CORE
Metadata, citation and similar papers at core.ac.uk
Provided by Brunel University Research Archive

A golden template self-generating method for patterned wafer inspection

Pin Xie and Sheng-Uei Guan
Department of Electrical Engineering, National University of Singapore
10 Kent Ridge Crescent, Singapore 119260
E-mail: eleguans@nus.edu.sg

Abstract

This paper presents a novel golden template self-generating technique for detecting possible defects in periodic two-dimensional wafer images. A golden template of the patterned wafer image under inspection can be obtained from the wafer image itself and no other prior knowledge is needed. It is a bridge between the existing self-reference methods and image-to-image reference methods.

Spectral estimation is used in the first step to derive the periods of repeating patterns in both directions. Then a building block representing the structure of the patterns is extracted using interpolation to obtain sub-pixel resolution. After that, a new defect-free golden template is built based on the extracted building block. Finally, a pixel-to-pixel comparison is all we need to find out possible defects.

A comparison between the results of the proposed method and those of the previously published methods is presented.

Key words: Wafer inspection - Golden template - Spectral estimation – PDI - Image-to-image reference method

1. Introduction

The phases of wafer inspection should include image acquisition, defect detection, and defect classification. The phase of image acquisition is described in *Section 3*. This paper focuses on the detection of defects in patterned wafers. Defect classification is an area that more issues are involved (Chou et al. 1997), which is not the focus of this paper.

1.1 Backgrounds

Most automatic inspection systems use one or a combination of two approaches: design-rule checking or image-to-image reference (Dom and Brecher, 1995). A pure design-rule system checks for the violation of a set of generic rules everywhere on the IC part. A design-rule-based PDI prototype system has been developed by NanYang Technical University, Singapore (Meisburger et al. 1992; Mital and Khwang, 1991). Most PDI systems use the image-to-image-reference approach. A pure reference system compares every pixel in the digital image under inspection with the corresponding pixel in the reference image, which is assumed to be perfectly registered with the image being analyzed. With this approach, image registration is a major problem.

Optical spatial filtering (Chin, 1988) can also be used in defect detection on masks and patterns. This method is very fast but a major disadvantage is that small defects cannot be recognised. Wavelet technique (Chen et al. 1998) has also been involved in wafer inspection. A fairly complete review of the related literature may be found in Babian (1986), Newman et al. (1995), Moganti et al. (1996, 1998a, 1998b).

1.2 Related work

All the above methods need a database of images or some prior knowledge. But in some cases, we have only a single image. A self-reference technique that

avoids the mentioned difficulties was developed by Dom et al.(1988), in which the comparison is made using the repeating cells in the image. This method was further developed by Khalaj et al.(1993) by proposing a technique to extract the building block of repeating patterns from the acquired image, and then detecting the defects by comparing the resulting building block with the image. In the first step of this method, the ESPRIT algorithm (Paulraj et al. 1985; Roy et al. 1989; Khalaj et al. 1994) is used in estimating the frequency components. Then a building block representing the constructive structure of the patterns is extracted using *Eq. 1* to obtain sub-pixel resolution. Please note that the size of the building block extracted from this approach is $(\mathrm{int}(T_x)+1)*(\mathrm{int}(T_y)+1)$, not the exact size of the repeating pattern.

Eq. 1:

$$B(k,l)=\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}[(1-r_i)(1-s_j)F(k_i+k,l_j+l)+r_i(1-s_j)F(k_i+k+1,l_j+l)$$
$$+(1-r_i)s_jF(k_i+k,l_j+l+1)+r_is_jF(k_i+k+1,l_j+l+1)]$$

Assume $F(k,l)$ is the value of the $N*N$ image at location (k,l), $B(k,l)$ is the value of the T_x*T_y building block at location (k,l), T_x and T_y are horizontal and vertical periods of the image respectively.

$$1\le k\le \mathrm{int}(T_x)+1, \qquad 1\le l\le \mathrm{int}(T_y)+1$$

$$n_1=\mathrm{int}(N/T_x), \qquad n_2=\mathrm{int}(N/T_y)$$

$$k_i=\mathrm{int}(T_x*i), \qquad l_j=\mathrm{int}(T_y*j)$$

$$r_i=T_x*i-k_i, \qquad s_j=T_y*j-l_j$$

By averaging among all of the blocks in the image, a good estimate of the building block is obtained in this way.

In the final step, each point in the original image is compared with the corresponding point in the building block. If the difference is larger than a threshold, then the point may be a possible defect. Because of the quantization effects at the edges of the image, each point is also compared with the eight neighboring points of its corresponding point in the building block. Among the nine absolute values of difference, the smallest one is kept which gives a measure of the probability that the point is a defect. This procedure can be stated by *Eq. 2*.

