PCB Inspection for Missing or Misaligned Components using Background Subtraction

K. Sundaraj
University Malaysia Perlis
School of Mechatronic Engineering
02600 Jejawi - Perlis
MALAYSIA
kenneth@unimap.edu.my

Abstract: Automated visual inspection (AVI) is becoming an integral part of modern surface mount technology (SMT) assembly process. This high technology assembly, produces printed circuit boards (PCB) with tiny and delicate electronic components. With the increase in demand for such PCBs, high-volume production has to cater for both the quantity and zero defect quality assurance. The ever changing technology in fabrication, placement and soldering of SMT electronic components have caused an increase in PCB defects both in terms of numbers and types. Consequently, a wide range of defect detecting techniques and algorithms have been reported and implemented in AVI systems in the past decade. Unfortunately, the turn-over rate for PCB inspection is very crucial in the electronic industry. Current AVI systems spend too much time inspecting PCBs on a component-by-component basis. In this paper, we focus on providing a solution that can cover a larger inspection area of a PCB at any one time. This will reduce inspection time and increase the throughput of PCB production. Our solution is targeted for missing and misalignment defects of SMT devices in a PCB. An alternative visual inspection approach using color background subtraction is presented to address the stated defect. Experimental results of various defect PCBs are also presented.

Key–Words: PCB Inspection, Background Subtraction, Automated Visual Inspection.

1 Introduction

Inspection has been an integrated process in human related activities since the dawn of barter trade up to the current production of high technology devices. The advancement of technology in the SMT industry is pushing human visual inspection to the twilight edge. This, coupled with fatigue and inconsistency, has made AVI systems the best choice as replacements [1]. In a SMT assembly process, AVIs are often used to check the functionality of a PCB assembly [2]. Today, PCBs have evolved to become more complex in design, multi layered and are assembled with increasingly miniaturized components. This has made quality control of PCBs more challenging and demanding. Currently, AVIs combined with in-circuit-testers (ICT) are the most prominent systems that are used to check the functionality of a PCB assembly. This is because AVI systems provide better quality control at lower costs [3]. Inspection systems placed at appropriate sections along an assembly line will reduce rework cost which would eventually provide better results during the electrical testing phase [4]. Many authors as cited in the vast reference list of [5] have repeatedly emphasized the importance of developing techniques and algorithms for an automatic inspection system in the electronic industry. However, to the best of our knowledge, there is no one-stop single AVI system to detect all known visual defects on a PCB assembly for a high-volume manufacturing assembly line. Components on an assembled PCB come in a wide range of sizes, colors, shapes and uniqueness. These variations seems to be the major bottleneck in most of the existing defect detection techniques.

Variations in the type of defects present on a PCB assembly and the numerous fabrication technology of electronic components have made development of AVI systems a challenging issue in the last few decades. In the development of such systems, several studies [6] [7] has outlined missing components as one of the top five common defects and the need for AVI systems to have suitable algorithms to detect these defects. It has been found in these studies that 20% of defects were missing electronic components, missing ball grid arrays (BGA) was at at 5% and missing in-chip devices were at 11%. This concern lead [8] to address the detection of missing and misalignment of components in their work. The technique applied in this work was pixel frequency summation of gray scale values. This method was found to be reasonably effective if inspection was done specifically for a single type and a fixed color component. Unfortunately, PCB components come in a varying range of colors and types. This condition was overlooked in their work causing high false call rates. As a remedy, [9] developed a PCB assembly inspection system employing color image processing with multi-lighting that switches the type of lighting in accordance with the type of inspection. This method would not be feasible in a high-volume manufacturing line, given that their time for inspecting a single mounted part to be around 40ms. A typical PCB assembly has approximately 800 to 1200 components, which would take an average of about 40s for the complete inspection of mounting parts. This does not include inspections of joint, wrong part, wrong polarity, etc. In their work also, the varying PCB background and component colors were not discussed. Using a colored slit to detect missing components is suitable for a specific component with constant color. It is not suitable for multiple components with a wide range of varying colors. [10] used thresholding to create a binary image to detect missing and misaligned components. But static thresholding was found to be unsuitable in dynamic environments. This method invited higher false call rates for the increasingly complex and dense PCB assemblies with varying component colors. Surface defects was analyzed using principle component analysis of images in [11] and a hardware optimized image processing algorithms which can be used in a PCB inspection industry is proposed in [12].

To the best of our knowledge, within the semiconductor industry, automated non-contact inspection of PCB assemblies using image processing techniques has not been done using color background subtraction. In this paper, we present the feasibility of such a procedure to detect missing or misaligned components.

2 Background Subtraction

In many computer vision applications, identifying objects of interest in a 2D image is of paramount importance. A common approach to this problem is background subtraction. It basically consists of comparing a current captured image with an estimated reference image (with no objects of interest) and removing common areas. In order to do this, two aspects of image processing are required; background modeling and subtraction technique. During run-time, these aspects are combined and performed by the background subtraction algorithm. However, in many applications, the occurrence of a static background is very rare, at least from the camera sensors point of view. Hence, background subtraction algorithms should be flexible to the following dynamic situations:

- Illumination changes that are gradual or sudden,
- Motion changes that are internal or external,
- Geometry changes that are intentional or unintentional.

