Defects and Diffusion Forum Vols. 230-232 (2004) pp 135-148
online at http://www.scientific.net
© (2004) Trans Tech Publications, Switzerland

Classification of Defects on Semiconductor Wafers using Priority Rules

N.G. Shankar[1], Z.W. Zhong[2] and N. Ravi[3]

[1]Euro Technology Pte Ltd, Singapore
[2]School of MPE, Nanyang Technological University, Singapore
[3]ISDD UK Ltd, UK

Abstract

This paper presents a template-based vision system to detect and classify the nonuniformaties that appear on the semiconductor wafer surfaces. Design goals include detection of flaws and correlation of defect features based on semiconductor industry expert's knowledge. The die pattern is generated and kept as the reference beforehand from the experts in the semiconductor industry. The system is capable of identifying the defects on the wafers after die sawing. Each unique defect structure is defined as an object. Objects are grouped into user-defined categories such as chipping, metallization peel off, silicon dust contamination, etc., after die sawing and micro-crack, scratch, ink dot being washed off, bridging, etc., from the wafer. This paper also describes the vision system in terms of its hardware modules, as well as the image processing algorithms utilized to perform the functions.

Keywords: machine vision, defect inspection, defect type, referential inspection, wafer.

1. Introduction

In the majority of wafer fabs, the visual inspection process of wafer surfaces depends on manual review by human experts. These experts separate the defects into two categories: die saw defects and defects from wafers. The inspection task requires extreme concentration, and the time that an inspector could continue the inspection task is quite limited. Still it tends to be quite slow and inaccurate. The decision instability of an inspector can be quite large against various defect classes and each inspector relies on different features and strategies [1]. Hence, one cannot expect uniform classification of defects from all the human inspectors. The availability of multiple data sources and the evolution of automated analysis techniques such as automatic defect classification and spatial signature analysis are providing solutions to convert basic defect, parametric, electrical data into useful prediction and control information before shipment of wafers to the IC (integrated circuit) manufacturing. These techniques did not have any knowledge about defects generated during wafer mounting, dicing, ink marking, etc.

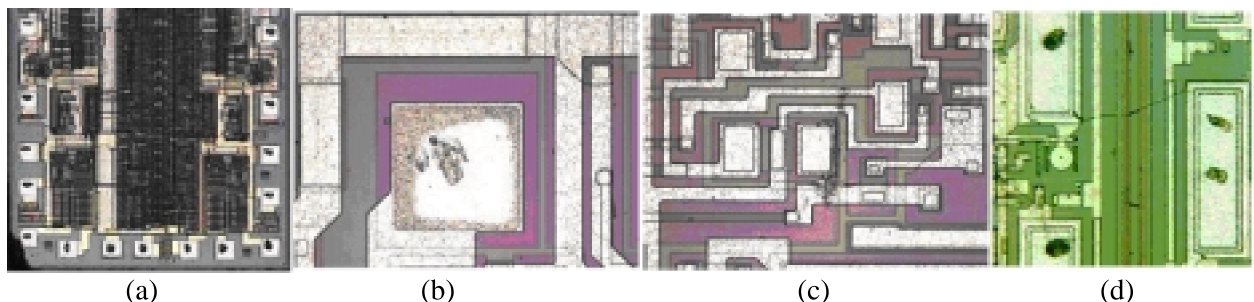
Studies have shown that up to 80% of the yield loss in the production of high-volume, very-large scale integrated (VLSI) circuits can be attributed to random visual particle and pattern defects [2]. The increasing demands for miniaturization, high electrical performance and high I/O pin count have resulted in intensive research and development of advanced IC (integrated circuit) packages [3, 4], which further results in much critical inspection task. Previous studies have demonstrated the ability to classify defects through the analysis of artificial neural nets (ANN) [5]. The objective of wafer inspection should include image acquisition, defect detection and defect classification. Issues based on defect classification were well explained by A.K. Jain et al. [6]. In general, automatic inspection systems use one or combinations of two approaches of either design rule checking or image-to-image comparison [7]. A pure design rule

All rights reserved. No part of contents of this paper may be reproduced or transmitted in any form or by any means without the written permission of the publisher: Trans Tech Publications Ltd, Switzerland, www.ttp.net. (ID: 130.203.133.34-15/04/08,14:20:52)

system should check and compare every pixel in the digital image under inspection with the corresponding pixel in the reference image. With this approach, image registration is a major problem. Optical special filtering [8] 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 recognized. A wavelet technique [9] has also been involved in wafer inspection. Complete reviews of the related literature were performed by Babian [10], Newman [11], Moganti [12] and Klaus [13]. The above methods need a database of images or some prior knowledge. A self-reference technique to overcome the mentioned difficulties was developed by Dom et al. [14], in which the comparison was made using the repeating cells in the image which was further developed by Khalaj et al. [15] by extracting the building block of repeating patterns from the acquired image. The defects are detected by comparing the resulting building block with the image and estimating the frequency components using the ESPRIT algorithm [16]. A golden-template self-generating method for patterned wafer image was discussed by Pin et al. [17] wherein a defect-free golden template was built based on the above [15] extracted building block. Rule-based inspections of semiconductor wafer surface have been reported [18, 19].