Eq. 2:

$$\min_{-1 \le i, j \le 1} \left| F(m,n) - B(k+i, l+j) \right|$$

where $k = m - \text{int}(m/T_x) * T_x$, $l = n - \text{int}(n/T_y) * T_y$

In this paper, we propose a novel golden template self-generating method that yields better results than the method described above with more accuracy and less computation. The main differences between the two algorithms are described in detail in *Section 4*. Preliminary reports were presented in the previous paper (Guan et al. 1999).

1.3 The structure of this paper

The algorithm is described in *Section 2*. In *Section 3*, we apply the developed techniques to some sample images that have repeating patterns and show the results. Finally, in *Section 4* we compare and contrast our approach with the approach described in *Section 1.2*. Possible applications of our approach are also covered.

2. Golden template self-generating method

This algorithm includes four steps. An important point that must be mentioned here is that all the images start from pixel $(1,1)$ instead of $(0,0)$ in this paper.

2.1 Estimating the periods of repetition

Spectral analysis is used to estimate the periods of a periodic image in the horizontal and vertical directions. Since the computations in two directions are quite similar, only the procedure used in the horizontal direction is illustrated.

The spatial projection in the horizontal direction is obtained by $H(j) = \frac{1}{m}\sum_{i=1}^{m} F(i, j)$. The resolution of frequency domain is not high enough to estimate the period of the projection vector at the order of sub-pixel accuracy, because it is determined by the number of points in the spatial domain. In order to improve the resolution in the frequency domain, the vector obtained from spatial projection is extended to length of 5000 (more than 10 times of the dimension of the spatial projection) by zero padding (Porat, 1997). The spectral profile of this 5000-element vector remains the same as the one in the original projection, while the normalized frequency unit is refined from 0.003 or above (the image size is assumed to be smaller than 300×300) to 0.0002, which allows us to find the frequency of the repeating pattern more accurately.

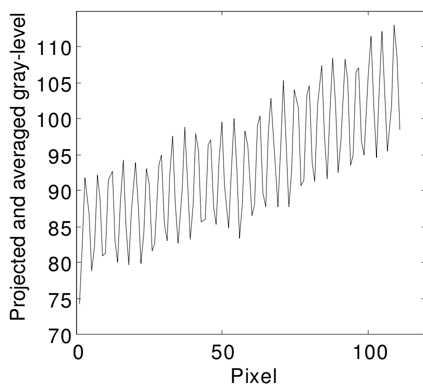

Fig. 1a Horizontal projection

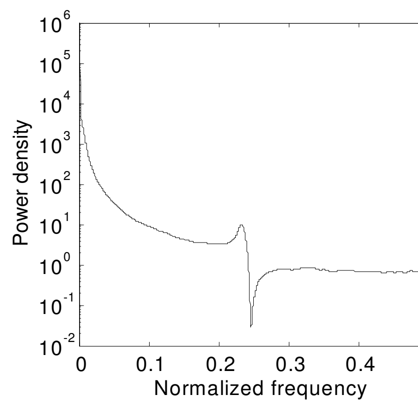

Fig. 1b Power spectrum of the projection

The above two figures are the horizontal projection of the image shown in *Fig. 6a* and its spectrum respectively. (The diagrammatic curve in *Fig. 1a* slants to the

upper-right because of the illumination non-uniformity in *Fig. 6a*. It has no effect on the calculation of the frequency.)

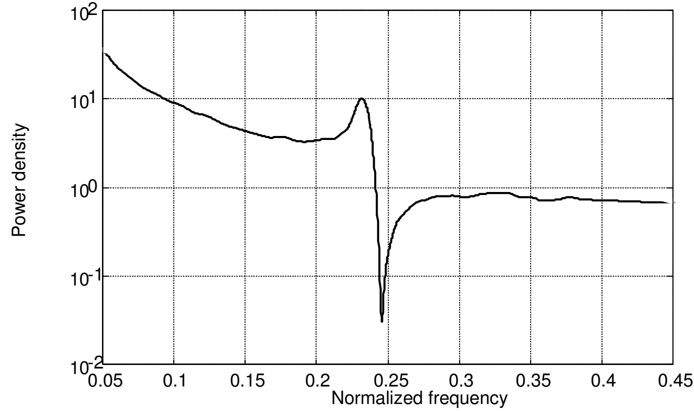

Fig. 1c Filtered spectrum

With respect to the normal size of the pattern periods (4 to 25 pixels), a reasonable band-pass filter (0.04-0.25) is used so that the frequencies of the repeating pattern could be found more accurately. Referring to *Fig. 1c,* the frequency of the repeating pattern in the horizontal direction corresponds to the main frequency component in the filtered spectrum. The most prominent peak in the concerned frequency range is located at 0.232 (normalized frequency), which means that the horizontal period of the repeating pattern is $T_x = 4.314$ (pixel). Following the same procedure, we can derive the vertical period T_y of the pattern under interest.