These types of dynamic situations are also present in a visual system used for PCB inspection. In such a system, illumination is generally provided by a fixed light source. A camera then zooms in and captures various sections of the PCB assembly to be processed. This can be done by either moving the PCB relative to the camera or vice versa. But due to the fact that the camera sensors are imperfect, the captured signals for a fixed section of the PCB assembly vary over time; meaning that they change from one PCB assembly to another. Inspection conditions are also dynamic; quality of lightings and surrounding surface reflectivity degrade and becomes non-uniform over time. These can cause variations in camera sensor readings. Small vibrations due to the motion of the mountings and stages will cause the image captured over time to be displaced back-and-forth vertically and horizontally by about 1 to 2 pixels. This may cause the camera sensor readings to change intermittently. In some case, the components in a PCB assembly may not be missing or misaligned, but merely not in the supposed geometrical form; for example not round but slightly ellipsoid or not straight but slightly curved. These situations must be considered in the development of a background subtraction algorithm for PCB inspection.

The process of considering these changes in an image is called background modeling. In this stage, the changes that might occur *a priori* in an image is mathematically modeled at various levels; pixel, regions, temporal, etc. Once background modeling has been completed, the result is normally stored as an image called a background reference image. This reference image is then used during run-time, where the current image is then subtracted from the reference image to detect displaced, missing or new objects. Depending on the application, the background reference image can be updated using information from each incoming frame during run-time or otherwise.

Several background subtraction algorithms have been proposed in the recent literature. All of these methods try to effectively estimate the background model from the temporal sequence of the frames. One of the simplest algorithms is frame differencing [13]. The current frame is subtracted from the previous frame. This method was extended such that the reference frame is obtained by averaging a period of frames [14] [15] also known as median filtering. A second extension applied a linear predictive filter to the period of frames [16] [17]. A disadvantage of this method is that the coefficients used (based on the sample covariance) needs to be estimated for each incoming frame, which makes this method not suitable for real-time operation. [18] and [19] proposed a solution to this using a much higher level algorithm for removing background information from a video stream. A further computationally improved technique was developed by [20] and it is reported that this method is successfully applied in a traffic monitoring system by [21]. In these types of time averaging algorithms, the choice of temporal distance between frames becomes a tricky question. It depends on the size and speed of the moving object. According to [22], background subtraction using time averaging algorithms, at best, only tell where the motion is. Though this is the simplest algorithm, it has many problems; interior pixels of a very slow-moving object are marked as background (known as the aperture problem) and pixels behind the moving object are cast as foreground (known as the ghost effect problem). Multi-model algorithms were then developed to solve this shortcoming. A parametric approach which uses a single Gaussian distribution [23] or multiple Gaussian distributions (MoG) [24] [25] can be found in the literature. Various improvements techniques like the Kalman filter to track the changes in illumination [26] [27], updating of Gaussian distributions [28] [29], inclusion of image gradient [30] and the modeling and detection of shadows and highlights [31] [32] have been done to improve the accuracy of background subtraction. Non-parametric approaches have also been attempted in [33] and [34]. These approaches use a kernel function to estimate the density function of the background images.

3 Experimental Setup

In this work, a simple setup was made using a digital camera connected to a desktop computer as shown in Fig. 1. The lighting for this setup was devised as a circular shape taking the cue from most AVI systems in the industry. This shape reduces shadow effects and provides good nominal illumination for image processing. The PCB was placed on a rigid optical table. Alignments and right-angle determination was made by using water levelers. The industrial camera was placed overhead normal to the surface under test. The digital CCD camera used was a Marlin FO33C progressive scan with a macro zoom lens. It was however placed at a particular height to the surface under test to give the required field of view (FOV). This was to simulate an industry AVI system. The computer used was a Pentium IV 1.8 GHz PC operating under a Windows XP environment to capture and process the incoming frames. The image processing software was programmed using Microsoft Visual C++ 6.0. All triggering and synchronizing of cameras were hardware controlled. The lighting conditions in this setup was however hand controlled. Given this setup, by controlling the type of lighting and the type of camera used, it is possible to perform a suitable test to model the sensors of the camera that will be incorporated into a suitable mathematical model which can be employed for the inspection of PCBs for the detection of missing or misaligned components.

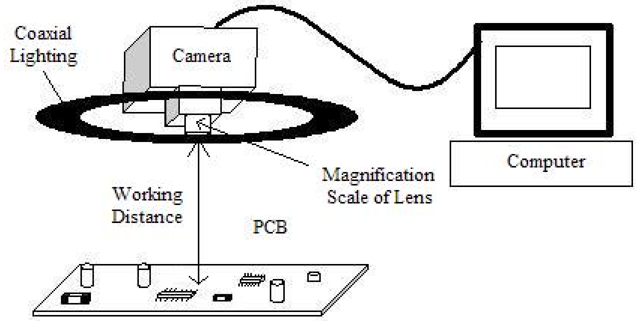


Figure 1: A simple image acquiring setup for PCBA inspection.

