A golden block based self-refining scheme for repetitive patterned wafer inspections

Sheng-Uei Guan, Pin Xie, Hong Li

Department of Electrical & Computer Engineering, National University of Singapore
10 Kent Ridge Crescent, Singapore 119260
Tel: (65) 68745153 Fax: (65) 67791103 E-mail: eleguans@nus.edu.sg

Abstract

This paper presents a novel technique for detecting possible defects in two-dimensional wafer images with repetitive patterns using prior knowledge. It has a learning ability that is able to create a golden block database from the wafer image itself, modify and refine its content when used in further inspections. The extracted building block is stored as a golden block for the detected pattern. When new wafer images with the same periodical pattern arrives, we do not have to re-calculate its periods and building block. A new building block can be derived directly from the existing golden block after eliminating alignment differences. If the newly derived building block has better quality than the stored golden block, then the golden block is replaced with the new building block. With the proposed algorithm, our implementation shows that a significant amount of processing time is saved. And the storage overhead of golden templates is also reduced significantly by storing golden blocks only.

Index terms: Wafer inspection - Golden template - PDI - Image-to-image reference method – Golden block

1. Introduction

Patterned wafer defect inspection (PDI) has evolved from a manual procedure to various automated inspection methods. It must be able to consistently capture real defects while minimizing the detection of false and nuisance events [1].

Two main categories of automatic inspection methods are described in the following paragraphs. Other methods such as optical spatial filtering methods [11], laser scattering methods [18] and wavelet technique [12] have also been involved in wafer inspection. A fairly complete review of the related literature may be found in [13-16].

Knowledge-based methods use one or a combination of two approaches: design-rule checking or image-to-image reference [2]. 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 [3]. 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 between the reference image and the target image is a major problem. A review of the related literature may be found in [6] [19].

All the above methods need a database of images or some prior knowledge. The problem is that sometimes we have only a single image that is under inspection. A self-reference technique that avoids the mentioned difficulties was developed by Dom et al. [7], in which the comparison is made using the repeating cells in the image. This method was further developed by Khalaj et al. [5] 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 [8, 9, 10] is used in estimating the frequency components. Then a building block representing the constructive structure of the patterns is extracted. In the final step, each point in the original image is compared with the corresponding point in the building block. If the difference is greater than a threshold, the point may be a possible defect.

The detailed description of this algorithm, its problems, and the improvement made by us were presented in the previous paper [17]. This paper is a continuous work of it, with the main differences described as follows. In [17], an algorithm was proposed to obtain a golden block from a single repeated patterned wafer image. In the current IC industry, thousands of wafer with the same pattern need to be inspected in sequence. The algorithm in this paper provides a solution to apply the obtained golden block continuously to all the incoming wafer images with the same repetitive pattern by taking care of alignment differences. The proposed scheme is also able to refine the registered golden block when a new image with fewer defects appears.

The new algorithm is described in *Sec. 2*. Some of the results images are shown in *Sec. 3*. Sec. 4 is our conclusions and further discussions.

2. The golden block based self-refining algorithm

In [17], we have described an algorithm that is capable of estimating the periods of repetition, extracting the building block, building a defect-free image and detecting possible defects. At the end of that process, we group the connected defect pixels together and get the resulting defect image by thresholding on the size of each possible defect.

Fig. 1a and *Fig. 1b* show us two resulting images before and after grouping with the final threshold. There are 145 groups in *Fig. 1a*, but only four of them contain more than three defect pixels. Three has been chosen as the threshold to detect defects in this case. The threshold should be a heuristic value set according to different criteria in different situations (see Section 3 and 4).

