

Received September 24, 2016, accepted November 1, 2016, date of publication November 22, 2016, date of current version June 27, 2017.

Digital Object Identifier 10.1109/ACCESS.2016.2631658

A Machine Vision Based Automatic Optical Inspection System for Measuring Drilling Quality of Printed Circuit Boards

WEI-CHIEN WANG¹, SHANG-LIANG CHEN², LIANG-BI CHEN³, (Senior Member, IEEE), AND WAN-JUNG CHANG³, (Member, IEEE)

¹School of Engineering and Information Technology, Federation University Australia, Gippsland campus, Churchill, VIC 3842, Australia

²Institute of Manufacturing Information and Systems, National Cheng Kung University, Tainan 70101, Taiwan

³Department of Electronic Engineering, Southern Taiwan University of Science and Technology, Tainan 71005, Taiwan

Corresponding author: Wan-Jung Chang (allenchang@stust.edu.tw)

This work was supported in part by the Ministry of Science and Technology, Taiwan, under Grant MOST-106-2632-E-2018-002 and Grant MOST-106-2622-8-218-003-TE2.

ABSTRACT In this paper, we develop and put into practice an automatic optical inspection (AOI) system based on machine vision to check the holes on a printed circuit board (PCB). We incorporate the hardware and software. For the hardware part, we combine a PC, the three-axis positioning system, a lighting device, and charge-coupled device cameras. For the software part, we utilize image registration, image segmentation, drill numbering, drill contrast, and defect displays to achieve this system. Results indicated that an accuracy of 5 μm could be achieved in errors of the PCB holes allowing comparisons to be made. This is significant in inspecting the missing, the multi-hole, and the incorrect location of the holes. However, previous work only focuses on one or other feature of the holes. Our research is able to assess multiple features: missing holes, incorrectly located holes, and excessive holes. Equally, our results could be displayed as a bar chart and target plot. This has not been achieved before. These displays help users to analyze the causes of errors and immediately correct the problems. In addition, this AOI system is valuable for checking a large number of holes and finding out the defective ones on a PCB. Meanwhile, we apply a 0.1-mm image resolution, which is better than others used in industry. We set a detecting standard based on 2-mm diameter of circles to diagnose the quality of the holes within 10 s.

INDEX TERMS Automatic optical inspection (AOI) system, drill inspection, drilling technique, printed circuits board (PCB), machine vision.

I. INTRODUCTION

Electronic devices have been made small to meet the requirement of the market. Under the Market-Driven theory, micro devices with smaller and lighter electronic components should be designed lighter, thinner, shorter, and smaller. To produce multichip modules, the design of the outlet of wiring and components on a PCB would be crowded and complicated. Thousands of thousands of subtle holes need to be drilled only on a single PCB. So various conductors can be plugged in. After drilling, the geometric features and precision of the location of the holes will affect the placement other electronic components.

Besides, multi-layer within PCB, including inner and outer layers, usually have four types of holes, as shown in Fig. 1. Their processing conditions are different according to their characteristics, which are described as follows:

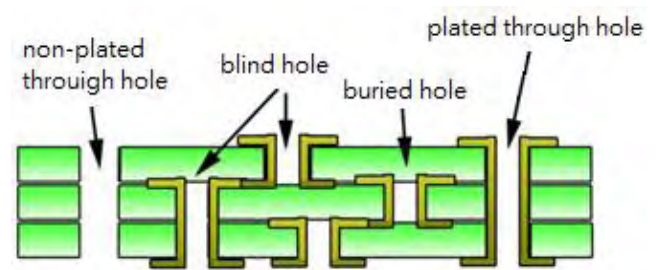


FIGURE 1. Four types of via: a non-plated through hole, a blind hole plated, a buried hole and a plated through hole.

1) *Buried Via Hole or Buried Hole*: It is defined as local via which focus on the connection of wiring between multiple inner layers instead of outer layers. A buried via cannot

be seen from the outside of a PCB because it only allows conductors to make contact with adjacent few inner layers not through all layers. This method is packing density, which means it does not need much board space but has lots of functionality.

2) *Blind Via Hole or Blind Hole*: The hole is made for connecting outer layers with one or many inner layers. For example, the first/outer layer connects with the following layers or the end of layers have a connection with previous layers.

3) *Via Hole or Plated Through Hole (PTH)*: The via that refers to making interconnections in every layer is drilled with copper plating through the whole board from the first layer to the end layer.

4) *Non-Plated Through Hole (NPTH)*: It is a hole which is similar with PTH and penetrates the entire board but without copper coated along it. The aperture of NPTH is usually bigger than PTH because it has no copper plating along the hole. Such as a screw hole.

It is tough for the traditional drilling technology to meet the needs of the market. Laser drilling has been widely used in order to drill smaller holes and the increasing number of holes on a PCB. There are two main types of lasers according to their processing approaches. The first is an infrared laser and the second is an ultraviolet laser. The infrared laser is also called thermal processing because its processing is to remove material by heating the surface of material until evaporation. There are two wavelengths based on the infrared laser, which are $10.6\mu\text{m}$ (Co2 laser) and 1.064 (Nd:YAG) separately. The ultraviolet laser breaks the molecular bond of material to separate the molecules from the body. It is also called cold working because it does not create high temperature during the processing stage. This kind of method includes a UV-YAG laser and an Excimer laser. Both lasers drill quickly and accurately. Currently, drilling on electronic components has been widely used, for example, filter plates with many tiny holes, medical vias and drilling on embedded PCBs. Hence, it is necessary to find a way of inspecting holes faster and efficiently.

Many traditional inspection methods have been replaced by automatic and contactless approaches. A PCB, for example, is commonly drilled in various size of holes. Hence, the precision of geometric dimensions and location of holes will significantly affect how IC or conductors are assembled. On the other hand, micro via on PCB need to be created as small as possible due to the number of them is huge. Designing of a faster and more precise inspection system to meet the requirement of production and quality in manufacturing is important. Automatic Optical Inspection (AOI) plays a significant role in manufacturing and processing in industry for checking defects on different kinds of electronic devices or products [1]–[3].

AOI technology is a key way of using precision automation machine vision to gradually replace manual visual inspection [4]. It has been demonstrated that AOI not only can improve the quality and productivity but also

increase competitiveness. Because of fast, accurate and low-cost AOI systems have been widely used in the inspection of a PCB.

AOI technology focuses mainly on, through machine vision, guiding the samples into inspection platform by analysing images captured from camera, moving the machine to the many sets of the coordinate. For the inspection of PCBs, we can check the images of the shape of the wiring and electronic components to find out the flaws. Then, we also can assess the correction of the location of each electronic components according to the relative positions. It is prominently important to find out the defect before completing the entire process which can increase the quality of product and reduce the costs. Except for the electrical test, appearance and surface defect are essential inspection items on PCBs. AOI not only can be used to control or manage the quality of products, but also even assist in monitoring performance. Therefore, if AOI system could find the problems and take repairing strategies as early as possible, this system will increase the productivity and reduce the wasting source.

Thus, we propose an automatic investigation system (AI) which is for drilling investigation in the PCB industry. This AI is developed based on an automatically optical investigation system (AOI) by integrating computer vision and automation. Besides, its specification is based on 2 mm borehole size, which provides competitive image processing ability in the market. Consequently, both the efficiency and quality of such investigation increase significantly, proving that it is viable to put it into practice.

The remainder of this paper is organized as follows: Section II of this paper describes the previous works on the design and development of AOI systems. The research framework and theory of this work are provided in section III. We present a description of experimental framework and procedure in section IV. Section V shows the result of inspection and discussed the related parameters, and following this, section VI presents our conclusions and future works.

II. RELATED WORKS

This work aims to identify and analyze the geometric features of defective the holes on the PCB by utilizing the AOI system. Several examples of research related to optical inspection on the PCB are described as follows:

Jackson and Kern's [5] remarks illustrated that it was extraordinarily important to examine the alignment of the aperture between inner and outer layers when creating multi-layer circuit boards. So, they proposed a microscope system which could measure the size of the board from 24in to 28in and the board thickness of the board from 0.1in to 0.3in. Furthermore, an aperture less than 0.001in can also be detected. The drawback of this method was that it only identified the aperture in one direction.

Manirambona et al. [6] used Resin Coated Copper (RCC) as the base material for an excimer laser on the multichip model. He mainly focused on using a UV laser to produce micro-holes (diameter $< 100\mu\text{m}$) of high-density arrays

on RCC. Simultaneously, he proposed the mask imaging method and conformal mask method. Lastly, he proofed that the depth of aperture was easily controlled when an excimer laser was etching on RCC based boards.

Lee and Lin [7] summed up the view that the several reasons may create obscure images, such as insufficient light, noises, and distortion images when taking a vertical picture from Charge-coupled Device (CCD). Thus, it might cause the error of extracting of the location of the holes. Mode method had been adopted to calculate the centers of the holes by calculating four reference points. Their method successfully overcame the problem of blurry images caused by reflection of light from the surface of the holes. They proved that it was viable through simulation as well as a practical test. Moreover, they could correctly measure the coordinates of the centers even if these centers were masked over 60 degrees.

Rau and Wu [8] were likeminded in their approach to focus mainly on flaw detection on inner layers on PCBs. They presented five testing stages: image rebuilding, inspection image normalization, image subtraction, defect classification, and separation. In the image subtraction stage, deficient parts could be isolated from the reference and the testing images. Following this was to segment the defective parts by using defect edge tracking method. Finally, through an outer defect boundary tracing method, they classified eight error types which were the open circuit, the short circuit, the missing hole and others. This method could not be used for further analyzing of other geometric features.

An algorithm was applied for assessment of incorrect drillings on the PCB by Opera *et al.* [9]. They utilized image processing technology but not source images as input. Taking MATLAB as the development environment was to extract the incorrect location of drilling resulted from the subtraction between the source images and the testing images. The testing images were obtained after using the grey level method, and thread method, and filling the right location parameters was used.

Huang *et al.* [10] proposed a virtual AOI system to inspect the PCB. The authors combined LABVIEW with image program library IMAQ to incorporate software and hardware parts. This system could recognize rotation, the inclination of components and missing components. Moreover, their system could improve the speed and accuracy of inspection. Most importantly, the detecting results were saved automatically at the end of the process. By comparison with traditional visual inspection machines, their system was more flexible to expand and keep the cost down.

Shetty *et al.* [11] detected the depth of blind via holes on turbine blades by lighting helium-neon laser on the surface of turbine blades. By moving the XZY framework, laser spots on blind via holes were reflected in the inspection system. Then, they calculated the depth of blind holes according to the information of this reflection and the geomantic mathematic model. The depth of examination of the range was from $5\mu\text{m}$ to $500\mu\text{m}$. One of the disadvantages of this approach

was it only could detect the depth of blind holes but no other diameters.

Ibrahim *et al.* [12] provided a controllable positive and negative image method to eliminate noise from image subtraction. Due to unbalance distribution and maladjustment of image binarization/threshold, there were many unnecessary noises. The authors chose a value as a threshold to remove extraneous noises which were produced from the result of subtraction of positive and negative images. Lastly, they performed the logical exclusive-or operation on those images to get the final result. Their advanced alignment equipment was too expensive to afford for many people or companies. Besides, their approach could only measure for the small or middle size of PCBs.

