BALL GRID ARRAY INSPECTION WITH X-RAY IMAGES

[1] Hong-Yi Chen (陳弘毅), Chi-Fu Lin (林奇賦), Yuh-Hsuan Hsiung (熊育萱), [2] Chiou-Shann Fuh (傅楸善)

[1] Dept. of Computer Science and Information Engineering,
National Taiwan University, Taiwan
E-mail: r04922099@ntu.edu.tw

[2] Dept. of Computer Science and Information Engineering,
National Taiwan University, Taiwan
E-mail: fuh@csie.ntu.edu.tw

ABSTRACT

A method of ball grid array inspection with x-ray image is proposed in this paper. We inspect the defect of solder ball on copper pillar flip chip. In this paper, we take geometry correction first. Use Integral projection to separate the image into several regions, remap, and do histogram equalization to make solder ball obvious. Apply Hough transform and connected component analysis to define the center of the solder ball on the binary image. After getting the average coordinates, we calculate the slope of the curve produced by applying vertical projection to each solder ball. Finally, use the threshold to find out the defective solder ball.

Keywords: *solder ball, geometry correction, integral projection, remap, Hough transform, connected component analysis*

1. INTRODUCTION

In electronics, heat and power issues are gating progress; especially in high-end, flip chip devices. System level cooling is limited by power requirements. A novel approach integrating thermal material into solder bumps allows for spot cooling at chip level, and enables a system solution to do its job efficiently [2]. Copper Pillar Bump (CPB) is one of the approaches to achieve the desired effect.

We aim to implement the solder inspection on copper pillar flip chip with X-ray image. Generally, many factors affect this issue, including X-ray intensity, the flatness of the copper pillar flip chip, the angle between X-ray tube and camera, and so on. These factors affect the quality of X-ray images.

In this paper, we mainly focus on solder defect inspection with complicated background. Our approach can separate into two parts: location and analysis.

The part of location consists of many measures because of the complexity of the input X-ray image.

Considering the copper pillar flip chip is on soft Printed Circuit Board (PCB) actually. It may not be flat on the *xy*-table. This may distort the X-ray image and affect the image quality. Thus, geometry correction is the first part to carry out, helping to define positions of solder balls afterward.

Copper pillar flip chip consists of many levels including many different materials, so the output image from two-dimensional transmission X-ray image may look like many components overlapping at the same locations. Therefore, we adopt remapping from the 16-bit image, mathematical morphology with binarization, and histogram equalization to filter the solder image.

The solder ball inspected in the X-ray image looks like a disk. Thus, we focus on disk detection with X-ray image. It is difficult because the solder ball is not a complete disk due to other overlapping electronic parts, and the darkest region is not in the middle of the disk. Thus, we implement three steps of detection with two methods: Hough transform and connected component.

In analysis, we use vertical projection to create a curve of each solder ball. The good solder ball on the image will change from deep to light in grey level, from the center to the boundary of the solder ball due to well solder wetting. Thus, the curve of the good solder ball will rise smoothly from center to the boundary, otherwise, the bad solder ball curve will show a step edge nearby the boundary.

Some important techniques used in our approach will be introduced in Section 2. The detailed procedure of our approach will be described in Section 3. Section 4 demonstrates the experimental results. Section 5 concludes this paper, and the final section is the list of references.

2. BACKGROUND

2.1. Geometry Correction

Geometry correction is a basic topic in image processing and computer vision. It is the process of digitally manipulating image data such that the image's projection precisely matches a specific projection surface or shape, and aims to make an image look visually correct after projection [6]. We can consider projection as warping the image. Forms of image warping can be found in many projector models. According to our detected case, we decide to use shearing mapping.

In plane geometry, a shear mapping [7] is a linear mapping that displaces each point in fixed direction, by an amount proportional to its signed distance from a line that is parallel to that direction.

Shear mappings must not be confused with rotations. Applying a shear mapping to a set of points of the plane will change all angles between them (except straight angles), and the length of any line segment that is not parallel to the direction of displacement. Therefore, it will usually distort the shape of a geometric figure, for example turning squares into non-square parallelograms, and circles into ellipses. However, a shearing does preserve the area of geometric figures, the alignment and relative distances of collinear points.

In our approach, we use horizontal shear which is a function that takes a generic point with coordinates (x, y) to the point $(x + my, y)$ where m is a fixed parameter called the shear factor.