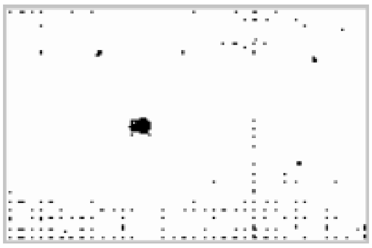


Fig. 1a Difference image

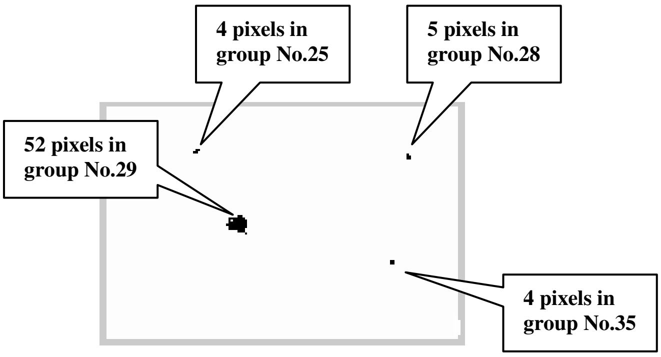


Fig. 1b Resulting image

In this step, we can easily get an important evaluation parameter using *Eq. 1*:

Eq. 1:

$$h = size_d / size$$

where $size_d$ is the total defect area of this image and $size$ is the total area of the image.

The building block extracted is saved as a golden block while the parameter h is saved as a quality metric for the golden block needed by the algorithm to be described in this paper.

Building block B has the collected characteristics of some pattern that we have learned from the wafer image in this inspection. B is called as a golden block, like the reference image or golden image in image-to-image reference methods. We keep the golden block B instead of the defect-free image F', through which the storage is significantly saved. For example, the size of the image and the extracted building block in *Fig. 2* is $365*175$ and $13*13$ respectively. The saving ratio is about 378.

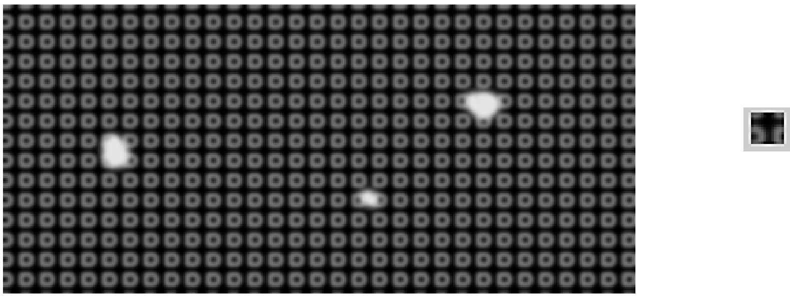


Fig. 2 One image and its building block

In the algorithm described below, we will explain how the knowledge obtained serves in later inspections.

2.1 Getting a new building block from the golden block

Now we describe how a new building block $B^{'}$ is obtained for a new incoming image F based on the golden block B we have stored for continuous patterned wafer inspections. The size and the illumination of an incoming image may be different from those of the former ones, but the size and the shape of the repeating pattern are exactly the same. So how to find out defects by using the golden block is a problem similar to the alignment problem in image-to-image reference methods except that the problem now has been localized to the block-level.

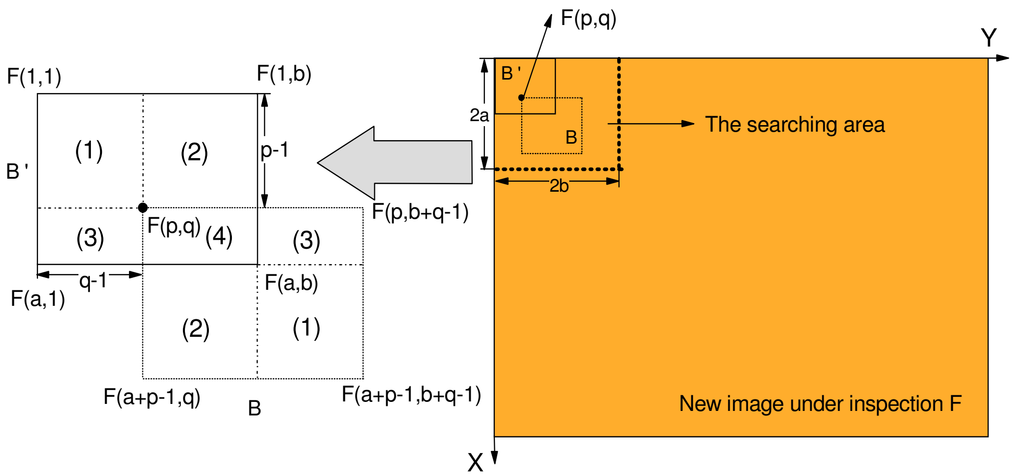


Fig. 3 Obtaining a new building block from the golden block

All the images start from pixel $(1,1)$ instead of $(0,0)$ in this paper. $F[x,y]$ denotes the pixel value of the pixel $F(x,y)$ in image F at location (x,y).

