

## Extracting PCB components based on color distribution of highlight areas

Zhou Zeng<sup>1</sup>, LiZhuang Ma<sup>1</sup>, and ZuoYong Zheng<sup>1</sup>

<sup>1</sup> The School of Electronic, Information and Electrical Engineering,  
the Shanghai Jiao Tong University, MinHang Campus,  
NO.800 Dong Chuan Road, Shanghai, 200240, P.R.China  
bluecourse@china.com, ma-lz@cs.sjtu.edu.cn, Zuoyong.Zheng@gmail.com

**Abstract.** This paper investigates methodologies for locating and identifying the components on a printed circuit board (PCB) used for surface mount device inspection. The proposed scheme consists of two stages: solder joint extraction and protective coating extraction. Solder joints are extracted by first detecting all the highlight areas, and then recognizing and removing the invalid highlight areas which are mainly markings and via-holes. We sum up three color distribution features. And the invalid highlight areas are recognized and removed by comparing the features of the target objects and the reference objects. The sequence of color distribution as a new clue has been applied to clustering solder joints. Each protective coating is extracted by the positions of the clustered solder joints. Experimental results show that the proposed method can extract most of components effectively.

**Keywords:** machine vision, component extraction, solder joints, protective coatings.

### 1. Introduction

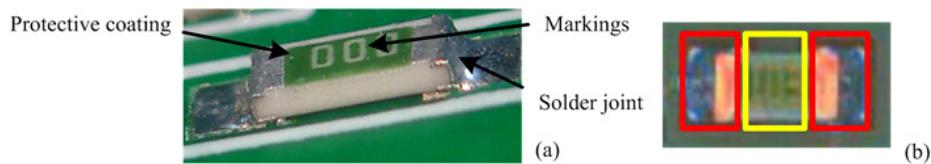
With the fast development of surface mounting technology, the need for automatic inspection has been ever increasing. A portion of PCB surface defect inspection involves inspecting the solder joints on the PCB and many different inspection approaches have been developed (Teoh et al. [13], Bartlet et al. [1], Kim et al. [6] [7]). Because of the complexity of the board surface, the identification of solder joint locations is relatively difficult. Loh et al [3] use a slant map to extract the shape information of a solder joint, which is based on the slant angle of the solder joint surface. But to their approach, there exist a number of factors that can affect the solder joint shape, including the gravity, the solder ability of the solder pad and so on. For these reasons, their approach is limited. The histogram based methods are used extensively to extracting solder joints. Kim et al. [7] take three frames of images which are sequentially captured as three layers of LEDs and turned on, one after another. From these images, soldered regions are segmented by using  $x/y$

projection and thresholding. This approach just works for the local areas where solder joints are concentrated. Wu and Wang [4] present an efficient approach for PCB components detection. The suggested method is based on multiple template matching and a modified species based particle swarm optimization (SPSO) with three specific acceleration strategies. This kind of approach has a high degree of accuracy for a specified object. But it needs to sample lots of templates for a complex PCB, which has many different components.

This paper explores automated object-recognition techniques for extracting multiple objects (e.g. IC, chip resistor, transistor, diode and etc.) in a PCB image. From the perspective of PCB image, a typical chip component consists of protective coating, markings and solder joints. The proposed approach detects solder joints based on their specular attribute and determines specular interval of gray levels by multilevel thresholding algorithm. The markings, via-holes and other invalid specular areas are recognized and removed by comparing color distribution features of them with those of reference object in  $(u', v')$  chromaticity coordinates. The sequence of color distribution as a new clue has been applied to clustering solder joints. Each protective coating is extracted by the positions of the clustered solder joints. According to the experimental results, the developed algorithm can effectively recognize most of components without any restrictions.

## 2. Illuminant System

We capture images by PCB inspection device VT-RNS (Omron Corp.). Three layers of ring-shaped LEDs with the different illumination angles and input camera are controlled by the host computer. The illumination design is similar to the equipment that is used for Kim and Cho [6]. We acquire an image when all LEDs are turned on in order to get specular areas of solder joints as many



**Fig. 1.** A typical component. (a) component structure; (b) image of the component under three layers of ring shaped LEDs illuminant system.

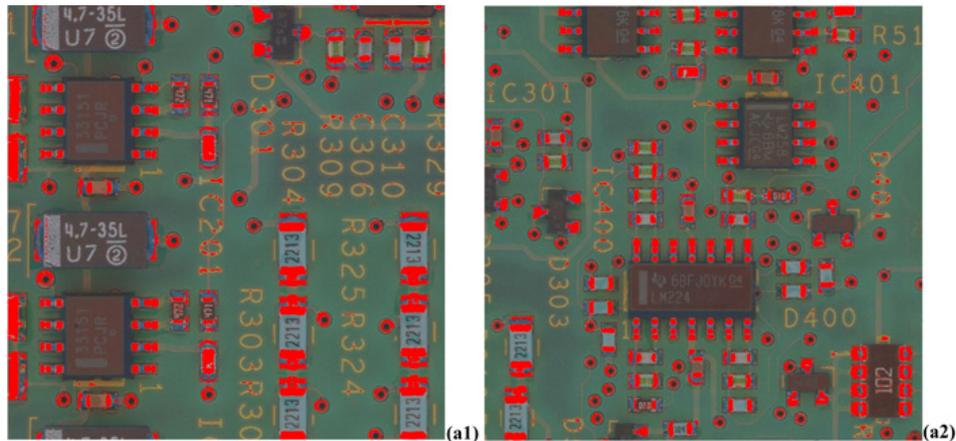
as possible. Fig.1 (a) shows a typical chip component. From the perspective of PCB image, it consists of a protective coating, multiple markings and more than two solder joints. Fig.1 (b) shows the projected image under illuminant system. The highlight area of solder joints shows up as the 2D projection of R, B and G three LEDs, and the color distribution of highlight area shows regularity. The proposed algorithm consists of two parts: One is to extract

solder joints. That is, to locate solder joints and to evaluate their ranges (see red rectangles in Fig.1 (b)). This is discussed in section 3. Another is to extract protective coatings. That is, to locate protective coatings and to evaluate their ranges (see a yellow rectangle in Fig.1 (b)). This is discussed in section 4.

