# A Smart Machine Vision System for PCB Inspection

Tie Qi Chen[1], JianXin Zhang[1], YouNing Zhou[2] and Yi Lu Murphey[1]

[1]Department of Electrical and Computer Engineering
University of Michigan - Dearborn, Dearborn, MI 48128
Phone: 313-5935028,
Email: yilu@umich.edu

[2]Jabil Circuit, Inc.
1700 Atlantic Boulevard
Auburn Hills, Michigan 48326

**Please address all correspondence to**

Yi Lu Murphey
Department of Electrical and Computer Engineering
University of Michigan-Dearborn
Dearborn, Michigan 48128-1491, U.S.A.
Voice: 313-593-5028, Fax: 313-593-9967
yilu@umich.edu

**Keywords: computer vision, automated problem solving, intelligent manufacturing**

**Abstract --** In this paper, we present a smart machine vision (SMV) system for printed circuit board (PCB) inspection. It has advantages over the traditional manual inspection by its higher efficiency and accuracy. This SMV system consists of two modules, LIF (Learning Inspection Features) and OLI (On-Line Inspection). The LIF module automatically learns inspection features from the CAD files of a PCB board. The OLI module runs on-line to inspect PCB boards using a high-resolution 2-D sensor and the knowledge provided by the LIF components. Key algorithms developed for SMV are presented in the paper. The SMV system can be deployed on a manufacturing line with a much more affordable price comparing to other commercial inspection systems.

**Keywords:** machine vision, automatic inspection, PCB inspection.

## 1 Introduction

The PCB(Printed Circuit Board) industry continues to adopt increasingly higher levels of integration and achieving higher and higher levels of component density. As a consequence, the tolerances on PCB assembly become tighter and tighter. This causes an increased need for reliable and accurate visual inspection of PCB boards[1,2,3]. The manufacturing of PCB circuits uses the SMT (Surface Mount Technology). The SMT circuit assembly consists of three major processes,
screen printing solder paste on the PCB, component placement and then solder re-flow in a convection oven. Correspondingly, there are three main tasks of vision inspection in PCB assembly:

- Solder paste inspection,
- Component placement, and
- Post-reflow inspection.

Placing a proper amount of solder paste on a pad is the key to prevent unwanted opens or shorts. Sometimes, it is possible to catch these unwanted opens or shorts using an in-circuit-test after all components are placed on the board, but most solder paste defects are impossible to catch after components are mounted.

The focus of this research is to develop a technology for inspection of solder paste on PCB's. Due to the development of the semiconductor technology, the electronic components are getting smaller and smaller, and more and more components can fit on one PCB board. A pad on a PCB can be as small as 0.01 inch (see Figure 1). Machine vision inspection of solder paste on PCB's is a non-trivial task. In this paper we describe a smart machine vision (SMV) system for inspecting defects of solder paste on PCB's. SMV was developed using machine learning technology combined with advanced machine vision techniques. SMV has been deployed on a manufacturing line and been tested on more than 2000 PCB boards, the accuracy of detection has exceeded 97%.

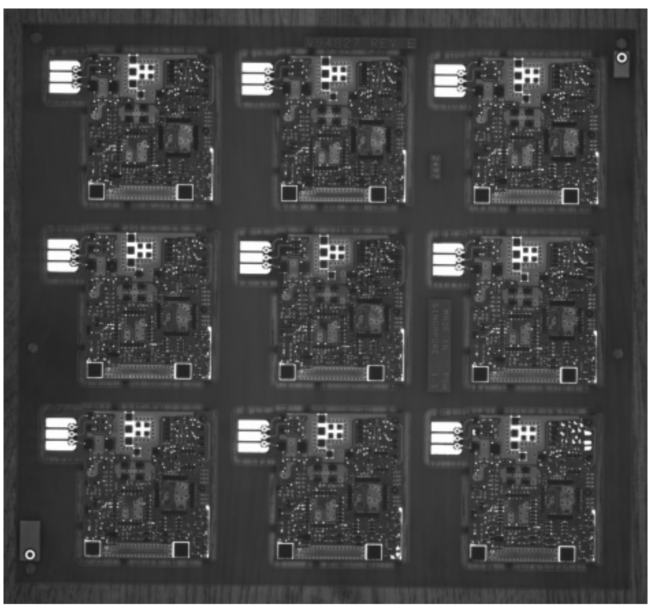

Figure 1 A PCB board array with thousands of pads.