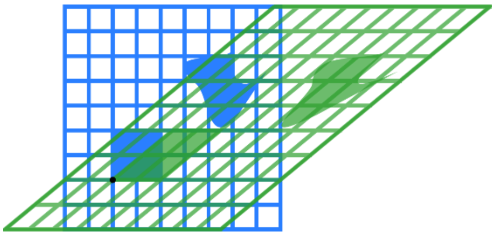


Fig. 2-1 A horizontal shearing of the plane with coefficient $m = 1.25$, illustrated by its effect (in green) on a rectangular grid and some figures (in blue). The black dot is the origin.

The effect of this mapping is to displace every point horizontally by an amount proportionally to its y coordinate. Any point above the x-axis is displaced to the right (increasing x) if $m > 0$, and to the left if $m < 0$. Points below the x-axis move in the opposite direction, while points on the axis stay fixed.

Straight lines parallel to the x-axis remain where they are, while all other lines are turned, by various angles, about the point where they cross the x-axis. Vertical lines, in particular, become oblique lines with slope $1/m$. Therefore, the shear factor m is the cotangent of the angle by which the vertical lines tilt, called the shear angle.

If the coordinates of a point are written as a column vector (a 2×1 matrix), the shear mapping can be written as multiplication by a 2×2 matrix:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x + my \\ y \end{pmatrix} = \begin{pmatrix} 1 & m \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

where x and y are Cartesian coordinates of the xy-plane, and m is shear factor [7].

To shear our detected case, we need to define the shear factor first. We apply connected component to find some obvious objects with the same y coordinates actually to be the reference of the bias.

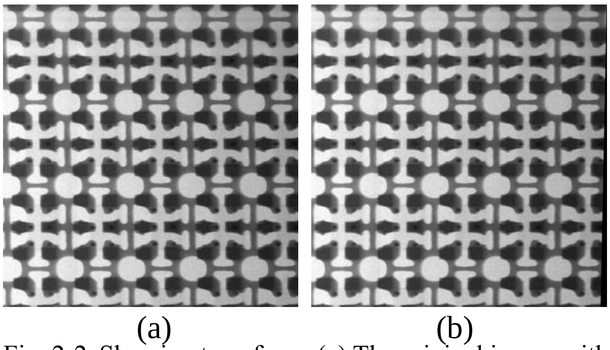


Fig. 2-2 Shearing transform. (a) The original image with distortion. (b) The sesult by applying shearing transform.

2.2. Remapping

The PCB consists of many layers, so different solder balls overlap with other different electronic components. It may cause different locations of the solder ball with different appearances on the image. Considering different look-up tables may come to different results, so that we can do remapping from 16-bit to 8-bit to enhance the contrast of the solder ball. Remapping is defined by:

$$S = \frac{(r - Min) \times NR}{OR}$$

where r is original intensities; s is new intensities; Min is minimal intensity (e.g. 2000); NR is 8-bit new range (e.g. 255-0); OR is 16-bit old range (e.g. 7000-2000); and range is the difference between maximal intensity and minimal intensity.

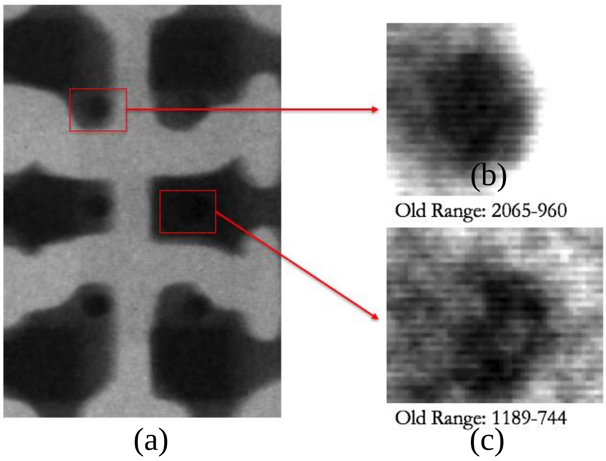


Fig. 2-3 The result of remapping by different look-up tables. (a) The original 8-bit image. (b)(c) The result by remapping by different look-up tables.

2.3. Histogram Equalization

Even the remapping can enhance the contrast, we still use histogram equalization [5] to get further more contrast enhancement. Histogram equalization is a method to adjust the contrast using the histogram of the image. Histogram equalization usually increases the global contrast of many images, especially when the usable data of the image are represented by low-contrast values. Through this adjustment, the intensities can be better distributed on the histogram. This allows for areas of lower local contrast to gain a higher contrast. Histogram equalization accomplishes this by effectively spreading the most frequent intensity values.