### 3. Extracting solder joints

In general, the algorithm of extracting solder joints is divided into two stages: one is to detect all highlight areas; another is to recognize and remove invalid highlight parts based on the color distribution features. Then the remained highlight areas only contain solder joints. In Sect.3.1, the specular detection algorithm is presented. In Sect.3.2, GMM based color distribution feature extraction and comparison approach is introduced. The implement details of extracting solder joints are described in Sect.3.3.

#### 3.1. Specular detection



**Fig. 2.** PCB images and specular detection results. (a1) and (a2) are two PCB images  $I_a$  and  $I_b$ . Red points are detected highlight pixels.

Many specular detection methods have been introduced in the past decade, such as [7][8][11][12]. Yen et al. [14] have proposed a new criterion for multilevel thresholding. We present a novel approach based on automatic multilevel thresholding for detecting specularity. Implementation details are described as follows: (1) Image preprocess. Decrease brightness and sharpen the contrast of input image by

$$p' = [(p + (b + 100) \cdot 1.27) - 127] \cdot [(c + 100) \cdot 0.01] + 127 \quad (1)$$

where  $p$  is a original pixel; brightness coefficient  $b \in (-100,100)$  and contrast coefficient  $c \in [0,100)$ . This operation is favorable for minimizing disturbance from background and makes the specular areas more obvious. (2) Convert color space from RGB to CMYK. CMYK produces a lower level of color details than RGB does. This helps to decrease the discontinuity of specular areas. (3) Construct the histogram of K channel and evaluate the specularity interval by automatic multilevel thresholding approach[14]. By minimizing the cost function, the classification number that the gray-levels of K channel should be classified and the threshold values can be determined automatically. The last classification belongs to specularity. We select two typical PCB images  $I_a$  and  $I_b$  to illustrate the experiment results. Fig.2 (a1) and (a2) show specular detection results of  $I_a$  and  $I_b$  respectively.

### 3.2. GMM based color distribution features extraction and comparison

Color cues have been shown to offer several significant advantages for certain tasks in visual perception, such as [5][9]. The proposed approach also takes color as a cue for recognizing the specified components based on Gaussian Mixture Model (GMM). A framework is developed for recognizing the target objects according to the following procedures: First, project all highlight pixels into  $U'V'$  space and model the color distribution of them by GMM. And then, sample several reference objects and all target objects, and extract their color distribution features. Finally, recognize the target object with a comparison between features of the target object and features of some reference objects.

#### 3.2.1. Modeling color distribution of highlight pixels in $U'V'$ space

In order to keep color constant, first, we convert all recognized highlight pixels into LUV space, and then project them into  $(u',v')$  chromaticity coordinates with the following transformation

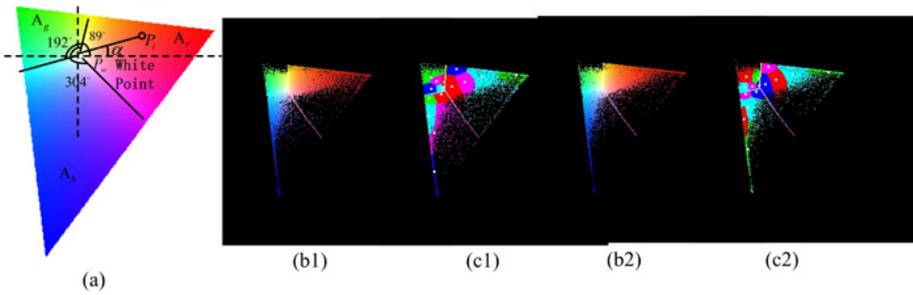
$$u' = u / (13l) + u'_n, v' = v / (13l) + v'_n \quad (2)$$

where  $l, u, v$  are LUV channels; the quantities  $u'_n$  and  $v'_n$  are the  $(u',v')$  chromaticity coordinates of a specified white point. In our case, we set  $u'_n = 0.200331$  and  $v'_n = 0.474959$  with standard illuminant C.

Fig.3 (a) shows the visible gamut in  $(u',v')$  perceptually uniform coordinates. The color distribution is modeled based on GMM. First, all highlight pixels are classified into three areas  $A_b$ ,  $A_g$  and  $A_r$ . In Fig.3 (a), three black lines segment  $(u',v')$  space into above three areas. Each highlight pixel is assigned to the specified area according to the following rule

$$\alpha = \arctan(|v_{\max} - v'_n - v'_i| / |u'_i - u'_n|) \cdot 180 / \pi, P_i \in \begin{cases} A_r, & \alpha \leq 89^\circ \cup \alpha \geq 304^\circ \\ A_g, & \alpha \in (89^\circ, 192^\circ) \\ A_b, & \alpha \in [192^\circ, 304^\circ) \end{cases} \quad (3)$$

where  $v_{\max}$  is the maximum value of coordinate axis  $v'$ ; chromaticity coordinates  $(u'_i, v'_i)$  and  $(u'_n, v'_n)$  locate highlight pixel  $P_i$  and white point  $P_w$  respectively; Angle  $\alpha$  between vector  $\overline{P_w P_i}$  and horizontal line (see Fig.3 (a)). Fig.3 (b1) and (b2) show color distributions of  $I_a$  and  $I_b$ .



**Fig. 3.** Color distributions. (a) CIE 1976 uniform chromaticity scale diagram. (b1) and (b2) are color distributions in chromaticity coordinates of  $I_a$  and  $I_b$  respectively; (c1) and (c2) are evaluated Gaussian mixture distributions with five mixtures of  $I_a$  and  $I_b$  respectively. Each kind of color represents a mixture.

After that, color distribution of each area is modeled by Gaussian mixture distribution. The EM (Expectation-Maximization) algorithm estimates the parameters of the multivariate probability density function in a form of the Gaussian mixture distribution with a specified number of mixtures (Bishop [2] and Render [10]). Consider the set of the feature vector  $\{x_1, x_2, \dots, x_n\}$ :  $n$  vectors from  $d$ -dimensional Euclidean space drawn from a Gaussian mixture

$$\rho(x|a_k, S_k, \pi_k) = \sum_{k=1}^m \pi_k P_k(x), \pi_k \geq 0, \sum_{k=1}^m \pi_k = 1 \quad (4)$$

$$\rho_k(x) = \varphi(x|a_k, S_k) = \frac{1}{(2\pi)^{d/2} |S_k|^{1/2}} \cdot e^{\{-\frac{1}{2}(x-a_k)^T S_k^{-1} (x-a_k)\}} \quad (5)$$