## 2 Overview of SMV system

The objective of the SMV system is to detection whether there is a sufficient amount of solder paste on a pad, or if there is smear on a solder pasted pad. In theory, the bare pads and the solder pastes on a PCB should have different reflection rate under direct illuminating (see Figure 2). The major challenging is the high density of PCB boards and low contrast bare pads and paste. Even with the highest resolution CCD cameras on the market, images of high density PCB boards typically have give less than five pixels for a small pad and the paste on a small pad can be as small as one or two pixels. The SMV system was developed and tested using images of a variety of PCB boards acquired by a high resolution 2Kx2K Kodak camera. Figure 3 gives an overall view of SMV system.

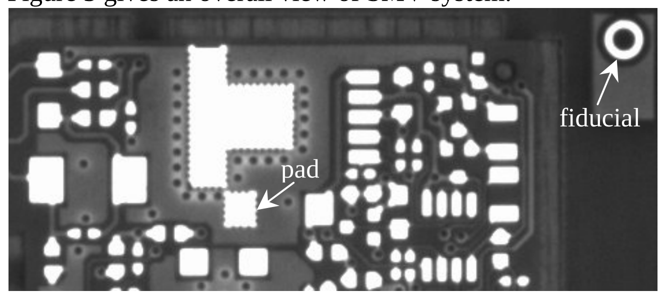


Figure 2a The pads without solder paste

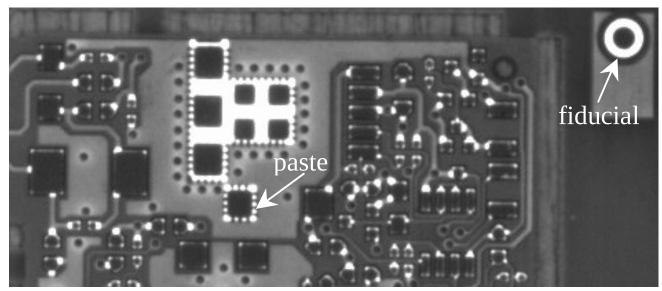


Figure 2b The pads with solder paste

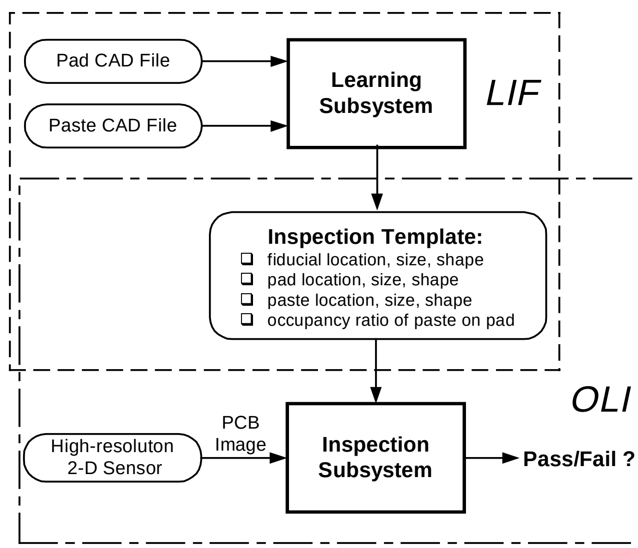


Figure 3 The SMV system

The SMV is composed of two modules: LIF (Learning Inspection Features) and OLI (On-Line Inspection). The LIF module was developed to learn visual inspection features of a PCB from CAD design data, and the OLI module applies the knowledge learnt by LIF to on-line PCB inspection. The LIF module automatically learns the inspection features and outputs the knowledge in a template file to be used in the on-line inspection program, OLI, during the manufacturing process. The learning program consists of algorithms for extracting inspection features from the input, image processing, and computing statistics. The OLI program consists of algorithms for image registration, image processing, comparing the features of the input image with the learnt features from the LIF. The output from the OLI will indicate whether the product has failed or pass the test. The exact format of the output from the OLI can vary depending on the task specification.

