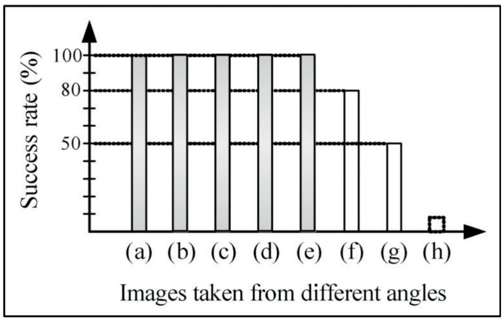


Figure 13. Success rate of the algorithm for different captured images.

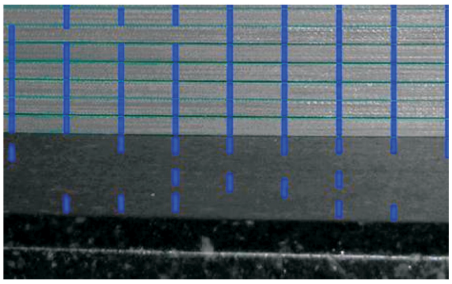
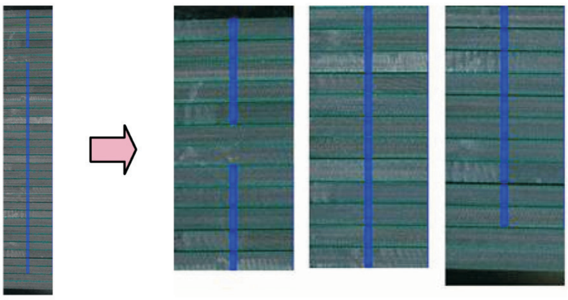
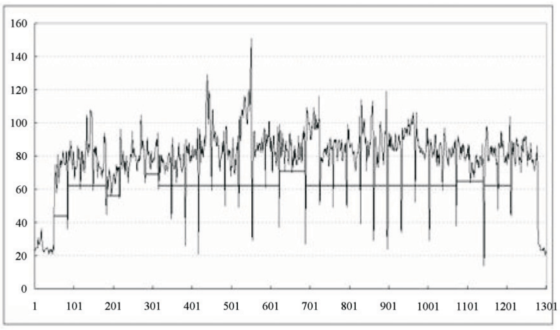


Figure 14. False PCB caused by reflection.



(a) The left part is the vertical sampling line of an image capture, where the right parts are the enlargements



(b) Result from indistinct partition feature in between PCBs

Figure 15. Result from indistinct partition feature in between PCBs.

there is no apparent partition feature between 200 and 300 as well as 1200 and 1300, so any adjustment of the horizontal level could not give us a correct result.

5. Conclusion

In this paper, we designed and carried out a visualization PCB counting system. The number of PCBs is calculated by giving a PCB picture with the same PCB specification on the moving conveyor belt using a digital camera. The system is highly effective, as it also quantifies the accuracy of the results, enabling human diagnosis in finding errors. In the future, a standard for input image will be drawn, while improving the overall effectiveness of our algorithm to minimize the effects brought by shadows and reflections. Since the complexity of our algorithm is rather simple, it could be employed in the embedded system or camera phones in practice.

Acknowledgement

The present work is partly supported by National Science Council Taiwan R.O.C. under the contract No. NSC 97-2511-S-236-001.

References

[1] Nan Ya PCB Corporation, http://www.nanyapcb.com.tw/, available online Setp. (2009).

[2] Compeq Manufacturing Co., LTD., http://www.compeq.com.tw/, available online Setp. (2009).

[3] Chang, C. Y., Xu, J. Y. and Cai, F. R., "The Applications of Position Monitoring with Enhanced Image Processing," *Journal of Cheng Shiu University,* Vol. 18, pp. 147–160 (2005).

[4] Windyga, P. S., "Fast Impulsive Noise Removal," *IEEE Transactions on Image Processing,* Vol. 10, pp. 173–179 (2001).

[5] Damil, N., Bakar, Z. A. and Sembok, T. M. T., "A Comparison of Noise Removal Techniques in Songket Motif Images," *Proc. of the International Conference on Computer Graphics, Imaging and Visualization,* pp. 139–143 (2004).

[6] Gonzalez, R. C. and Woods, R. E., *Digital Image Processing,* Prentice Hall, London (2002).

[7] Kuo, C. H., Wu, P. H. and Wang, T. S., "On the Con-

struction of a Color Model for Real-time Visual Moving Objects Segmentation," *The Tamkang Journal of Science and Engineering,* Vol. 8, pp. 87−98 (2005).

[8] Terada, K., Yoshida, D., Oe, S. and Yamaguchi, J., "A Method of Counting the Passing People by Using the Stereo Images," *International Conference on Image Processing,* Vol. 2, pp. 338−342 (1999).

[9] Tsuchikawa, M., Sato, A., Koike, H. and Tomono, A., "A Moving-Object Extraction Method Robust Against Illumination Level Changes for a Pedestrian Counting System," *International Symposium on Computer Vision,* pp. 563−568 (1995).

[10] Chen, T. H., "An Automatic Bi-Directional Passing-People Counting Method Based on Color Image Processing," *International Carnahan Conference on Security Technology,* pp. 200–207 (2003).

[11] Tremsin, A. S., Siegmund, O. H. W., Vallerga, J. V., Hull, J. S. and Abiad, R., "Cross Strip Readouts for Photon Counting Detectors with High Spatial and Temporal Resolution," *IEEE Transactions on nuclear science,* Vol. 51, pp. 1707−1711 (2004).

Manuscript Received: Nov. 28, 2007
Accepted: Jun. 30, 2009