4 Mathematical Model

The main techniques for background modeling can be classified into two main categories; non-recursive and recursive. Non-recursive techniques require a huge storage space and hence do not seem to be an interesting option. Hence, we opted for a recursive technique. Within the various recursive techniques, we decided to use the **Mixture of Gaussian** (MoG) method. This is probably the most widely used technique. A density function that is a combination of different Gaussian components, each with mean intensity, standard deviation and appropriate weights, is maintained for each pixel. The reason for this choice can be understood from an empirical point of view. During the tests of the cameras used, the variation of the sensor readings taken from the camera at each pixel was observed. An example of this observation is shown in the histogram as seen in Fig. 2. From this histogram, we can see that the variations can be modeled as a Gaussian density function. More precisely, in this case, only a single Gaussian function is required due to the single peak. We note that, effectively this choice is a function of the sensor readings and this in turn is related to the chosen test environment and the type of camera used. A second reason for this choice is related to the cost function of PCB inspection. This depends on the number of components, size of the board and the type of components. In view of this, we believe that a simple (low computational cost) but yet reasonable accurate (low false calls) model is required to meet the high-volume throughput expected in the PCB assembly inspection industry. By considering all these constraints, we believe that the MoG method should perform well.

In the MoG technique, we first need to decide on the components of the pixel vector which are going to be observed. From the sensors of the camera, the output vector at pixel level is as follows:

$$I = (R, G, B) \tag{1}$$

which separates the red (R), green (G) and blue (B) channels coded in a (8:8:8) ratio. Thus, we have 8 bits of color information for each of the three channels. This will become our vector which we will learn by observation. The variation in the observed vectors at pixel I, can be modeled separately by a mixture of K Gaussians. The probability P then of a pixel I belonging to the background is given by:

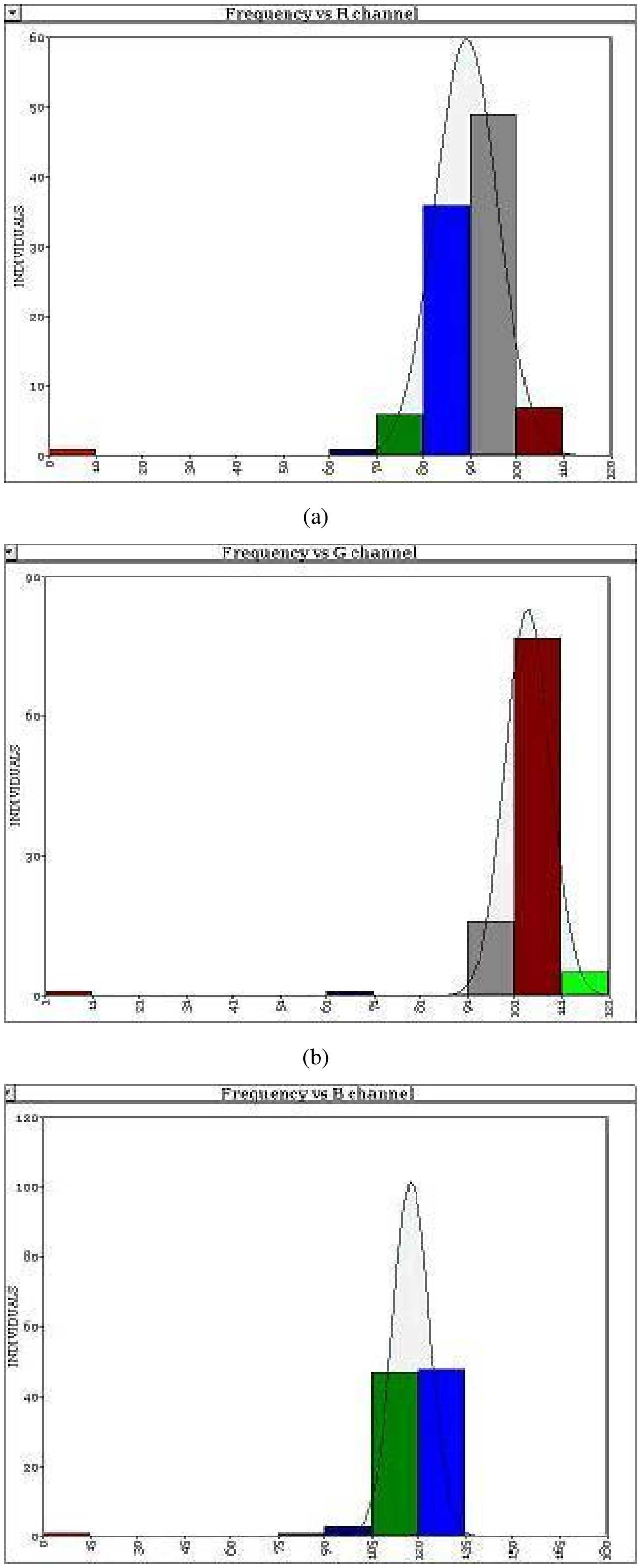
$$P(I) = \sum_{i=1}^{K} \omega_i \, f(\, I, \; \mu_i \,, C_i \,) \tag{2}$$

where K denotes the number of Gaussian, μ the mean value of the Gaussian and C the covariance matrix associated with this Gaussian. Since we have chosen a single Gaussian to model the background, $K = i = \omega = 1$. This model is will be used to learn the input vector from the camera sensors during run-time.

Having chosen a suitable mathematical model, we can proceed to describe the background subtraction process during run-time. We assume that all required memory for image processing has been allocated sufficiently. From equation (2), we can see that in order to use a Gaussian density function, we need to obtain the mean value μ and the covariance matrix C for each pixel. In our implementation, this is done in three stages. Each stage is outlined in the following sections.