Wang *et al.* [13] proposed a method which took advantages of combination the global and local approaches. The global approach was to identify the possible cracking holes through the entire images. However, for the local approach, they utilized information of the holes, such as Hessian and shape, to detect the potential crack holes in a limited area. They demonstrated their method was efficiency. Nevertheless, their approach still had to be improved by enhancing the ability to extract the ridge information, increasing pre-processing steps to remove noises, and detecting cracks on the non-flat surface.

Huang *et al.* [14]'s observations underlined the need for X-ray imaging in the electronic industry. They took advantages of X-ray imaging techniques to detect defects on the PTH. X-ray computer tomography (CT) allows researchers to look at the internal structures and defects without destructing the PCB. Moreover, Wang [15] mentioned that this technology only needed few samples. They combined the information of both 2D and 3D X-ray inspection. A large number of 2D X-ray images were processed for reconstruction by 3D. So, unfortunately, processing those lots of high-resolution 2D images not only is very time-consuming but also requires a lot of computer power.

Besides, the inspection of the holes on PCBs, there are many kinds of research on IC solder joints on PCBs. Barnich and Van Droogenbroeck [16] and Van Droogenbroeck and Barnich [17] applied visual background extraction (ViBe) algorithm to detect objects. Gregory [18] improved this method to fit for the defect inspection. At first, they used a series of standard solder joint images to train six templates as their model. Then, they found out the difference between the input images and the templates. Lastly, they created an assessment to judge the qualities of the solder joints. Their experimental results demonstrated their method was more accurate than traditional approaches.

Loh and Lu [25] developed a computer-vision-based inspection system which adopted structured lighting for solder joint inspection on PCB. This work can determine good, bad, and bridged for joint solders. Wu and Zhang [26] also developed an AOI algorithm for solder joints on PCBs, which was divided into some sub-regions as shape characteristics and its relative digital characteristics were extracted. As their

experimental results, 137 PCBs with defects had inspected correctly by using this AOI algorithm.

Li and Yang [22] designed an AOI-based system for detecting PCB defects, including open circuit, short circuit, wire gaps, voids, scratch defects, etc. Moreover, this work achieved $15\mu\text{m}$ resolution with over 95% detection success rate. Wu *et al.* [27] presented a 2-stage (Bayesian and SVM) classifier with characteristics selection approach for inspecting solder joints. Experimental results showed that this work effectively increased the recognition rate and reduced the number of needed extracted characteristics. Mak *et al.* [28] proposed a Bayesian-based AOI expert system for solder jet ball in the head gimbal assembly process, which was consisted of a three augmented Naïve Bayes network (TAN-BN) probabilistic classifier, a check classifier, and an image processing technique. Their proposed system can achieve 91.52% accuracy. Furthermore, Long *et al.* [29] proposed an AOI system for inspecting fixture probes and drilling holes. This work also further analyzed the quality of pin contacts and drilling holes by x/y deviated positions.

In summary, there are few studies in the inspection of the holes on PCBs using the AOI system. Nevertheless, their methods only could measure one or two dimensions of the holes. We believe there are still many spaces for developing AOI system to check flaw holes.

Hence, we consider the increasing needs of an AOI in manufacturing and processing industry and propose an AOI system based on machine vision to detect the holes on PCBs. We implement an AOI system by using image registration, geometrical image transformation, labelling and matching methods.

Firstly, after applying pre-processing method on extracted images, we calculate the centers, radius, round and other diameters of the holes. Subsequently, we obtain the difference between the standard and testing images by comparing both of them. Finally, we are able to precisely find out the flaw drillings by analysing and showing the target plots and bar charts, as a fundament for improving the quality of drillings.

III. THE THEORIES, METHODOLOGIES, AND ITS DESIGN FLOW OF THE PROPOSED AOI SYSTEM

We develop an automated optical inspection system which integrates the lighting device, CCD cameras and linear motors. Then, we use C# to create a user interface (UI) to detect the area and location of the holes on PCB. Flow chart of automated optical inspection system is shown in Fig. 2. First of all, we control a CCD by a computer to move to the desired position. The CCD is moved from the upper left to the bottom right to take the 144 sub-images. The image registration method is to merge all sub-images to get an entire image. The CCD continually takes each sub-image on the PCB from the most left corner to the most right corner on the first row (Column sub-image grabbed). Then, the CCD returns to the most left corner on the second row and starts shooting pictures until the end of the second row. The CCD grabs sub-images by repeating this approach from the first row to the final row

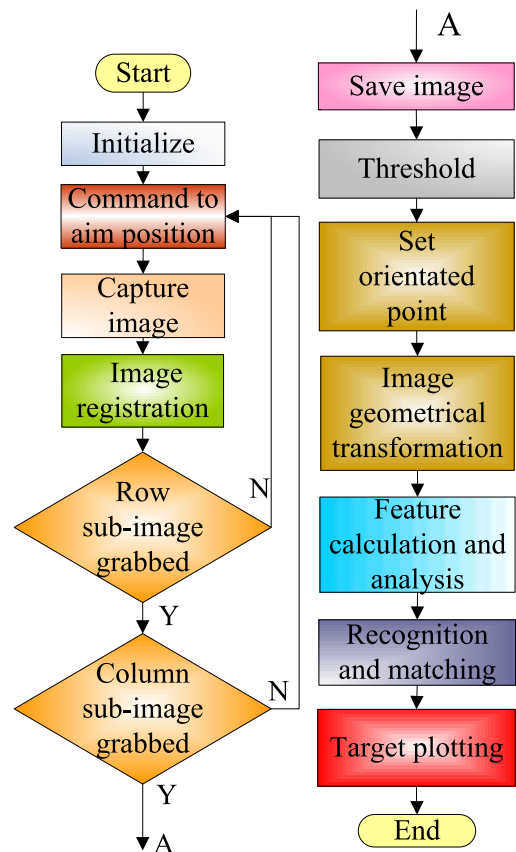


FIGURE 2. Flow chart of automated optical inspection system.

(Row sub-image grabbed). Then, we use threshold method to get rid of noises. After setting oriented points on the grabbed images, we calculate geometrical transformation of the images and features of the holes. Additionally, we classify those defect drillings into the multi-hole, the missing holes, and the aperture error by using recognition and matching methods. Finally, target plot is showed the statistics of errors of coordinates of the holes.

A. IMAGE REGISTRATION

Image registration technology is to merge two or more adjacent sub-images into a large field of vision of the image. The program gives each image a sequential number according to the movement of the CCD. These number can be sequenced to combine sub-images. The sub image that CCD takes at the most left corner of the PCB is combined with the sub image that CCD takes when it moves 10mm on the x-axis. We overlap each sub images from the most left to the most right on the every row. The two overlapping areas of adjoining sub-images, such as (i, j) and $(i+1, j)$ or (i, j) and $(i, j+1)$, are $10 \times 37\text{mm}^2$ and $10 \times 47.5\text{mm}^2$, respectively. Where $(i+1, j)$ and $(i, j+1)$ are the two sub-images which are closest to sub-image (i, j) , as shown in Fig. 3.

B. THRESHOLD

The thread method is to choose a value to isolate the gray level on the grayscale image. By way of illustration, thread

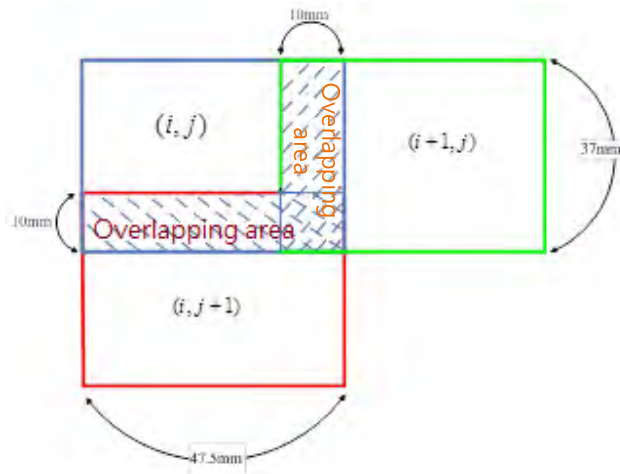


FIGURE 3. Two overlapping areas between adjacent sub-images.

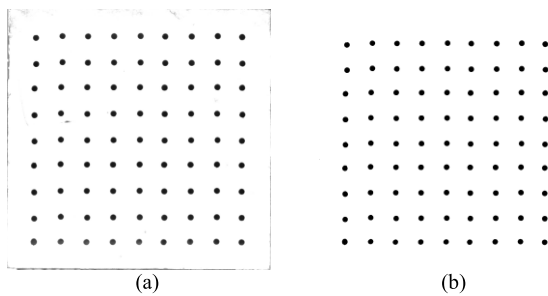


FIGURE 4. The result of before (a) and after (b) thread method.

method/Image binarization is defined as following:

$$g(x, y) = \begin{cases} 1 & f(x, y) > T \\ 0 & f(x, y) \leq T \end{cases} \quad (1)$$

where $f(x, y)$, $g(x, y)$ and T are the original image, binary image and threshold respectively. For instance, the $g(x, y)$ is 0 if a gray level $f(x, y)$ of a pixel on the image is less than a threshold value T . However, if a gray level $f(x, y)$ of a pixel is large than T , we obtain a $g(x, y)$ that is equal to 1. The thread method can highlight the objects that we desire on the images. Furthermore, the other advantages are easy saving and quick processing. In this work, after taking several experiments, we found a threshold of 128 was the appropriate value to separate the holes and the background on the PCB. Fig. 4(b) shows the result of before (a) and after (b) thread method.

C. SET ORIENTATED POINT

Oriented points have to be set before capturing the sub-images. The purpose of positioning is to obtain the transformation relation, such as distance and angle, between the image coordinates and the PCB. Moreover, the images are calibrated according to these transformations. There are three types of orientation model: horizontal alignment, vertical alignment, and three-point positioning.

In this paper, we apply three-point positioning measurement as the preliminary correction. As can be seen in Fig. 5,

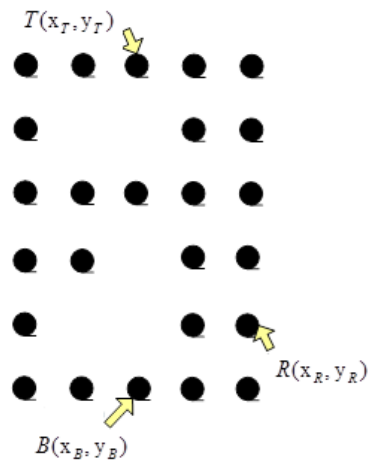


FIGURE 5. Three orientated points.

this method transform the image by taking arbitrarily three non-collinear points, namely top $T(X_T, Y_T)$, bottom $B(X_B, Y_B)$ and reference $R(X_R, Y_R)$ points from an image.