where  $m$  is the number of mixtures;  $\rho_k$  is the normal distribution density with the mean  $a_k$  and covariance matrix  $S_k$ ; and  $\pi_k$  is the weight of the  $k$ th mixture. Given the number of mixtures  $m$  and the samples  $\{x_1, x_2, \dots, x_n\}$  the algorithm finds the maximum-likelihood estimates (MLE) of the all the mixture parameters, i.e.,  $a_k$ ,  $S_k$  and  $\pi_k$ . Corresponding to pixels within  $A_r$ ,  $A_b$  and  $A_g$ , three groups of GMM parameters,  $\{a_{k,r}, S_{k,r}, \pi_{k,r}\}$ ,  $\{a_{k,b}, S_{k,b}, \pi_{k,b}\}$  and  $\{a_{k,g}, S_{k,g}, \pi_{k,g}\}$  are evaluated for modeling color distribution of all highlight

pixels. For simplicity, the GMM parameters are denoted as  $\{a_{k,c}, S_{k,c}, \pi_{k,c}\}, c \in \{r, g, b\}$ . Fig.3 (c1) and (c2) show the modeling results of  $I_a$  and  $I_b$  respectively with  $m = 5$ .

### 3.2.2. Extracting distribution features of the specified object

Three color distribution features of each specified object are evaluated:

(1) Mean distribution probability of each mixture. A  $3 \times m$  matrix  $M_{prob}$  is defined to store  $3m$  mean probabilities, and  $M_{prob}(i, j)$  denotes the mean probability of  $j$ th mixture in  $i$ th area.

(2) Mean distance between pixels in each mixture and white point. A  $3 \times m$  matrix  $M_{dist}$  is defined to store  $3m$  mean distances, and  $M_{dist}(i, j)$  denotes the mean distance of  $j$ th mixture and in  $i$ th area.

(3) The ratios of pixel number of three areas. A vector  $V_{ratio}$  is defined to store three ratios, and  $V_{ratio}(i)$  denotes the ratio of pixel number of  $i$ th area.

Some terms are used to describe the steps of evaluating color distribution features, they are defined as follows:  $\psi_{tar}$  and  $\psi_{ref}$  denote the color distributions of the target object and the reference object respectively;  $A_c$  denotes the area and  $c \in \{r, g, b\}$ ;  $\tau_{c,i}$  denotes the  $i$ th mixture in  $A_c$ ;  $N(c, i)$  records pixel number of the  $i$ th mixture and in  $A_c$ ;  $\rho_k(c, i)$  denotes the normal distribution density of pixel  $P_k$  in the  $i$ th mixture and in  $A_c$ ;  $d_k$  denotes the distance between  $P_k$  and white point.

Suppose that there are  $n$  highlight pixels  $P = \{P_k | P_1, P_2, \dots, P_n\}$  in an object. In order to evaluate three features, first, the  $3 \times m$  matrix  $M_{dist}$ ,  $M_{prob}$  and a vector  $V_{ratio}$  are initialized as zero. And then,  $P_k$  is assigned to the specified area  $A_c$  by (3). For each pixel  $P_k$ , computes its  $m$  probabilities  $\rho_k(c, i)$  by (5) with evaluated parameters  $\{a_{i,c}, S_{i,c}, \pi_{i,c}\}, i \in [1, m]$  and computes its distance  $d_k$ . Finally, mean probabilities, mean distances and pixel number ratio are evaluated by:

$$N_c = \sum_{i=1}^m N(c, i), c \in \{r, g, b\}, i \in [1, m]$$

$$\text{For all } P_k, \text{ and } P_k \text{ within } A_c, M_{prob}(c, i) = \sum_{k=1}^{N_c} \rho_k(c, i) / N_c \quad (6)$$

$$\text{For all } P_k, \text{ and } P_k \text{ within } \tau_{c,i}, M_{dist}(c, i) = \sum_{k=1}^{N(c,i)} d_k / N(c, i)$$

$$V_{ratio}(c) = N_c / n$$

To summarize, we express differences between the color distributions of two objects in three levels.  $V_{ratio}$  describes the difference between areas,  $M_{dist}$  describes the difference between mixtures and  $M_{prob}$  describes the difference within single mixture. Only by working together, three features can effectively describe the color distribution. It's likely to causing ambiguity just relying on one or two of features.

### 3.2.3. Recognizing the target object by feature comparison

In order to determine whether  $\psi_{ref}$  and  $\psi_{tar}$  are of the same type, there are three cases are considered: (1)  $\psi_{tar}$  and  $\psi_{ref}$  share the same mixtures; (2) All pixels of  $\psi_{tar}$  within the subset of the mixtures of  $\psi_{ref}$ ; (3) Some pixels of  $\psi_{tar}$  are outside the mixtures of  $\psi_{ref}$ . The relative difference on mean distribution probability is evaluated according to formula

In case1 and 2: if  $M'_{dist}(c, i) > 0$ ,

$$f_{prob}(c, i) = \frac{|M_{prob}(c, i) - M'_{prob}(c, i)|}{f_{non\_zero}(M_{prob}(c, i), M'_{prob}(c, i))}$$

In case3: if  $M'_{dist}(c, i) = 0$  and  $M_{dist}(c, i) > 0$ ,

$$f_{prob}(c, i) = \left( \frac{|M_{prob}(c, i) - M'_{prob}(c, i)|}{f_{non\_zero}(M_{prob}(c, i), M'_{prob}(c, i))} \right) \cdot \mu, \mu > 1.0,$$

$$\lambda_{prob}(c) = \left( \sum_{i=1}^m f_{prob}(c, i) \right) / m$$

where  $M_{prob}$ ,  $M_{dist}$  and  $V_{ratio}$  are features of  $\psi_{tar}$ ,  $M'_{prob}$ ,  $M'_{dist}$  and  $V'_{ratio}$  are features of  $\psi_{ref}$ ; and  $\lambda_{prob}(c)$  denotes the relative difference on mean distribution probability in  $A_c$ ; function  $f_{non\_zero}$  is used to select a nonzero variable; function  $f_{prob}(c, i)$  denotes a relative difference on mean distribution probability of the  $i$ th mixture and in  $A_c$ . In general, we sample multiple reference objects simultaneously in order to maintain stability of reference features. It might lead to the distribution scope of  $\psi_{tar}$  that is less than or equal to the scope of  $\psi_{ref}$ . So case 1 and case 2 are considered normal, but there is a huge variation between  $\psi_{tar}$  and  $\psi_{ref}$  in case 3. In order to show this variation, a penalty  $\mu$  is attached to enlarge the relative differences. After that, the relative differences on mean distances is evaluated in three areas according to formula