2.2 Extracting the building block

In the self-reference method described in *Section 1.2*, the building block is constructed by shifting a window of a proper size throughout the image and adding the corresponding pixel values together. Since the periods of the repeating patterns are not integer numbers in pixels under most cases, the size of the building block defined in *Section 1.2* is not the actual size of the repeating patterns. In this paper, we define a simulated building block whose size is exactly $T_x * T_y$. The values of $B(k,l)$, where

$1 \le k \le \text{int}(T_x)+1$ and $1 \le l \le \text{int}(T_y)+1$, are obtained by linear interpolation. *Eq. 1* is used to calculate the values of the integer points in the building block.

The noise and defects in the building block have been reduced significantly because the above equation is actually an averaging process. As mentioned, we imagine that the building block is exactly $T_x * T_y$ in size. The values of those integer points in the building block have been calculated. The role the fractional part plays in detecting defects will be explained in details in *Section 2.3* (and the Appendix).

2.3 Building a defect-free image

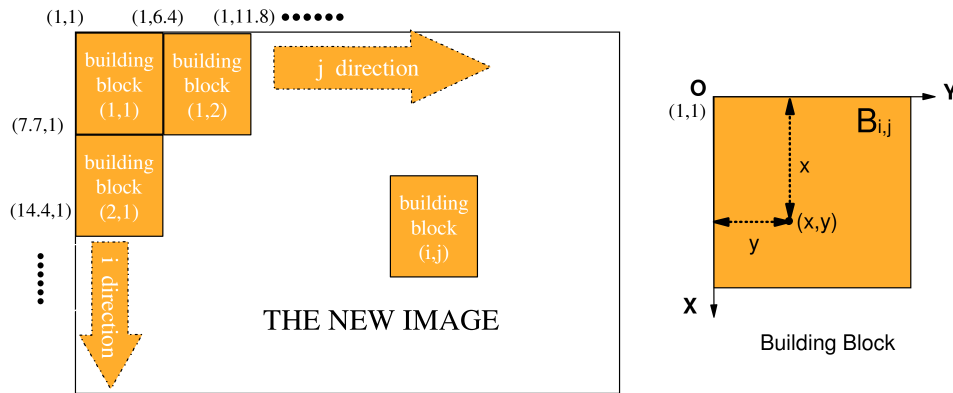

Fig. 2 The structure of a defect-free image F' with $T_x = 6.7$, $T_y = 5.4$

A defect-free image F' can be built based on the building block extracted. Looking at *Fig. 2* above, assume $T_x = 6.7$, $T_y = 5.4$ and imagine that the defect-free image in the same size as the original one is built up by building blocks of the size $6.7*5.4$. Here we assume each building block has its own coordinate system XOY and (x, y) is relative to the local origin $(1,1)$ (see *Fig. 2*). The gray level value of each point (x, y) within the (i, j)th building block is denoted by $B_{i,j}(x, y)$. The corresponding coordinates (k,l) in F' for any $B_{i,j}(x, y)$ can be obtained from *Eq. 3*.

Eq. 3:

$$k = (i-1) * T_x + x, \qquad l = (j-1) * T_y + y$$

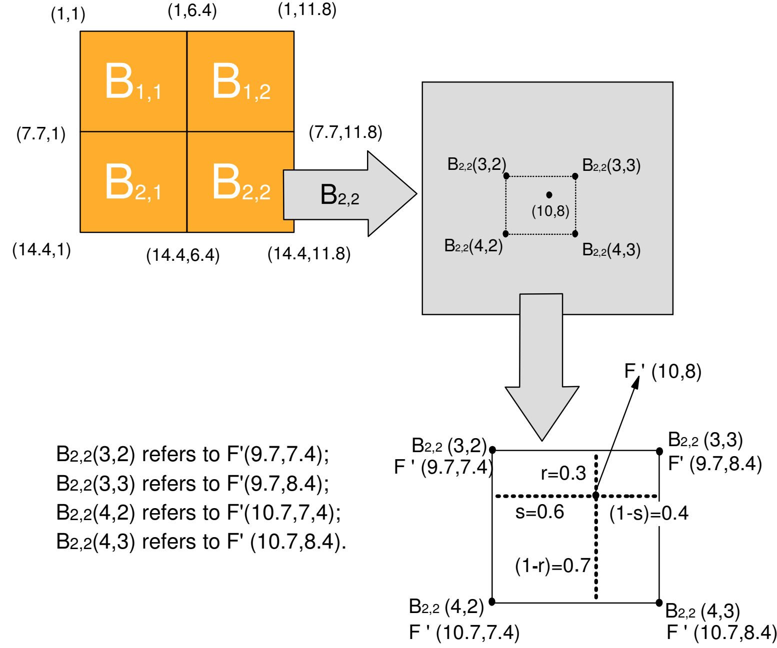

Fig. 3 Constructing a $M * N$ defect-free image from building blocks for an image with $T_x = 6.7$, $T_y = 5.4$