## 3 Learning inspection feature (LIF) module

The primary function of the LIF is to learn the solder features of a PCB from its CAD design files. PCBs are designed using CAD tools and the design information is contained in a CAD file. A PCB CAD file contains a set of instructions that, when interpreted, enable a photoplotter or laser imager to produce an image of the PCB on paper or other media. One example of such CAD file format is GERBER, a non-proprietary superset of Electronic Industries Association Standard RS-274D. A CAD design language typically has more than 400 different operators (or commands). We implemented 25 operators that are necessary for learning inspections features. The challenge of implementing these operators lies in the dynamics of these operators. For example, a number of operators may dynamically change the appearance of a line, the shape of the joint of two lines; and the target coordinates. In order to effectively inspecting PCB images, the LIF module needs to learn the following features from the CAD file:

- number of pads on each array,
- location of each pad,
- shape of each pad,
- fill or no-fill status of each pad, and
- location of every fiducial points on each PCB image.

A learning algorithm has been developed for the LIF module and it has three major steps, detecting components, finding bounding boxes, and computing occupancy ratio. The occupancy ratio is a measurement to be used in the on-line inspection procedure and is critical to the result of inspection.

A component on a PCB is a region that can be a pad, a paste, or a fiducial. Fiducials(see Figure 2) used to map CAD data to the images captured on line can be in any shape. A component in a CAD file typically has one or more closed paths, each of them consists of a number of strokes. A stroke can be as simple as a straight line, or as complicated as a part of Bézier curve. The information we are most interested in is the bounding box, which tells us the exact location of a component, and the occupancy ratio, which will be used in paste inspection in the OLI module.

The algorithm repeatedly searches for the bounding boxes of the strokes that are connected. The stroke bounding boxes are then merged to form the bounding boxes of the closed paths. Finally the bounding boxes of overlapped closed paths are merged to form the bounding box of a component.

The occupancy ratio is defined as the ratio of the solder paste area verses the bounding box of a pad. In order to inspect the solder paste during the manufacturing, accurate occupancy ratio is the key. The computation of occupancy ratio is a nontrivial task. In this algorithm we represent complicated curves by straight-line segments, in which the area of a stroke is quite close to the area of its bounding box. The occupancy ratio is calculated by computing the sum of the areas of the bounding boxes of the strokes with the overlapping regions subtracted. The following outlines the computational steps used to compute occupancy ratio.

Let $R$ be an array of $N$ rectangles, $A_n$ be the total area of $R[1]$, $R[2]$, …, $R[n]$, and $B_n$ be the area of $R[n+1]$ subtracting the part where it overlaps with the first $n$ rectangles. Thus we have,

$$A_N = A_{N-1} + B_{N-1} = \ldots = A_1 + \sum_{n=1}^{N-1} B_n \qquad (1)$$

where $A_1$ is the area of $R[1]$, which is easy to calculate. In order to calculate $B_n$, we first compare $R[n+1]$ with $R[1]$. If they overlap, we split $R[n+1]$ into up to 4 rectangles denoted as $r_1$, $r_2$, $r_3$ and $r_4$ (see Figure 4). Then we compare $r_1$, $r_2$, $r_3$ and $r_4$ with $R[2]$ one by one. They split if necessary. This procedure continues until $R[n]$ is split and its four rectangles are compared. $B_n$ equals to the sum of the areas of the rectangles from final splitting. It is guaranteed that the rectangles from splitting after comparison with $R[i]$ do not overlap with any rectangle of $R[1]$, …, $R[i]$, and they do not overlap with each other either of course.

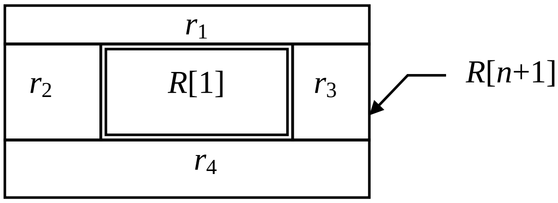


Figure 4 $R[n+1]$ splits into up to 4 rectangles if $R[n+1]$ and $R[1]$ overlap

In order to facilitate engineers to view the information of the PCB and edit the information for robust inspection, we developed a graphics user interface accompanied by the following functions:

- Bird's Eye View of the PCB.
- Zoom-in function to allow user to view different part of the board in detail.
- The Component-Clipping window. Some time, CAD files contain irrelevant information, e.g. captions and frames. We developed a Component-Clipping window function that allows user to define a region of interest to exclude irrelevant information. Only the component inside the region of interest will be learned by the LIF algorithm.
- The Component-Editing window. The Component-Editing window allows user to edit the information such as the component name, the occupancy ratio, and the threshold of test. User can also specify the fiducials by checking the radio button. An individual name will help an engineer to find which component fails the test.

