GRAYLEVEL ALIGNMENT BETWEEN TWO IMAGES USING LINEAR PROGRAMMING

Edmund Y. Lam

Department of Electrical and Electronic Engineering,
University of Hong Kong,
Hong Kong.
E-mail: elam@eee.hku.hk

ABSTRACT

A critical step in defect detection for semiconductor process is to align a test image against a reference. This includes both spatial alignment and grayscale alignment. For the latter, a direct least square approach is not very applicable because the presence of defects would skew the parameters. Instead, we use a linear programming formulation which has the advantage of having a fast algorithm, while at the same time can produce better alignment of the test image to the reference. Furthermore, this is a flexible algorithm capable of incorporating additional constraints, such as ensuring that the aligned pixel values are within the allowable intensity range.

1. INTRODUCTION

Defect identification is an indispensable step in the semiconductor manufacturing process [1]. From photomask manufacturing to final packaging, careful inspection is needed to avoid malfunctioning of the integrated circuits (IC) due to contamination. Because of the ever-shrinking feature size and ever-increasing complexity of the IC, advanced machine vision techniques are constantly needed to improve the inspection capability. Although the ultimate test of an IC is its electronic behavior, visual inspection is equally important because it pinpoints the areas with defects that may cause the problems.

An efficient setup for inspecting a circuit is as follows. A reference circuit is first carefully examined to ensure that it does not contain any defect. A test circuit is then compared against the reference to see if there exists any visual difference. This comparison can be done on a pixel-by-pixel basis. Usually, some post-processing step is required to classify the potential defects into some known defect types. Clearly, the success of this defect identification and classification depends on an accurate comparison between the test and the reference.

Usually, the inspection system is carefully designed to ensure that images of the test and the reference are taken under identical conditions. In reality, there are some inevitable variations, and the two images will have some shifts in location and intensity. In this paper, we focus our attention on the latter. This problem emerges because the lighting on the test and the reference may differ. In section 2, we discuss a straightforward way to cope with such difference, and the drawbacks of this method. Then, in section 3, we explain our algorithm that uses linear programming to tackle this problem. Simulation results are shown in section 4 to demonstrate the improvement in performance, and we discuss the implications and conclusions in section 5.

2. A LEAST SQUARE APPROACH

Let $f_r(x,y)$ denote the reference, and $f_t(x,y)$ denote the test image. (x,y) are the spatial coordinates relative to a certain marker on the circuits, so the two are aligned spatially. Ideally, if their grayscale values are also aligned, then $f_r(x,y) = f_t(x,y)$ except in areas where there are defects. Therefore,

$$\delta(x,y) = f_t(x,y) - f_r(x,y) \tag{1}$$

is a map for the defect locations.

Now assume that the graylevels of the two images have not been aligned, but it is known that the test does not contain any defect. Furthermore, we assume that the graylevel variation between the two are linearly related, *i.e.*,

$$f_r(x,y) = \sum_{i=1}^{M} a_i f_{t,i}(x,y), \tag{2}$$

where $f_{t,i}(x,y)$ are images that depend on $f_t(x,y)$, or a plain image to capture shift in mean intensity. Some possibilities of $f_{t,i}(x,y)$ include $xf_t(x,y)$, $yf_t(x,y)$, $f_t(x,y) - f_t(x-1,y)$, and $\sqrt{x^2+y^2}f_t(x,y)$. a_i's are the design parameters. We can use the least square method to find the optimal a_i's, denotes as $\mathbf{a}^*$. Let $\mathbf{f_{t,i}}$ and $\mathbf{f_r}$ be the lexicographical ordering of $f_{t,i}(x,y)$ and $f_r(x,y)$ respectively.

The optimal value $\mathbf{a}^*$ can be found by

$$\mathbf{a}^* = \arg\min_{\mathbf{a}} \left\| \begin{bmatrix} \mathbf{f_{t,1}} & \cdots & \mathbf{f_{t,M}} \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_M \end{bmatrix} - \mathbf{f_r} \right\|_2, \quad (3)$$

where $\|\cdot\|_2$ denotes the L_2 norm. It is known that the residue on the right hand side behaves like a white Gaussian noise.

In reality, we deal with a situation where there are both defects and graylevel variations. If we just compute equation 3, the values of $\mathbf{a}^*$ will be affected by the defects. We can minimize the contribution of the defects by calculating equation 3 iteratively. At each stage, we identify places where the residual is still large. We then mark those pixels as defect candidates, and omit those pixels in the subsequent stages of computing equation 3. It is hoped that, after a number of iterations, the defect pixels will be mostly identified and only the valid pixels are used to calculate the correct values of the grayscale alignment factors.

Speed and accuracy are the primary concerns in such an inspection system. The advantage of using equation 3 is that it is fast. In fact, analytic solution is readily available for least square calculation. However, accuracy is compromised by the presence of the defects, and we compensate for that by using iteration. Furthermore, a direct least square approach makes no guarantee that the resulting pixels in $\sum a_i f_{t,i}(x,y)$ will not create pixel values that are outside the allowable range, such as below zero intensity or above maximum intensity. We propose an improved formulation of the problem in the next section that addresses these issues.

