1—3

Practical Pattern Detection from Distributed Defect Points on a Semiconductor Wafer

Hisae Shibuya [1]
Image Recognition and Inspection System Dept.
Production Engineering Research Laboratory
Hitachi Ltd.

Yuji Takagi [2]
Image Recognition and Inspection System Dept.
Production Engineering Research Laboratory
Hitachi Ltd.

Abstract

A spatial pattern recognition algorithm is proposed to determine root causes of defects on semiconductor wafers by visually identifying regional defect point set patterns. The algorithm classifies defects into either random patterns or regional patterns. Four different types of regional patterns are identified: rings, blobs, lines and arcs. Ring and blob patterns are detected by template matching techniques, while line and arc patterns are detected by utilizing their geometric properties.

The proposed algorithm was evaluated using 193 sample wafers with regional defect patterns. 182 of the samples (94.3%) were processed correctly. Processing time for a wafer containing 10,000 defect points was 6 sec using the Pentium^{TM} IV 1.4 GHz microprocessor.

1 Introduction

High-speed and high-sensitivity visual inspection tools are currently in practical use for semiconductor wafer inspection for defects. Defects detected are then analysed visually during a defect review. This review step is important for determining the types of defects that have occurred and analyzing the root causes of critical defects. In practice, defect reviews are performed using scanning electron microscopy.

Since the number of defects identified increases as pattern dimensions decrease, it becomes impractical to review all detected defects within a given length of time. This problem can be addressed to some extent by classifying defects as either "regional defects" or "random defects" based on spatial distributions of defects on the wafer [1]. This makes it possible to characterize defect occurrence of an entire wafer in a statistical manner by reviewing a small number of defect samples chosen from each defect class.

Furthermore, in many cases it is possible to determine the root cause of defects from regional defect patterns, as regional defects are strongly related to process abnormalities. It is well known that serious process abnormalities, such as mechanical scratching, wafer-edge film exfoliation and tool-triggered contamination, produce characteristic spatial defect patterns.

A method to extract regional defect patterns from defect distribution maps using image processing has been proposed [2,3,4]. A problem with this method is that it is difficult to determine the optimum pixel size for various point set densities. A method to extract regional defect patterns as a cluster by applying robust estimation has been reported [5]. However, it is difficult to apply this method practically because of the long computational times involved.

The present authors have developed a spatial pattern recognition algorithm for practical use in semiconductor fabrication processes. This algorithm distinguishes regional defect patterns from random patterns. Figure 1 shows examples of maps of detected defects (wafer maps) to be analyzed. Taking into account variations in regional defects patterns and semiconductor fabrication process mechanisms, four major defect pattern classes have been chosen for examination; ring, blob, line and arc. For detection of ring and blob patterns, the proposed algorithm creates a "density image" with an optimized pixel size and applies template-matching techniques to the binarized density image. For line and arc patterns, variations in shape and location are so large that template-matching techniques are impractical from a computational cost point of view. Therefore, geometric properties are utilized in order to identify line and arc patterns efficiently.

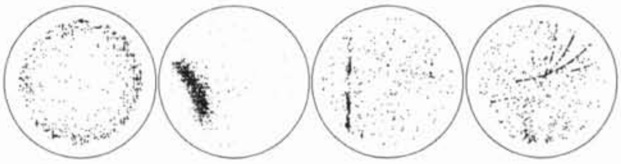


Figure 1 Maps of distrbuted defects points

2 Ring and Blob Pattern Detection

2.1 Voronoi Diagram

The ring and blob pattern detection algorithm is based on an analysis of point set densities. Voronoi diagrams are applied to calculate densities at each point. The Voronoi diagram is defined as follows.