4.1 Background Learning

In this stage, for each section of a **perfect** PCB assembly that is to be processed for missing or misaligned components, we capture N continuous frames for learning purposes. Let the observed vector at pixel level be denoted as I. At pixel level, we store the number of samples n, the sum of the observed vector a and the sum of the cross-product of the observed vector b.



(a)

(b)

(c)

Figure 2: The observed R, G and B values of a randomly sampled pixel tabulated in a histogram.

$$n = \sum_{i=1}^{N} 1 \tag{3}$$

$$a = \sum_{i=1}^{N} I_i \tag{4}$$

$$b = \sum_{i=1}^{N} (I_i \times I_i^T) \tag{5}$$

This stage will be defined as the learning phase and is necessary for initialization. From our experiments, about 100 frames is necessary to sufficiently learn the variations in the background. This corresponds to about 3 to 4 seconds of initialization.

4.2 Parameter Estimation

At the end of the learning phase, the required variables for the Gaussian model needs to be calculated. They are the mean value μ, the covariance matrix C and the inverse covariance matrix C^{-1}. Each of these variables are computed at pixel level in a very efficient manner using optimized codes. The inverse covariance matrix is obtained from the inversion of the 3×3 C matrix.

$$\mu = \frac{a}{n} \tag{6}$$

$$C = \frac{1}{n}(b) - \frac{1}{n^2}(a \times a^T) \tag{7}$$

$$C^{-1} = inv(C) \tag{8}$$

4.3 Pixel Classification

Once the required variables of the model has been calculated, for each predefined section of an incoming PCB assembly, we can perform the discrimination between background (BG) and foreground (FG). Let us begin by expanding equation (2) in the case when $K = i = \omega = 1$.

$$\begin{aligned} P(I) &= f(\, I,\, \mu\,, C\,) \\ &= (\, 2\pi^{\frac{K}{2}} \times \sqrt{|C|}\,)^{-1} \\ &\quad \times exp\,\{\, (I-\mu)^T \\ &\quad \times C^{-1} \times (I-\mu)\,\} \end{aligned} \tag{9}$$

To compute this probability, it is not entirely necessary to evaluate the whole expression. The first term is a constant. Hence, the decision making process is streamed down to the following:

$$P(I) = f\{(I-\mu)^T \times C^{-1} \times (I-\mu)\} \tag{10}$$

This expression is popularly known as the Mahalanobis Distance. To evaluate it, a threshold ϵ is used to decide if a pixel belongs to the background (BG) or foreground (FG). The determination of ϵ is done empirically because this constant is dependent on the camera, lighting, mounting setup and the type of PCB being inspected.

$$\begin{aligned} P(I) &= BG \;\text{ if }\; (I-\mu)^T \times C^{-1} \times (I-\mu) > \epsilon \\ &= FG \;\text{ otherwise} \end{aligned} \tag{11}$$

These steps will enable us to decide if a pixel belongs to the background or otherwise. If a high percentage of pixels do not belong to the background, the presence of a missing or misaligned component has been detected. To illustrate this point, in the next section, we will present the experimental results from our prototype PCB assembly AVI system.

5 Experimental Results

A sample PCB assembly was used in our prototype AVI experimental setup as shown in Fig. 1 with a working distance of about 500mm and a lens magnification scale of $\times 3$. The PCB assembly was ensured to contain various components of different types, sizes, colors and fittings. During the experiment, firstly, the defect free PCB assembly was used in the learning process to obtain the reference background image. This corresponds to capturing about 100 frames of the PCB assembly board. A coaxial light source and lens were used in the experimental setup to minimize distortion of the images. After the learning phase, a similar PCB but with defect characteristics such as missing or misaligned components is mounted and its image captured. We note here that the both the used PCBs must be aligned in the same orientation with high accuracy. It is for this reason we used an optical table in our experiments. In the PCB inspection industry, this is trivial and is achieved with the help of positioning lasers. The newly captured image is the subtracted from the reference background image and the result is a binary mask denoting foreground (FG) and background (BG) pixels. A large presence of FG pixels will indicate some form of defect in the suspected location. The results presented in this section is arranged in the following manner whereby the top most image is the reference image that is used in the learning process, the center image is the captured image of an incoming PCB assembly with some form of defect and finally the bottom most image in our results is the processed image or the background subtracted image.