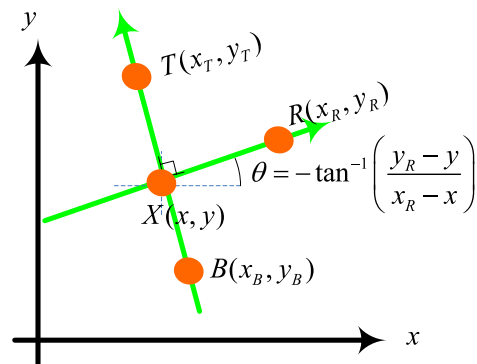


FIGURE 6. Transformation of three-point positioning.

Relating to finding the best three points are to calculate the greater distance between them and good roundness of each of them. The transformation formula related to this work can be derived from $T(X_T, Y_T)$, $B(X_B, Y_B)$, and $R(X_R, Y_R)$, and are shown in Fig. 6.

Where S_1 is the slope of point $T(X_T, Y_T)$ and point $B(X_B, Y_B)$, θ is angle of rotation, $X(x, y)$ is a perpendicular bisector, and $R(X_R, Y_R)$ is a point along the perpendicular line of \overline{TB} .

To calculate the perpendicular bisector $X(x, y)$, first of all, we use the perpendicular equation:

$$(x_T - x_B, y_T - y_B) \bullet (x - x_R, y - y_R) = 0 \quad (2)$$

And point-slope form:

$$\frac{y - y_B}{x - x_B} = \frac{y_T - y_B}{x_T - x_B} = S_1 \quad (3)$$

$$(x_T - x_B)(x - x_R) + (y_T - y_B)(y - y_R) = 0 \quad (4)$$

$$x_T x - x x_B - x_R x_T + x_B x_R + y_T y - y y_B - y_T y_R + y_B y_R = 0 \quad (5)$$

$$x(x_T - x_B) - x_R(x_T - x_B) - y(y_B - y_T) - y_R(y_T - y_B) = 0 \quad (6)$$

$$x(x_T - x_B) - x_R(x_T - x_B) - y_R(y_T - y_B) = y(y_B - y_T) \quad (7)$$

$$y = x \frac{(x_T - x_B)}{(y_B - y_T)} - x_R \frac{(x_T - x_B)}{(y_B - y_T)} - y_R \frac{(y_T - y_B)}{(y_B - y_T)} \quad (8)$$

$$S_1 = \frac{y_T - y_B}{x_T - x_B} \quad (9)$$

$$-\frac{1}{S_1} = \frac{(x_T - x_B)}{(y_B - y_T)} \quad (10)$$

According to Eq. (9) and (10), we obtain two other equations of (11) and (12):

$$y = s_1 x - s_1 x_B + y_B \quad (11)$$

$$y = -\frac{1}{s_1} x - \frac{1}{s_1} x_R + y_R \quad (12)$$

$$\left(S_1 + \frac{1}{S_1}\right)x = S_1 x_B + \frac{1}{S_1} x_R - y_R - y_B \quad (13)$$

We work out the coordinate of perpendicular bisector:

$$x = \frac{S_1 x_B + \frac{1}{S_1} x_R + y_R - y_B}{\left(S_1 + \frac{1}{S_1}\right)} \quad (14)$$

$$y = S_1 \frac{S_1 x_B + \frac{1}{S_1} x_R + y_R - y_B}{\left(S_1 + \frac{1}{S_1}\right)} - S_1 x_B + y_B \quad (15)$$

And angle of rotation is:

$$\theta = -\tan^{-1} \left(\frac{y_R - y}{x_R - x} \right) \quad (16)$$

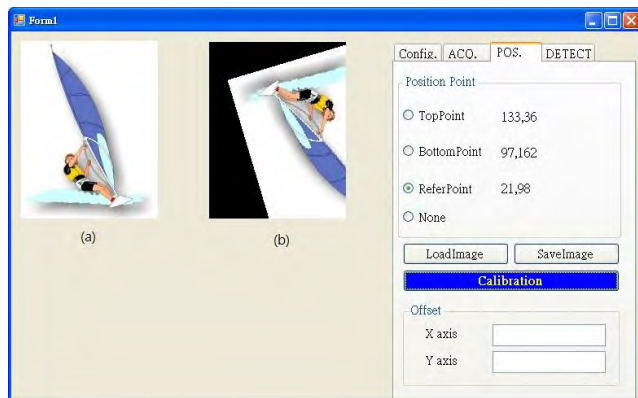


FIGURE 7. Result of before (a) and after (b) transformation.

Fig. 7 shows the example when we set $T(X_T, Y_T)$ is (133, 36), $B(X_B, Y_B)$ is (97, 162) and $R(X_R, Y_R)$ is (21, 98).

D. GEOMETRIC TRANSFORMATION OF IMAGES

Geometric transformation is to calculate the projection of each pixel in the images onto another space. This method mainly focuses on image restoration especially when the images are geometric distortions. This technology also has

been applied for the overlapping positions, the aspect ratio of correction of the images, rotation, zoom mirror and distortion of the images. Geometric transformation is usually divided into two types: the first category is resampling, which is to transform the space coordinates (x, y) of original images into new image space coordinates (x', y') . However, this new coordinate of pixel that is matched to the initial one usually does not locate correctly at a pixel. We should utilize gray-level interpolation to calculate the correct of (x, y) . The other type is space transformation, and the main idea is shown in (17). Space transformation refers to rearranging pixels of images. There are three kinds of space transformation: 1) scaling method; 2) rotating method; 3) translation method.

$$I(x, y) \rightarrow O(x', y') \quad (17)$$

1) SCALING METHOD

When image processing, the size of an image is too small to detect or too large for the window we have. Hence, we have to take enlargement and reduction of the images, and the equations are shown in (18) and (19), separately. Where x and y are the coordinate of the pixel of the original image. X'' and y'' are the coordinate of the pixel after doing scaling method. s_x and s_y are the scaling factors on the x-axis and y-axis respectively.

$$x'' = S_x x \quad (18)$$

$$y'' = S_y y \quad (19)$$

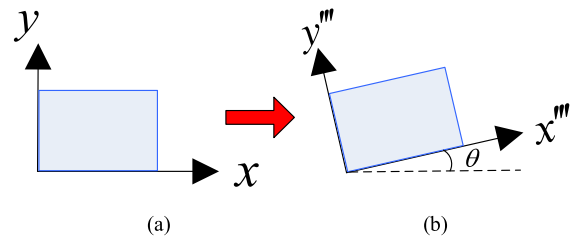


FIGURE 8. Result of before (a) and after (b) transformation.

2) ROTATING METHOD

When matching two images, we have to calculate the relative position and angle of these two images. It is common to combine translation and rotating methods to match images. Rotating method as shown in Fig. 8, we can get a new position (x''', y''') of each pixel by using (20) and (21) with the center (X_c, Y_c) of each image. After rotation, we have to add gray-scale interpolation into the pixel of the original image to acquire the new pixel value if this new pixel is not an integer. Where x''' and y''' are the new coordinates after the rotating method.

$$x''' = x_c + (x - x_c) \cos \theta + (y - y_c) \sin \theta \quad (20)$$

$$y''' = y_c + (y - y_c) \cos \theta - (x - x_c) \sin \theta \quad (21)$$

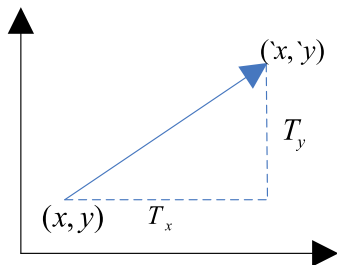


FIGURE 9. Translation method.

3) TRANSLATION METHOD

We can move images to move left, right, up, or down by using translation method which is acquired from (22) and (23). Translation method is shown in Fig. 9. Where x , and y are the coordinates of the pixel of the original image. x' and y' are the coordinates of the pixel after doing translation method. T_x and T_y are the amount of horizontal and vertical movement separately.

$$x' = x + T_x \tag{22}$$

$$y' = y + T_y \tag{23}$$

Three methods can be utilized at the same time when processing images. If we have to use both translation and rotation methods we do translation method first and then rotation method. Our work combines translation, rotation, and translation methods, and the matrix operations are show

on from (24) to (30). Where $\begin{bmatrix} x_{\Delta} \\ y_{\Delta} \\ z_{\Delta} \\ 1 \end{bmatrix}$ is the coordinate of

original image and $\begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{bmatrix}$ is the coordinate after translation,

rotation, and translation methods.

Result:

$$\begin{bmatrix} x_{\Delta} \\ y_{\Delta} \\ z_{\Delta} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & -x \\ 0 & 1 & 0 & -y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{bmatrix} \tag{24}$$

Multiply both sides of the equals sign by

$$\begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 & 0 & -x \\ 0 & 1 & 0 & -y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Result:

$$\begin{bmatrix} 1 & 0 & 0 & -x \\ 0 & 1 & 0 & -y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{\Delta} \\ y_{\Delta} \\ z_{\Delta} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -x \\ 0 & 1 & 0 & -y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{bmatrix} \tag{25}$$

And then multiply both sides of the equals sign by

$$\begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Result:

$$\begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -x \\ 0 & 1 & 0 & -y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{\Delta} \\ y_{\Delta} \\ z_{\Delta} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -x \\ 0 & 1 & 0 & -y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{bmatrix} \tag{26}$$

Finally, multiply both sides of the equals sign by

$$\begin{bmatrix} 1 & 0 & 0 & -x \\ 0 & 1 & 0 & -y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Result:

$$\begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & -x \\ 0 & 1 & 0 & -y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{\Delta} \\ y_{\Delta} \\ z_{\Delta} \\ 1 \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{bmatrix} \tag{27}$$

Result in understandable form:

$$\begin{bmatrix} \cos \theta & -\sin \theta & 0 & -x \cos \theta + y \sin \theta + x \\ \sin \theta & \cos \theta & 0 & -x \sin \theta - y \cos \theta + y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{\Delta} \\ y_{\Delta} \\ z_{\Delta} \\ 1 \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{bmatrix} \tag{28}$$

$$x_0 = \cos \theta x_{\Delta} - \sin \theta y_{\Delta} - x \cos \theta + y \sin \theta + x \tag{29}$$

$$y_0 = \sin \theta x_{\Delta} + \cos \theta y_{\Delta} - x \sin \theta - y \cos \theta + y \tag{30}$$

where (x_0, y_0) is the new coordinate of the pixel related to the original pixel after translation, rotation, and translation methods.