Basic studies reveal that each defect object has different surface intensity, shape, color, texture, etc. Each and every time the user has to teach the system to identify and recognize the defect objects by some means. Secondly the user has to teach the system for various matrix tolerances and it should be able to tell the user about the shape, size, and type of defects and how to segregate into various classes for future references. As manufacturing processes mature, the occurrence of anomalies arises from particle contamination, mechanical damages and process variations due to improper equipment calibration or miss-out calibration or poor maintenance of the equipment. Equipment with implemented defect pattern recognition algorithms had no prior knowledge of newly developed defect events and also the accuracy of detecting equipment slowly degrades due to aging. At present most of the front-of-line process equipment in the semiconductor assembly utilizes the defect data supplied by the wafer manufacturer as wafermaps. These wafermaps have no prior knowledge of the defects generated during the assembly process. Hence there is a necessity to detect these defects along with wafer manufacturer defect data (wafermap) before the die-bond process in the semiconductor assembly.

In general, it is difficult to detect defects because of the inherent variability of wafers in the assembly line. Although much of the work reported above has yielded results that are good in a quantitative sense, evaluation suggests that the defect analysis results could be improved in a qualitative sense by refining the resulting shapes and extent of detected defect regions in the images.



(a) (b) (c) (d)

Figure 1: Examples of various defect image patterns identified by manual inspectors in the assembly area, (a) chipping defect, (b) metallization peel-off defect, (c) scratch defect, and (d) bridging defect

For example, small spurious defect regions may have only a small effect on pixel-wise statistical classification accuracy, but are objectionable to the human observer. In many cases, such regions can be identified and eliminated easily. Figure 1 shows representative examples of the variety of defect images in semiconductor manufacturing. Figure 1 (a)-(d) show examples of individual defect pattern identified using optical and electron microscope by manual inspectors in the assembly area. Hence, it is clear that the defects that arise after wafer die sawing is discarded during die bonding. The goal of this paper is to describe the defect detection method and to accomplish defect classification after die saw by automated means.

2. System Description

The system works on the principle of comparing two images while identifying the faults. One image is called the Golden Master (GM) or Reference Image (RI), while the other image is the actual image under inspection. To have both images compared, they must be in the same position for comparison. The GM is stored in the referred lookup table, and an image for comparison is acquired on regular basis for inspection. The defect classification system can be separated into the following major steps.

- Golden Master (GM) or Reference Image Selection
- Matching of the GM and acquired Image
- Granulometry for defect’s segmentation

2.1 Golden Master (GM) or Reference Image Selection

The GM image selection is performed to identify the Region of Interest (ROI) in a wafer die image. Normally the ROI serves as valuable data used to match the GM with the acquired image using correlation. Also using a single frame for comparison to the actual image would end up with more “defects”, which may not be defects. There are many conditions, which can prevent two identical images acquired during two different points of time. Some of the critical factors, which play a major role in the selection of GM images, are illumination during the acquiring of images, accuracy in converting the analog video signal from CCD camera to digital data, external factors like heat, dust etc. This would reflect change in the pixel values at given x, y positions for two identical over-lapped images. A one to one comparison of pixel values would only lead to unknown errors. A built-in tolerance system is used to overcome changes in the GM image when two identical images acquired at two different points of time.

2.2 Matching of GM and Live Images Using Subtractive Correlation

The next process is to match the GM image to the actual image because the actual image is not always in synchronous with the GM image. There is always some shift in the actual images. Before doing any comparison both the images should be overlapped properly. For this we use a correlation technique to overlap both GM and actual images. The following steps explain the overlapping process.

Step 1

The ROI coordinates (x_{1r}, y_{1r}) and (x_{2r}, y_{2r}) are used for the analysis. The ROI window identified with the above coordinates is moved ± 10 pixels both in x and y directions in the actual live image I_{live} as shown in Figure 2.