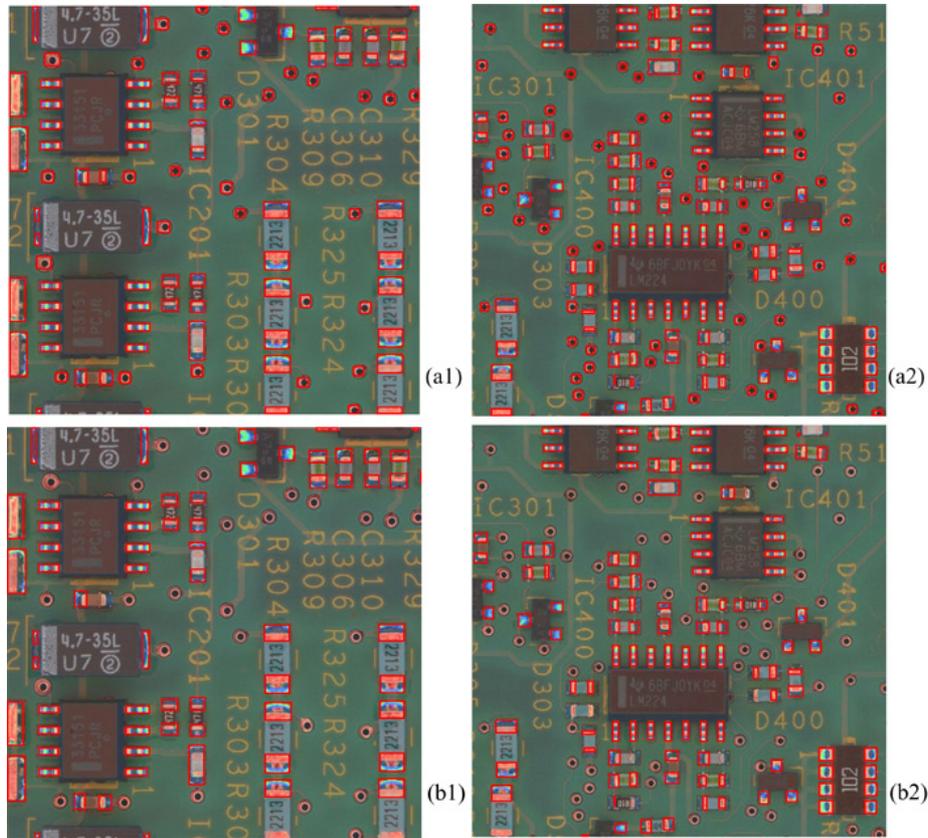
$$\lambda_{dist}(c) = \left( \sum_{i=1}^m \frac{|M_{dist}(c, i) - M'_{dist}(c, i)|}{f_{non\_zero}(M_{dist}(c, i), M'_{dist}(c, i))} \right) / m \quad (8)$$

where  $\lambda_{dist}(c)$  denotes the relative difference on mean distance in  $A_c$ . Finally, evaluate the total difference of three features between  $\psi_{ref}$  and  $\psi_{tar}$  by

$$f_{ratio}(c) = \frac{|V_{ratio}(c) - V'_{ratio}(c)|}{\sum_{c=1}^3 |V_{ratio}(c) - V'_{ratio}(c)|}, \lambda = \sum_{c=1}^3 (\alpha \cdot \lambda_{prob}(c) + (1-\alpha) \cdot \lambda_{dist}(c)) \cdot f_{ratio}(c) \quad (9)$$

where  $f_{ratio}(c)$  denotes a relative difference on ratio of pixel number in channel  $c$ ,  $\lambda$  denotes total difference, coefficient  $\alpha$  is used to adjust the ratio between difference of mean probability distribution and difference of mean distance, and  $\alpha \in [0,1]$ . Usually, we set  $\alpha = 0.6$ . If  $\lambda < 0.5$ , then  $\psi_{ref}$  and  $\psi_{tar}$  are of the same type.

### 3.3. Implement details of extracting solder joints



**Fig. 4.** Recognize and remove results of PCB images  $I_a$  and  $I_b$ . (a1) and (a2) show the results after removing the markings and merging the remaining clusters; (b1) and (b2) show the results after removing via-holes.

The five steps of extracting solder joints are described as follows:

First, project all highlight pixels in  $u'v'$  chromaticity coordinate and model their color distribution based on GMM (refer to Sect. 3.2.1).

Second, cluster all highlight pixels. We traverse all highlight pixels and classify them into many clusters according to their differences in  $K$  channel of CMYK color space by eight neighborhood search. We define  $c$  is a single cluster of cluster set  $\{C\}$ .

Third, evaluate features  $M_{prob}$ ,  $M_{dist}$  and  $V_{ratio}$  of each cluster  $c$  by (6).

Forth, remove markings. Sample multiple typical markings as a reference object  $\psi_{ref}$ , and evaluate distribution features  $M'_{prob}$ ,  $M'_{dist}$  and  $V'_{ratio}$  of  $\psi_{ref}$  by (6). Make each cluster  $c$  as the target object  $\psi_{tar}$  and evaluate total difference  $\lambda$  between  $\psi_{ref}$  and  $\psi_{tar}$  by (7), (8) and (9). If  $\lambda < 0.5$ , then  $c$  is a marking and remove it from set  $\{C\}$ .

Fifth, merge the remaining clusters of  $\{C\}$  according to their distances. The specular areas of a solder joint may be not continuous because of the changes in surface normal and light source position. It makes the highlight pixels of a solder joint are usually classified into multiple clusters, thus leads to recognition errors. We alleviate this problem by merging adjacent clusters before removing via-holes.

Finally, remove via-holes. Sample some typical via-holes as the reference object, and then remove via-holes from  $\{C\}$  in the same way as step forth. Now the remaining clusters of  $\{C\}$  only contain solder joints.

Fig.4 shows solder joints extraction results of PCB images  $I_a$  and  $I_b$ . In Fig.4 (a1) and (a2), all markings of  $I_a$  and  $I_b$  have been removed and the merging clusters have been merged. Fig.4 (b1) and (b2) show the results that via-holes have been removed and the remaining clusters of  $\{C\}$  only contain solder joints.