E. FEATURE CALCULATION AND ANALYSIS

There are several basic dimensions of measurements, such as center, radius, angle and width of the holes. We can calculate more other complex dimensions based on those core dimensions. I discuss how we calculate the 1) center; 2) radius; and 3) round, as follows:

1) COORDINATES OF A CENTER (x_{center}, y_{center})

We label the same number on every pixel within a hole, which means the pixels inside a hole are labelled the same number. An area of a hole is acquired by summing all pixels that are marked the same number. For an ideal hole, the coordinate of a center is an average of this area. The formulas are shown in (31) and (32).

$$x_{center} = \frac{\sum_{i=1}^n (x_i)}{n} \quad i = 1, 2, 3, \dots, n \quad (31)$$

$$y_{center} = \frac{\sum_{i=1}^n (y_i)}{n} \quad i = 1, 2, 3, \dots, n \quad (32)$$

2) THE AREA OF A HOLE (x_{area}, y_{area})

The area of a hole can be calculated by the sum of all pixels in a hole. The equations (33) and (34) show the standard calculation of the area of a circle.

$$x_{area} = \sum_{i=1}^n (x_i) \quad i = 1, 2, 3, \dots, n \quad (33)$$

$$y_{area} = \sum_{i=1}^n (y_i) \quad i = 1, 2, 3, \dots, n \quad (34)$$

3) ROUNDNESS (ROUND)

Roundness is a normal judgment for checking the quality of the holes. First, we obtain the coordinate of a center through (31) and (32). Then, we calculate the eight boundary points of this center in the coordinates of 0 degree, 45 degrees, 90 degrees, 135 degrees, 180 degrees, 225 degrees, 270 degrees, and 315 degrees. Using (35), shown at the bottom of this page, we calculate the distances, as the radii, between the center and the coordinates of 0 degree, 90 degrees, 180 degrees, and 270 degrees. Similarly, with (36), also shown at the bottom of this page, we obtain the distances, as the radii, between the center and

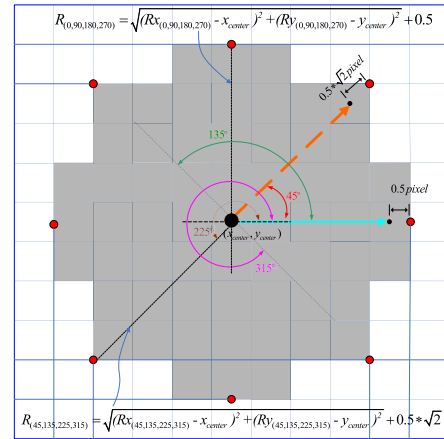


FIGURE 10. Radius compensation of eight boundary points.

coordinates of 45 degrees, 135 degrees, 225 degrees, and 315 degrees. Following this, we add the value of 0.5 and $0.5 * \sqrt{2}$ into (35) and (36), separately, to make reparation when an endpoint is not precisely located at one pixel. Radius compensation of eight boundary points is shown in Fig. 10. Where $R_{(0, 90, 180, 270)}$ are the radii in the coordinates of 0 degree, 90 degrees, 180 degrees, and 270 degrees. $R_{(45, 135, 225, 315)}$ are the radii in the coordinates of 45 degrees, 135 degrees, 225 degrees, and 315 degrees.

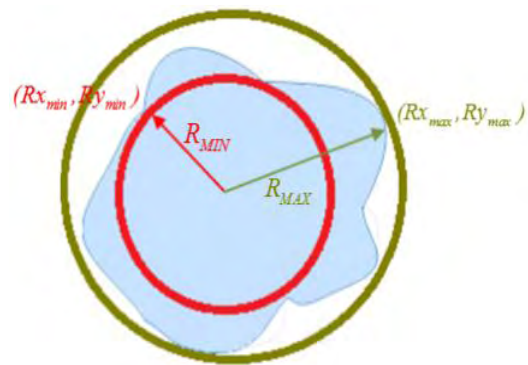


FIGURE 11. The farthest and the nearest points for calculating roundness.

We acquire the largest distance R_{MAX} by calculating the difference of the coordinate of the center of the hole and the farthest point $(R_{X_{max}}, R_{Y_{max}})$. In the same way, we require the smallest distance R_{MIN} by calculating the difference between the coordinate of the center of the hole and the nearest point $(R_{X_{min}}, R_{Y_{min}})$, as shown in Fig. 11.

$$R_{(0,90,180,270)} = \sqrt{(Rx_{(0,90,180,270)} - x_{center})^2 + (Ry_{(0,90,180,270)} - y_{center})^2 + 0.5} \quad (35)$$

$$R_{(45,135,225,315)} = \sqrt{(Rx_{(45,135,225,315)} - x_{center})^2 + (Ry_{(45,135,225,315)} - y_{center})^2 + 0.5 * \sqrt{2}} \quad (36)$$

$$round = \sqrt{(R_{x_{man}} - x_{center})^2 + (R_{y_{max}} - y_{center})^2} - \sqrt{(R_{x_{min}} - x_{center})^2 + (R_{y_{min}} - y_{center})^2} \quad (37)$$

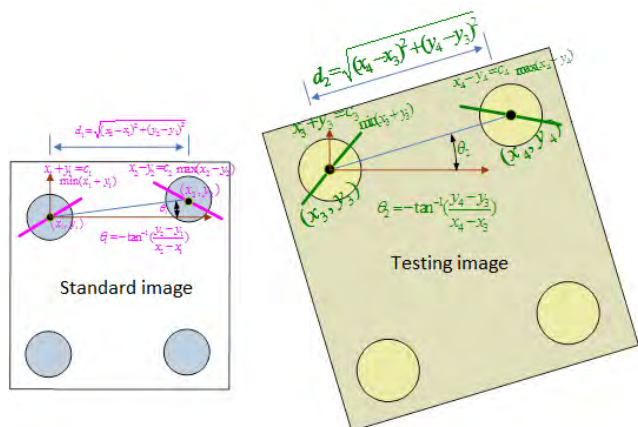


FIGURE 12. Calculation of the ratio between the standard and the testing images.

F. RECOGNITION AND MATCHING

Before comparing or matching, we have to calculate the ratio and angle of rotation between the testing and the standard images and then modify them until they are in the same direction and at the same angle. In this work, we set the centers of the holes in the most upper left and most upper right as two datum points for both testing and standard images. In other words, we make the smallest x-coordinate and y-coordinate of the center of the hole as the most upper left datum point. For the most upper right datum point, we set the largest x-coordinate but smallest y-coordinate. The ratio between the testing and standard images is calculated by d_1 and d_2 , as shown in Fig. 12. d_1 is the distance between two datum points on the standard image. d_2 is the distance between two datum points on the testing image.

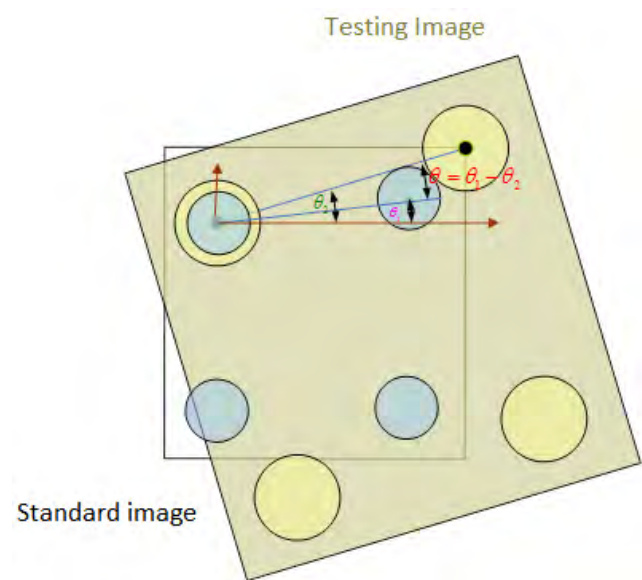


FIGURE 13. Calculation of the angle of rotation between the standard and the testing images.

The angle θ of rotation of the testing image can be obtained from $\theta_1 - \theta_2$, as shown in Figs. 12 and 13. θ_1 and θ_2 are

the correction angles for both testing and standard images, respectively.

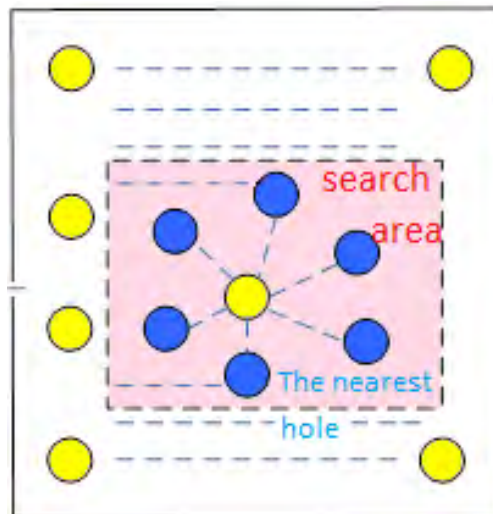


FIGURE 14. Search area on a testing image.

Following this, we match the standard images with the testing images that are calibrated. We first use a coordinate of the center of a hole on the standard images to find the corresponding coordinate on the testing image. Based on this coordinate, we search for the nearest center of a hole within the particular range on the testing images. The searching range should be less than the distance between two centers of the holes on the standard images. As shown in Fig. 14, there are six centers of the holes we found within the particular range. We record and compare those six coordinates to find out the nearest coordinate as the center of the testing image.

TABLE 1. The judgment of three types of holes.

A standard image (Black color)	A testing image (Red color)	Types
Yes	Yes	Correction (Overlapping black w/ red)
Yes	No	Missing hole (Green)
No	Yes	Multi-holes (Blue)

After searching all holes on the testing images, we classify them as the missing holes if there are holes on the standard images but not on the testing image. However, if there is not any hole on the standard image but are holes on the testing images we define them as multi-hole, as shown in Table 1.

G. TARGET PLOT

We can analyze the accuracy and precision of the holes from the target plot. In this work, the target plot shows, as in Fig. 15, the distribution of displacement error of the holes. Three sub functions were included:

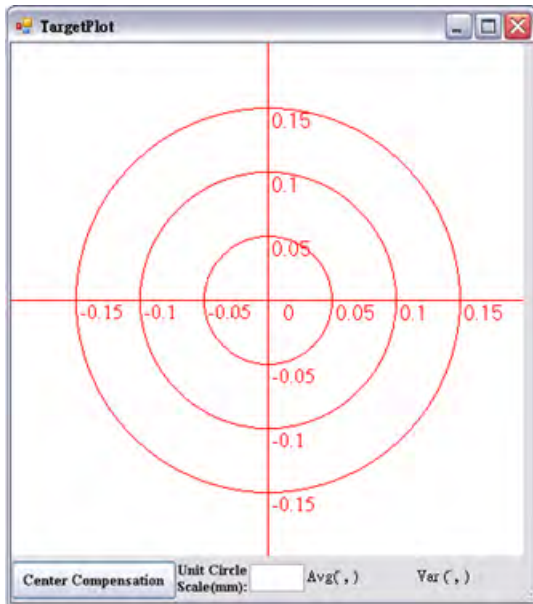


FIGURE 15. The target plot.

1) AVG(AVERAGE) AND VAR(VARIANCE)

We first calculate the total number of holes. Then, we compute the average and variance of x and y coordinates of the center of the holes.

2) CENTER COMPENSATION

We first set the position of the most upper left hole which has zero bias error. The positions are obviously offset after matching between the testing and the standard images. Hence, we use the Center Compensation function.

3) UNIT CIRCLE SCALE

This function allows us to display flexibility of the distribution of offsets based on different scales of the red circles. For instance, we set the value of the Unit Circle Scale was 0.5 on the target plot function and it shows the distribution of offsets on red circles with radius of multiples of 0.5.