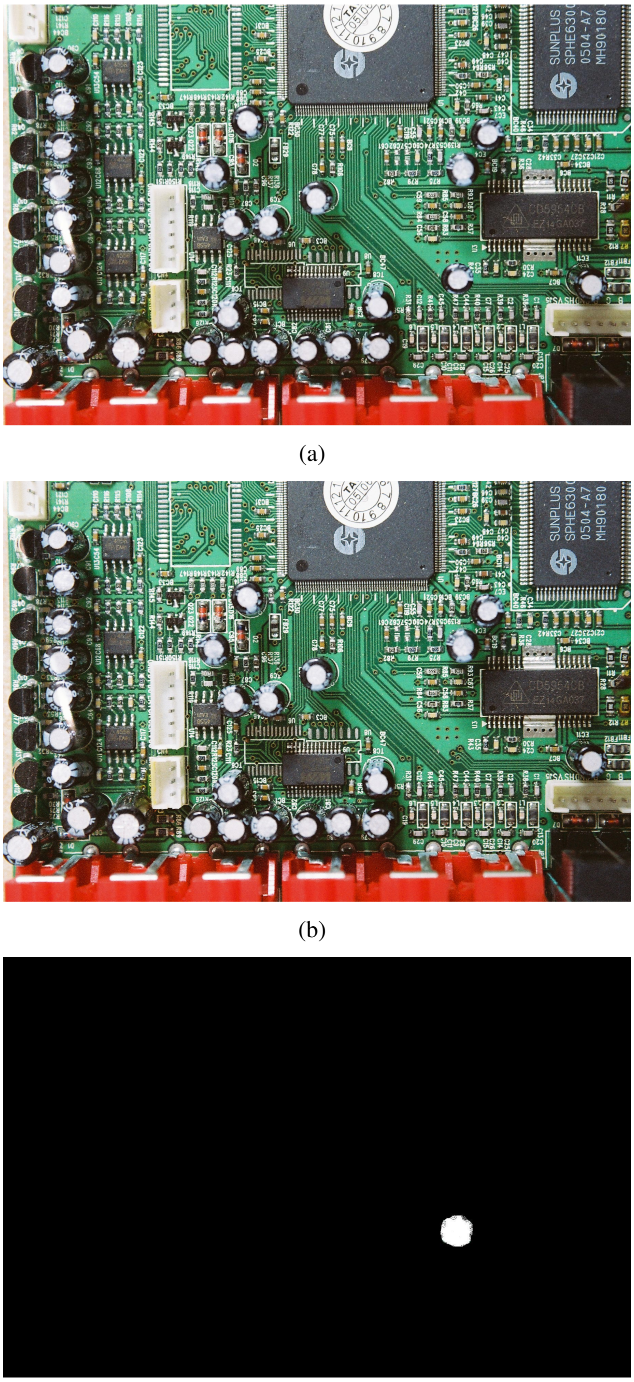
In the first experiment, as shown by the three images in Fig. 3, a capacitor was removed (missing component). The proposed defect detection technique has successfully identified the absence of this component. The three images in Fig. 4, shows the results of the second experiment. The first image with a perfect PCB assembly was learned. However, the incoming PCB assembly (second image) had a missing IC. Again, the proposed defect detection technique has successfully detected this absence. In the third experiment, as shown by the three images in Fig. 5, a capacitor was soldered in the wrong place (misaligned component). In this case, the background subtraction algorithm produced two white circular areas; the missing component and the misplaced component. In our final experiment, as shown by the three images in Fig. 6, an IC was displaced in the incoming PCB assembly to simulate a false soldering process. The result shows that this defect has been successfully identified.

A study was done to quantify the error in the proposed algorithm. In the case of missing components, we manually quantified the pixels that belong to the component from the defect PCB by counting the number of pixels of that particular component color and by comparing with the number of FG pixels in the background subtracted image. However, in the case of misaligned components, we studied the misalignment by comparing component pixel location (x, y) in the reference image, defect image and FG pixels in the background subtracted image of the PCB assembly. The experiments were repeated about 20 times with different lighting conditions and with different types of electronic components. The average percentage of error in the values obtained are tabulated in Table 1.

Table 1: Average error rate (%) in the conducted experiments.

Experiment	Error (%)
Missing Capacitor	7.4
Missing IC	5.6
Misaligned Capacitor	5.4
Misaligned IC	6.8

The results we have presented show that it is possible to find missing or misaligned component defects using background subtraction. Although some error is found in the results, a large number of FG pixels $(> 90\%)$ indicate some form of defect in the PCB assembly. In some cases, white noise can be present in the subtracted image, but these are removed by changing the threshold or by improving lighting conditions.