#### 4. Clustering solder joints and extracting their protective coatings

We have found that the color distribution of solder joints shows regularity. The regularity can be summed up as follows: for each solder joint, the color distribution of a part or a whole highlight region of a solder joint is changed in a sequence of {R, B, G} in the direction from protective coating to solder joint. According to this principle, we can determine the direction from solder joint to the protective coating that it is connected to. The determined directions offer important clues for extracting protective coatings. The main procedure is as follows: first, determine the direction of each solder joint based on above regularity (see section 4.1). Second, cluster all solder joints of each component by directions of each component and the gray level interval of PCB's background. Finally, the protective coating of each component is

extracted according to the positions of all connected solder joints (see section 4.2).

#### 4.1. Determining the direction of each solder joint

The components mount to the PCB along a horizontal ( $U$  axis) or vertical ( $V$  axis) direction. So, first of all, we determine the axis of color distribution. And then, we determine the direction of axis.

##### 4.1.1. Determining the axis of color distribution

The color distribution is changing with its axis. Therefore, the dispersion of color distribution along with axis is greater than the dispersion along a line perpendicular to the axis. We define the dispersion by the coefficient of variance. Suppose that the size of a solder joint is  $w \times h$ , and  $\gamma_u$  and  $\gamma_v$  are defined as the mean coefficients of variance of data that sample along with the  $U$  and  $V$  axes respectively. Symbol  $\phi_{axis}$  denotes the direction of axis. In order to determine  $\phi_{axis}$ , first, we sample  $h$  rows of data separately from  $U$  and  $V$  channels along with the  $U$  axis. The coefficient of variance of each row of data is evaluated by (10) and stored into arrays  $\rho_{uL}$  and  $\rho_{uU}$  respectively, and  $\rho_{uL}$  and  $\rho_{uU}$  with the length  $h$ .

$$\mu = \left( \sqrt{\sum_{i=1}^n (D(i) - \bar{D})^2 / n} \right) / \bar{D} \quad (10)$$

where  $\mu$  denotes coefficient of variance;  $D$  and  $\bar{D}$  denote the sample data and its mean value respectively;  $n$  expresses the number of sample data. After that, we sample  $w$  columns of data separately from  $U$  and  $V$  channels along with the  $V$  axis. The coefficient of variance of each column of data is evaluated by (10) and stored into arrays  $\rho_{vL}$  and  $\rho_{vU}$  respectively with the length  $w$ .  $\gamma_u$  and  $\gamma_v$  are evaluated by

$$\gamma_u = \sum_{i=1}^h \sqrt{\rho_{uL}(i)^2 + \rho_{uU}(i)^2} / h, \quad \gamma_v = \sum_{i=1}^w \sqrt{\rho_{vL}(i)^2 + \rho_{vU}(i)^2} / w \quad (11)$$

If  $\gamma_u \geq \gamma_v$ ,  $\phi_{axis} = 0$ , else  $\phi_{axis} = 1$ . Where 0 and 1 denote the color distribution along with the  $U$  and  $V$  axes respectively.

##### 4.1.2. Determining the direction of axis

According to the color distribution regularity mentioned above, we define the direction of axis by four steps: First of all, sample color distribution data along with the determined axis; Then, filter the useless data; After that, select some

valid data from the sample data along with the forward and reverse directions of axis respectively, and fit them by Dose-response curve; Finally, determine the direction of axis by analyzing the fitting parameters. Implementation details are described as follows:

*Step 1: sample color distribution data.* We decrease brightness and sharpen the contrast of original PCB image by (1) for enhancing the features of color distribution. At the same time, channels R, B and G are encoded in integer 3, 2, and 1 respectively. For each pixel included in  $w \times h$  region, we compare its three channels and store its channel code with maximal value into a  $w \times h$  matrix  $M_p$ . Then, the color distribution data is evaluated and stored in a vector  $V_{mean}$ . If  $\phi_{axis} = 0$ , then the length of  $V_{mean}$  is  $w$ ; otherwise, the length of  $V_{mean}$  is  $h$ . Color distribution data is evaluated by the following rules:

$$\begin{aligned} \text{If } \phi_{axis} = 0, \text{ for } i = 1 \text{ to } w, V_{mean}(i) &= \sum_{j=1}^h M_p(i, j) / h, \\ \text{If } \phi_{axis} = 1, \text{ for } i = 1 \text{ to } h, V_{mean}(i) &= \sum_{j=1}^w M_p^T(i, j) / w \end{aligned} \quad (12)$$

Fig.5 (a1)-(a3) shows three typical solder joints. In Fig.5 (b1)-(b3), the black curve shows the color distribution data.

*Step 2: filter the useless data.* In some cases, the color distributions of some solder joints become more complex and make their color distribution curves not smooth enough, because of the variances in surface normal and angle of illumination incidence. It may lead fitting to a failure in the next step. Therefore, we remove some useless data in order to smooth the curve effectively. First, we map each data  $V_{mean}(i)$  to three intervals based on the following rules and save it into  $V'_{mean}(i)$ .

$$V'_{mean}(i) = \begin{cases} 1, & \text{if } V_{mean}(i) \in [1, 1.5) \\ 2, & \text{if } V_{mean}(i) \in [1.5, 2.3) \\ 3, & \text{if } V_{mean}(i) \in [2.3, 3.0] \end{cases} \quad (13)$$