H. LIGHTING

A suitable lighting system and the way of lighting are definitely the importing keys to detect the holes successfully. The combination of light sources and lighting method has to highlighting the holes as possible. We can easily distinguish between the holes and unnecessary noises only if the combination ensures enough brightness and contrast enhancement. So the location of the holes may not easily affect the image quality assessment.

IV. THE PROPOSED AOI SYSTEM

The AOI system, as shown in Fig. 16, includes hardware and software parts. The hardware part consists of optical modules and the three-Axis positioning system. For the software part, controlling of the linear motors and image processing are included. The operating processes are discussed at the end of this section.

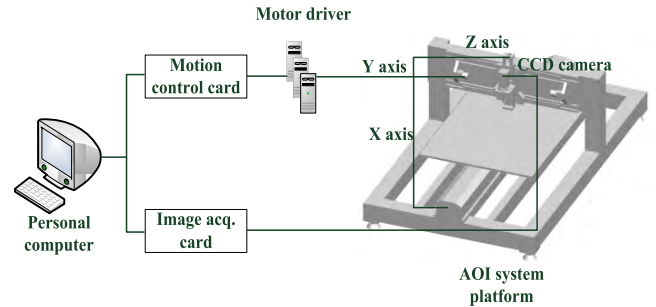


FIGURE 16. The structure of the AOI System.

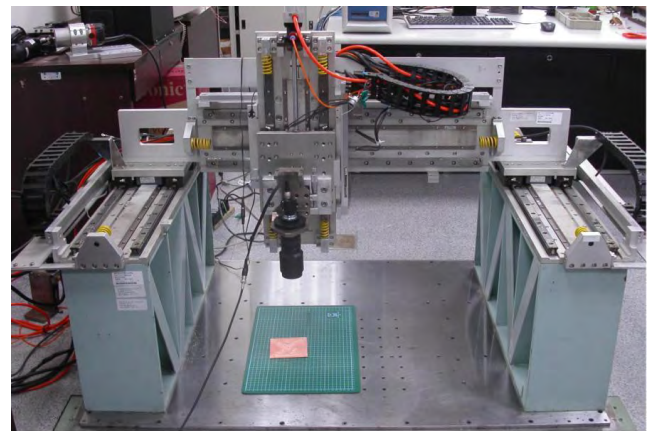


FIGURE 17. The three-axis positioning system.

A. HARDWARE COMPONENTS

The three-axis positioning system, as shown in Fig. 17, includes a servo motor, a linear motor and a stepping motor in x-axis, y-axis and z-axis, respectively. The XYZ positioning system is controlled by a PC through the motion control card (EPCIO) and the driver (M2J60A) to deliver commands. At first, we set the desired positions through the computer to a driver motor, and we check the situation of moving of the motor by returning the encoder of position, speed or moment. The accuracy of measurement of x-axis, y-axis and z-axis are $0.5\mu\text{m}$, $0.05\mu\text{m}$ and $0.2\mu\text{m}$, separately.

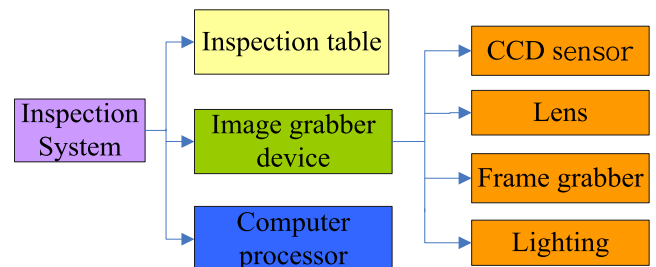


FIGURE 18. The hardware structure of the AOI System.

Optical modules are comprised of a PC, Sony lens group, contact image sensor (CIS), camera link PCI frame grabber, the linear scales, and the led ring light, as shown in Fig. 18.

The CIS is a Line/Area-Scan sensor that is used for industry. We use CIS with resolution of 75.75Um/pixel to grab each size of 640×640 sub-image and save them on PC. Through image registration written by C#, we combine all sub-images into a size of 7680×5760 image.

B. SOFTWARE PARTS

For the controlling motors, we move linear and servo motors on the x-axis and the y-axis to the chosen position and adjust stepping motor on the z-axis with a CCD to be in focus. The camera is fixed on the z-axis. Through the computer vision, we modify the coordinate errors until the right position is found.

For the image processing, we apply Microsoft Visual Studio (C#) as the development environment. This designed software can automatically measure various dimensions of the holes, such as the aperture, the roundness and the number of holes on a PCB. Image processing functions, including image registration, geometrical image transformation, and other functions, are mentioned in the previous section. In this section, three flow charts are discussed, and there are 1) labelling; 2) calculation of the centers and the areas of the holes; and 3) the target plot.

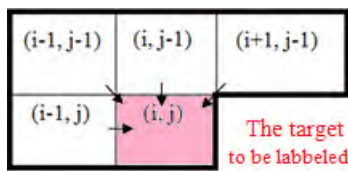


FIGURE 19. Connected components labeling algorithm.

1) LABELLING

We need to classify each pixel to different holes. We use Connected Components Labeling Algorithm [18], [19] to label each pixel. We set the target as the one we want to label and its grey level is 0 (black). As shown in Fig. 19, if the target is at (i, j) we have to make a decision about the label of this target according to the label of upper left $(i - 1, j - 1)$, upper $(i, j - 1)$, upper right $(i + 1, j - 1)$, and left $(i - 1, j)$.

This search is started from the most left to the most right on the first row. Following this, this algorithm starts searching from the most left to the end of the second row. We repeat this method until completion of searching an entire image. We can see from Fig. 20(a)-(d) there are two kinds of judgement of labelling depending on the labels of four adjacent pixels. The first type: We set the new label to the target if the four neighboring pixels are non-labelled. The second type, however, is when the four adjacent pixels are more than two different labels. We appoint the label of the target to the least number within those different labels. Simultaneously, for each target, we recorded those different labels into an array, as shown in Fig. 21. Given this, we can easily find the least number. We repeat these two kinds of assessment of labelling through the whole image. Finally, we can successfully label the same

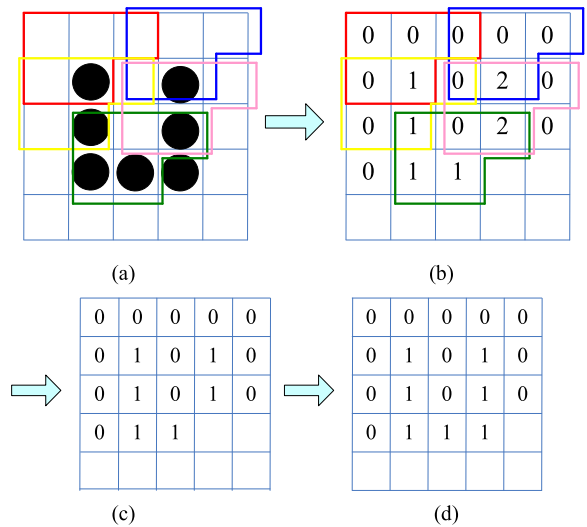


FIGURE 20. Labelling. (a) The result of after threshold; (b) After comparing more than two different labels; (c) Labelling the same hole; (d) Labelling for next pixel.

P[3]---	0	0	0	0	0
P[2]---	0	0	0	0	1
P[1]---	0	0	0	0	2
P[0]---	0	0	1	2	1

FIGURE 21. Record adjacent pixels of labels for each target.

number of pixels if they were closed enough. The flow chart of labelling can be seen in Fig. 22.

2) CALCULATION OF THE CENTER AND THE AREAS OF THE HOLES

We used (31) and (32) to calculate the coordinate of the center of the holes. The areas of the holes are obtained by using (33) and (34). Fig. 23 shows the flow chart of calculation of the center and area of holes from the software view.

3) TARGET PLOT

We input the difference between the testing image and the standard images to the target plot function. Then we can adjust the scales to show our errors or displacements. Fig. 24 shows the flow chart of calculation of target plot.

C. THE OPERATING PROCESSES

In this section, the operating processes of our designed AOI system are discussed, as shown in Fig. 25. There are six main steps of operating processes as describe as follows:

- 1) Move the Machine to the Right Position
 - (a). Open the light.
 - (b). Set the three-axis positioning system to the origin of the coordinate.
 - (c). Load motion files to the computer to move the motor with CCD until to the most left hole.
 - (d). Move camera on the z-axis up and down to focus it.

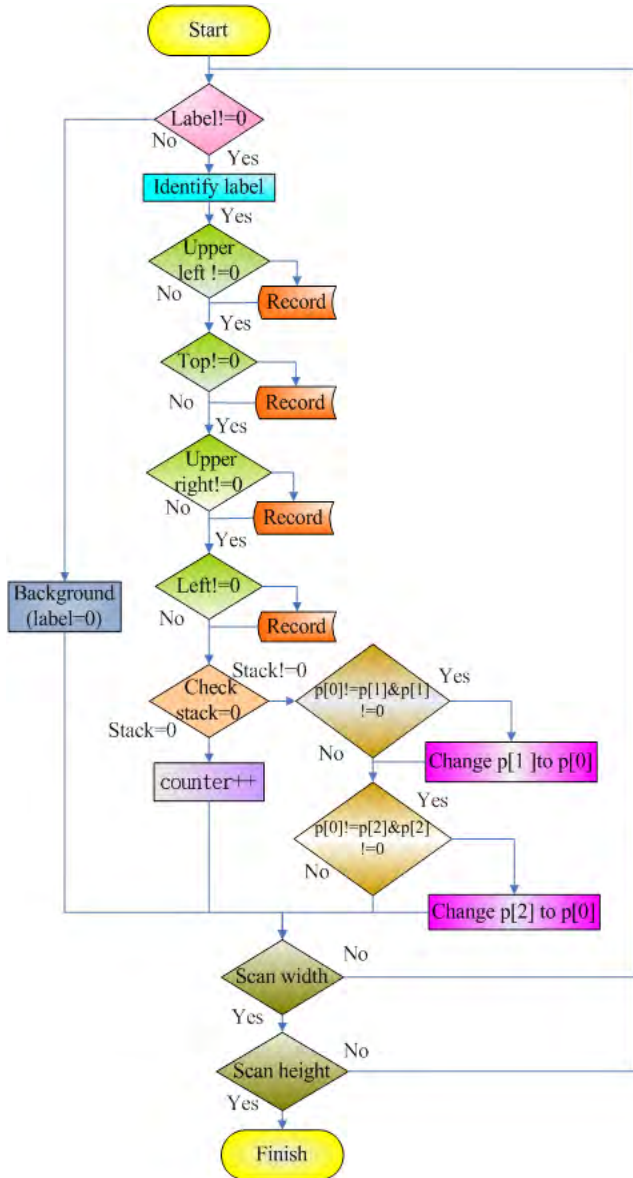


FIGURE 22. The flow chart of labelling.