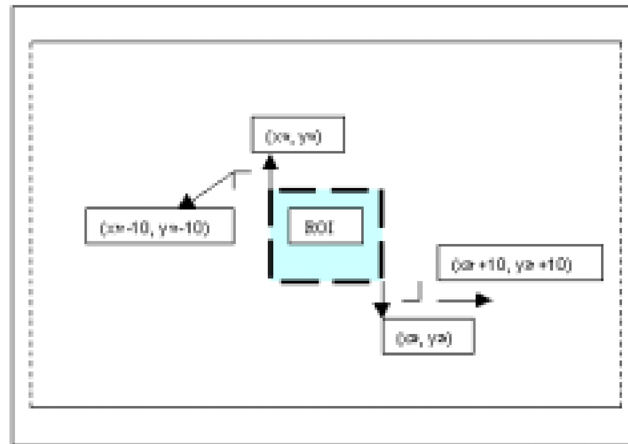


Figure 2. ROI in the actual live image

Step 2

For each position starting from (x_{1r} -10, y_{1r} -10) the pixels within the window area are subtracted to the pixels in ROI region of the GM image I_{avg} . For all window movements the pixels in the GM ROI would be the same and only the pixels in the actual image I_{live} would change for each window movement.

Step 3

The subtracted pixel values are accumulated for each window position and this step is repeated till ± 10 pixels movement in x and y directions.

Step 4

A typical loop for a given location x_1, y_1 and x_2, y_2 is calculated using correlation subtraction and it is expressed as,

$$for(i = x_1, k = x_{1r}; i <= x_2; i++, k++)$$
{
$$for(j = y_1, i = y_{1r}; j <= y_2; j++, i++)$$

$$\{$$
$$dr_{xy} = abs(I_{ref}(k,1) - I_{live}(i,j));$$
$$\}\,\} \qquad (1)$$

where $I_{refr}(k,1)$ is the GM image pixel intensity in the R plane at a given $(k,1)$ position, $I_{liver}(i,j)$ is the live image pixel intensity in the R plane at a given (i,j) position and dr_{xy} is the pixel intensity difference in the R plane in x and y directions.

In this process the window position where the pixel difference is minimum, identifies the position where the actual live image and GM image are in synchronization. The x and y coordinates of this position in the actual live image identified as $x_1^{'}, y_1^{'}$ are compared to the values x_{1r}, y_{1r} of the GM image to find the exact shift in terms of pixel values.

For example: let us assume that the x_{1r}, y_{1r} and x_{2r}, y_{2r} for the ROI in the GM image are 50, 50 and 100, 100 assuming a window dimension of 50×50. Now the movement of window in the actual live image would be 50 ± 10 in the x direction and 50 ± 10 in the y direction i.e., in the x direction we would start from 30 to 60 and in the y direction the same way. Assuming that we get the minimum difference at position 45, 45 i.e., the image for comparison has been shifted 5 pixels left and 5 pixels up with respect to the GM image ROI position. This shift in x and y directions is called very critical defect detection. If we were comparing pixels at 10, 10 in the GM image then the corresponding pixels in the actual image for defect detection would be from position 5, 5.

At certain instant the shift in actual image is not exactly in pixels, which means that sub pixel correlation has to be attempted to bring the images in synchronization.

2.3 Granulometry for Defect's Segmentation

Mathematical morphology operations can be used to modify image shapes, reduce noise and detect features of interest. Here we consider only binary images and binary morphology. In this case only two different pixel values are possible, often called "foreground" and "background" levels. It is possible to represent a binary image as a set of (row, column) coordinate locations for all of the foreground points. Most morphology operations involve a *structuring element,* which is another set of (row, column) pairs. The structuring element is typically quite small, and its shape has a direct impact on the results. Two fundamental operations of mathematical morphology are *dilation* and *erosion.* Intuitively, these operations tend to enlarge and reduce the sizes of foreground regions in images.

The morphological operators combine the pixel values in a neighborhood of given shape (square, rectangular or circular) and replace the central pixel of the neighborhood by the resultant shape. The combining function is non-linear, and in most cases it is a rank filter. Consider the N values in the given neighborhood, sort them increasingly and select the K_{th} largest. Three special cases are most often used: K

can be 1 (minimum of the set), N (maximum) or $\frac{N}{2}$ (median). They correspond to the so-called erosion, dilation and median filter operations.

Erosion, dilation, opening, closing and gradient operations performed in this work all use rectangular kernels as shown in Figure 3. It is important to well understand how the kernel size is specified the kernels are kernel half height and kernel half width. The actual kernel width and height are (2 × Kernel Half Width + 1) and (2 × Kernel Half Height + 1). Most of the morphological operations (except median filters) can be applied destructively, i.e. the destination image being the same as the source, or not. The destructive operations are faster.