Let us have a look at *Fig. 3* that tells us how to calculate the gray level value of pixel $(10,8)$ in the defect-free image. $(10,8)$ is located in building block $B_{2,2}$. It is horizontally *0.6* pixel away from $B_{2,2}(i,2)\ (i = 3,4)$, *0.4* pixel away from $B_{2,2}(i,3)\ (i = 3,4)$, and vertically *0.3* pixel away from $B_{2,2}(3, j)\ (j = 2,3)$, 0.7 pixel away form $B_{2,2}(4, j)\ (j = 2,3)$. Its gray level value can be obtained from the values of $B_{2,2}(3,2)$, $B_{2,2}(3,3)$, $B_{2,2}(4,2)$ and $B_{2,2}(4,3)$ using linear interpolation.

To sum up, the gray level value of a pixel in the new image F' is calculated with the following equations, which can be easily obtained from basic geometric deduction.

Eq. 4:

$$F'(k,l) = (1-r)(1-s)B_{i_1,j_1}(x_1, y_1) + r(1-s)B_{i_2,j_1}(x_2, y_1) + s(1-r)B_{i_1,j_2}(x_1, y_2) + rsB_{i_2,j_2}(x_2, y_2)$$

Assume $F^{'}(k,l)$ is the value of the $M*N$ new image at location (k,l). When the four interpolating pixels of $F^{'}(k,l)$ sit within the same building block, we have:

$$i_1 = i_2 = \text{int}(k/T_x)+1, \qquad j_1 = j_2 = \text{int}(l/T_y)+1$$

$$x_1 = \text{int}(k - \text{int}(k/T_x)*T_x), \quad x_2 = x_1 + 1$$

$$y_1 = \text{int}(l - \text{int}(l/T_y)*T_y), \quad y_2 = y_1 + 1$$

$$r = \frac{k-k_1}{k_2-k_1} = k - \text{int}(k/T_x)*T_x - x_1, \quad s = \frac{l-l_1}{l_2-l_1} = l - \text{int}(l/T_y)*T_y - y_1$$

k_1, k_2, l_1 and l_2 are calculated from x_1, x_2, y_1, and y_2 using *Eq. 3*.

Note that *Eq. 4 and Eq. 1* in *Section 1.2* are symmetric as they represent reverse procedures relating $B(x,y)$ to $F(k,l)$ and $F^{'}(k,l)$.

It should be mentioned that there are special cases arising when using *Eq. 4*. They occur only when any of the four interpolating points of a pixel in $F^{'}$ are sitting on the borders of two or four adjacent building blocks. *Eq. 4* needs to be modified under such special cases. We explain these special cases in the Appendix.

2.4 Detecting the possible defects

Once the defect-free image is constructed, we simply compare each pixel in the original image with the corresponding pixel in the new image. If the difference is larger than a pre-chosen threshold, the pixel may be involved in a possible defect. We group the connected defect pixels together and get the resulting defect image by thresholding the size of each possible defect.

Finally, if the sizes of the defects in the original image are so large that the defect-free image obtained is not a good estimation of a golden template, we will receive many false alarms in the final results. The real defects might not be located accurately, but we will still be warned that this is a defective product anyway.

3. Results of the algorithm running over some samples

A Leitz Metallux 3 Microscope is used to magnify the IC chip, and an image grabber card with a PC (Pentium II 266) to catch the magnified IC images. With a 5 objective rate, the IC images is digitized into 9*4 pictures (640 pixels *480 pixels), and the resolution is about 8 micrometers per pixel. After adjustment, the uniformity of the illumination tested on a piece of uniform silicon substrate material is within ± 5 levels (out of 256 levels). (Note that our 8-bit gray-level, two dimensional sample images, which have repeating patterns, are only part of the digitized images of the whole IC chips, so their sizes are not 640 *480 pixels but smaller.)

The algorithm proposed in this paper is used to analyze the following sample images. The original image, defect-free image and the resulting defect image of each sample are shown.

- Sample 1 (size: 210×125)

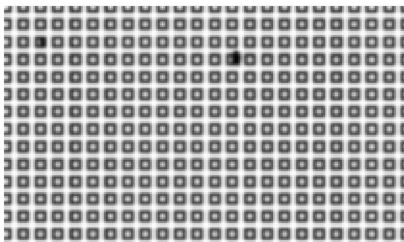

Fig. 4a Image from a chip with defects

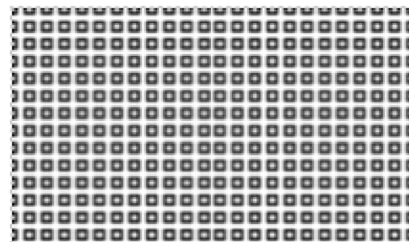

Fig. 4b Defect-free image

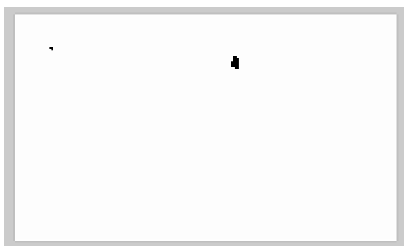

Fig. 4c Resulting defect image

- Sample 2 (size: 120×199)