As the result of learning, LIF module stores the following attributes for every component on a PCB, component name, location, occupancy ration, and test threshold. The test threshold is determined by the size of the component. In addition, the location and size of each fiducial is detected for use in the inspection module. The file that contains the knowledge of inspection features is referred to as a template file.

## 4 On-line inspection (OLI)

The OLI module inspects PCBs on a manufacturing line under the guidance of the inspection knowledge provided by the LIF module. The major processes within the OLI include:

1) Finding the fiducials in the image of a PCB
2) binarizing the image;
3) Detecting the tilt angle and then correct the image;
4) Mapping the components based on the information provided by the LIF to the image;
5) Detecting the paste occupancy on each component in the image.

The accuracy of fiducial finding algorithm is critical to the inspection result, because the tilt angle and the mapping scale and the mapping offset all come from the fiducials. The fiducials are usually in symmetric shapes such as round shape.

According to the information from the template we can get the approximate locations of the fiducials on the image. Around these approximate locations we set searching areas for the fiducials. In each searching area, the least-square fitting method is applied to calculate the center of the round-shape fiducial.

The objective function of the least square fitting is defined as:

$$E(x_c, y_c, r) = \sum_{i=1}^{n}\left[\rho(x_i, y_i)\left(\sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} - r\right)^2\right] = \min \qquad (2)$$

where the weight $\rho(x_i, y_i)$ is the intensity value of ($x_i$, $y_i$) on the original image.

In order to avoid using non-linear optimization method to solve Eq.(2), another objective function is defined:

$$E'(x_c, y_c, r) = \sum_{i=1}^{n}\left\{\rho(x_i, y_i)\left[(x_i - x_c)^2 + (y_i - y_c)^2 - r^2\right]^2\right\} = \min \qquad (3)$$

Eq.(3) can be rewritten as:

$$E'(x_c, y_c, r) = \sum_{i=1}^{n}\left\{\left(\sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} + r\right)^2 \cdot \left[\rho(x_i, y_i)\left(\sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} - r\right)^2\right]\right\} = \min \qquad (4)$$

From Eq.(4) we can see, Eq.(3) equals to Eq.(2) multiplied by a scale $k$.

$$k = \left(\sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} + r\right)^2 \qquad (5)$$

If $k$ is a constant, the solution of Eq.(2) and that of Eq.(3) will be same. If $k$ is not a constant, but we can get better approximate values for $x_c$, $y_c$ and $r$, then Eq.(3) can be changed into:

$$E''(x_c, y_c, r) = \sum_{i=1}^{n} \frac{\rho(x_i, y_i)\left[(x_i - x_c)^2 + (y_i - y_c)^2 - r^2\right]^2}{\left(\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} + r_0\right)^2} = \min \qquad (6)$$

An iteration algorithm has been developed for calculating $x_c$, $y_c$ and $r$. First, we used Eq.(3) to obtain the original values for $x_0$, $y_0$ and $r_0$. Then we used Eq.(6) to calculate $x_c$, $y_c$ and $r$, and replaced $x_0$, $y_0$ and $r_0$ by these new $x_c$, $y_c$ and $r$ in the next calculation of Eq.(6) until the difference between $x_0$, $y_0$, $r_0$ and $x_c$, $y_c$, $r$ was very small.

This fiducial finding algorithm is very robust. The fiducials can be found even if many pads are within the searching areas.

After the fiducials are accurately located on the image, the global bounding box of the inspection area on the image is calculated, and the following binarization operation is applied on the image content within the global bounding box.

Based on the analysis of PCB images, an effective binarization algorithm is developed, which consists of the following steps:

[Step-1] Generate the histogram V(X) of the PCB image, where V(X) is the number of pixels at pixel value X).

[Step-2] Smooth the histogram using an averaging filter. In most cases, the peaks around $P_L$ and the peaks around $P_H$ will not be removed.

[Step-3] Find the highest peak $P_L$. Suppose $P_L$ locates at $X_L$.

[Step-4] Define $V_{MIN}(X)$, for $X \in (X_L, 255)$, $V_{MIN}(X)$ equals to the minimum V(X'), $X' \in (X_L, X]$.