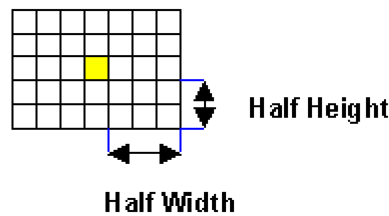


Figure 3: Rectangular kernel

2.3.1 Area sieves

There are many more applications that morphology can be applied to. Morphology has been widely researched for use in image and video processing, but with no application to defect separation. One of the properties not utilized for defect separation so far is that of 'Granules'. A granule is an element that falls through a 'Sieve' in the same way that small stones fall through a sieve [20, 21]. If a set of structuring elements is used with increasing size, $\{B(r): r > 0\}$ with the property that $B(t)$ is $B(r)$ opened for $t \geq r$. This just means that each structuring element is bigger than the previous. Using this, an opening can be defined to use these structuring elements:

$$\alpha_r(E) = X \circ B(r) \quad (2)$$

where, α_r is the area sieve with scale factor r, E is the error image, B is the structuring element, $\circ$ is the open operator and t is the granule target size, if equation (2) satisfies the property that $\alpha_r \alpha_t = \alpha_t \alpha_r = \alpha_t$ where $t \geq r$, then the openings are called a '**Granulometry**'. The morphology techniques have already proven that it is good for filtering. However, some time it is found that the images show a more noticeable degradation than others. This is due to the fact that the filters operate on the entire image having no knowledge of the image statistics. To overcome this a filter that will act on the entire image, but that only changes local minimum and maximum points is needed. The method used to describe such a filter is known as 'Area Morphology'.

It is difficult to see from the above definition, exactly what Granulometry does. The opening used above is good, but a more useful method would be to use and Open-Close, which will remove both dark and light spots. Thus, an Open-Close can be defined for a sieve as:

$$S_r(E,B) = (E \circ B_r) \bullet B_r \tag{3}$$

where, r is the size of sieve and $\bullet$ is the close operator. From equation (3) it is clear that the size will determine the defect data to be removed. For example, if $r = 3$, the corresponding structuring element will remove runs of length 3. However, for a true sieve, this must be done in order. For example, for a target size of 4, the input would first be processed to a size of 1. The result would then be processed to 2 and so on until the target is reached. This is defined as:

$$R_t(E) = S_t(S_{t-1}(...(S_1(E,B_1)...), B_{t-1}), B_t) \tag{4}$$

where, t is the target size of sieve. This can be seen more clearly in Figure 4.

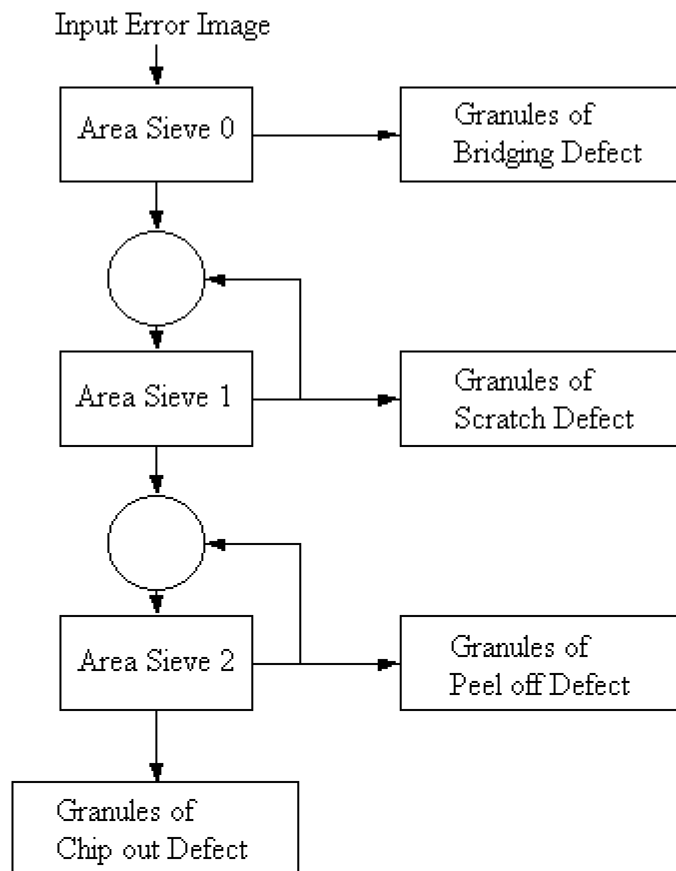


Figure 4: Example of a 2D sieve structure using area morphology

Hence, granules of the classification system are simply the result subtracted from the input error image and it is expressed as:

$$G_t(E) = S_t(E) - S_{t-1}(E) \quad (5)$$

2.3.2 Priority Rules

The defect classification is performed using priority rules such that each defect assigns a label ("chipping," "peel off," "scratch", "bridging," or "clear die") to each pixel in the error image. The classification system described here uses higher-level, domain-dependent knowledge to refine initial classification results. There are several motivations for such an approach. First, the initial classification depends primarily on information from very small image neighborhoods, and identifies defects on a pixel-by-pixel basis. It therefore ignores information such as defect shape, size, and position within the die. Second, the priority rules that we describe here can employ fundamentally different types of rules for different defects, whereas the previous approaches used are restricted to a single topology for all classes that it can identify.