From left to right:

Fig. 5a Image from a chip with different repeating patterns in rows and columns
Fig. 5b Defect-free image **Fig. 5c** Resulting defect image

- Sample 3 (size: 152×111)

Fig. 6a Image from a chip with similar repeating Patterns in rows and columns **Fig. 6b** Defect-free image

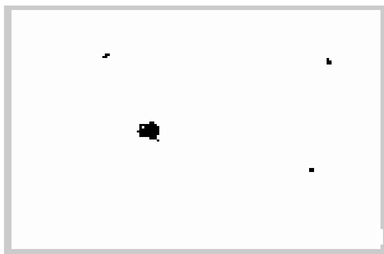

Fig. 6c Resulting defect image

- Sample 4 (size: 247×153)

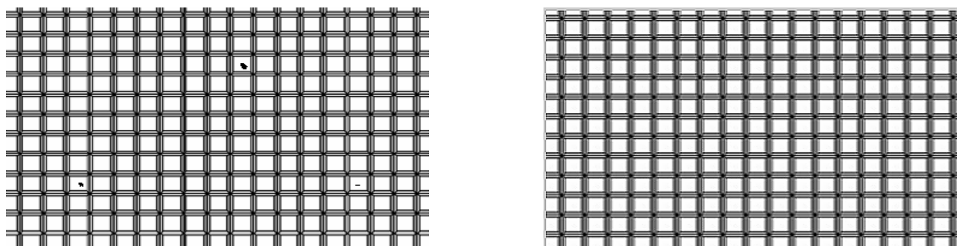

Fig. 7a Yet another image with defects **Fig. 7b** Defect-free image

Fig. 7c Resulting defect image

4 Discussions and conclusions

The proposed golden template self-generating method shows better results than Khalaj's algorithm because of the following improvements we have made.

- The definition of the size of building block has been made precise.

  Instead of defining an $m * n$ building block ($m = \text{int}(T_x) + 1,\ n = \text{int}(T_y) + 1$) with integer dimensions, we define a simulated building block whose size is $T_x * T_y$, which is the exact size of the repeating pattern.

- A defect-free image is generated using *Eq. 4*, which is consistent with the one (*Eq. 1*) through which we derive the values of the integer points in the simulated building block.

- Since the periods can be resolved with an accuracy at the order of 0.01 pixels and the size of the sample images in this paper are all smaller than 300×300, any small-scale shifting of the defect-free image compared to the original image can be ignored.

- In the final step of the algorithm described in *Section 1.2*, each pixel in the original image is compared with the corresponding point in the building block and its eight neighbors to relieve from the quantization effects at the edges of the image (see *Eq. 2*). Choosing the right point from those nine neighboring points in the building block is actually a procedure of mapping one pixel in the

original image to one point in the building block, which is quite time-consuming because it repeats for every point in the original image.

- One questionable point in this procedure is that it may result in errors such as two connected pixels in the original image are often not mapped to adjacent points in the building block as they are supposed to be. We can see this in *Table 1* by listing those points in the building block mapped from the pixels from (1,1) to (15,15) in the original image (*Fig. 6a*) using the approach in *Section 1.2*. This happens because of those approximate calculations in *Eq. 1*. The questionable point does not happen in our method, because a defect image $F^{'}$ is built from the building block using interpolation and the values of $B_{i_1,j_1}(x,y)$ and $B_{i_2,j_2}(x,y)$ ($i_1 \neq i_2$ or $j_1 \neq j_2$) are different in most cases.

In this paper, a defect-free image is generated using the simulated building block. The way we define the simulated building block is consistent with the way we build the defect-free image. In the final step, each pixel in the original image is directly compared with the corresponding pixel in the defect-free image. This method avoids the possible errors incurred from the above-mentioned algorithm in *Section 1.2* and the computation time is greatly saved.

The resolution of all the samples in our experiments is *8* micrometers per pixel. We have conducted experiments contrasting our throughput with that of the scheme in *Section 1.2*. The throughput of the self-reference methods described in *Section 1.2* is about $2.85cm^2/\min$. The throughput of the algorithm proposed in this paper is about $6.3cm^2/\min$ (This is an average value obtained from our samples), with a defect rate of 98% and a false alarm rate of 0.3/image on the defects whose size is larger than

$\frac{\text{size of building block}}{15}$. The performance of our approach is more than two times better, and the defect-free image in our approach once generated can be repeatedly used in future inspections while the approach in *Section 1.2* needs to recalculate almost everything in each inspection.