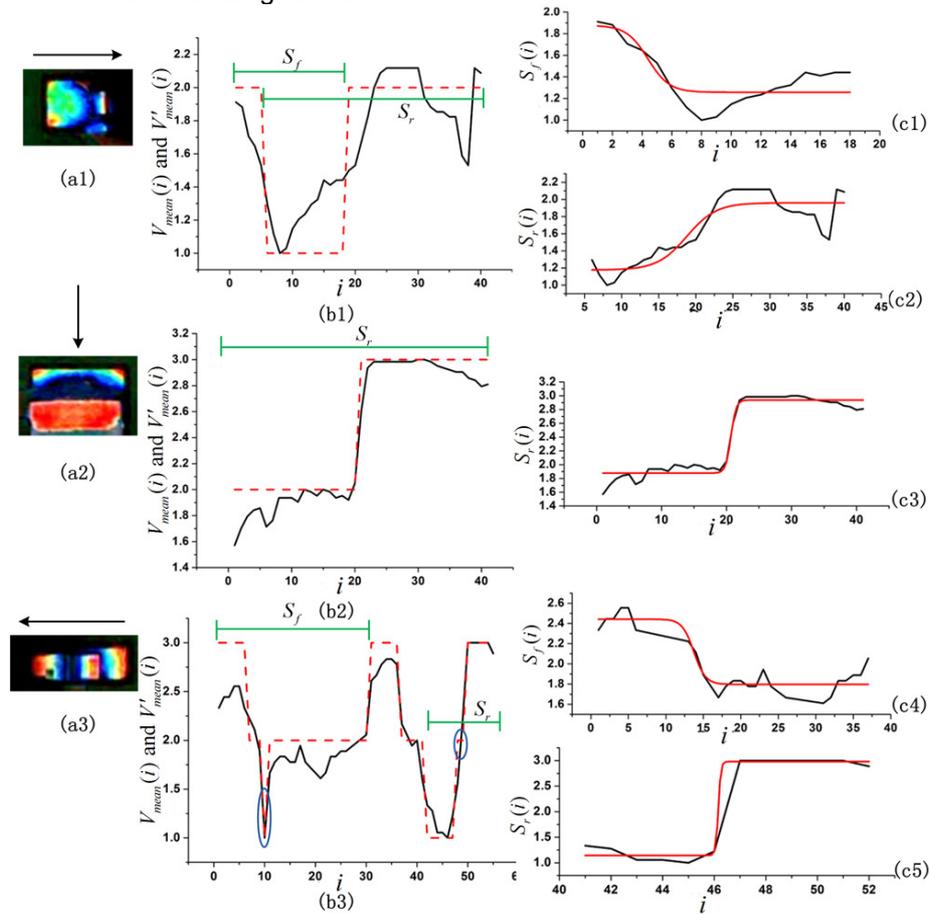
In Fig.5 (b1) to (b3), the red dotted line shows the mapped data. Furthermore, we check the contiguous data with the same value in  $V'_{mean}$ . If the scale of contiguous data is less than 2, this kind of data is useless and ought to be removed. For example, in Fig.5, the 11<sup>th</sup>, 12<sup>th</sup> and 48<sup>th</sup> data in (b3) are useless. They are enclosed with blue ellipses.

*Step 3: fit color distribution data.* The color distribution curve can be fitted by some growth models, such as dose-response function. We define  $S_f$  and  $S_r$  as two data segments. Check data along with the forward direction of axis, if  $V'_{mean}(i) \geq V'_{mean}(i+1)$ , add  $V'_{mean}(i)$  into  $S_f$ . Check data along with the reverse direction of axis, if  $V'_{mean}(i) \geq V'_{mean}(i-1)$ , add  $V'_{mean}(i)$  into  $S_r$ . Data segment with length is less than four is invalid because dose-response function has four parameters. In Fig.5 (b1)-(b3), the picked data segments are marked by the

green lines. Where  $S_f$  of (b2) is invalid. After that, we fit  $S_f$  and  $S_r$  by dose-response function. A standard dose-response function is defined by four parameters

$$y = A_1 + (A_2 - A_1) / (1 + 10^{(LOGx0-x)p}) \quad (14)$$

where  $A_1$  is the baseline response (Bottom);  $A_2$  is the maximum response (Top);  $p$  is the slope and  $LOGx0$  is the drug concentration that provokes a response halfway between baseline and maximum. In Fig.5 (c1) to (c5), the red curve shows fitting result.



**Fig. 5.** Color distributions of three typical solder joints. (a1) to (a3) are three typical solder joints; (b1) to (b3) are color distribution curves; (c1) and (c2) are fitting curve of (a1); (c3) is a fitting curve of (a2); (c4) and (c5) are fitting curves of (a3).

*Step 4: determine the direction of axis based on the fitted parameters. A variable  $\phi_{dir}$  denotes the direction of axis. It uses 0 to indicate the reverse of*

axis, 1 for the forward of axis. The direction of axis is determined according to the following three cases:

In case 1: Only one of  $S_f$  and  $S_r$  is valid. If  $S_f$  is valid,  $\phi_{dir} = 0$ , otherwise,  $\phi_{dir} = 1$ . In (b2), only  $S_r$  is valid, so,  $\phi_{dir} = 1$ , the direction of (a2) is along the forward of axis (see black arrow in (a2)).

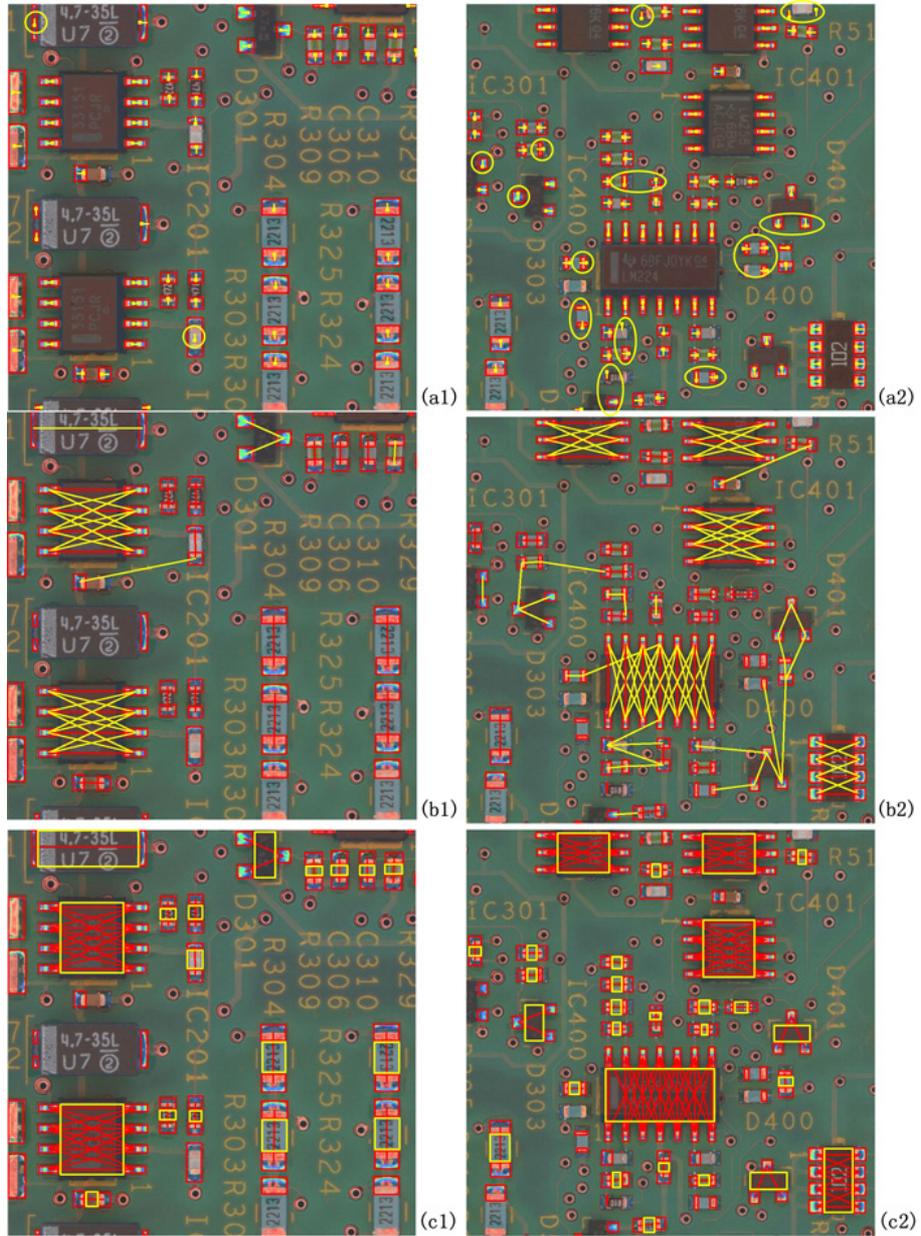
In case 2: Both  $S_f$  and  $S_r$  are invalid.  $\phi_{dir}$  is determined by comparison with the length of two data segments. If the length of  $S_f$  is larger than that of  $S_r$ , then  $\phi_{dir} = 0$ , otherwise,  $\phi_{dir} = 1$ .