Area morphology can be used to create a 2D sieves. The basic sieve structure is used. Figure 4 shows how a sieve works and how the defect granules fall through the sieve. Assigning a label to each pixel in the error image the labels are “chip out”, “peel off, scratch”, “bridging”, or “clear die” performs defect pattern classification.

Table 1. Priority rules to resolve pairs of overlapping labels. The unshaded labels in the table represent the "winners" for cases that two labels (one shaded label from the left and one shaded label from the bottom) have been assigned to a single pixel.

Diffusion Layer	Diffusion Layer				
Scratch	Metal Scratch	Scratch			
Bridging	Bridging	Bridging	Scratch		
Peel off	Metal Lifting	Diffusion Layer	Peel off	Bridging	
Chip Out	Chip Out	Chip Out	Chip Out	Chip Out	Chip Out
	Clear Die	Diffusion Layer	Scratch	Bridging	Peel off

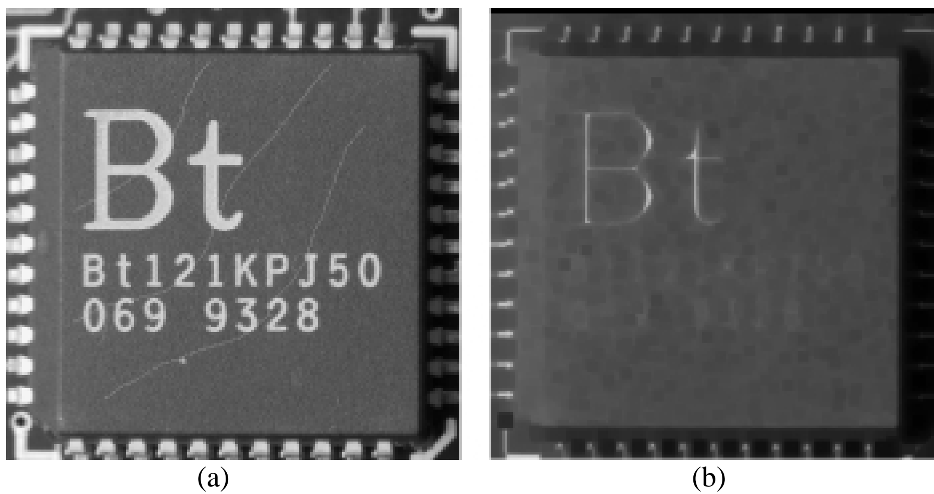
The classification system utilized depends heavily on mathematical morphology; morphological operations are performed to remove spurious regions and refine defect region shapes from the error images. However, some defect regions will have grown, causing some other regions from different layers to overlap. To resolve those overlaps, we have developed priority rules (Table 1) to resolve conflicts when a pixel is assigned different labels by separate processing steps. For example, the region-filling algorithm that is applied to clear die (die without any defect) will create overlaps with diffusion layer (a layer by which electrical isolation of one or more active circuits) regions. Because it is important to

distinguish diffusion layer fault, we give those regions preference over clear die regions. Whenever clear die overlaps another defect type, clear die surrenders its pixel label at that specific point. Because voids and metal scratch can appear inside of chip out, they are given priority over diffusion layer whenever a conflict occurs.