[Step-5] Search from $X_L$ to 255, find the location corresponding to the maximum of $[V(X)/V_{MIN}(X)]$. We denote it as $X_H$. We consider $X_H$ the location of $P_H$ because this peak has the largest peak-to-valley ratio than any other $X \in (X_L, 255)$. By finding the peak according to $[V(X)/V_{MIN}(X)]$, we can skip the peaks around $P_L$ which are usually higher than $P_H$.

[Step-6] Find the location X where $V(X) = V_{MIN}(X_H)$. We denote it as $X_t$. $X_t$ is the binarization threshold we will use. The reason is that it is the location of the deepest valley between $P_L$ and $P_H$ and should be the most suitable border between $\{P_{L'}\}$ and $\{P_{H'}\}$.

On a manufacturing line, a PCB is not always mounted perfectly on the test table. It may tilt in a small angle. But this small tilt angle may cause big errors while the template mapping on the image. So the PCB image has to be untilted first.

The least mean square method is used in estimating the tilt angle. Suppose there are $n$ fiducials, their locations in the learning template are $(X_i, Y_i)$, and their locations in the image are $(x_i, y_i)$, where $i$=1, 2, ... $n$. Suppose the tilt angle of the image relative to the template is $\alpha$, the scaling factor is $k$, and $(x_0, y_0)$ are offset, then we should have,

$$\begin{cases} x_i \approx k\left(X_i \cos\alpha - Y_i \sin\alpha\right) + x_0 \\ y_i \approx k\left(Y_i \cos\alpha + X_i \sin\alpha\right) + y_0 \end{cases} \quad (i=1, 2, ..., n) \qquad (7)$$

Clearly, $\alpha$, $k$ and $(x_0, y_0)$ can be determined by minimizing the following objective function $E(\alpha, k, x_0, y_0)$:

$$E(\alpha, k, x_0, y_0) = \frac{1}{n}\sum_{i=1}^{n}\Big\{\left[k\left(X_i \cos\alpha - Y_i \sin\alpha\right) + x_0 - x_i\right]^2 + \left[k\left(Y_i \cos\alpha + X_i \sin\alpha\right) + y_0 - y_i\right]^2\Big\} = \min \qquad (8)$$

By letting $\frac{\partial E}{\partial \alpha} = \frac{\partial E}{\partial k} = \frac{\partial E}{\partial x_0} = \frac{\partial E}{\partial y_0} = 0$, we will get the following solution:

$$\begin{cases} \tan\alpha = \dfrac{\overline{xY} - \overline{yX} - \left(\bar{x}\bar{Y} - \bar{y}\bar{X}\right)}{\overline{xX} + \overline{yY} - \left(\bar{x}\bar{X} + \bar{y}\bar{Y}\right)} \\ k = \dfrac{\left[\overline{xX} + \overline{yY} - \left(\bar{x}\bar{X} + \bar{y}\bar{Y}\right)\right]\cos\alpha - \left[\overline{xY} - \overline{yX} - \left(\bar{x}\bar{Y} - \bar{y}\bar{X}\right)\right]\sin\alpha}{\overline{X^2 + Y^2} - \left(\bar{X}^2 + \bar{Y}^2\right)} \\ x_0 = \bar{x} - k\left(\bar{X}\cos\alpha - \bar{Y}\sin\alpha\right) \\ y_0 = \bar{y} - k\left(\bar{Y}\cos\alpha + \bar{X}\sin\alpha\right) \end{cases}$$

(9)

where $\bar{f} = \frac{1}{n}\sum_{i=1}^{n} f_i$ is the mean.

After the tilt angle is estimated, the image is untilted then the new locations of the fiducials in the modified image are re-calculated. These new locations of the fiducials will be used in mapping the template onto the image. Because untilting the image is the most time consuming part in the whole process, efforts have been made to speed it up:

1. Only the part of the image inside the global bounding box is untilted;
2. Functions “sin” and “cos” are called only once;
3. Floating point multiplication/division has been reduced to minimum;
4. The lower limit of the tilt angle is estimated before untilting, the image is untilted only when it is necessary (i.e. the tilt angle is above the lower limit).