Assume the size of the new image under inspection F is $M*N$, the size of the golden block B is T_x*T_y, and the size of the integer part of B is $a*b$. We know that $a=\text{int}(T_x)$ and $b=\text{int}(T_y)$. The procedure to find the building block for F is the following:

1. Define a $(2*a)*(2*b)$ rectangle on the upper-left corner of F as the searching area (see *Fig. 3*). We will construct the building block $B^{'}$ for F in this searching area using the golden block B. This searching area starts from the upper-left corner of F because the defect-free image built from $B^{'}$ should have exactly the same pattern orientation as that of F. Moving B in this searching area until it covers a sub-area starting with $F(p,q)$ ($p\le a, q\le b$), which meets the following requirements:

   (a) $\sum_{i=p}^{p+a-1}\sum_{j=q}^{q+b-1} difference(i-p+1, j-q+1)/(a*b)$ is minimum

   where $difference(i-p+1, j-q+1) = abs(F[i,j]-B[i-p+1, j-q+1])$,

   $1\le p\le a+1$ and $1\le q\le b+1$.

   The *difference* set has the same size as that of the integer part of B. When $B's$ upper left corner moves to $F(p,q)$ in the current searching area, we place the difference between each corresponding pixel value into the *difference* set. Requirement (a) aims to find out a block in the current searching area starting from $F(p,q)$, with a size being the same as that of B and the average difference in pixel values between B and $B^{'}$ being the smallest.

(b) $\sigma < T$

where σ is the standard deviation of all the elements in *difference*. T is a pre-chosen threshold. $p \le i \le p+a-1$, $q \le j \le q+b-1$.

When a block satisfying requirement (a) is found, we check whether it also satisfies requirement (b). σ should be smaller than a certain threshold. Because if this block starting from $F(p,q)$ is the corresponding part in F for B, the elements in *difference* should be the illumination difference between the golden block B and the newly arrived image F. We assume here that the light source used during image scanning does not vary its intensity too much in its life-span.

2. The new building block $B^{'}$ is obtained through the following equations:

Eq. 2:

$$B^{'}[i,j] = B[a-p+i+1, b-q+j+1] + C$$

$$where \quad 1 \le i \le p, \quad 1 \le j \le q \qquad \ldots\ldots\ldots\ldots\ldots (1)$$

$$B^{'}[i,j] = B[a-p+i+1, j-q+1] + C$$

$$where \quad 1 \le i \le p, \quad q < j \le b \qquad \ldots\ldots\ldots\ldots\ldots (2)$$

$$B^{'}[i,j] = B[i-p+1, b-q+j+1] + C$$

$$where \quad p < i \le a, \quad 1 \le j \le q \qquad \ldots\ldots\ldots\ldots\ldots (3)$$

$$B^{'}[i,j] = B[i-p+1, j-q+1] + C$$

$$where \quad p < i \le a, \quad q < j \le b \qquad \ldots\ldots\ldots\ldots\ldots (4)$$

where

$$C = \sum_{i=p}^{p+a-1} \sum_{j=q}^{q+b-1} (F[i,j] - B[i-p+1, j-q+1]) / (a * b)$$

C is the averaged illumination difference between the new image F and the golden block B.

It must be mentioned that there is a constraint in the above algorithm that no defect larger than B appears in the searching area. Cases without this constraint will be described later.

The new image under inspection shown in *Fig. 4a* has the same pattern as the one shown in *Fig. 2*, but the image size is different and there is some shifting of the pattern in the image. Using *Eq. 2*, we can obtain the new building block from the golden block without re-calculating its periods and doing the interpolation again. The defect-free image built from the new building block and the resulting defect image are shown in *Fig. 4b* and *Fig. 4c* respectively. The result is satisfactory and the computation effort is saved.