Histogram equalization is useful in images with backgrounds and foregrounds that are both bright or both dark. A key advantage of histogram equalization is that it is a fairly straightforward technique and an invertible operator. Thus in theory, if the histogram equalization function is known, then the original histogram can be recovered. The calculation is not computationally intensive. A disadvantage of the method is that it is indiscriminate. It may increase the contrast of background noise, while decreasing the usable signal [5]. It is defined by:

$$S_k = 255 \times \sum_{j=0}^{k} \frac{n_j}{n}$$

where k = 0~255; n_j is number of pixels with intensity j; n is the total number of pixels, for every pixel with intensity k the result pixel with intensity S_k.

The underlying mathematics behind histogram equalization involves mapping one distribution (the given histogram of intensity values) to another distribution (a wider and, ideally, uniform distribution of intensity values). That is, we want to spread out the *y*-values of the original distribution as evenly as possible in the new distribution. It turns out that there is a good answer to the problem of spreading out distribution values: the remapping function should be the cumulative distribution function. However, cumulative density can be applied to any distribution; it is just the running sum of the original distribution from its negative to its positive bounds.

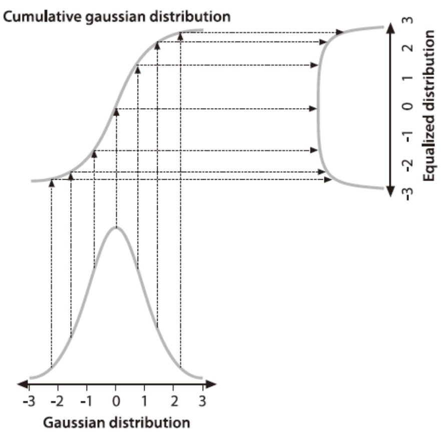


Fig. 2-4 Using the cumulative density function to equalize a Gaussian distribution [1].

We may use the cumulative distribution function to remap the original distribution to an equally spread distribution (in Figure 2-4) simply by looking up each *y*-value in the original distribution and seeing where it should go in the equalized distribution [1].

2.4. Circle Detection

There are many ways to detect circle. Since solder ball is not a complete disk due to other overlapping electronic parts, it would get low accuracy with only one method. In our approach, we apply Circle Hough Transform and connected component analysis to get higher accuracy.

2.4.1. Circle Hough Transform

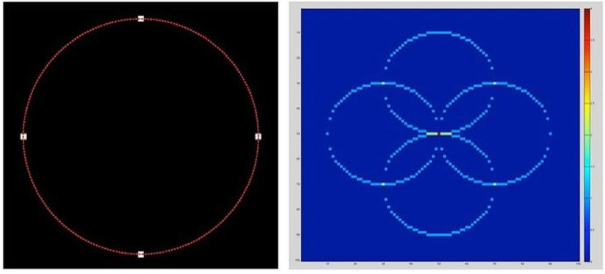
The Circle Hough Transform (CHT) [4] is a basic technique used in digital image processing for detecting circular objects in a digital image. It is a specialization of Hough transform. The purpose of the technique is to find circles in imperfect image inputs. CHT ability to perform with the incomplete circle is the main reason. The circle candidates are produced by “voting” in the Hough parameter space and then selecting the local maxima in a so-called accumulator matrix.

In a two-dimensional space, a circle can by described by:

$$(x-a)^2 + (y-b)^2 = r^2$$

where (*a*, *b*) is the center of the circle, and *r* is the radius. The parameter space would be three-dimensional, (*a*, *b*, *r*). All the parameters satisfying (*x*, *y*) would lie on the surface of an inverted right-angled cone whose apex is at (*x*, *y*, 0). In the 3D space, the circle parameters can be identified by the intersection of many conic surfaces that are defined by points on the 2D circle. This process can be divided into two stages. The first stage is fixing radius then finding the optimal center of circles in a 2D parameter space. The second stage is to find the optimal radius in a one-dimensional parameter space.

Since we have assigned the radius of the solder ball, the parameter space would be reduced to 2D (the position of the circle center). For each point (*x*, *y*) on the original circle, it can define a circle centered at (*x*, *y*) with radius *r* according to the fomula above. The intersection point of all such circles in the parameter space would correspond to the center point of the original circle (in Figure 2-5).



(a) (b)

Fig. 2-5 Points on a circle in (a) the original image [4]. The circle Hough transform is shown in the right (b). The intersection of all the circles shows the center of the circle in (a).

In practice, an accumulator matrix is introduced to find the intersection point in the parameter space. First, we need to divide the parameter space into "buckets" using a grid and produce an accumulator matrix according to the grid. The element in the accumulator matrix denotes the number of "circles" in the parameter space passing through the corresponding grid cell in the parameter space. The number is also called "voting number". Initially, every element in the matrix is zero. Then for each "edge" point in the original space, we can formulate a circle in the parameter space and increase the voting number of the grid cell where the circle passes through. This process is called "voting". After voting, we can find local maxima in the accumulator matrix. The positions of the local maxima correspond to the circle centers in the original space.