The least mean square method is also used in mapping the learning template onto the image. Suppose we have $n$ fiducials, their locations in the learning template are ($X_i$, $Y_i$), and their locations in the image are ($x_i$, $y_i$), where $i$=1, 2, ... $n$. Suppose the scaling factor is $a$. Then we should have,

$$\begin{cases} x_i \approx aX_i + x_0 \\ y_i \approx aY_i + y_0 \end{cases} \quad (i=1, 2, ..., n) \quad (10)$$

where ($x_0$, $y_0$) are translation. Both $a$ and ($x_0$, $y_0$) can be determined by minimizing the following objective function $E(a, x_0, y_0)$:

$$E(a, x_0, y_0) = \frac{1}{n}\sum_{i=1}^{n}\left[(aX_i + x_0 - x_i)^2 + (aY_i + y_0 - y_i)^2\right] = \min \quad (11)$$

By letting $\frac{\partial E}{\partial \alpha} = \frac{\partial E}{\partial k} = \frac{\partial E}{\partial x_0} = \frac{\partial E}{\partial y_0} = 0$, we will get the following solution:

$$\begin{cases} a = \dfrac{\overline{xX + yY} - \left(\bar{x}\bar{X} + \bar{y}\bar{Y}\right)}{\overline{X^2 + Y^2} - \left(\bar{X}^2 + \bar{Y}^2\right)} \\ x_0 = \bar{x} - a\bar{X} \\ y_0 = \bar{y} - a\bar{Y} \end{cases} \quad (12)$$

where $\bar{f} = \frac{1}{n}\sum_{i=1}^{n} f_i$ is the mean.

Finally, the occupancy ratio of paste on each pad is calculated. The occupancy ratios calculated at the LIF stage are the occupancy ratios of the pastes themselves. And it is the bare pads (not the pastes) that reflect the light while the CCD camera captures the image. Therefore, if the occupancy ratio of a component is above the test threshold, then it implies a missing or damaged paste.

For a 2K×2K image with more than 4000 components, the processing time of automatic inspection is about 6 seconds on a Pentium Pro computer.

# 5 Conclusions

We have presented an automatic solder paste inspection system, SMV (Smart Machine Vision), which relieves human test operators from a stressful and unrealistic inspection task. The SMV system has two modules, namely the LIF (Learning Inspection Features) and the OLI (On-Line Inspection). During the off-line learning process, the LIF learns from the CAD files of the PCB and generates an inspection template for every new type of PCB layout. The OLI module runs on the production line accurately and efficiently inspects PCBs. The key algorithms for supporting the SMV system are introduced. This automatic solder paste inspection system finds defects at the early stage on the production lines, which can significantly reduce the manufacturing cost. The whole system has been tested over 8000 boards on a manufacturing line and the detection accuracy was above 97%.

## 6. Acknowledgements

This work is supported in part by a contract from Jabil Circuit, Inc. In particular, we would like to thank Mr. Anthony Tisler for introducing the problem and his contribution in the knowledge of printed circuit board manufacturing.

## 7. References

[1] F. J. Langley, Imaging systems for PCB inspection, Circuit Mauf. 25(1), pp50-54, 1985.

[2] M. Beck and D. Clark, SMT inspection strategies: Maximizing cost effectiveness, in Proceedings of Technical Program: NEPCON West'91, pp1075-1081, 1991.

[3] B. R. Taylor, Automatic inspection in electronics manufacturing, SPIE Autom. Opt. Inspection 654l, pp157-159, 1986.

[4] S. Mukai, PCB continuous line system proceeds from manufacturing to inspection, J. Electron. Eng. 29(305), pp34-39.

[5] J. W. Foster III, P. M. Griffin, S. L. Messimer and J. R. Villalobos, Automated visual inspection: A tutorial, Comput. Ind. Eng. 18(4), pp493-504, 1990.

[6] S. Yu, W. Cheng and S. C. Chiang, Printed circuit board inspection system PI/1, SPIE Autom. Insp. High Speed Vision Archit. II 1004, pp126-134, 1988.

[7] E. Bin-Nun, Automatic optical inspection focuses on defects, Electron. Packag. Prod. pp82-87, 1984.
[8] J. Ragland, Automating innerlayer inspection, Circuits Manuf. Pp91-94, 1995.
[9] H. W. Markstein, Automatic optical inspection improves multilayer yields, Electron. Packag. Prod. Pp60-64, 1983.
[10] W. Wu, M. Wang and C. Liu, Automated inspection of printed circuit boards through machine vision, Computers in Industry 28, pp103-111, 1996.
[11] Yi L et al "Machine Vision Algorithms Using Interactive Learning For VFD Inspection," submitted to Journal of Applied Intelligence, 2000