For n points $p_i = (x_i, y_i)$ in a plane, the tile $V(p_i)$ of point p_i is defined as the set

$$V(p_i) = \{p = (x,y) \mid d(p,p_i) < d(p,p_j), \forall j \neq i\} \cdots (1)$$

where $d(p,p_i)$ is the Euclidean distance between points p and p_i. The diagram consisting of all $V(p_i)$ is called the Voronoi diagram. Point p_i is called the generator of tile $V(p_i)$. Figure 2 shows an example of a Voronoi diagram. Each

[1] Address: 292 Yoshida-cho, Totsukaku, Yokohama 244-0817 Japan. E-mail: **shibuya.hisae@gm.perl.hitachi.co.jp**

[2] Address: : 292 Yoshida-cho, Totsukaku, Yokohama 244-0817 Japan. E-mail: **takagi@perl.hitachi.co.jp**

boundary between 2 tiles is a segment of the perpendicular bisector of the generators of the tiles.

Local density $D(p_i)$ of point p_i is given by

$$D(p_i) = \frac{1}{S(V(p_i))} \cdots (2)$$

where $S(V(p_i))$ is area of the tile $V(p_i)$.

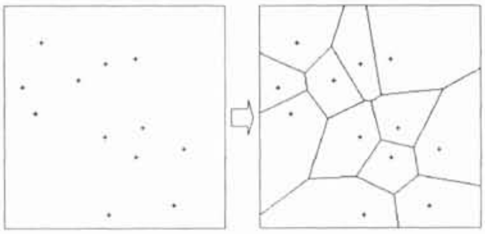


Figure 2 An example of aVoronoi diagram

2.2 Pattern Extraction

In order to meaningfully visualize a wafer for the purpose of regional defect classification, the density image that expresses the distribution of the point densities over the wafer is created. Optimization of pixel size is essential for correctly extracting and identifying regional defect patterns.

Figure 3 shows a schematic of the ring and blob pattern detection algorithm. After local densities $D(p_i)$ have been defined by a Voronoi diagram, a histogram of point densities is calculated to determine the modal density value D_{mode}. Pixel size of the density image is then set to be

$$PIxel\ size = \sqrt{n / D_{mode}} \cdots (3)$$

where n is an arbitrary number. Thus a higher density point set produces a smaller pixel size.

The pixel value is defined as the number of defect points falling within the pixel. However, in cases where many points are concentrated within a few pixels, the levels of other pixels are depressed when the pixel set is normalized to achieve a maximum pixel value, such as 255.

To reduce this undesirable effect, every point is weighted in proportion to the square of the nearest neighbor distance.

Regional defect patterns are then extracted by binarizing the density image in a statistical manner.

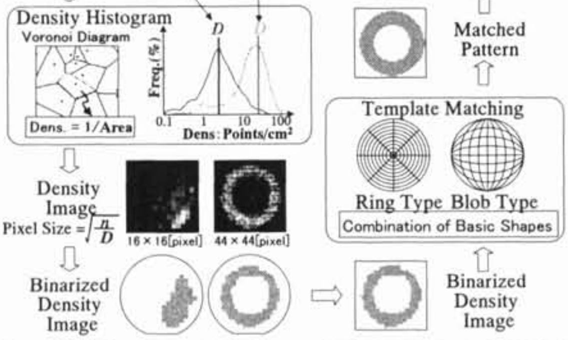


Figure 3 Algorithm for ring and blob pattern detection

2.3 Pattern Identification Using Template Matching

Patterns are identified by matching the binarized density image against a set of templates. To avoid a blowout in computational time caused by a large number of templates, templates are resolved into a set of smaller components of common basic shapes. Sets of common basic shapes are provided for both ring and blob type patterns.

Figure 4 shows the method used to construct a set of basic shapes for ring type templates. Basic shapes are generated by dividing a circle into N rings of concentric circles (division A) and dividing the same circle into M fans (division B). Templates are produced by combination of division A and division B shapes. The total number of templates $C(N,M)$ is then given by

$$C(N,M) = \frac{N(N+1)}{2} \bullet \{M(M+1)+1\} \cdots (4)$$

In our study, N is set to 10 and M is set to 8, giving a total of 3135 template variations.