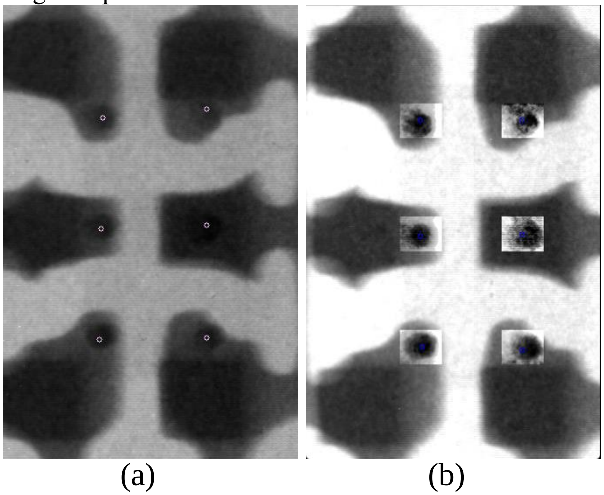


Fig. 2-6 Circle Hough Transform. (a) The result by circle Hough transform with original image. Pink points show centers of solder balls. (b) The result by Circle Hough Transform with enhanced image. Blue points show centers of solder balls.

2.4.2 Connected Component Analysis

Connected Component Analysis [3] is a method defining components as regions of adjacent pixels that have the same input value (or label). To compute the connected components of an image, we first (conceptually) split the image into horizontal runs of adjacent pixels, and then color the runs with unique labels, reusing the labels of vertically adjacent runs whenever possible. In a second phase, adjacent runs of different colors are then merged (in Figure 2-7).

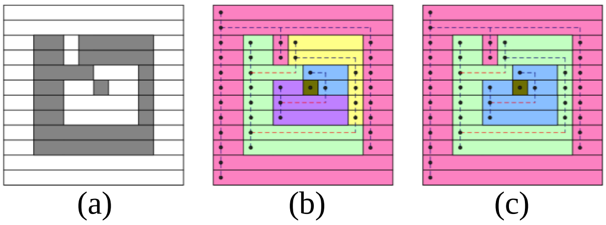


Fig 2-7 Connected component computation. (a) Original gray scale image. (b) Horizontal runs (nodes) connected by vertical (graph) edges (dashed blue)—runs are pseudo colored with unique colors inherited from parent nodes. (c) Re-coloring after merging adjacent segments.

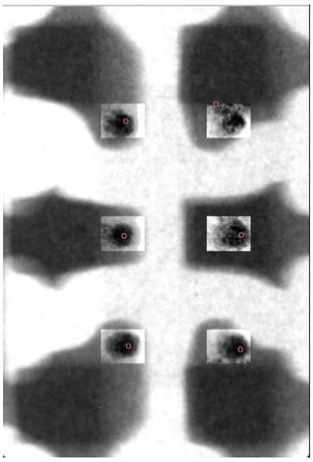


Fig. 2-8 The result by connected component analysis with enhanced image. Orange points show centers of solder balls.

2.5. Integral Projection

Integral projection is a method to calculate the sum of the intensity of all the pixels in each column or each row by vertical integral projection or horizontal integral projection.

The horizontal integral projection (HIP) is defined by:

$$\mathrm{HIP}(i) = \sum_{i=0}^{n} I(i,j)$$

where *I(i, j)* is the intensity of the image with *n* pixels in each column.

The vertical integral projection (VIP) is defined by:

$$\mathrm{VIP}(ij) = \sum_{j=0}^{n} I(i,j)$$

where *I(i, j)* is the intensity of the image with *n* pixels in each row.

3. METHODOLOGY

The flow of our proposed algorithm is as follows:

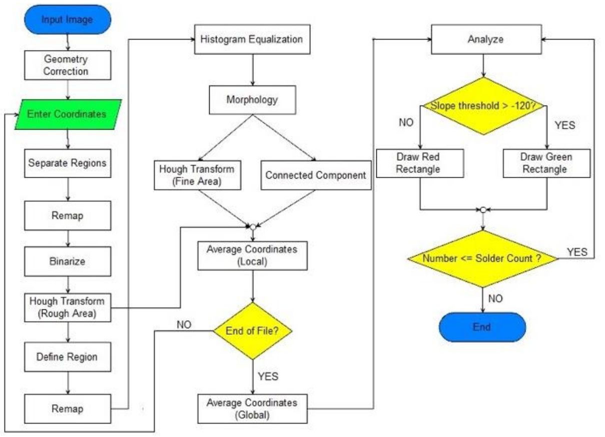
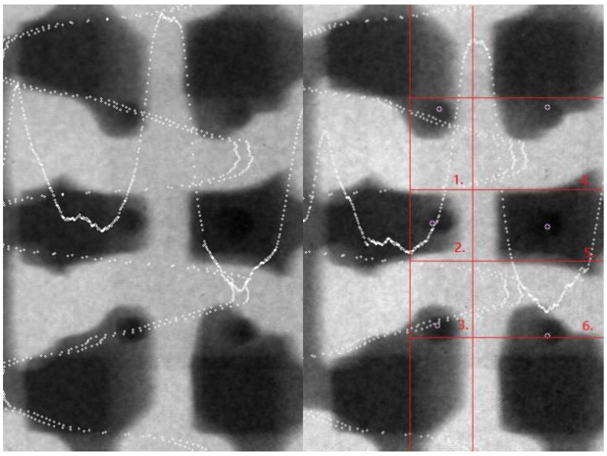


Fig. 3-1 The flow of our proposed algorithm.

In our approach, we take geometry correction (shearing transform) first. Considering the features of the pattern, we use extreme points of horizontal and vertical projection to cut our interested part out of the pattern to filter many irrelevant parts which could affect the detection.



(a) (b)

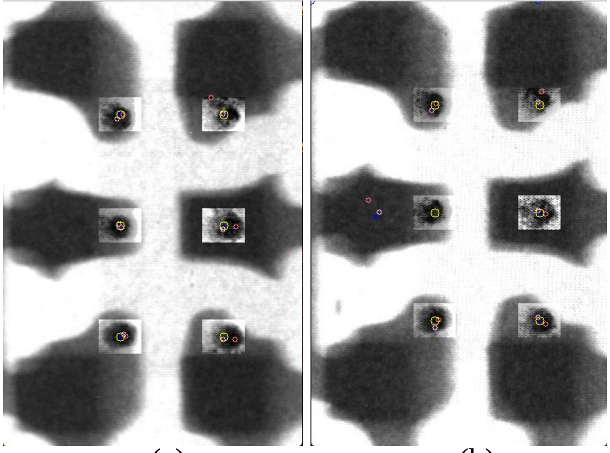
Fig. 3-2 The result of separating region. (a) The image with the curves made by horizontal and vertical projection. (b) The Regions 1 to 6 are the regions we focus on.

After separating region, we remap and do histogram equalization to make solder ball obvious in each region and apply the first circle detection by Hough transform on rough area in binary image with dynamic threshold, depending on whether finding a circle, to define the rough destination of the solder balls. Thus we can define fine regions whose center is the same as the center of the circle found by Hough transform and make sure each region contains a solder ball. To each fine region, we remap, do histogram equalization, and use opening operation in mathematical morphology to enhance again, then go to second circle detection.

In the second circle detection, we use Hough transform as well on fine area. In addition, the only one obvious component in the region is the solder ball. Thus we can also use connected component analysis to find the circle in the region.

Since the solder ball is not a complete disk as we mentioned before, there are few wrong locations at each time we apply circle detection. Considering the features of the pattern, the *y* coordinate of each solder ball is the same in each row, and the *x* coordinate of each solder ball is also the same in each column. Thus we decide to take all position average to let the few wrong locations be corrected by most of correct locations (in Figure 3-3). Thus we use many times of circle detection.

The average horizontal coordinates of left three positions are made by the average of the horizontal coordinates of circles detected by Hough transform and connected component analysis on the left side. The horizontal coordinates of right three positions are set to the minimum of the vertical projection. The vertical coordinates of top two positions are made by the average of the vertical coordinates of circles detected by Hough transform and connected component analysis on the top. The vertical coordinates of bottom two positions are made in the same way. The middle two are set to the value of the half of the difference between the top and the bottom vertical coordinates. The circle may not be chosen, if the difference between its horizontal or vertical coordinate and others is more than ten pixels.



(a) (b)

Fig. 3-3 The result by averaging the coordinates locally. (a) The pink and blue points show the centers of the circles made by Hough transform. The orange points show the centers of the circles made by connected component analysis. The yellow points are the positions after averaging. (b) Even getting wrong locations during detection, we still get correct location after averaging.

In order to get more samples to get higher accuracy, we extend the scope where we get samples from each pattern to the whole input image containing 12 patterns at 6 solder balls/pattern.