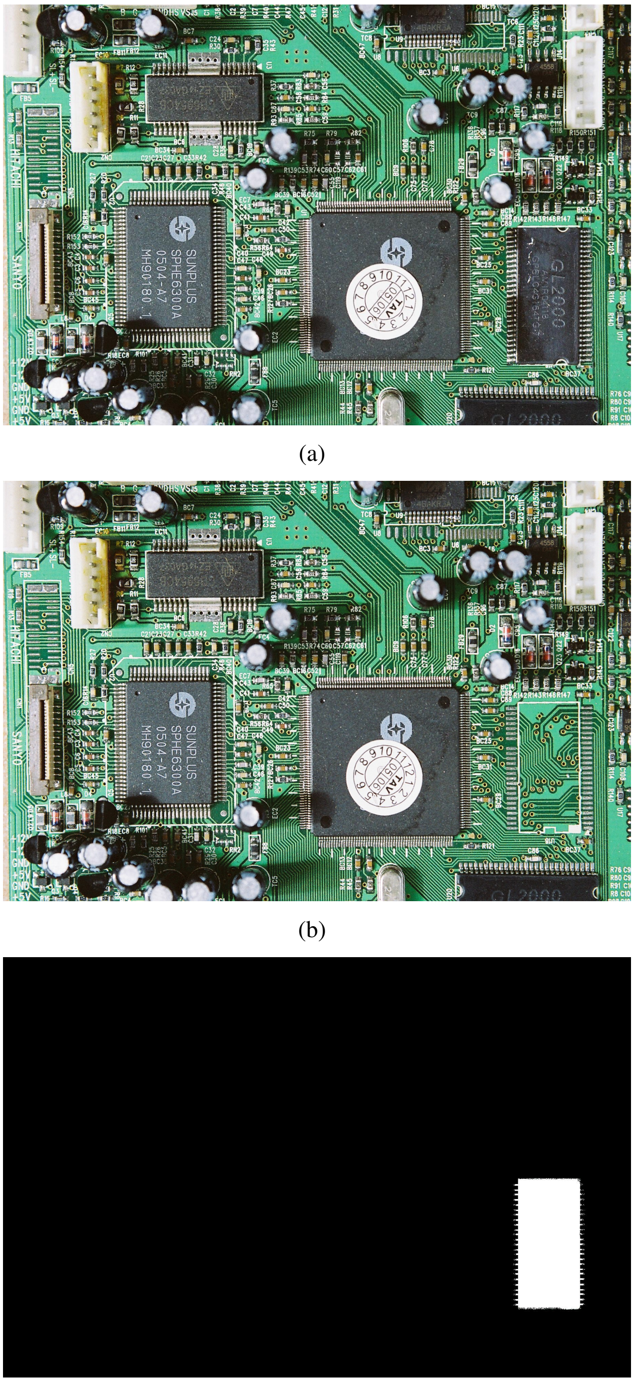


(a)

(b)

(c)

Figure 3: Detecting missing capacitors using background subtraction.

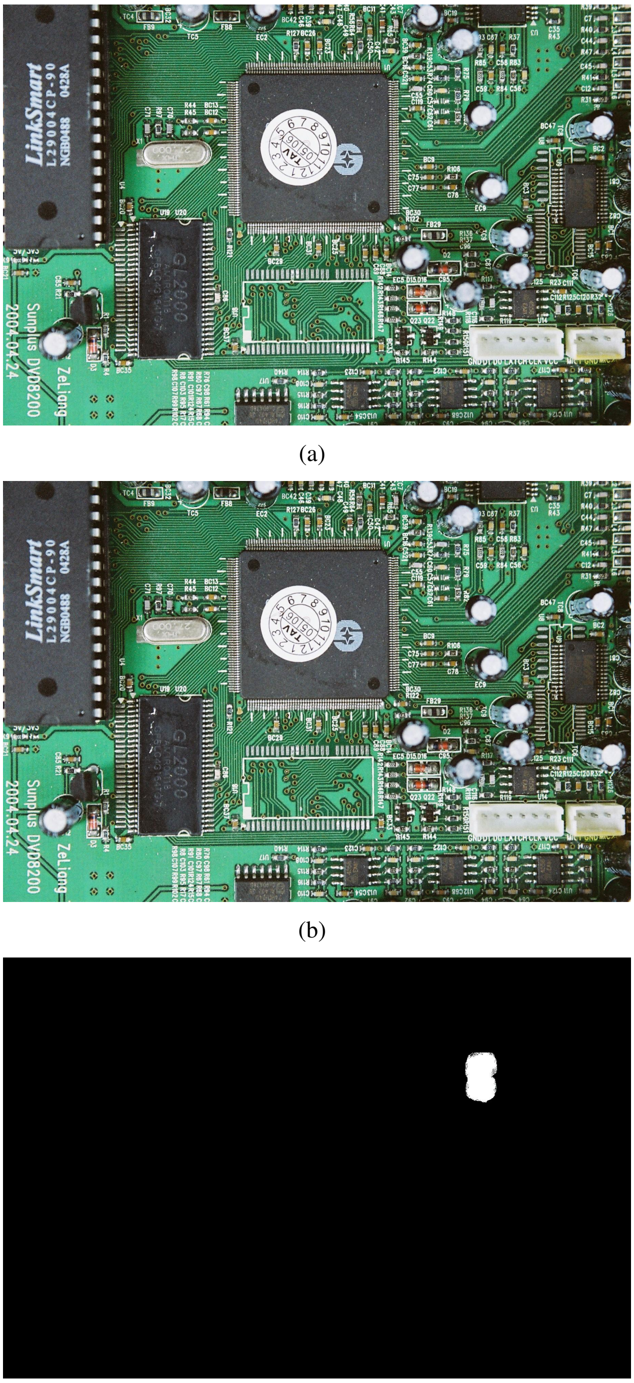


(a)

(b)

(c)

Figure 4: Detecting missing ICs using background subtraction.

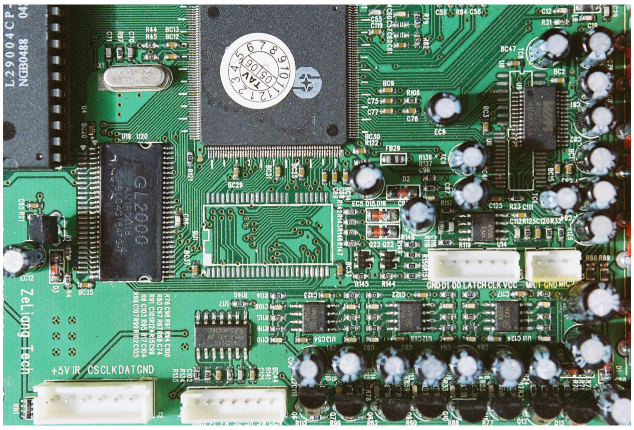


(a)