3. A LINEAR PROGRAMMING APPROACH

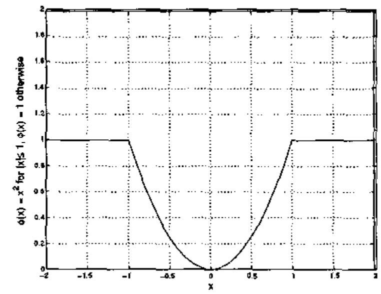
Equation 3 can be cast as the following optimization problem:

$$\text{minimize} \quad \phi(g_1) + \phi(g_2) + \ldots + \phi(g_k)$$

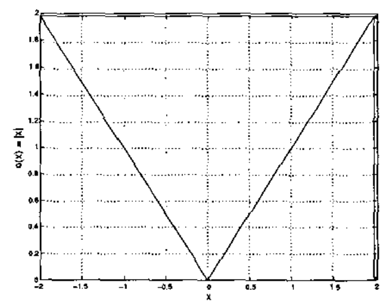
$$\text{subject to} \quad \mathbf{g} = \begin{bmatrix} \mathbf{f_{t,1}} & \cdots & \mathbf{f_{t,M}} \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_M \end{bmatrix} - \mathbf{f_r}, \quad (4)$$

where g_i denotes the ith component of the vector $\mathbf{g}$ (with a total of k elements), and $\phi(g_i) = g_i^2$. ϕ is called the penalty function [2]. Since defects usually result in great residuals in g_i, a quadratic penalty for these terms lead to a solution that seeks to distribute the errors across more terms but with smaller residuals. In other words, $\mathbf{a}^*$ will be significantly affected by the defects.

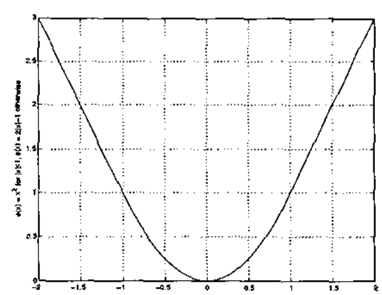
Therefore, a better approach is to use a penalty term that is smaller for large values of its argument. Figure 1 plots a few possibilities. 1(a) is obtained with a clipping of the maximum value of the quadratic penalty function, (b) is a linear penalty function, while (c) is a combination of a linear and quadratic penalty functions, called the Huber penalty function. If we use (b) or (c) in equation 4, the optimization problem is convex, which has fast algorithm such as interior point method [2]. Furthermore, if we use (b), the problem can be converted to linear programming. In this case, we can make use of fast algorithm readily available for linear programming.



(a) $\phi(x) = x^2$ for $|x| \leq 1$, $\phi(x) = 1$ otherwise.



(b) $\phi(x) = |x|$.



(c) $\phi(x) = x^2$ for $|x| \leq 1$, $\phi(x) = 2|x| - 1$ otherwise.

Fig. 1: Some example penalty functions.

To see how this can be done, we introduce a new variable $\mathbf{t} = |\mathbf{g}|$. Let

$$\mathbf{b} = \begin{bmatrix} \mathbf{a} \\ \mathbf{t} \end{bmatrix} \quad (5)$$

be our parameters. Equation 4 becomes

$$\begin{aligned} \text{minimize} \quad & [\ \mathbf{0} \quad \mathbf{1}\ ]\mathbf{b} \\ \text{subject to} \quad & [\ \mathbf{f_{t,1}} \ \cdots \ \mathbf{f_{t,M}} \quad \mathbf{0}\ ]\,\mathbf{b} - \mathbf{f_r} \leq \mathbf{t} \\ & [\ \mathbf{f_{t,1}} \ \cdots \ \mathbf{f_{t,M}} \quad \mathbf{0}\ ]\,\mathbf{b} - \mathbf{f_r} \geq -\mathbf{t}. \end{aligned} \tag{6}$$

We can further rearrange the terms to be

$$\begin{aligned} \text{minimize} \quad & [\ \mathbf{0} \quad \mathbf{1}\ ]\mathbf{b} \\ \text{subject to} \quad & \begin{bmatrix} \mathbf{f_{t,1}} & \cdots & \mathbf{f_{t,M}} & \begin{matrix} -1 & & 0 \\ & \ddots & \\ 0 & & -1 \end{matrix} \end{bmatrix} \mathbf{b} \leq \mathbf{f_r} \\ & \begin{bmatrix} -\mathbf{f_{t,1}} & \cdots & -\mathbf{f_{t,M}} & \begin{matrix} -1 & & 0 \\ & \ddots & \\ 0 & & -1 \end{matrix} \end{bmatrix} \mathbf{b} \leq -\mathbf{f_r}. \end{aligned} \tag{7}$$

Moreover, we can add the constraints

$$\begin{aligned} [\ \mathbf{f_{t,1}} \ \cdots \ \mathbf{f_{t,M}} \quad \mathbf{0}\ ]\,\mathbf{b} & \leq \mathbf{f_r} + \mathbf{i}_{\max} \\ [\ -\mathbf{f_{t,1}} \ \cdots \ -\mathbf{f_{t,M}} \quad \mathbf{0}\ ]\,\mathbf{b} & \leq -\mathbf{f_r} - \mathbf{i}_{\min} \end{aligned} \tag{8}$$

to ensure that the adjusted pixel values fall between $[\mathbf{i}_{\min}, \mathbf{i}_{\max}]$, while still maintaining the linear programming formulation.