After getting the average coordinates, considering the good solder ball on the image will change from deep to light gradually in grey level, from the center to the boundary of the solder ball due to well solder wetting, the curve of the good solder ball will rise smoothly from center to the boundary. Otherwise, the bad solder ball

Experimental Environment

1. **OS: OS X El Capitan 10.11.4**
2. **CPU: 2.7 GHz Intel Core i5**
3. **Memory: 8 GB**
4. **Platform: Xcode 7.1 with OpenCV 2.4.9**

curve will show a step edge and sudden rise nearby the boundary. According this feature, we calculate the slope of the curve produced by applying vertical projection to each solder ball. The threshold is set by -120 gray level / pixel to distinguish the good solder and defective solder ball (in Figure 3-5).

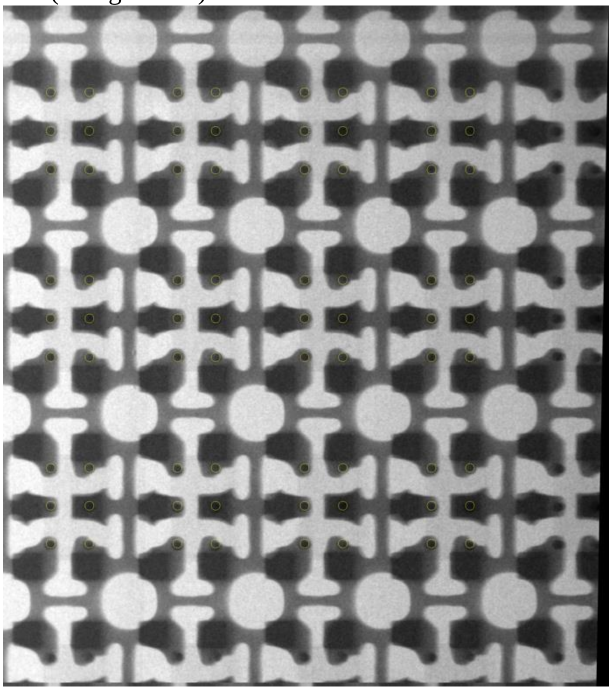
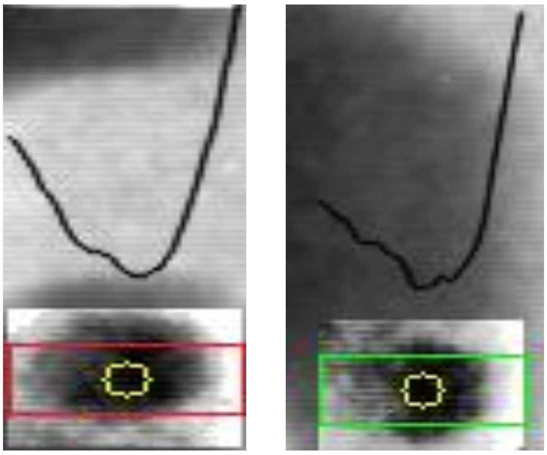


Fig. 3-4 Yellow circles show circle center of the solder ball by computing the average coordinates of all the circles found.



(a) (b)

Fig. 3-5 The result after analysis. (a) The red rectangle shows the defective solder ball steeper than -120 gray level / pixel. (b) The green rectangle shows the good solder ball smoother than -120 gray level / pixel. The rectangle is 20 vertical pixels × 50 horizontal pixels.

4. EXPERIMENTAL RESULT

We have four source images, and all images are 2000 × 2249 pixels at 16-bit/pixel images. There are 12 complete patterns at 6 solder balls/pattern in each image. In each pattern, we know that the solder ball on the left-top is with flux to enhance solder melting and the solder ball on the left-bottom is without flux. Thus we can use these four images to test our approach.

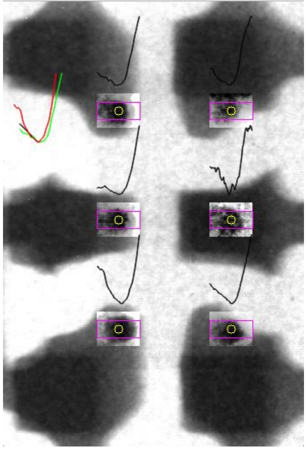


Fig. 4-1 Pattern 1 in output image with projection curve. Purple rectangle is the range for projection (20 vertical pixels x 50 horizontal pixels). The green curve shows the solder without defect in the region on the top-left, and the red curve shows the solder with defect in the region on the bottom-left. The black curve is the result after projection of the solder below. (Slope threshold: -120 gray level / pixel.)