In case 3: Both  $S_f$  and  $S_r$  are valid. Over all, there are five kinds of colour sequence. Through observing and analyzing a great deal of color distributions, we set the priority for each sequence.  $\phi_{dir}$  is determined mainly through the priorities. Sequence is evaluated by the fitted parameters  $A_1$ ,  $A_2$  and some constraints. Table 1 shows five sequences and their corresponding priority and constraints.

**Table 1.** Sequence type, priority and constraints

Sequence	Priority	Constraints
{R, B}	3	$A_1 \in [1.5, 2.3) \wedge A_2 \in [2.3, 3.0]$
{R, B, G}	3	$A_1 \in [1, 1.5) \wedge A_2 \in [2.3, 3.0] \wedge (\exists j)(V_{mean}(j) \in [1.5, 2.3)$
{B, G}	2	$A_1 \in [1, 1.5) \wedge A_2 \in [1.5, 2.3)$
{R, G}	1	$A_1 \in [1, 1.5) \wedge A_2 \in [2.3, 3.0] \wedge (\forall j)(V_{mean}(j) \notin [1.5, 2.3)$
others	0	

We evaluate  $L_f$  and  $L_r$ , the sequence priority of  $S_f$  and  $S_r$  respectively. If  $L_f > L_r$ , then  $\phi_{dir} = 0$ . If  $L_f < L_r$ , then  $\phi_{dir} = 1$ . If  $L_f = L_r$ ,  $\phi_{dir}$  is determined by comparison with the length of  $S_f$  and  $S_r$ . For example, in the case of Fig.5 (b1),  $L_f = L_r = 2$ , but the length of  $S_f$  is less than  $S_r$ . Therefore,  $\phi_{dir} = 1$ , the direction of (a1) is along the forward of axis (see black arrow in (a1)). In the case of Fig.5 (b3),  $L_f = 3$  and  $L_r = 1$ . Therefore,  $\phi_{dir} = 0$ , the direction of (a3) is along the reverse of axis (see black arrow in (a3)). Fig.6 (a1) and (a2) show the results of determining the direction of solder joints of  $I_a$  and  $I_b$ , the direction of solder joint is represented by the yellow arrow. Observing the results, we can know that many directions are determined legitimately, and there are still some minor mistakes (error directions are marked by the yellow ellipses). The errors are caused by two reasons: one is the highlight area of a solder joint is incomplete, and it's the main reason; another is the color distribution of some solder joints is so special that we cannot determine its color sequence correctly. However, this is rarely the case.



**Fig. 6.** Clustering and extracting results of  $I_a$  and  $I_b$ . (a1) and (a2) show the results after determining the direction of solder joints. (b1) and (b2) show the results after clustering solder joints in the first step; (c1) and (c2) show the results after clustering solder joints in the second step and extracting protective coatings.

#### 4.2. Implement details of clustering solder joints and extracting protective coatings

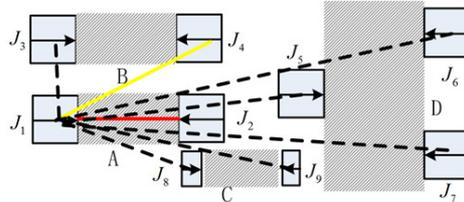


Fig. 7. Some constraints of connections.

The clustering algorithm is divided into two steps: In the first step, we take all solder joints as nodes of a graph, and find all connections between nodes based on the direction of solder joints under some constraints. In the second step, some invalid connections are removed by using the thresholds of PCB background. Finally, the protective coatings are extracted based on the position of clustered solder joints. Some details of clustering algorithm are introduced as follows:

*Step 1: Find all connections between solder joints.* Suppose that there are two solder joints  $J_i$  and  $J_j$ , their centers are  $(u_i, v_i)$  and  $(u_j, v_j)$  respectively, their sizes of highlight area are  $(w_i, h_i)$  and  $(w_j, h_j)$  respectively.  $C(i, j)$  denotes a connection from  $J_i$  to  $J_j$ . Then  $C(i, j)$  is determined by the following constraints: (1) Angle constraint. Compute the angle between  $J_i$  and  $J_j$  along the direction of  $J_i$ , and the angle within  $[-45^\circ, 45^\circ]$ . (2) Size constraint. Observing many PCB images, we find that all solder joints, which belong to the same component, have the similar size of highlight area. As a result, the relative error  $|w_i - w_j| / w_i$  and  $|h_i - h_j| / h_i$  ought to be less than  $\varepsilon_1$ . In our experiment, we set  $\varepsilon_1 = 0.35$ . (3) Shortest distance constraint. There are two kinds of the shortest distances. One is along the direction of  $J_i$ , we denote it as  $D_{axis}$ . Another is satisfy constraint (1) but isn't along the direction of  $J_i$ , we denote it as  $D_{normal}$ . If all solder joints of a component are distributed symmetrically, then each solder joint of this component has  $D_{axis}$ . Otherwise, each solder joint of this component has  $D_{normal}$ . For example, in Fig.7, all solder joints of components A, B and C have  $D_{axis}$ , all solder joints of component D have  $D_{normal}$ . Consequently, the distance between  $J_i$  and  $J_j$ ,  $\sqrt{(u_i - u_j)^2 + (v_i - v_j)^2}$  must be less than  $D_{axis}$  or  $D_{normal}$ .