3. Experiments and Results

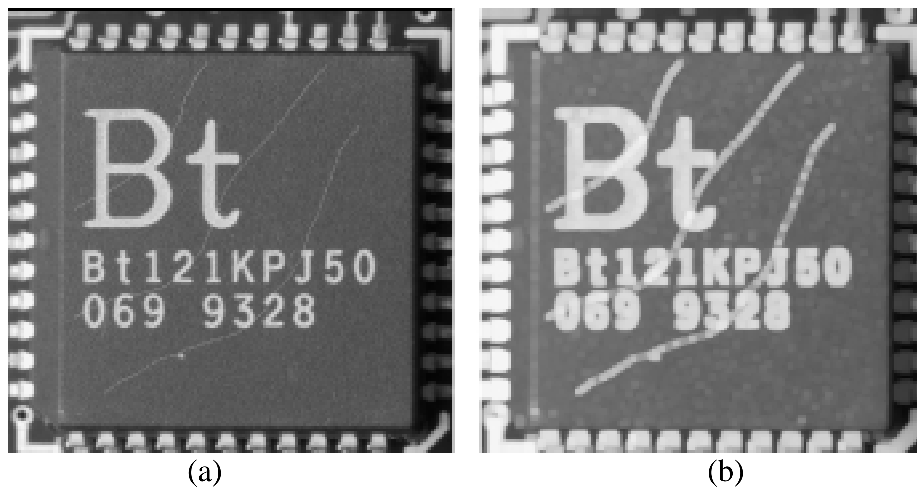
3.3.1 Erosion and dilation

Erosion thins the white objects by removing one layer of pixels along the object edges. When the kernel size gets large, the tiny white objects completely disappear and the black ones get fatter.



(a) (b)

Figure 5: (a) Image before erosion; (b) Image after erosion



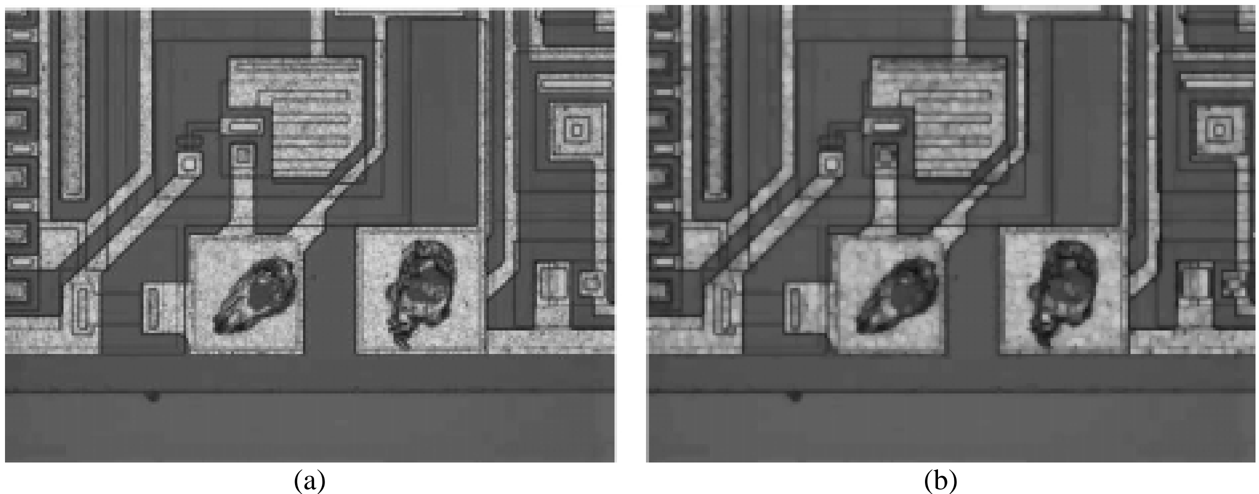
(a) (b)

Figure 6: (a) Image before dilation; (b) Image after dilation

Dilation is the dual of erosion. Dilation thickens the white objects by adding one layer of pixels along the object edges. When the kernel size gets large, the white objects get fatter and the tiny black ones disappear. Figures 5 and 6 show examples how both erosion and dilation work on an image.