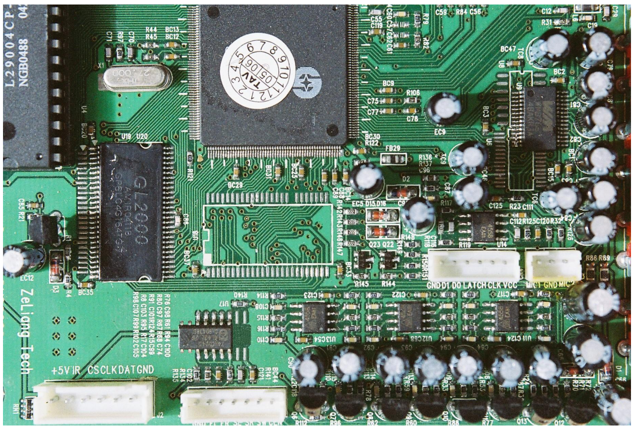
(b)

(c)

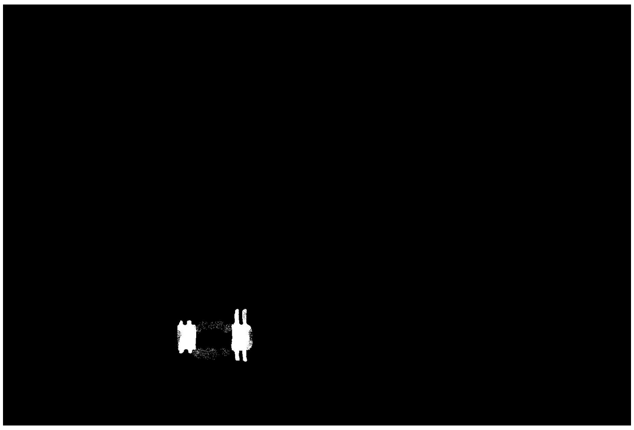
Figure 5: Detecting misaligned capacitors using background subtraction.



(a)



(b)



(c)

Figure 6: Detecting misaligned ICs using background subtraction.

6 Conclusion

We have presented the implementation details for a prototype AVI system for missing or misaligned components using color background subtraction. The accuracy of our proposition is however hardware dependent. A technologically advanced camera would probably give more reliable sensor readings as compared to a normal industrial camera and a high resolution camera would probably require lesser image sections to be processed for a given PCB size. On the other hand, the computational cost would be higher as compared to the latter for a pixel-per-pixel algorithm.

The background reference model which is obtained from the learning phase contributes to the efficiency of this defect detection technique. We have kept this model as simple as possible. Again, we must be cautious; in this technique, the algorithm can be fed with information that might lead to wrong learning or too little information leading to insufficient learning. A possible improvement is to update the learned model with presumably correct new information. It is not really clear if this suggestion will always work well. Sometimes, the tendency of the snowball effect arises; learning wrong information may lead to learning more wrong information and this should be avoided at all cost. In another aspect, the input vector to the learning phase plays a key role. The possibility of other vector components such as disparity and texture have yet to be experimented exhaustively within the context of PCB assembly inspection.

During our experiments, the proposed system was able to operate at about 20Hz for an image size of about 640×480 pixels. This image size is generally much bigger than the input images reported by other authors. Although our algorithm is a pixel-by-pixel based algorithm, we still managed to speed up the entire process. Much of this is due to the optimized structures in the coding of the defect detection technique. However, we should note that more computational operations would be carried out at foreground (FG) pixels as compared to background (BG) pixels. Hence, the system should theoretically run much faster is a defect free environment.

This system also has its limitations. At the moment, we do not get satisfactory results if the component has a color which is of the same color as the PCB background (board color). But this problem can be rectified in the PCB industry and electronic component industry. It is well known that AVI service providers dictate some form of color coding to electronic component manufacturers to avoid such problems. Hence in real life situations, matching color problems can be avoided altogether.

References:

[1] A. Sprague, M. Donahue, and S. Rokhlin, "A method for automatic inspection of printed circuit boards," *Graphical Model and Image Processing*, vol. 54, no. 3, pp. 401–415, 1991.

[2] Y. Hara, N. Akiyama, and K. Karasaki, "Automatic inspection system for printed circuit boards," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 5, no. 6, pp. 623–630, 1983.

[3] F. J. Langley, "Imaging systems for PCB inspection," *Circuit Manufacturing*, vol. 25, no. 1, pp. 50–54, 1985.

[4] S. Mukai, "PCB continuous line system proceeds from manufacturing to inspection," *Electronic Engineering*, vol. 39, no. 305, pp. 34–39, 1992.

[5] M. Moganti, F. Ercal, C. H. Dagli, and S. Tsunekawa, "Automatic PCB inspection algorithms: A survey," *Computer Vision and Image Understanding*, vol. 63, no. 2, pp. 287–313, 1996.

[6] P. Chou, A. Rao, M. Sturzenbecker, F. Wu, and V. Brecher, "Automatic defect classification for semiconductor manufacturing," *Machine Vision and Applications*, vol. 9, no. 4, pp. 201–214, 1997.

[7] B. Dom and V. Brecher, "Recent advances in the automatic inspection of integrated-circuits for pattern defects," *Machine Vision and Applications*, vol. 8, no. 1, pp. 5–19, 1995.