- 2) Load Excellon files by pressing the LoadExcellon button
- 3) Press Constant velocity X Motion or Constant velocity Y Motion button to set the number of images that will be grabbed on x-axis and y-axis.
- 4) Take the image registration by pressing the Combine button.
- 5) Press Analysis bottom to load results from steps 1)-4). Then, the machine takes labels, calculate the areas, the centers and the roundness of the holes.
- 6) Press Target Plot button to draw the target plot according to the results from step 5).

V. MEASUREMENTS AND EXPERIMENTAL RESULTS

We use C# as the development environment to design a programming interface of an AOI system to put theory

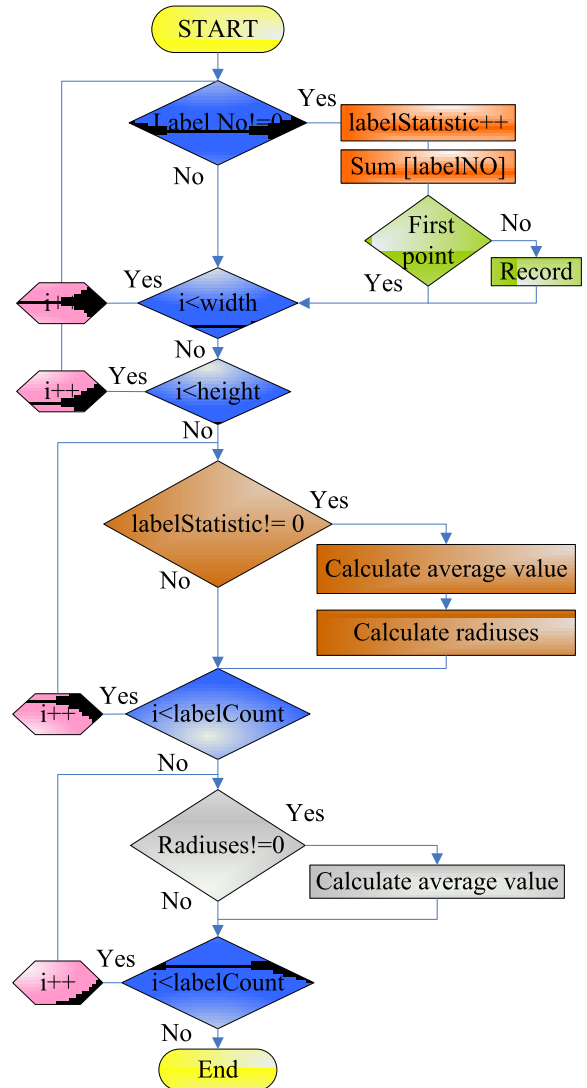


FIGURE 23. The flow chart of calculation of the centers and the areas of the holes.

into practice. The programming interface is shown in Fig. 26. Some of the functions are discussed as follows:

- 1) XRepeat and YRepeat are the number of images we want to capture in the x- or y-direction. For instance, if we set XRepeat and YRepeat at 12, it means the camera takes 12 pictures on both x-axis and y-axis.
- 2) XAxis Jog, or YAxis Jog is the movement of each shooting. In this paper, we set XAxis Jog, or YAxis Jog to 10 mm so the machine moves 10mm on each x- axis and y-axis to grab images.
- 3) The GetImage function allows us, through CCD, to look at the scene at the present moment.
- 4) Consuming label time shows how much time taken to label all the holes
- 5) Consuming Center time shows how much time taken to calculate the centers of the holes.

Fig. 27 shows the PCB we take experiments on, and its distance between two centers of two holes is 10mm.

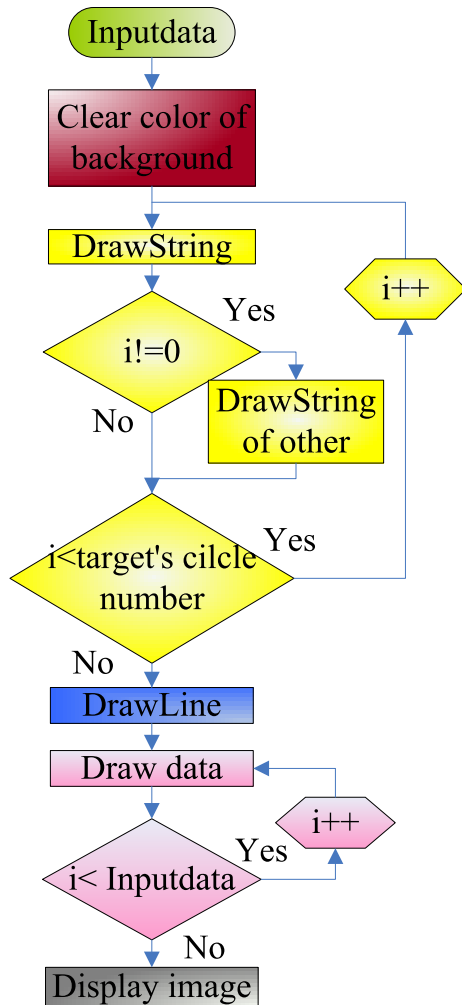


FIGURE 24. The flow chart of calculation of the target plot.

A. HARDWARE AND SOFTWARE CALIBRATION

The correction is the key to influence the calculation later. Both hardware and software correction are used. First, we move the machine on the xyz-axis to take preliminary correction. Then, precise correction is taken through software.

1) HARDWARE CORRECTION

The camera is paralleled with the z-axis to avoid taking each inclined sub-image. Besides, the cached images should parallel with the direction of the motion on the y-axis. By doing those, all sub-images are combined correctly. Fig. 28(a) and (b) show the results of before and after correcting. The lines in Fig. 28(a) are discontinuous. However, the lines are more continuous in Fig. 28(b).

2) SOFTWARE CORRECTION

The hardware calibration usually cannot correct perfectly. One common problem is when the linear motor moves too quickly to loosen the connection between the CCD and the linear motor. To improve this issue, we can take many sub-images and overlap large areas within images when using the

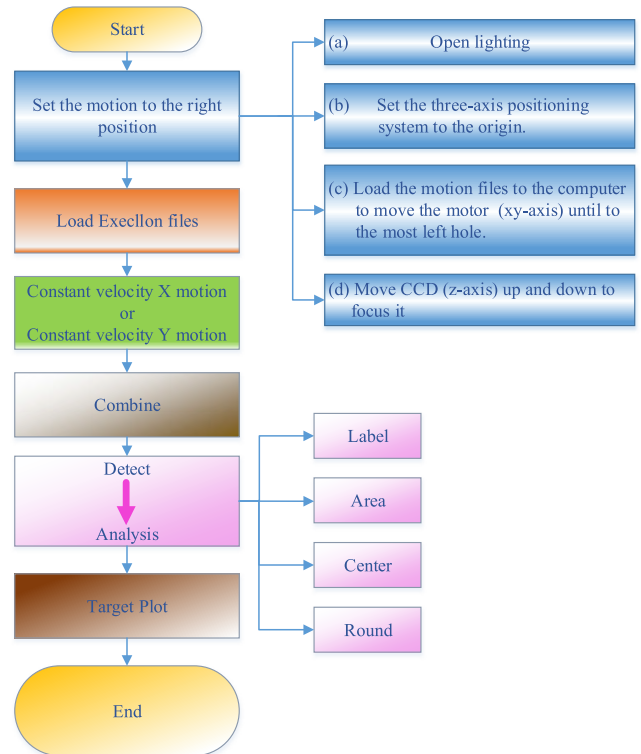


FIGURE 25. The operating processes.

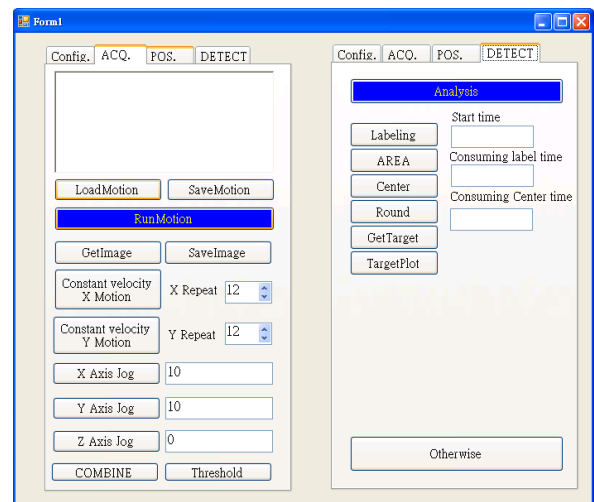


FIGURE 26. The program user interface.

image registration method. Besides, there is a ratio between the displacement of the linear motor and the captured image. This rate will cause distortion in the junctions between images. We have to take this ratio into consideration because this parameter will affect the precision of the inspection of the holes. The width between two lines, as shown within the red rectangle in Fig. 29(a), is larger than in Fig. 29(b). Fig. 29(a) and (b) show the results when we set the ratio to 13.5 and 14.5 respectively.

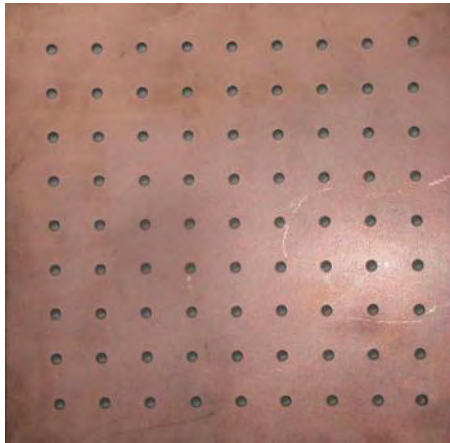


FIGURE 27. The PCB for experiment.

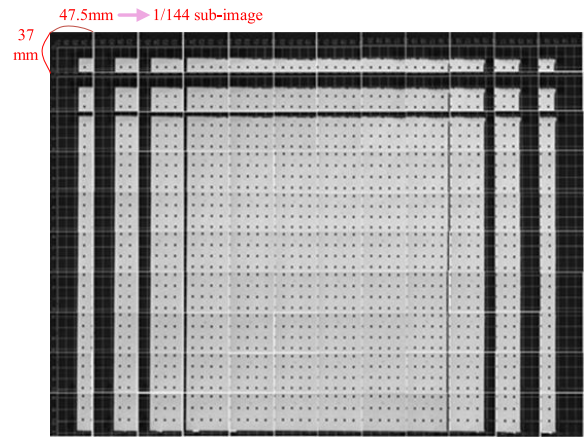


FIGURE 30. The shooting of a sub-image (35×45mm).

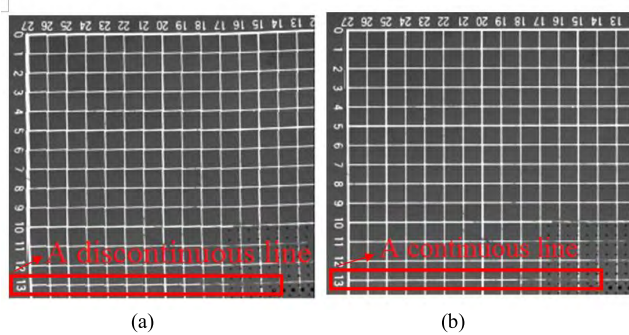


FIGURE 28. The results of before and after correcting. (a) The result of before correcting; (b) The result of after correcting.