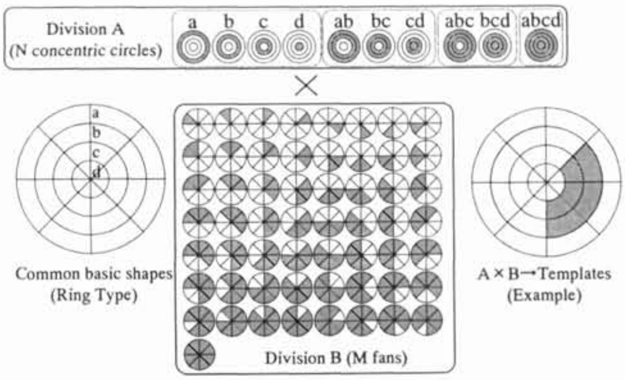


Figure 4 Construction of templates

The matching score S_k of the k-th template is calculated by

$$S_k = \sum_i a_i^{(k)} s_i \cdots (5).$$

where i is the index of the i-th basic shape, s_i is the matching score of the i-th basic shape, and $a_i^{(k)}$ is set to 1 when the i-th basic shape is part of the k-th template or set to -1 otherwise. The score s_i is defined as the total number of matched pixels minus the total number of unmatched pixels, with all s_i obtained by a single image scan. This enables fast template matching. Figure 5 shows computational times of this method compared to a conventional method for various image sizes. The conventional method calculates a matching score each time for all templates. The conventional method does not include template generation time, while our method does. It can be seen that computational time does not increase significantly with image size when our method is used, with computational time dropping to 25% of that of the conventional method in the case of a 32x32 image.

Next, the matched template region is checked to see if it is actually a random pattern instead. Such template matches need to be rejected. Density contrast C is used to check for this condition, and is defined as

$$C = \frac{\omega_p \omega_b (\mu_p - \mu_b)^2}{\omega_p \sigma_p^2 + \omega_b \sigma_b^2} \quad \cdots (6)$$

where ω_p (ω_b) is the number of pixels within (outside of) the template pattern, μ_p (μ_b) is the average pixel value within (outside of) the template pattern and σ_p (σ_b) is the standard deviation of pixel value across pixels within (outside of) the template pattern. Template patterns are rejected if density contrast is lower than a predefined threshold.

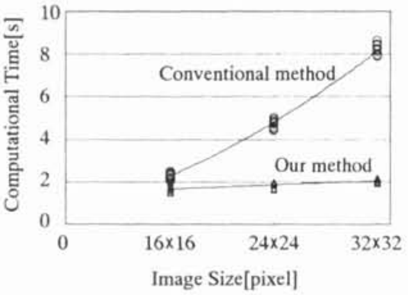


Figure 5 Comparison of computational times of different template matching methods

3 Arc Pattern Detection

Geometric properties are used to efficiently identify line and arc patterns. The Hough transformation is used to identify line patterns, while arc patterns are identified by forming a voting image from geometrical parameters of possible arcs.

Figure 6 shows the arc pattern detection algorithm. The proposed algorithm is comprised of two steps; the first step calculates the center of the arc and the second step determines the arc bounds.

In the first step, the geometric property that the perpendicular bisector of two arbitrary points of an arc crosses the center of curvature of the arc is utilized. Perpendicular bisectors of all sets of two arbitrary defect points are drawn onto a voting image V. The center of the arc is then set to be the peak in the voting image V. To reduce processing time, the number of points that contribute to voting is adjusted to be less than 500. To achieve reliable detection, the voting contribution of a certain pair of points is determined by the following rules.

(1) If the nearest neighbor distance of a given point is more than d_{max}, the contribution is set to 0, because isolated points cannot be a part of a defect pattern.

(2) If all of the neighbor distances of a point are less than d_{min}, the contribution is set to 0 in order to exclude extremely concentrated defects.