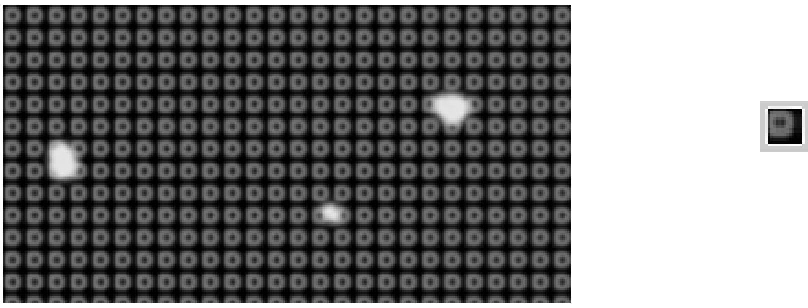


Fig. 4a New image under inspection and its building block calculated from Eq. 1

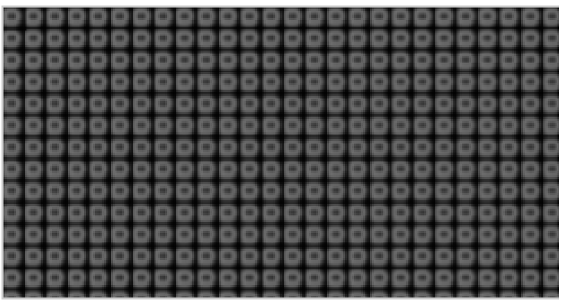


Fig. 4b Defect-free image

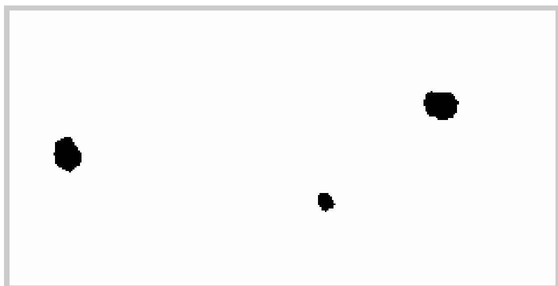


Fig. 4c Resulting image

Before ending this part of the algorithm, we calculate the evaluation parameter h' using *Eq. 1* for this new implementation.

If there happens to be a defect larger than B on the upper-left corner of F, we will not be able to find any (p,q) that meets the two requirements mentioned above. In this case, the searching area shouldn't start from the upper-left corner of the image any more. We have to shift it in the whole image until such (p,q) is found (See *Fig. 5a*).

Assuming the current searching area starts from $F(s_x, s_y)$, the revised algorithm is:

1. $while(s_x + 2a \le M)$

   { $while(s_y + 2b \le N)$

   { Search in the current searching area for a point (p,q) that meets the two requirements. (see *Fig. 5a*).

   if found go to Step 2;

   else $s_y = s_y + 2b;$

   }

   $s_x = s_x + 2a;$

   }

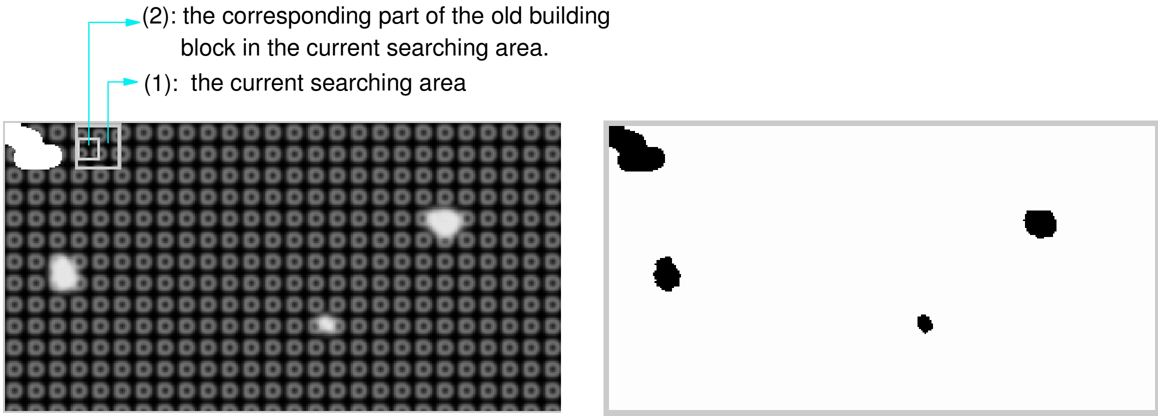
   If no such (p,q) can be found when the loop ends, we mark F as a bad image and go to Step 3.