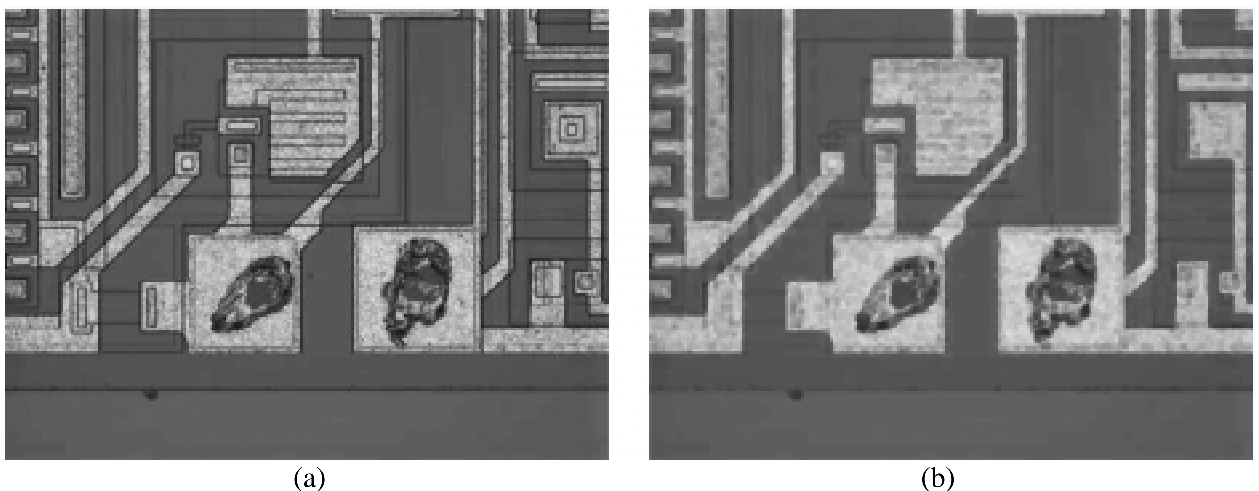
3.3.2 Opening and closing

Opening is erosion followed by dilation. The global effect is to preserve the overall shape of objects, while dropping the white details smaller than the kernel size. This provides an efficient way to remove white dust or other tiny objects.



(a) (b)

Figure 7: (a) Image before open operation; (b) Image after open operation



(a) (b)

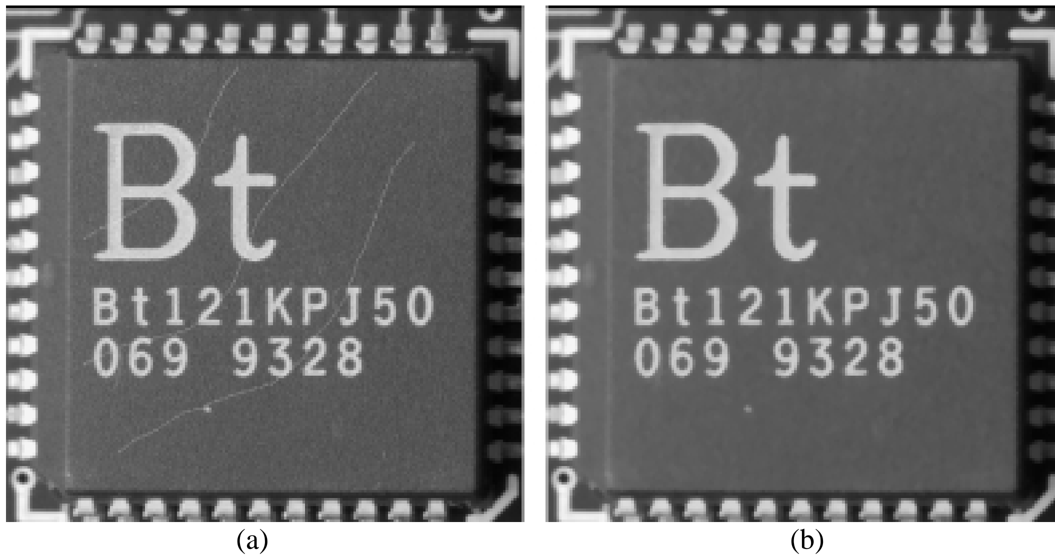
Figure 8: (a) Image before close operation; (b) Image after close operation

Closing is the dual of opening. It is a dilation followed by erosion. The global effect is to preserve the overall shape of objects, while dropping the black details smaller than the kernel size. This provides

an efficient way to remove black dust or other tiny holes. Figures 7 and 8 show example of how both close and open work on an image.

3.3.3 Median filter

The median filter replaces every pixel by the median (central value) of its neighbors in a 3x3 square kernel. This way, outlier pixels are discarded. The net effect is a very effective removal of impulse noise, while edges and image sharpness are well preserved. Figure 9 shows an example how the median filter works on an image.



(a) (b)

Figure 9: (a) Image before median filter operation. (b) median filtered image

3.3.4 Refinement of Chip Out Region Using Priority Rules

In order to show how domain-dependent rules can be developed and applied, we begin with an example concerning chip out region of the die under inspection. In this application, it is reasonable to assume that chip out should lie on the outside boundary of a die. It is therefore possible to state this simple rule: Retain only those chip out regions in the image that lie at the outer boundary of the die. The high-level processing strategy is first to determine which point lie on the outside edge of the die, and then to retain only those chip out regions that overlap one or more boundary points.