(3) Otherwise, the contribution w is calculated from $w = \exp\left(-(d - d_{st})^2 / k\sigma^2\right) \cdots (7)$

where d is the distance between two points and d_{st} ,σ and k are control parameters. The purpose of this equation is to depress the contribution when the distance d is too short or too long, because the error of the slope of the perpendicular bisector is large when the distance d is too short, and it is less likely that two points exist on the same arc when the distance d is too long.

Values of the parameters d_{max}, d_{min}, d_{st} , σ, and k are determined by trial and error.

In the second step, coordinates of all defects are transformed into polar coordinates with origin at the center of curvature of the arc. Consequently, the arc pattern on the wafer is transformed into a horizontal line in the image T expressed in polar coordinates. This transformation makes arc defect detection much easier. To determine arc parameters, the following procedure is used

(1) The peak position R is detected by projecting the point data in polar coordinates onto the radial (r) axis.

(2) The bounds of the interval r_0=R in the image T are found by searching for a roof-edge along the angular (θ) direction.

(3) The longest roof-edge [θ_1,θ_2] is extracted. The arc pattern in the wafer is identified by transforming the roof-edge, i.e. section [θ_1,θ_2], back into Cartesian coordinates.

The above procedure is applied repeatedly until no more roof-edges can be found.

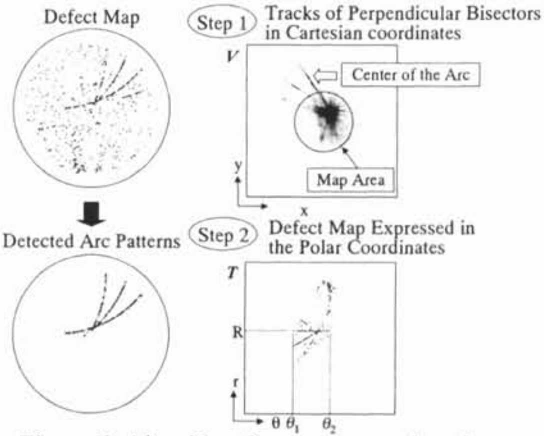

Defect Map
Step 1
Tracks of Perpendicular Bisectors in Cartesian coordinates
V
Center of the Arc
Map Area
y
x
Detected Arc Patterns
Step 2
Defect Map Expressed in the Polar Coordinates
T
R
r
θ θ_1 θ_2

Figure 6 Algorithm for arc pattern detecion

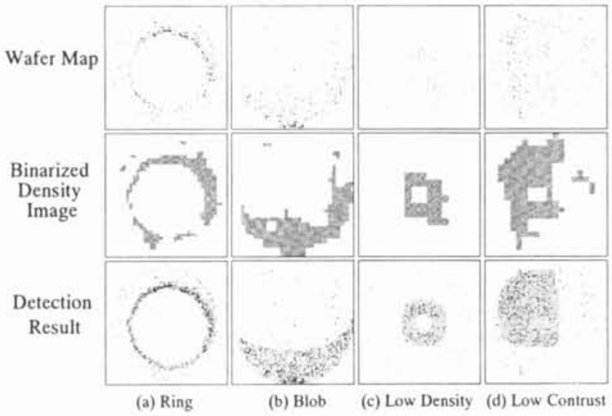
4 Evaluation

The proposed algorithm was tested using 193 test samples. Examples of correct identification of ring and blob patterns are shown in Figure 7. It is shown that correct detection is done without regards to density variations. Examples of correct identification of arc patterns are shown in Figure 8. The performance of this pattern detection algorithm is detailed in Table 1. Accuracy varied from 87.5% to 100% depending on the pattern class. The total accuracy was 94.3%.

As Figure 7 shows, ring and blob patterns shown in (a) and (b) both have high densities, the ring pattern shown in (c) has a low density and the blob pattern shown in (d) shows a low contrast pattern. In other words, various densities and contrasts can be found in a single pattern class. The line and arc patterns have specifically identified geometric properties, while blob and ring patterns allow for density variations and pattern contrast variations.