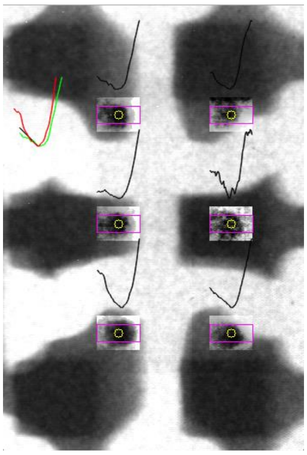


Fig. 4-2 Pattern 2 in output image with projection curve. Purple rectangle is the range for projection (20 vertical pixels x 50 horizontal pixels). The green curve shows the solder without defect in the region on the top-left, and the red curve shows the solder with defect in the region on the bottom-left. The black curve is the result after projection of the solder below. (Slope threshold: -120 gray level / pixel.)

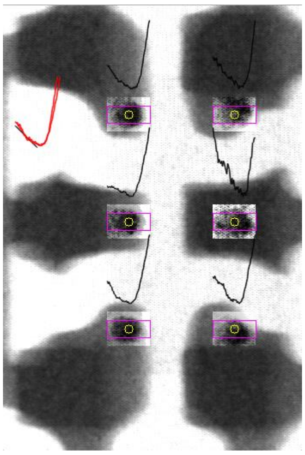


Fig. 4-3 Pattern 3 in output image with wrong result. Purple rectangle is the range for projection (20 vertical pixels x 50 horizontal pixels). The two red curves show the solders with defect in the regions on the top-left and bottom-left. There should be no defect in the region on the top-left. The black curve is the result after projection of the solder below. (Slope threshold: -120 gray level / pixel.)

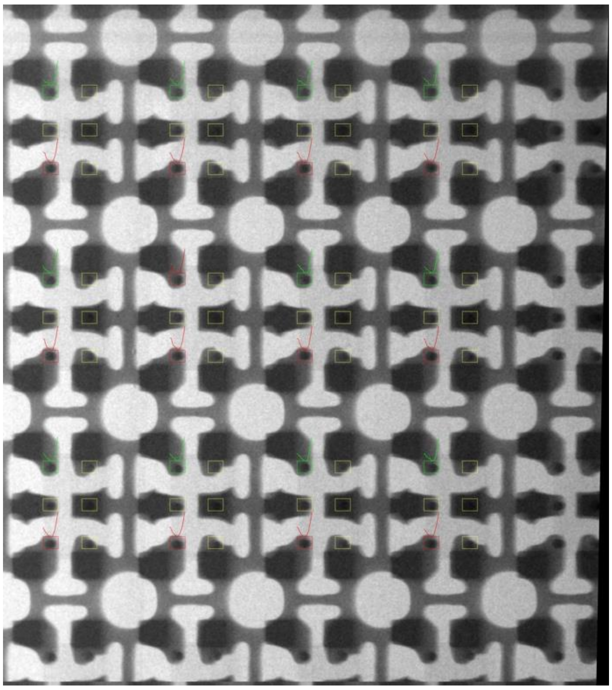


Fig. 4-4 The Output Image 1 with the projection curve. Green curves and rectangles show the good solder ball. Red curves and rectangles show bad solder ball. Yellow rectangles show undetermined result.

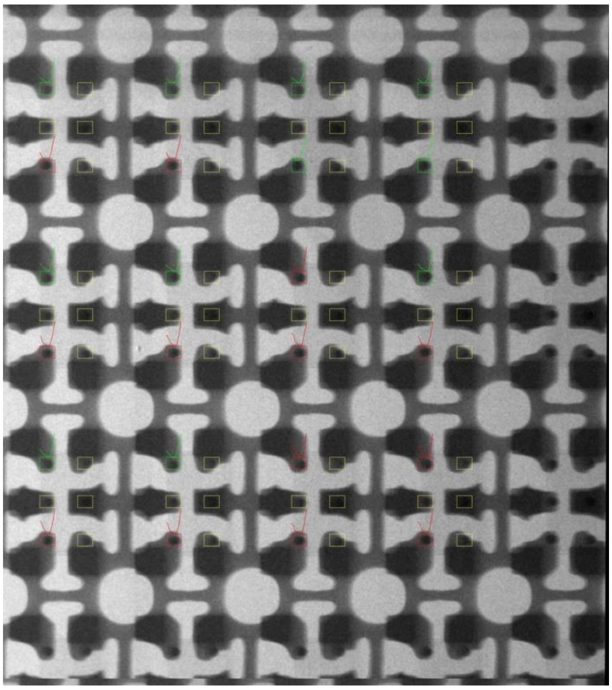


Fig. 4-5 The Output Image 2 with the projection curve. Green curves and rectangles show the good solder ball. Red curves and rectangles show bad solder ball. Yellow rectangles show undetermined result.