[8] E. K. Teoh, D. P. Mital, B. W. Lee, and K. L. Wee, "An intelligent robotic vision system for inspection of surface mount PCBs," in *IEEE International Conference on Systems and Cybernetics*, 1991, pp. 13–17.

[9] S. Kishimoto, N. Kakimori, Y. Yamamoto, and Y. Takahashi, "A printed circuit board (PCB) inspection system employing the multi-lighting optical system," in *Symposium on Electronic Manufacturing Technology*, 1990, pp. 120–129.

[10] W. Yokho, "Automated optical inspection of surface mount components using 2D machine vision," *IEEE 15th Annual Conference of IECON*, vol. 3, pp. 584–589, 1989.

[11] C. Y. C. Hong Dar Lin and W. T. Lin, "Principal component analysis based on wavelet characteristics applied to automated surface defect inspection," *WSEAS Transaction on Computer Research*, vol. 3, no. 4, pp. 193–202, 2008.

[12] M. K. J. Velten and A. Kummert, "Hardware optimized image processing algorithms for industrial high speed applications," in *WSEAS International Conference on Electronics, Control and Signal Processing*, 2003.

[13] N. Friedman and S. Russel, "Image segmentation in video sequences: A probabilistic approach," in *Proceedings of IEEE International Conference on Uncertainty in Artificial Intelligence*, San Francisco, USA, 1997, pp. 175–181.

[14] D. Hong and W. Woo, "A background subtraction for a vision-based user interface," in *Proceedings of IEEE International Conference on Information, Communications and Signal Processing*, vol. 1, 2003, pp. 263– 267.

[15] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, "Detecting moving objects, ghosts and shadows in video streams," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1337–1342, 2003.

[16] I. Haritaoglu, R. Cutler, D. Harwood, and L. Davis, "Backpack - Detection of people carrying objects using silhouettes," in *Proceedings of Computer Vision and Image Understanding*, vol. 81, no. 3, New York, USA, 2001, pp. 385–397.

[17] K. M. Cheung, T. Kanade, J. Bouguet, and M. Holler, "A real-time system for robust 3D voxel reconstruction of human motions," in *Proceedings of Computer Vision and Pattern Recognition*, vol. 2, 2000, pp. 714 – 720.

[18] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "WallFlower - Principles and practice of background maintenance," in *Proceedings of International Conference on Computer Vision*, California, USA, 1999, pp. 255–261.

[19] C. Ianasi, V. Gui, F. Alexa, and C. I. Toma, "Noncausal adaptive mode tracking estimation for background subtraction in video surveillance," *WSEAS Transactions on Signal Processing*, vol. 2, no. 1, pp. 52–59, 2006.

[20] N. J. B. McFarlane and C. P. Schofield, "Segmentation and tracking of piglets in images," *Machine Vision and Applications*, vol. 8, no. 3, pp. 187–193, 1995.

[21] S. C. S. Cheung and C. Kamath, "Robust background subtraction with foreground validation for urban traffic video," *EURASIP Journal of Applied Signal Processing*, vol. 2005, no. 1, pp. 2330–2340, 2005.

[22] Y. Chen and T. S. Huang, "Hierarchical MRF model for model-based multi-object tracking," in *Proceedings of IEEE International Conference on Image Processing*, vol. 1, Thessaloniki, Greece, 2001, pp. 385–388.

[23] C. R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780–785, 1997.

[24] J. Heikkila and O. Silven, "A real-time system for monitoring of cyclists and pedestrians," in *Proceedings of IEEE Workshop on Visual Surveillance*, vol. 22, no. 7, Washington, USA, 2004, pp. 563–570.

[25] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Background modeling and subtraction by codebook construction," in *Proceedings of International Conference on Image Processing*, vol. 5, 2004, pp. 3061– 3064.

[26] D. Koller, J. Weber, and T. Huang, "Towards robust automatic traffic scene analysis in real-time," in *IEEE International Conference on Decision and Control*, 1994, pp. 3776–3782.

[27] P. Roth, H. Bischof, D. Skovcaj, and A. Leonardis, "Object detection with bootstrapped learning," in *Proceedings 10th Computer Vision Winter Workshop*, 2005, pp. 33–42.

[28] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 1999, pp. 246–252.

[29] N. Li, D. Xu, and B. Li, "A novel background updating algorithm based on the logical relationship," in *WSEAS Transactions on International Conference on Signal, Speech and Image Processing*, 2007, pp. 154–158.

[30] O. Javed, K. Shafique, and M. Shah, "A hierarchical approach to robust background subtraction using color and gradient information," in *Proceedings of IEEE Workshop on Motion and Video Computing*, Washington, USA, 2002, pp. 22–27.

[31] T. Horprasert, D. Harwood, and L. Davis, "A statistical approach for real-time robust background subtraction and shadow detection," in *International Conference on Computer Vision*, Kerkyra, Greece, 1999, pp. 1–19.

[32] T. M. Su and J. S. Hu, "Robust background subtraction with shadow and highlight removal for indoor surveillance," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, pp. 14–25, 2007.

[33] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *Proceedings of the European Conference on Computer Vision*, 2000.

[34] A. Elgammal, R. Duraiswami, D. Harwood, and L. Davis, "Background and foreground modeling using non-parametric kernel density estimation for video surveillance," *Proceedings of the IEEE*, vol. 90, no. 7, pp. 1151–1163, 2002.