- Since a defect-free image can be obtained, this algorithm acts as a bridge between self-reference methods and image-to-image reference methods. That is, the simulated building block for each self-repeating pattern can be stored in a database. In the later inspection for the same repeating pattern, we can use image-to-image reference methods.

Table 1: Points in the building block corresponding to the pixels from $(1,1)$ to $(15,15)$ in the original image (*Fig. 6a*) based on the technique in *Section 1.2*

(2,2)	(2,3)	(1,2)	(2,4)	(2,1)	(2,3)	(1,2)	(2,4)	(2,1)	(1,1)	(1,2)	(2,4)	(1,1)	(1,1)	(2,3)
(2,2)	(2,3)	(1,3)	(3,4)	(3,2)	(2,3)	(1,3)	(2,4)	(3,2)	(2,3)	(1,3)	(2,3)	(2,1)	(2,2)	(2,3)
(4,1)	(4,3)	(4,3)	(4,4)	(2,1)	(4,1)	(4,3)	(4,4)	(2,1)	(4,1)	(4,3)	(4,3)	(4,1)	(2,1)	(4,3)
(4,1)	(4,3)	(4,3)	(4,4)	(3,1)	(4,1)	(4,3)	(4,4)	(3,1)	(4,1)	(4,3)	(4,4)	(3,1)	(3,1)	(4,3)
(2,1)	(2,3)	(2,4)	(2,4)	(2,1)	(1,1)	(2,3)	(2,4)	(2,1)	(2,1)	(2,4)	(2,4)	(1,1)	(2,1)	(2,1)
(2,2)	(2,3)	(1,3)	(3,4)	(3,2)	(2,3)	(1,3)	(3,4)	(3,2)	(1,1)	(1,3)	(2,3)	(3,1)	(3,2)	(3,3)
(4,2)	(4,3)	(4,3)	(4,4)	(3,1)	(4,3)	(4,3)	(4,4)	(3,1)	(2,1)	(4,3)	(4,3)	(4,1)	(3,1)	(3,1)
(3,1)	(4,3)	(4,3)	(4,4)	(3,1)	(4,3)	(4,3)	(4,4)	(3,1)	(2,1)	(4,3)	(4,4)	(3,1)	(3,1)	(3,1)
(1,1)	(1,1)	(1,4)	(1,4)	(1,1)	(1,1)	(1,4)	(1,4)	(1,1)	(1,1)	(1,4)	(1,4)	(1,1)	(1,1)	(1,1)
(2,2)	(2,3)	(1,3)	(2,4)	(2,1)	(2,3)	(1,3)	(2,4)	(2,1)	(2,3)	(1,3)	(1,3)	(1,1)	(2,2)	(2,3)
(1,1)	(2,3)	(1,3)	(3,4)	(3,2)	(2,3)	(1,3)	(3,3)	(3,2)	(1,1)	(1,2)	(3,3)	(3,1)	(1,1)	(2,3)
(4,1)	(3,1)	(4,4)	(4,4)	(3,1)	(3,1)	(4,4)	(4,4)	(3,1)	(3,1)	(4,4)	(4,4)	(3,1)	(3,1)	(3,1)
(1,1)	(1,1)	(1,4)	(1,4)	(1,1)	(1,1)	(1,4)	(1,4)	(1,1)	(1,1)	(1,4)	(1,4)	(1,1)	(1,1)	(1,1)
(1,1)	(2,3)	(1,3)	(2,4)	(2,1)	(2,3)	(1,3)	(2,4)	(2,1)	(2,3)	(1,3)	(2,4)	(1,1)	(1,1)	(2,3)
(2,2)	(2,3)	(1,3)	(3,4)	(3,2)	(2,3)	(1,3)	(2,4)	(3,2)	(2,2)	(1,3)	(2,3)	(3,1)	(2,2)	(2,3)

References

[1] Paulraj A., Roy R., Kailath T. (1985) Estimation of signal parameters by rotational invariance techniques (ESPRIT). *Proc. of 19th Asilomar Conference on Circuits, Systems and Comp.*

[2] Babian F. (1986) Optical defect detection limits in semiconductor wafers and msks. *PhD thesis*, Stanford University, Stanford, Calif.