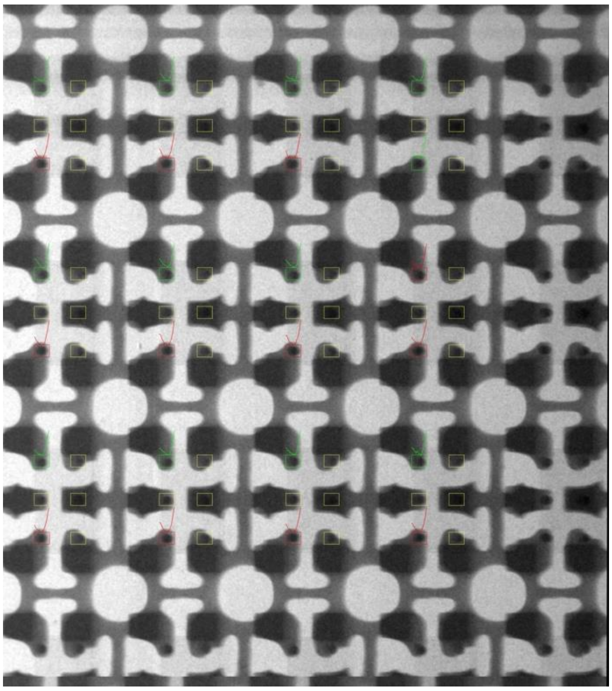


Fig. 4-6 The Output Image 3 with the projection curve. Green curves and rectangles show the good solder ball. Red curves and rectangles show bad solder ball. Yellow rectangles show undetermined result.

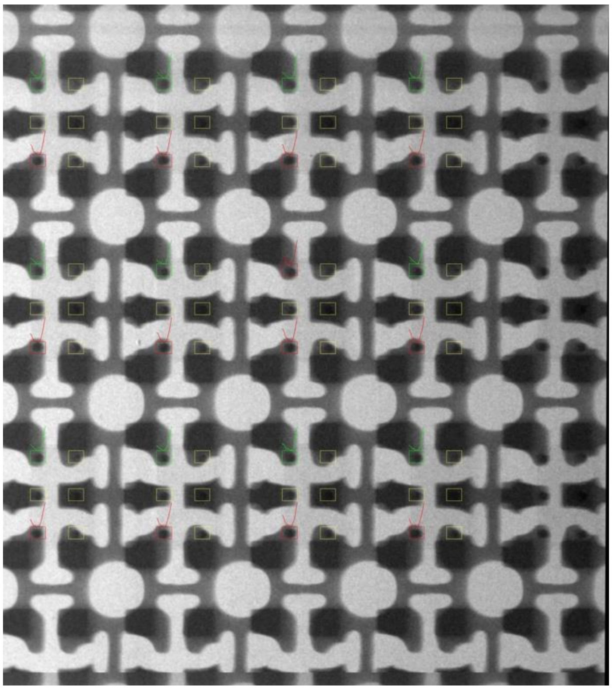


Fig. 4-7 The Output Image 4 with the projection curve. Green curves and rectangles show the good solder ball. Red curves and rectangles show bad solder ball. Yellow rectangles show undetermined result.

5. CONCLUSION

The reliability of solder joints is a critical element for the successful application of flip chip technology. The solder ball detection is a challenging problem due to the complex background. In our work, we propose a method for this purpose based on shearing transform to do geometry correction, remapping, and histogram equalization to enhance the image, Hough transform and connected component analysis to detect circle, and integral projection to analyze the solder ball. Our experimental results show that the proposed method can reach 81.25% accuracy for solder balls of copper pillar.

ACKNOWLEDGEMENT

This research was supported by the Ministry of Science and Technology of Taiwan, R.O.C., under grants MOST 103-2221-e-002-188 and 104-2221-e-002-133-my2, and by Test Research, Lumens Digital Optics, and Lite-On.

REFERENCES

[1] G. Bradski and A. Kaehler, *Learning OpenCV*, 1st ed, O’Reilly, Sebastopol, CA, 2008.

[2] P. A. Magill, “The Thermal CPB An Approach to Thermal and Power Management”, http://electroiq.com/blog/2008/01/the-thermal-cpb-an-approach-to-thermal-and-power-management/, 2008.

[3] R. Szeliski, *Computer Vision: Algorithms and Applications*, Springer, London, 2010.

[4] Wikipedia, “Circle Hough Transform,” https://en.wikipedia.org/wiki/Circle_Hough_Transform, 2016.

[5] Wikipedia, “Histogram equalization,” https://en.wikipedia.org/wiki/Histogram_equalization, 2016.

[6] Wikipedia, “Image geometry correction,” https://en.wikipedia.org/wiki/Image_geometry_correction, 2016.

[7] Wikipedia, “Shear mapping,” https://en.wikipedia.org/wiki/Shear_mapping, 2016.