2. Keep subtracting T_x from p until $p < a$, let $p = round(p)$;

   Keep subtracting T_y from q until $q < b$, let $q = round(q)$.

   The new building block B' is obtained through *Eq. 2*.

3. End.



From left to right: **Fig. 5a** Searching in a shifted searching area for (p, q)

Fig. 5b Resulting image of the image shown in Fig. 5a

In fact, we have solved the problem of alignment in this part of algorithm.

2.2 Refining the golden block

Both the golden block B and the new building block $B^{'}$ represent the structure of a single pattern. The better one should be placed into the golden block database. That is the reason why we keep h together with the golden block and why we calculate $h^{'}$ in *Sec. 2.1* before the algorithm ends.

Since a smaller h means a better wafer quality, it also means that the building block calculated from this image can represent the pattern better. So if $h < h^{'}$, the same B and h are kept in the database. Otherwise, we know that we have a better wafer image comparing to all those we have processed. So we re-calculate the golden block B based on the new image *F'* without repeating the time consuming step "estimating the periods of repetition".

In this way, the golden block can be refined continuously.

3. Results of the algorithm

The IC chip is magnified by a Leitz Metallux 3 microscope. The resolution of the captured images is about 8 micrometers per pixel.

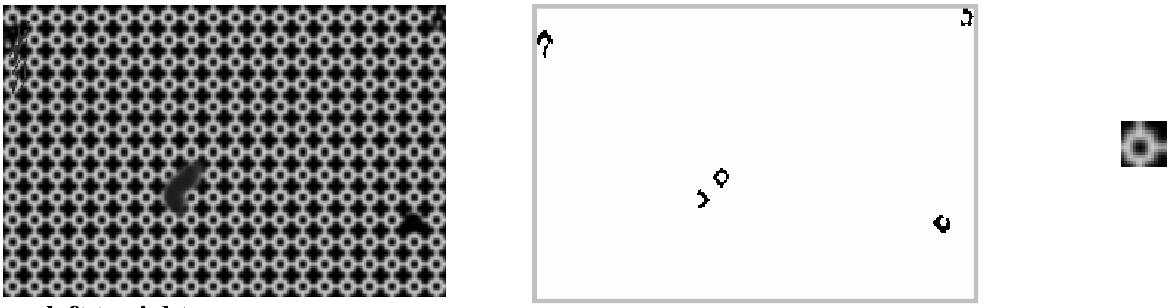
The above algorithm is used to analyze the following sample images. The accuracy of the algorithm is comparable to that of paper [17]. The sensitivity is 98% and the false alarm rate is 0.3/image. The threshold for the minimum defect size is $\frac{\text{size of building block}}{15}$. It has been derived empirically from our experiments to distinguish defects from noises. And this threshold is the only metric applied to consider whether a group of connected pixels is a defect compared to the regular pattern. So what affects the detection result is the defect size rather than the defect variability. The image size of *Fig. 4a* is 310*161. If it is processed with the algorithm described in the paper [17], it takes 0.31 seconds, which means that the throughput is about $6.3cm^2/\min$. However if it is processed with the algorithm described in this paper, it takes only 0.12 seconds, which means a throughput of about $16cm^2/\min$.

Five samples are shown in this section. We use image (a) to derive the original golden block, apply the algorithm described in this paper to image (b) to get the new building block, and then decide whether it is necessary to refine the existing golden block.