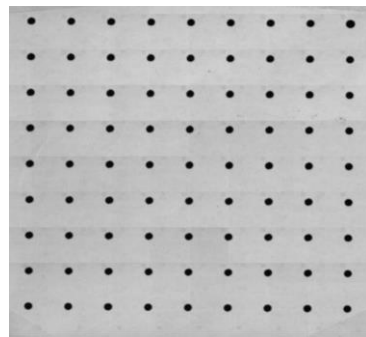


FIGURE 31. The result of image registration.

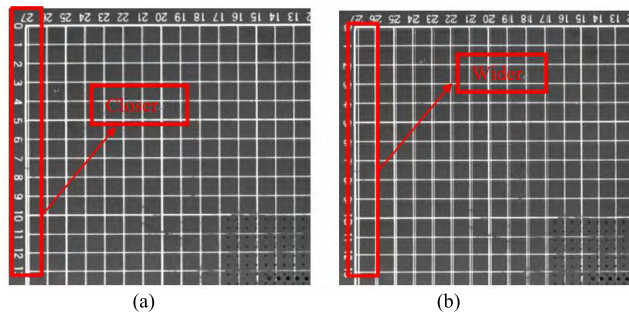


FIGURE 29. The result of different ratios. (a) The result set the ratio of 13.5; (b) The result set the ratio of 14.5.

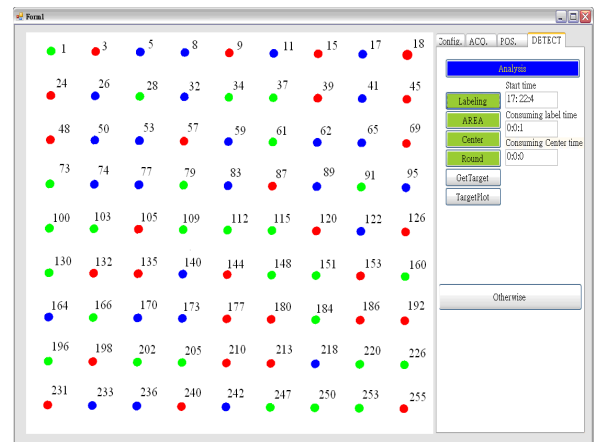


FIGURE 32. The result of labelling.

B. IMAGE REGISTRATION

As shown in Fig. 30, we take 12×12 sub-images, and each sub-image is 37mmx 47.5mm.

We move the CCD 10 mm each time after shooting on x-axis and y-axis separately. So the overlapping area along the x-axis and y-axis is 10mm × 37mm and 10mm × 47.5mm, respectively. The result of the image registration can be seen in Fig. 31.

C. LABELING

The holes are labeled, as shown in Fig. 32, by coloring them green, red, and blue. We first divide the labelling number by 3 and color the holes according to the remainder. We mark

the holes red when the labelling number is a multiple of 3, green when the remainder is 1, and blue when the remainder is 2. Simultaneously, we calculate and show the areas, the centers and the roundness of the holes. The consuming time of labelling is 0.01 second.

D. LABELING CENTER

For each hole we label, we obtain the position of the center by averaging the all coordinates of pixels. The red points in Fig. 33 show the coordinates of the centers of the holes

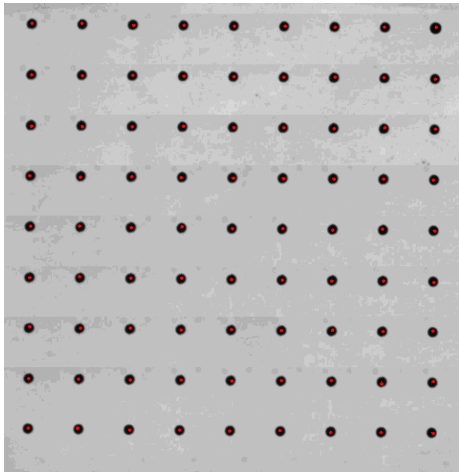


FIGURE 33. The result of calculation of the coordinates of the centers of the holes.

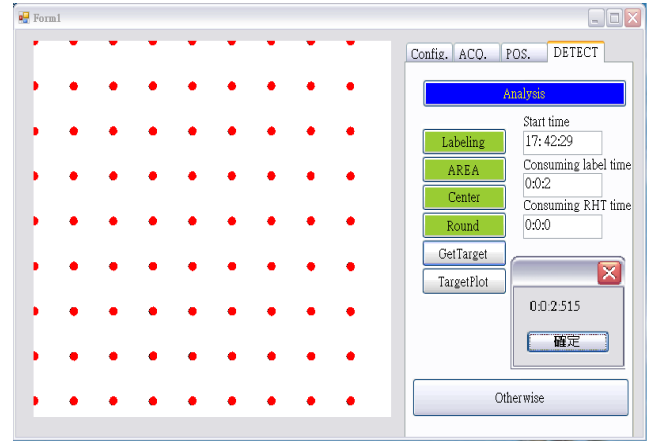


FIGURE 35. The matching result.

TABLE 2. Calculation of ratio and transform angle of the testing and standard images.

	A Standard Image	A Testing Image
The first datum point (The most upper right point as the minimum of $(x+y)$)	(0, 0)	(73, 55)
The second datum point (The most upper left point as the maximum of $(x-y)$)	(80, 0)	(1131, 187)
The distance between two datum points	$d_1=80(\text{mm})$	$d_2 = 1058.57$ (pixel)
The angle of rotation	$\theta_1=0$	$\theta_2=0.541$

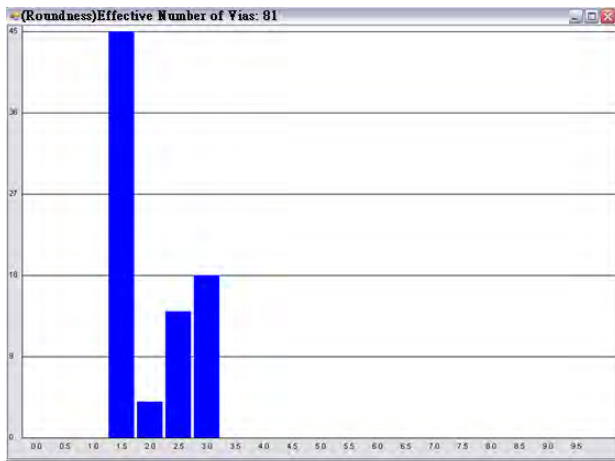


FIGURE 34. The statistics of roundness.

we found. Moreover, the area of each hole is the number of pixels belonging to the same label.

E. THE ROUNDNESS

The bar chart, as shown in Fig. 34, shows the statistics of the roundness. The highest number, when the roundness is 1.5 pixel, is 45. The second highest number is 18 where the roundness is 3.0 pixels.

F. THE CALCULATION OF RATIO AND TRANSFORM ANGLES OF THE TESTING AND STANDARD IMAGES

In this work, we set the minimum of $(x + y)$ as the first datum point and the maximum of $(x - y)$ as the second datum point. Based on those two datum points, the differences in distance between two datum points on the testing image and on the standard image are d_1 and d_2 , separately. The ratio between d_1 (80) and d_2 (1058.057) is 1/13.22. As shown in Table 2, θ_1 and θ_2 are the angles of rotation for both the testing and the standard images, respectively.

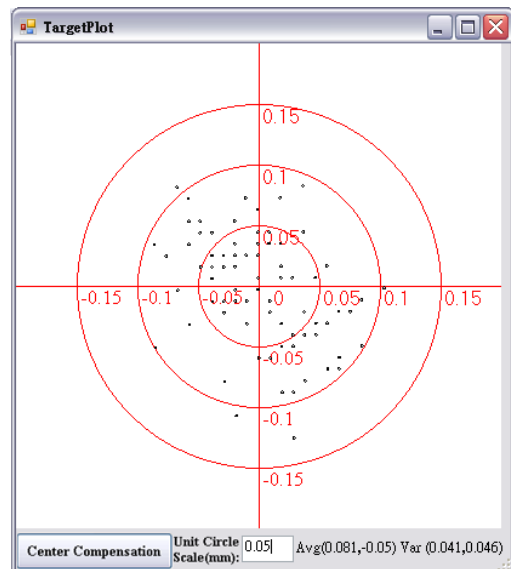


FIGURE 36. The target plot.

G. THE MATCHING RESULT, AND THE TARGET PLOT

The matching result shows the displacement between the testing and the standard images. We draw three colors to identify the normal, the error and the missing holes. We display black pixels to represent the standard image, and red pixels for the

testing image. Moreover, error holes are colored blue, and missing holes are expressed by green. As shown in Fig. 35, we can see the tiny difference between the testing (red color) and the standard images (black), but not error and missing holes. We set Unit Circle Scale to 0.05, as shown in Fig. 36. In other words, the distribution of offsets is displayed according to the red circles with the radius of multiples of 0.05. Most of the errors are within 0.1mm. The averages of the x coordinate and the y coordinate are 0.081 and -0.05 respectively. The variance of the x coordinate is 0.041 and the y coordinate is 0.046.

VI. CONCLUSIONS AND FUTURE WORKS

A. CONCLUSIONS

The precision of the holes on the PCB has played a major role in manufacturing industries. The increased number of the holes have to be drilled on smaller and smaller PCBs. In this work, we have used linear motors, and a CCD to achieve an AOI system to detect the positions and defects of holes on the PCB. We have designed C# as the programming interface to automatically taking 144 sub-images and combine all of them by using the image registration method. Then, we have calculated some basic dimensions, such as the area, the roundness, the center, and the aperture of the holes after labelling. Furthermore, based on those foundational dimensions and the matching method, we have identified the multi-hole, missing holes, aperture error, and displacement of the holes. Finally, we show the distribution of offset between the testing and standard images on the target plot. The total inspection time is 2.5 seconds. The defined accuracy of around $5\mu\text{m}$ was achieved.

B. DIRECTIONS OF FUTURE WORKS

We plan to conduct the research in three aspects regarding AOI systems in future.

1) SOFTWARE CORRECTION

For software correction, we manually calculate the ratio between the testing and standard images. We aim to automatically do software correction through images that have been taken.

2) HOLE MATCHING

We use the most upper left and right points as datum points. It would cause errors because if one or two of them are missing the result will fail. We have to figure out other methods to calculate ratio and the angle of rotation based on these two datum points. Or, we plan to substitute the missing holes to enhance the matching method.

3) CALCULATION OF OTHER DIMENSIONS

We hope to be able to calculate and analyze other dimensions such as Via, or even other electronic components on the PCB.

ACKNOWLEDGMENT

The authors would like to thank Dr. Martin Davies and Mr. Leigh Albon from the Federation University Australia, Victoria, Australia, for proofreading the English of the draft and giving many valuable comments and suggestions to this work.