The accuracy of the blob and ring detection algorithm was tested against wafers obtained in a real semiconductor process. The evaluation was done using 111 sample wafers with ring and blob patterns flagged for scrutiny with respect to process anomalies. 110 of the samples (99.1%) were processed correctly.

The processing time for a wafer containing 10,000 defect points was 6 sec using the Pentium^{TM} IV 1.4 GHz microprocessor.



(a) Ring (b) Blob (c) Low Density (d) Low Contrust

Figure 7 Examples of detected ring and blob patterns

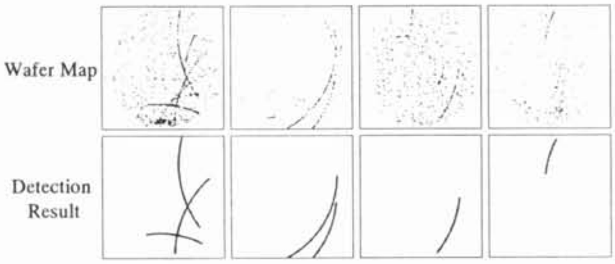


Figure 8 Examples of detected arc patterns

Table 1 Performance of pattern detection

		Visual Detection					
		Random	Blob	Ring	Line	Arc	Total
Automatical Detection	Random	67	0	0	0	2	69
	Blob	3	45	0	0	0	48
	Ring	0	0	24	0	0	24
	Line	0	2	0	32	0	34
	Arc	0	1	0	3	14	18
	Total	70	48	24	35	16	193
	Accuracy	95.7%	93.8%	100%	91.4%	87.5%	94.3%

5 Conclusion

A spatial pattern recognition algorithm was proposed to determine root causes of semiconductor wafer defects from visually recognized regional defect point set patterns of the wafers. The algorithm classifies defects into either random patterns or regional patterns. The regional patterns include rings, blobs, lines and arcs. Ring and blob patterns are identified by template matching techniques. Line and arc patterns are identified by utilizing geometric properties of these shapes.

The proposed algorithm was evaluated using 193 sample wafers with regional defect patterns. 182 of the samples (94.3%) were processed correctly. Furthermore, the blob and ring pattern detection algorithm was tested against various defect density and pattern contrast conditions by selecting wafers flagged for scrutiny with respect to process anomalies. The test was performed on 111 sample wafers, and correctly processed 110 samples (99.1%).

The processing time for a wafer containing 10,000 defect points was 6 sec using the Pentium^{TM} IV 1.4 GHz microprocessor. This execution time is fast enough to apply to a real production line.

Detected regional patterns reveal useful information related to the root causes of wafer defects. Thus, small number of defect samples chosen from every regional pattern is enough to characterize defect occurrence by defect review. Consquently, this algorithm could be applied to effective defect review and analysis that would aid in a more reliable determination of the root causes of wafer defects.

References

[1] F. Mizuno, S. Isogai, “A method for yield prediction using a SEM-ADC”, In-line Characterization, Yield Reliability, and Failure Analyses in Microelectronic Manufacturing, pp. 332-342, SPIE. Vol. 3743, 1999

[2] S. S. Gleason, K. W. Tobin, “Directional-based Dilation for Connection of Piece-wise Objects: A Semiconductor Manufacturing Case Study”, International Conference on Image Processing, Sep. 1996

[3] K. W. Tobin, S. S. Gleason, T. P. Karnowski, et al., “Feature Analysis and Classification of Manufacturing Signatures on Semiconductor Wafers”, Int'l Soc. for Optical Eng. 9th Symposium on Electronic Imaging, San Jose CA, Feb. 1997

[4] M. Ono, K. Nemoto, M. Ariga, “An Effective Method for Yield Enhancement Using Zonal Defect Recognition”, Proc. Int'l Symposium on Semiconductor Manufacturing '97, pp. E25-E28, 1997

[5] J. M. Jolion, P. Meer and S. Bataouche, “Robust clustering with application in computer vision”, IEEE Trans. PAMI. 13(8), pp. 791-802, 1991