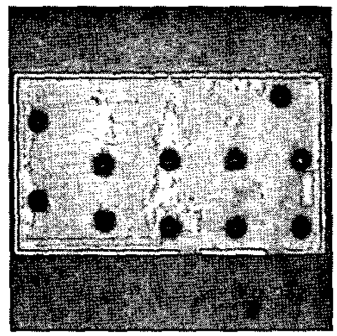
4. SIMULATIONS

We apply the linear programming approach to graylevel alignment on bump inspection, which is a critical process in die bonding [1]. Bumps are the electrical and mechanical connection between the die and the substrate, and are formed from processes such as paste-deposition and electroplating. As such, the shape of the bumps may vary, and may have a few potential defects such as missing bumps, bridged bumps, contaminants on or between bumps, and incorrect bump volumes or heights. Image processing techniques are frequently employed to identify these defects [3]. Figure 2 shows the test images we are using. 2(a) is a reference image without any defect. (b) has a patch of ink upon it, and visually it bridges two bumps together. On the other hand, (c) has some missing bumps. In addition to having defects, both (b) and (c) have intensity variation with respect to the reference image (a). Our goal is to adjust the grayscales of (b) and (c) to match that of (a).

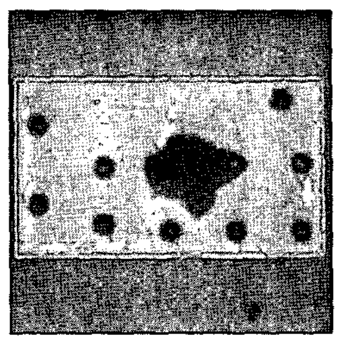
Table 1 shows the improvement in signal-to-noise ratio (SNR) gain with the least square approach and the linear programming approach in grayscale alignment. SNR is defined to be [4]

$$\mathrm{SNR}(f, \hat{f}) = 10 \log_{10} \left(\frac{\sum_x \sum_y (f(x,y))^2}{\sum_x \sum_y \left(f(x,y) - \hat{f}(x,y) \right)^2} \right), \tag{9}$$

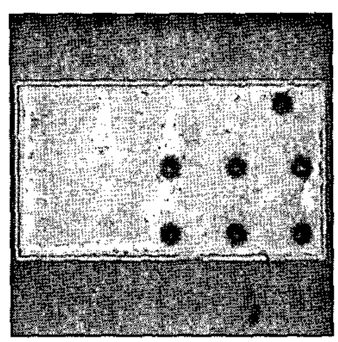
where $f(x,y)$ is the original image and $\hat{f}(x,y)$ is the adjusted image. In our case, $f(x,y)$ is always taken to be the reference image. The gain is with respect to the SNR for the unprocessed image. Also, we discard 5% of the pixels with the greatest errors after the grayscale adjustment, since those pixels are mostly likely defects. The percentage can change with *a priori* knowledge about the proportion of defects in the image. As seen in the table, the linear programming approach outperforms the least square approach in both cases. The improvement for the bridged bump case is greater, because the contamination covers a greater area and therefore the effect on the alignment for least square is more pronounced.



(a) Reference image.



(b) Test image with bridged bump.



(c) Test image with missing bump.

Fig. 2: Sample images from bump inspection.

Signal-to-noise ratio (SNR) gain		
	Least square	Linear programming
bridged bump	1.49dB	3.43dB
missing bump	2.31dB	2.38dB

Table 1. Simulation results in bump inspection.

5. CONCLUSIONS

In this paper, we have described a linear programming formulation of performing grayscale alignment between two images. This is seen to produce images of better quality even in the presence of defects. This is also a fast approach because there exists efficient algorithm for tackling linear programming, making it suitable in view of the stringent speed requirement in defect inspection applications.

However, we have not yet taken into consideration the locations of the defects. When a pixel is identified as a potential defect, the likelihood that its surrounding pixels are also defects is rather high. We are investigating an improvement to our approach where this information can help speed up the rejection of defect areas for grayscale alignment.

6. REFERENCES

[1] Peter Van Zant, *Microchip Fabrication: A Practical Guide to Semiconductor Processing*, McGraw-Hill, New York, fourth edition, 2000.

[2] Stephen Boyd and Lieven Vandenberghe, "Convex optimization," to be published in 2003.

[3] Steve Hibert, "Micro inspection for wafer bumping," *Advanced Packaging*, vol. 11, no. 2, pp. 19–22, February 2002.

[4] Anil K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, Englewood Cliffs, New Jersey, 1989.