Furthermore, because of the wrong direction of solder joints, not all determined connections are valid. Consequently, we define that if there is a connection  $C(i, j)$ , and  $J_j$  is located along the direction of  $J_i$ , and the



where  $N$  is the number of pixels, which are within the connection area and also within PCB background. If  $N \leq \varepsilon_3$ , then  $C(i, j)$  is valid. In our experiment, we set  $\varepsilon_3 = 0.35$ .

Finally, we extract protective coatings. Clustering solder joints according to the valid connections. Each cluster contains all solder joints of one of components. So, the approximate position of a protective coating is evaluated according to the positions of all solder joints in a cluster. Fig.6 (c1) and (c2) show results of clustering solder joints in the second step (see red lines) and extracting protective coatings (see yellow rectangles).

Based on the experimental results, we regard that the introduced algorithm can extract most of the electronic components accurately, except for a few special cases: (1) some components at the sides of image may be incomplete because of the image real-estate constraints. (2) specular area of some solder joints cannot be correctly detected.

## 5. Conclusion

We present an effective method for extracting components of PCB. The advantages of this method are as follows: First, in order to extract solder joints, we detect all highlight areas first, and then recognize and remove invalid highlight areas which are mainly markings and via-holes. Because the color distribution of markings and via-holes is much simpler than that of solder joints, the proposed approach has higher extraction accuracy in most PCB images. Second, recognizing color object is always a bottleneck, particularly the variation of color within a certain range. We summarize three fundamental features to describe the color distributions of reference object and target object in  $(u', v')$  chromaticity coordinate. So, invalid specular areas can be correctly recognized by comparing the distribution features. Finally, the sequence of the color distribution as a new clue has been applied to clustering solder joints. This improves the speed and accuracy of clustering. According to the experimental results, the developed algorithm can effectively recognize most of the chip components. The process of modeling all highlight pixels by GMM is time consuming. Therefore, future research may focus on improving the efficiency of algorithm.

## 6. Acknowledgement

This project is partly supported by National Science Fund for Creative Research Groups (60821062) 973 Plan (2006CB303105) and NSF of China (60873136)

## 7. References

1. Bartlet, S.L., Besl, P.J., Cole, C.L., Jain, R., Mukherjee, D., Skifstad, K.D., Automatic solder joint inspection, IEEE Trans. PAMI 10 (1), 31-41 (1988).
2. Bishop, C., Neural Networks for Pattern Recognition. Oxford Uni. Press, (1995).
3. Horng-Hai Loh, Ming-Sing Lu, Printed circuit board inspection using image analysis, *Indust. Automat. and Control: Emerging Technologies*, 673-677 (1995).
4. Chun-Ho Wu, Da-Zhi Wang, A particle swarm optimization approach for components placement inspection on printed circuit boards, *J Intell Manuf*, (2008).
5. WANG Tao, HU Shi-Min, SUN Jia-Guang. Image retrieval based on color-spatial feature[J]. *Journal of Software* ,13(10):2031-2036 (2002).
6. Kim, T.H., Cho, T.H., Moon, Y.S., Park, S.H., Automatic Inspection of Solder Joints Using Layered Illumination, IEEE DOI 9600, pp. 645-648 (1996)
7. Kim, T.H.[Tae-Hyeon], Cho, T.H.[Tai-Hoon], Moon, Y.S.[Young Shik], Park, S.H.[Sung Han], Visual inspection system for the classification of solder joints, *PR*(32), pp. 565-575 (1999).
8. Lee, S.W., Bajcsy, R., Detection of Specularity Using Colour and Multiple Views, *IVC*(10), pp. 643-653 (1992).
9. Tao Lin-Mi, Peng Zhen-Yun, Xu Guang-You. The Features of Skin Color,[J]. *Journal of Software* ,Vol.12, No.7 (2001).
10. Render, R.A., WalkerH.F., Mixture densities, maximum likelihood and the em algorithm. *SIAM Review*, 26(2):195-239, (1984).
11. Shafer,S. A. "Using color to separate reflection components," *Color Res. Appl.*, vol. 10, pp. 210–218, (1985).
12. Lin,S. , Li,Y. , Kang,S. , Tong,X. and Shum,H. "Diffuse-specular separation and depth recovery from image sequences," in *Lecture Notes in Computer Science*, vol. 2352, New York: Springer-Verlag, (2002).
13. Teoh, E.K., Mital, D.P., Lee, B.W. and Wee, L.K., Automated visual inspection of surface mount PCBs, in *Proceedings of IEEE 16th Annual Conference Industrial Electronics Society*, Singapore, pp. 576–580 (1990).
14. Yen, J.C., Chang, F.J. and Chang, S., A new criterion for automatic multilevel thresholding. *IEEE Trans. Image Process. vIP-4*. 370-378, (1995).

**Zhou Zeng** System Analyst. PhD candidate in Department of Computer Science and Engineering, Shanghai Jiaotong University, China. Reserach interests include computer graphics, image processing and computer vision.

**Li Zhuang Ma** Full Professor, PhD tutor, and the head of Digital Media Technology and Data Reconstruction Lab. at the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China since 2002. His research interests include computer aided geometric design, computer graphics, scientific data visualization, computer animation, digital media technology, and theory and applications for Computer Graphics, CAD/CAM.

**Zuo Yong Zheng** PhD candidate in Department of Computer Science and Engineering, Shanghai Jiaotong University, China. Reserach interests include computer graphics and rendering.

Received: April 28, 2009; Accepted: July 04, 2009.