In the first three samples, the different quality of the two images in (a) and (b) gives us different h values (see Eq. 1 in Sec. 2.1). We use them to show how we refine the golden block.

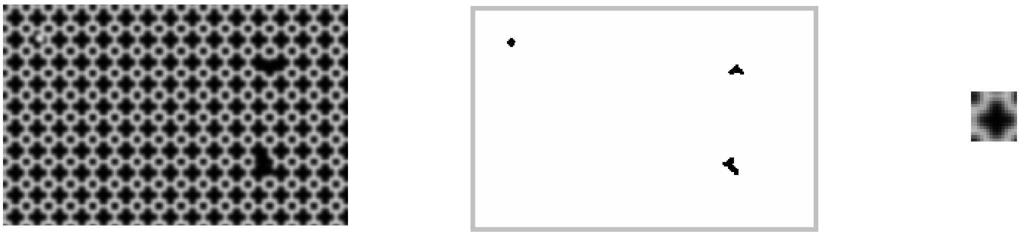
In the last two samples, Fig. 9(b) and Fig. 10(b) use the same wafer image as Fig. 9(a) and Fig. 10(a) respectively. We took the images again using the same equipment under the same condition. Only the position of the wafer and the illumination may change. We use them to show how we solved the alignment problem and get the new building block.

- Sample 1



From left to right:

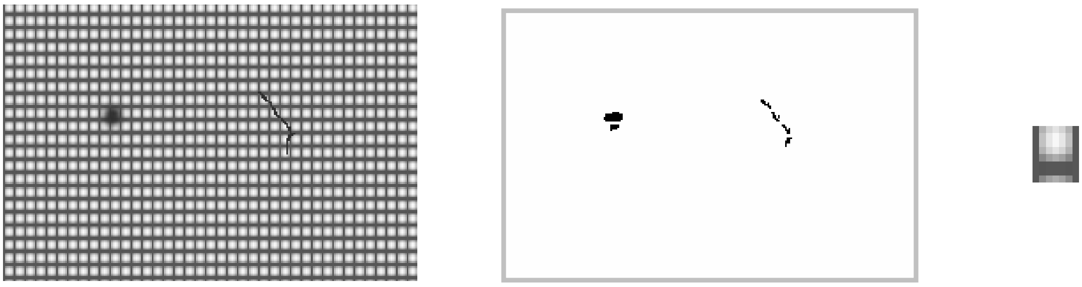
Fig. 6(a) Image used in the first time's learning (image size = 197*300)
Fig. 6(b) Defect image of Fig. 6(a) $h = 0.00500$
Fig. 6(c) Golden block extracted from Fig. 6(a)



From left to right:

Fig. 6(d) New image under inspection (image size =149*233)
Fig. 6(e) Defect image of Fig. 6(d) $h' = 0.00360$
Fig. 6(f) New building block extracted from Fig. 6(d)
$h > h'$, the building block shown in Fig. 6(f) becomes the new golden block. The original golden block shown in Fig. 6(c) is discarded.

- Sample 2



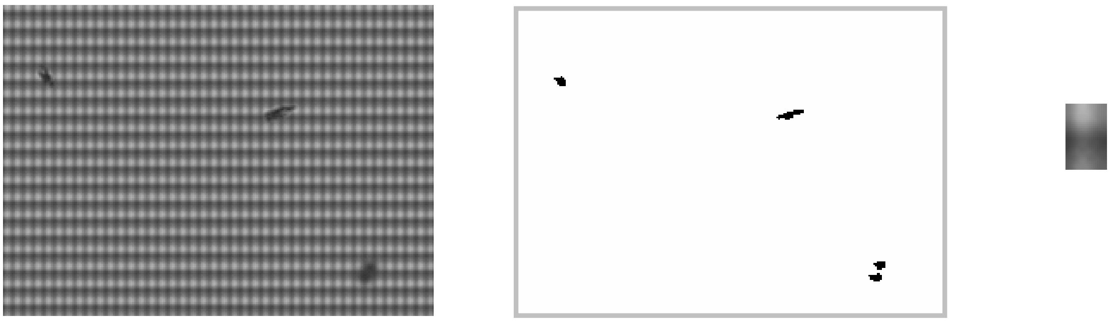
From left to right:

Fig. 7(a) Image used in the first time's learning (image size =168*234)
Fig. 7(b) Defect image of Fig. 7(a) $h = 0.00147$
Fig. 7(c) Golden block extracted from Fig. 7(a)

From left to right:

Fig. 7(d) New image under inspection (image size =140*203)
Fig. 7(e) Defect image of Fig. 7(d) $h' = 0.00054$
Fig. 7(f) New building block extracted from Fig. 7(d)
$h > h'$, the building block shown in Fig. 7(f) becomes the new golden block. The original golden block shown in Fig. 7(c) is discarded.

- Sample 3



From left to right:

Fig. 8(a) Image used in the first time's learning (image size =180*250)
Fig. 8(b) Defect image of Fig. 8(a) $h = 0.00309$
Fig. 8(c) Golden block extracted from Fig. 8(a)

From left to right:

Fig. 8(d) New image under inspection (image size =134*224)
Fig. 8(e) Defect image of Fig. 8(d) $h' = 0.00333$
Fig. 8(f) New building block extracted from Fig. 8(d)
$h < h'$, continue using the building block shown in Fig. 8(c) as the golden block.

- Sample 4

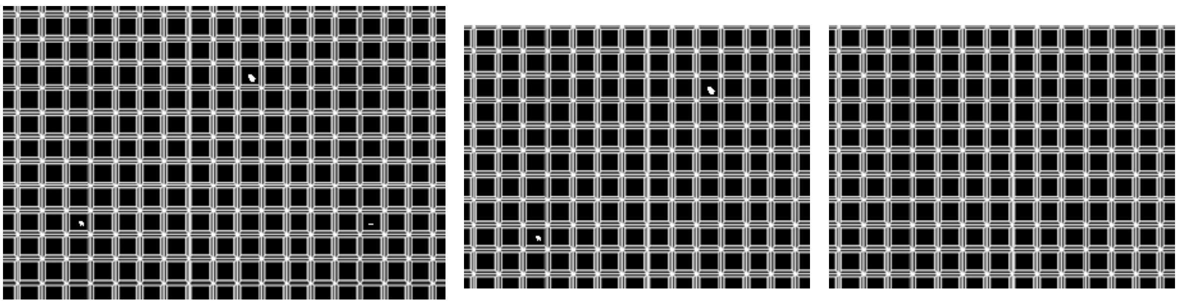
From left to right:

Fig. 9(a) Image used in the first time's learning (image size = 199*120)
Fig. 9(b) New image under inspection (image size = 185*113)
Fig. 9(c) Defect-free image of Fig. 9(b)
Fig. 9(d) Defect image of Fig. 9(b) h' = 0.00047
* The image in Fig. 9(a) contains the same defects, but the image size is larger. We have h = 0.00041.

From left to right:

Fig. 9(e) Golden block extracted from Fig. 9(a)
Fig. 9(f) New building block extracted from Fig. 9(b)
$h < h'$, continue using the building block shown in Fig. 9(e) as the golden block.

- Sample 5



From left to right:

Fig. 10(a) Image used in the first time's learning (image size = 232*350)
Fig. 10(b) New image under inspection (image size = 205*270)
Fig. 10(c) Defect-free image of Fig. 10(b)

From left to right:

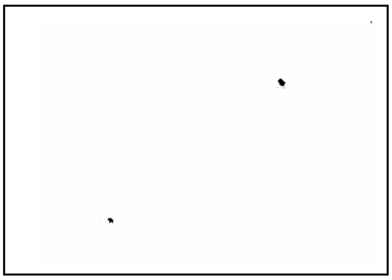
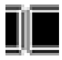
  

Fig. 10(d) Defect image of Fig. 10(b) h = 0.00053
*The image in Fig. 10(a) contains the same defects, but the image size is bigger. We have h = 0.00036
Fig. 10(e) Golden block extracted from Fig. 10(a)
Fig. 10(f) New building block extracted from Fig. 10(b)
$h < h'$, continue using the building block shown in Fig. 10(e) as the golden block.