[3] Dom B.E., Brecher V.H., Bonner R., Batchelder J.S., Jaffe R.S. (1988) The P300: A system for automatic patterned wafer inspection, *Machine Vision and Applications* 1(3):205-221

[4] Chin R. T. (1988) "Survey Automated Visual Inspection: 1981 to 1987". *Computer Vision, Graphics and Image Processing*, 41: 346-381

[5] Roy R., Kailath T. (1989) ESPRIT: Estimation of signal parameters via rotational Invariance Techniques. *IEEE Trans. On ASSP*, 37(7):984-995

[6] Mital D.P., Khwang T.E. (1991) Microcomputer based low cost vision system for wafer inspection. *Intell. Robotics Proceedings of the International Symposium Bangalore, India SPIE* 1571:200-214

[7] Meisburger W, Brodie A, Desai A (1992) Low-voltage electronic-optical system for the high-speed inspection of integrated circuits. *J. Vacuum Sci. Technol.* B 10:2804-2808

[8] Khalaj B.H., Aghajan H.K., Kailath T. (1993) Digital image processing techniques for patterned wafer inspection. *SPIE* 1926

[9] Khalaj B. H., Aghajan H. K., Kailath T. (1994) Patterned wafer inspection by high resolution spectral estimation techniques. *Machine vision and applications*, 7:178-185

[10] Dom B.E., Brecher V. (1995) Recent advances in the automatic inspection of integrated circuits for pattern defects. *Machine Vision and Applications* 8:5-19

[11] Newman T.S., Jain A.K. (1995) A survey of automated visual inspection, *Computer Vision, Graphics, Image Processing*, 61(2): 231-262

[12] Moganti M., Ercal F., Dagli C.H., Tsunekawa S. (1996) Automatic PCB inspection algorithms: a survey, *Computer Vision Image Understanding*, 63(2): 287-313

[13] Chou P.B., Rao A.R.., Sturzenbecker M.C., Wu F.Y., Brecher V.H. (1997) Automatic defect classification for semiconductor manufacturing. *Machine Vision and Applications*, 9:201-214.

[14] Porat B. (1997) "A Course in Digital Signal Processing". *John Wiley & Sons, Inc.*, Chapter 4.4, pp.104-107

[15] Moganti M. Ercal F. (1998a) A subpattern level inspection system for printed circuit boards, *Computer Vision Image Understanding*, 70(1): 51-62

[16] Moganti M. Ercal F. (1998b) Segmentation of printed circuit board images into basic patterns, *Computer Vision Image Understanding*, 70(1): 74-86

[17] Chen C. H., Cheng T. H., Wu W. T., Driscoll S. (1998) "Machine Vision Algorithms for Semiconductor Wafer Inspection: A Project with Inspex", Proceedings of SPIE, 3521:221-228

[18] Guan S.U., Xie P. (1999) "A golden block self-generating scheme for continuous patterned wafer inspections", Proceedings of the 10th International Conference on Image Analysis and Processing, Sep. 1999, pp. 436-441

Appendix: Special cases in building a defect-free image (see Chapter 2.3)

The special cases occur only when any of the four interpolating points of a pixel in $F^{'}$ are sitting on the borders of two or four adjacent building blocks. *Eq. 4* needs to be modified under such special case. We will explain with two examples (*Fig. 8a* and *Fig. 8b*) in the following.

Let's look at $F^{'}(14,9)$ in *Fig. 8a*. It falls on the borders of $B_{2,2}$ and $B_{3,2}$. Substituting $k = 14$ and $l = 9$ into *Eq. 3*, we know that its value should be obtained from the values of $B_{3,2}(0,3)$, $B_{3,2}(1,3)$, $B_{3,2}(0,4)$, and $B_{3,2}(1,4)$ using linear interpolation. But $B_{3,2}(0,3)$ and $B_{3,2}(0,4)$ don't exist, they actually represent points

sitting in an adjacent building block, i.e. $B_{2,2}$. We should use $B_{2,2}(7,3)$, $B_{3,2}(1,3)$, $B_{2,2}(7,4)$, and $B_{3,2}(1,4)$ to calculate the value of $F^{'}(14,9)$. We know from *Eq. 3* that $B_{2,2}(7,3)$ and $B_{3,2}(1,3)$ correspond to coordinates $(13.7,8.4)$ and $(14.4,8.4)$ in $F^{'}$ respectively. So in *Fig. 4a* the distance from point *(1)* to point *(3)* is *0.7* pixel instead of *1* pixel in normal situations.