REFERENCES

- [1] J. Jiang, J. Cheng, and D. Tao, "Color biological features-based solder paste defects detection and classification on printed circuit boards," *IEEE Trans. Compon. Packag. Manuf. Technol.*, vol. 2, no. 9, pp. 1536–1544, Sep. 2012.
- [2] S. Yammen and P. Muneesawang, "An advanced vision system for the automatic inspection of corosions on pole tips in hard disk drives," *IEEE Trans. Compon. Packag. Manuf. Technol.*, vol. 4, no. 9, pp. 1523–1533, Sep. 2014.
- [3] M. E. Scaman and L. Economikos, "Computer vision for automatic inspection of complex metal patterns on multichip modules (MCM-D)," *IEEE Trans. Compon. Packag. Manuf. Technol. B*, vol. 18, no. 4, pp. 675–684, Nov. 1995.
- [4] S. Shirmohammadi and A. Ferrero, "Camera as the instrument: The rising trend of vision based measurement," *IEEE Instrum. Meas. Mag.*, vol. 17, no. 3, pp. 41–47, Jun. 2014.
- [5] R. N. Jackson and W. R. Kern, "Determining outer layer registration via a concentricity measuring microscope," *Electron. Packag. Prod.*, vol. 19, no. 11, pp. 143–145, 1979.
- [6] B. Manirambona, J. De Baets, and A. Vervaeke, "Excimer laser microvia-technology in multichip modules," *Appl. Surf. Sci.*, vols. 208–209, no. 1, pp. 171–176, Mar. 2003.
- [7] K.-C. Lee and C.-C. Lin, "A vision-measuring method for an imperfect circle," *J. Chin. Inst. Ind. Eng.*, vol. 22, no. 1, pp. 38–45, 2005.
- [8] H. Rau and C.-H. Wu, "Automatic optical inspection for detecting defects on printed circuit board inner layers," *Int. J. Adv. Manuf. Technol.*, vol. 25, no. 9, pp. 940–946, May 2005.
- [9] S. Oprea, I. Lita, I. B. Cioc, and D. A. Visan, "Determination of misplaced drill holes on a PCB," in *Proc. 30th Int. Seminar Electron. Technol.*, May 2007, pp. 406–409.
- [10] Z. Huang, P. Wang, and L. Ma, "Automatic PCB inspection system based on virtual instruments," *Proc. SPIE*, vol. 6150, p. 61501T, May 2006.
- [11] D. Shetty, T. A. Eppes, N. Nazaryan, J. Kondo, and C. Campana, "Optical inspection of holes in jet engine blades," *Proc. SPIE*, vol. 6382, p. 638208, Oct. 2006.
- [12] Z. Ibrahim, N. K. Khalid, I. Ibrahim, M. S. Z. Abidin, M. M. Mokji, and S. A. R. S. A. Bakar, "A noise elimination procedure for printed circuit board inspection system," in *Proc. 2nd Asia Int. Conf. Modeling Simulation*, May 2008, pp. 332–337.
- [13] Y. Wang et al., "Vision based hole crack detection," in *Proc. IEEE 10th Conf. Ind. Electron. Appl. (ICIEA)*, Jun. 2015, pp. 1932–1936.
- [14] X. Huang, S. Zhu, X. Huang, B. Su, C. Ou, and W. Zhou, "Detection of plated through hole defects in printed circuit board with X-ray," in *Proc. IEEE 16th Int. Conf. Electron. Packag. Technol. (ICEPT)*, Aug. 2015, pp. 1296–1301.
- [15] S. Wang, "X-ray imaging tools for electronic device failure analysis," *Microelectronics Failure Analysis, Desk Reference*, 6th ed. OH, USA: EDFAS, ASM International, 2011, pp. 529–535.
- [16] O. Barnich and M. Van Droogenbroeck, "ViBe: A universal background subtraction algorithm for video sequences," *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1709–1724, Jun. 2011.
- [17] M. Van Droogenbroeck and O. Barnich, "ViBe: A disruptive method for background subtraction," in *Background Model. Foreground Detection for Video Surveillance*, vol. 7. London, U.K.: Chapman & Hall, 2014, pp. 7.1–7.23.
- [18] A. Gregory, *Digital Image Processing: Principles and Applications*. New York, NY, USA: Wiley, 1994.
- [19] L. Di Stefano and A. Bulgarelli, "A simple and efficient connected components labeling algorithm," in *Proc. IEEE 10th Int. Conf. Image Anal. Process.*, Sep. 1999, pp. 322–327.
- [20] W.-B. Horng and C. W. Chen, "Optimizing region of support for boundary-based corner detection: A statistic approach," *IEICE Trans. Inf. Syst.*, vol. E92-D, no. 10, pp. 2103–2111, 2009.

- [21] S.-Y. Huang, C.-W. Mao, and K.-S. Cheng, "Contour-based window extraction algorithm for bare printed circuit board inspection," *IEICE Trans. Inf. Syst.*, vol. E88-D, no. 12, pp. 2802–2810, 2005.
- [22] Z. Li and Q. Yang, "System design for PCB defects detection based on AOI technology," in *Proc. 2011 4th Int. Congr. Image Signal Process. (CISP)*, vol. 4, Oct. 2011, pp. 1988–1991.
- [23] S. Härter, T. Klinger, J. Franke, and D. Beer, "Comprehensive correlation of inline inspection data for the evaluation of defects in heterogeneous electronic assemblies," in *Proc. Pan Pacific Microelectron. Symp. (Pan Pacific)*, Jan. 2016, pp. 1–6.
- [24] F. Xie, A. H. Dau, A. L. Uittenboger, and A. Song, "Evolving PCB visual inspection programs using genetic programming," in *Proc. 2013 28th Int. Conf. Image Vis. Comput. New Zealand (IVCNZ)*, Nov. 2013, pp. 406–411.
- [25] H.-H. Loh and M.-S. Lu, "Printed circuit board inspection using image analysis," *IEEE Trans. Ind. Appl.*, vol. 35, no. 2, pp. 426–432, Mar. 1999.
- [26] F. Wu and X. Zhang, "Feature-extraction-based inspection algorithm for IC solder joints," *IEEE Trans. Compon. Packag. Manuf. Technol.*, vol. 1, no. 5, pp. 689–694, Mar. 2011.
- [27] H. Wu, X. Zhang, H. Xie, Y. Kuang, and G. Ouyang, "Classification of solder joint using feature selection based on Bayes and support vector machine," *IEEE Trans. Compon. Packag. Manuf. Technol.*, vol. 3, no. 3, pp. 516–522, Mar. 2013.
- [28] C. W. Mak, N. V. Afzulpurkar, M. N. Dailey, and P. B. Saram, "A Bayesian approach to automated optical inspection for solder jet ball joint defects in the head Gimbal assembly process," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 4, pp. 1155–1162, Oct. 2014.
- [29] A.-J. Long, D. Tsai, K. Lien, and S. Hsu, "Tolerance analysis of fixture fabrication, from drilling holes to pointing accuracy," in *Proc. IEEE Int. Test Conf. (ITC)*, Oct. 2015, pp. 1–4.



WEI-CHIEN WANG received the B.S. degree in mechatronic engineering from Huaan University, Taipei, Taiwan, in 2006, and the M.S. degree in manufacturing information and systems from National Cheng Kung University, Tainan, Taiwan, in 2008. She is currently pursuing the Ph.D. degree in engineering and information technology with Federation University Australia, VIC, Australia. She was a Full-time Research Assistant of the National Taiwan Normal University, Taipei, from 2009 to 2010. She was a Software Engineer with Hi-Lo System Research Co., Ltd., Taiwan, from 2010 to 2011. She was also a software engineer with the Software Design Center, Foxconn International Holdings, Ltd., Foxconn Technology Group, FIH Taiwan Design Center, Hon Hai Precision Ind. Co., Ltd., from 2011 to 2012. Since 2011, she has been with Academia Sinica, Taiwan, as a part-time Research Assistant. Her research interests include image processing, machine learning, and deep learning.



SHANG-LIANG CHEN received the B.S. and M.S. degrees in mechanical engineering from National Cheng Kung University, Tainan, Taiwan, in 1978 and 1984, respectively, and the Ph.D. degree in mechanical engineering from the University of Liverpool, U.K., in 1992. He is currently a Full Professor with the Institute of Manufacturing Information and Systems, National Cheng Kung University. He is also a Consultant of the Locks Association of Taiwan. He was a President of Taiwan Shoufu University from 2013 to 2015.



LIANG-BI CHEN (S'04–M'10–SM'16) received the B.S. and M.S. degrees in electronic engineering from the National Kaohsiung University of Applied Sciences, Kaohsiung, Taiwan, in 2001 and 2003, respectively. He is currently pursuing the Ph.D. degree in computer science and engineering with the National Sun Yat-sen University, Kaohsiung. From 2004 to 2011, he served as a Teaching and Research Assistant with National Sun Yat-Sen University. In 2008, he had an internship with the Department of Computer Science, National University of Singapore, Singapore. He was also a Visiting Researcher with the Department of Computer Science, University of California at Irvine, Irvine, CA, USA, from 2008 to 2009, and the Department of Computer Science and Engineering, Waseda University, Tokyo, Japan, in 2010. In 2012, he joined BXB Electronics Co., Ltd., Kaohsiung, as a Research and Development Engineer. In 2013, he was transferred from Executive Assistant to Vice-President. In 2016, he was transferred to the Department of Administration, as a Project Management Executive. In 2016, he joined the Southern Taiwan University of Science and Technology, Tainan, Taiwan, as a Senior Technical Manager. His research interests include VLSI design, power/performance analysis for embedded mobile applications and devices, power-aware embedded systems design, low-power systems design, digital audio signal processing, engineering education, project-based learning education, SoC/NoC verification, and system-level design space exploration. Since 2013, he has served as a Section Editor Leader, an Associate Editor, and a Guest Editor-in-Chief of the IEEE Technology and Engineering Education. He has also served as a TPC member, an IPC member, and as a Reviewer of many IEEE/ACM international conferences and journals. He has led many student teams to receive over 20 awards in national/international contests. He was a recipient of the 2014 IEEE Education Society Student Leadership Award and the 2015 IICM S.M. Cho IT Student Leader Award. He received the Honorable Mention Award at the ICAAI'16, the 1st Prize of Excellent Demo! Award at the IEEE GCCE'16, and the 2nd Prize of Excellent Poster Award at the IEEE GCCE'16. He is a senior member of the IEEE Consumer Electronics Society and the IEEE Education Society, and a member of the IEICE and the PMI.



WAN-JUNG CHANG (M'16) received the B.S. degree in electronic engineering from the Southern Taiwan University of Technology, Tainan, Taiwan, in 2000, the M.S. degree in computer science and information engineering from the National Taipei University of Technology, Taipei, Taiwan, in 2003, and the Ph.D. degree in electrical engineering from National Cheng-Kung University, Taiwan, in 2008. He is currently an Assistant Professor with the Electronic Engineering Department, Southern Taiwan University of Science and Technology. His research interests include the areas of cloud/IOT systems and applications, protocols for heterogeneous networks, and WSN/high-speed networks design and analysis. He received the 1st Prize of Excellent Demo! Award at the IEEE GCCE'16. He is a member of the IEEE Consumer Electronics Society.

...