4. Discussions

- How do we decide which periodical pattern/golden block should be used for an incoming new wafer image? We assume it can be decided from some pre-specified information attached with each image, i.e. wafer ID/product ID/wafer type, etc.

- The simulated building block for each self-repeating pattern can be stored in a database as the golden block of some pattern. It is the knowledge we learned from one implementation. The storage of a golden template in our approach has been reduced to the storage cost of a golden block. The ratio of saving is $(M * N)/(T_x * T_y)$ which can be significant over a large database of golden templates.

- When a new wafer image with the same continuous pattern comes, its building block can be constructed directly from the golden block in the database. We don't have to repeat all the steps in the first part of the algorithm described in [17], which means a significant amount of computation can be saved and the throughput is about 2.6 times better than that in [17].

- We use a minimum defect size threshold of approximately $\frac{\text{size of building block}}{15}$, because, it minimizes the detection of nuisance defects, efficiently sorts major defects from nuisance defects and image noises in all our data sets.
- An evaluation parameter $h = size_d / size$, which is defined to grade the golden block as a measure of confidence. The golden block is refined by comparing the stored evaluation parameter h to $h^{'}$ calculated from the new image. Why does the parameter $h\,(h^{'})$ describe the quality of the building block? Although the building block seems to be a small portion of the image, it was derived through an averaging process [17] to eliminate the effect from noises and defects. So the building block itself is a representative of all pattern blocks in the image.
- The method is superior to the conventional golden template method because it avoids all the issues involved in alignment of the golden template and wafer images. Also the effect of uneven illumination has been removed from the defect-free images because it is made from the golden block, which is an average result.

5. Conclusions

In this paper, we present an algorithm for detecting possible defects in two-dimensional, periodic wafer images. It has a learning ability that is able to create a golden block database from the wafer image itself, modify and refine its content when used in further inspections. The extracted building block is stored as a golden block for the detected pattern. Thus we do not have to re-calculate its periods and building block every time. A new building block can be derived directly from the existing golden block after eliminating alignment differences. The stored golden block is refined every time when a newly derived building block with better quality appears. With the proposed algorithm, our implementation shows that a significant amount of processing time is saved. And the storage overhead of golden templates is also reduced significantly by storing golden blocks only. Although we simulated an industrial environment in a university lab, more evaluation needs to be done before it can be accepted by industry. Furthermore, this algorithm is for defect detection only. For future work, defect classification can be considered.

References

[1] J. Harrigan, M. Stoller (1991) - “Automated wafer inspection in the manufacturing line”, *Solid State Technology* ,Oct.

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

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

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

[5] B.H. Khalaj, H.K. Aghajan, T. Kailath (1993) - “Digital image processing techniques for patterned wafer inspection”, *Proc. Of SPIE*, vol.1926

[6] F. Babian (1986) - “Optical defect detection limits in semiconductor wafers and masks”, *Ph.D. thesis*, Stanford University, Stanford, Calif

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

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

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

[10] B. H. Khalaj, H. K. Aghajan, T. Kailath (1994) - “Patterned wafer inspection by high resolution spectral estimation techniques”, *Machine Vision and Applications*, 7: 178-185

[11] R. T. Chin (1988) - “Survey automated visual inspection: 1981 to 1987”. *Computer Vision, Graphics and Image Processing*, 41: 346-381

[12] C. H. Chen, T. H. Cheng, W. T. Wu, S. Driscoll (1998) - “Machine vision algorithms for semiconductor wafer inspection: a Project with Inspex”, Proceedings of SPIE, 3521: 221-228

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

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

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

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

[17] P. Xie and S. U. Guan (2000) - "A golden template self-generating method for patterned wafer inspection", *Machine Vision and Applications,* 12: 149-156

[18] R. S. Howland, K. B. Wells and B. M. Trafas (1996) – “High-speed detection of pattern defects using laser scattering”, *Solid State Technology,* Vol. 38 Issue 11: 123-126

[19] (1996) – “A quick look at patterns [IC wafer inspection]”, *European Semiconductor,* Vol. 18 Issue 3: 57-58