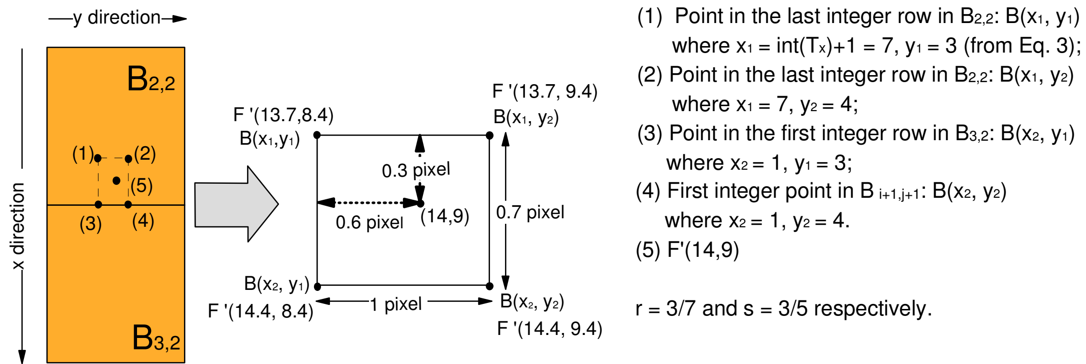

Fig. 8a A special case when two building blocks are involved in interpolation

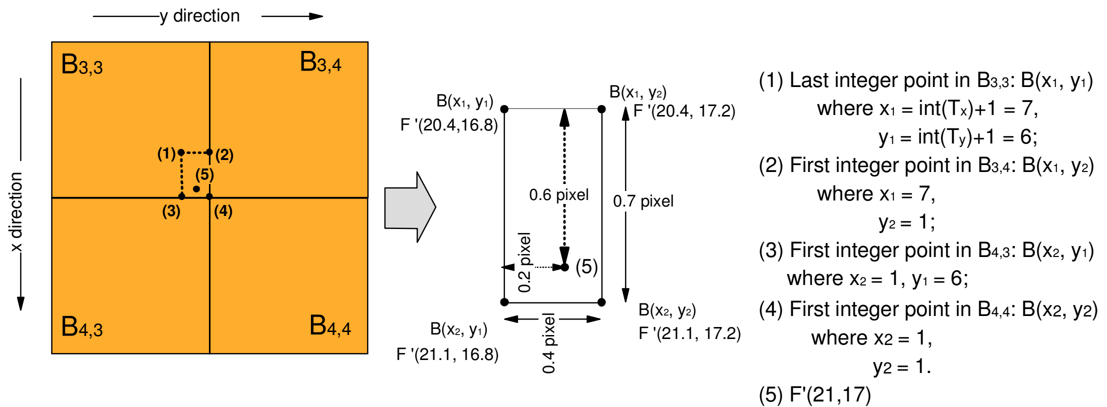

Fig. 8b Another special case of *Eq. 4* when four building blocks are involved in interpolation

$F^{'}(21,17)$ is another special case (See *Fig. 8b*). From *Eq. 4*, we know that it falls on the borders of four adjacent building blocks $B_{3,3}$, $B_{3,4}$, $B_{4,3}$ and $B_{4,4}$. As described above, its value comes from $B_{3,3}(7,6)$, $B_{3,4}(7,1)$, $B_{4,3}(1,6)$, and $B_{4,4}(1,1)$. We know from *Eq. 3* that $B_{3,3}(7,6)$, $B_{3,4}(7,1)$ and $B_{4,3}(1,6)$ correspond to coordinates

$(20.4,16.8)$, $(20.4,17.2)$ and $(21.1,16.8)$ in $F^{'}$ respectively. So in *Fig. 4b* the distance from point *(1)* to point *(2)* is *0.4* pixel and the distance from point *(1)* to point *(3)* is *0.7* pixel.

In summary, when we have any one of x_1 and y_1 in zero, or both x_1 and y_1 are zeroes, we should replace the corresponding parts of *Eq. 4* with the following respectively.

if $\text{int}(k - \text{int}(k / T_x) * T_x) = 0$, then

$$i_1 = \text{int}(k / T_x), \qquad i_2 = i_1 + 1, \qquad x_1 = \text{int}(T_x) + 1, \qquad x_2 = 1$$

$$r = \frac{k - k_1}{k_2 - k_1} = \{k - [\text{int}(k / T_x) - 1] * T_x - x_1\} / (T_x - \text{int}(T_x))$$

if $\text{int}(l - \text{int}(l / T_y) * T_y) = 0$, then

$$j_1 = \text{int}(l / T_y), \qquad j_2 = j_1 + 1, \qquad y_1 = \text{int}(T_y) + 1, \qquad y_2 = 1$$

$$s = \frac{l - l_1}{l_2 - l_1} = \{l - [\text{int}(l / T_y) - 1] * T_y - y_1]\} / (T_y - \text{int}(T_y))$$

Pin Xie received her B.E. in Automatic Control from University of Science and Technology of China in 1995. After two years' working in related area, she is now studying in the Electrical Engineering department of National University of Singapore as a postgraduate student.

Sheng-Uei Guan received his B.S. in mathematics from Tsing Hua University in 1979 and his M.S. and Ph.D. in computer science from the University of North Carolina at Chapel Hill in 1989. He was an associate professor of the Department of Computer Engineering and Science at Yuan-Ze Institute of Technology, Taiwan from 1992-1996. He was with the School of Computer Science and Computer Engineering, La Trobe University in Australia from 1996-1998. He is now with the Department of

Electrical Engineering at National University of Singapore. His research interests include Intelligent Machines and Vision, Computer Networks, Multimedia Systems, and E-Commerce.