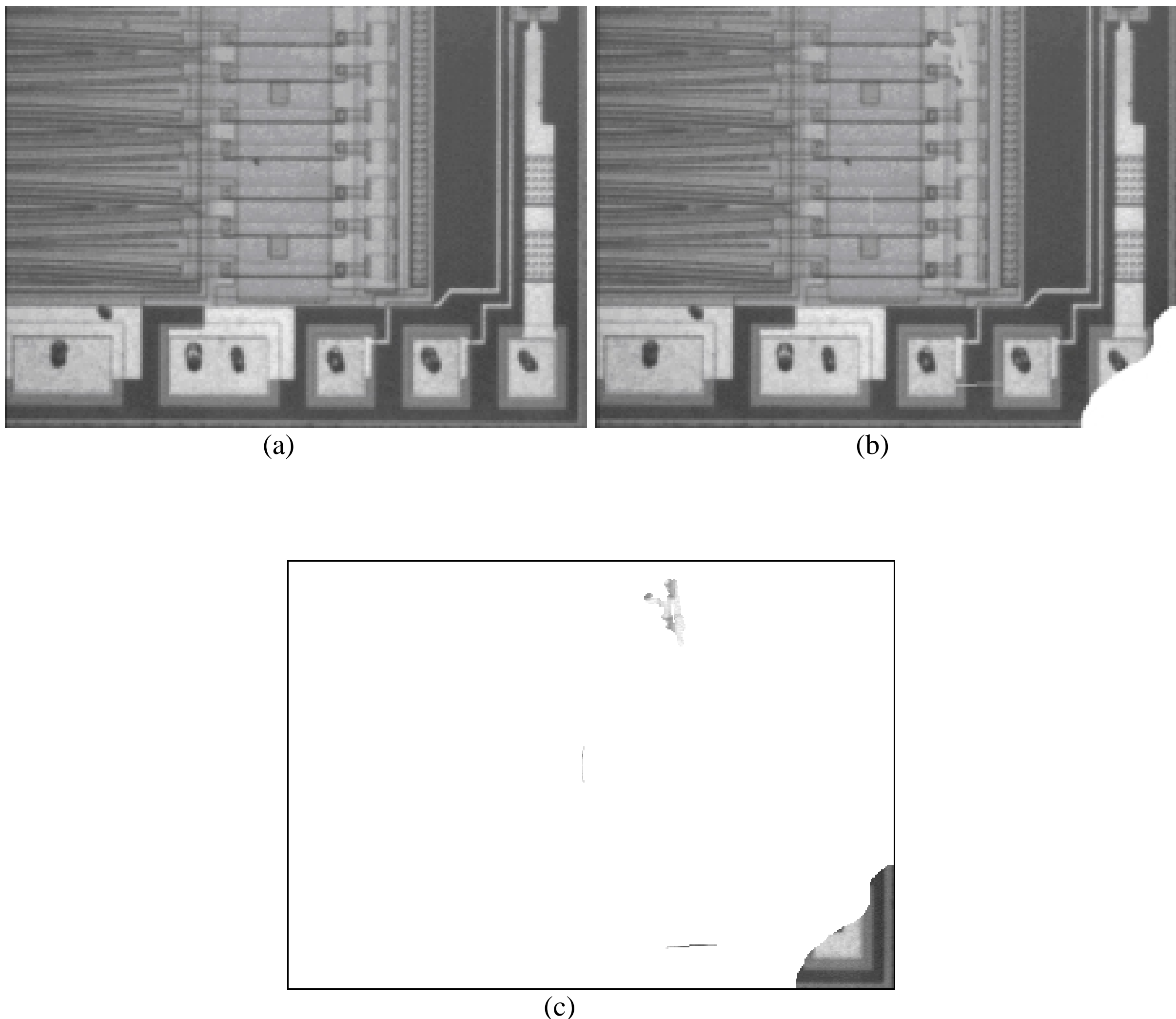
Reference circuit image in Figure 10 (a) is matched to the image under test in Figure 10 (b) and the resultant image is the error image as shown in Figure 10 (c). Figure 10 (c) contains defects such as chip out, bridging and scratch. Now it is therefore required to separate these defects by applying domain-dependant rules. Let us first consider morphological operations that can be used to obtain the boundary. A binary die error image in which dark foreground points represent the die boundary is extracted. This binary error image is filtered (median) to remove the two bridging defect components in the error image using a 3 x 3 structuring element and the resultant image is shown in Figure 11. Further the binary error image is dilated using a 3 x 3 structuring element, and points associated with scratches in the image are

removed and the resultant error image contains only chip out region is shown in Figure 12 from the dilated version.

Algebraically, the procedure can be written as

$$C = (E \oplus B) \setminus E \tag{6}$$

where $\oplus$ is the dilation operator, \ is the intersection operator, E is the binary error image, B is the structuring element, and C is the result. The structuring element can be expressed formally as $B = \{(-1, -1), (-1, 0), (-1, 1), (0, -1), (0, 0), (0, 1), (1, -1), (1, 0), (1, 1)\}$. Dilation by this structuring element causes A to be enlarged by one pixel in every direction. The dilations and intersections continue until no change occurs.



(a) (b)

(c)

Figure 10: (a) reference circuit image, (b) image under test, and (c) error image

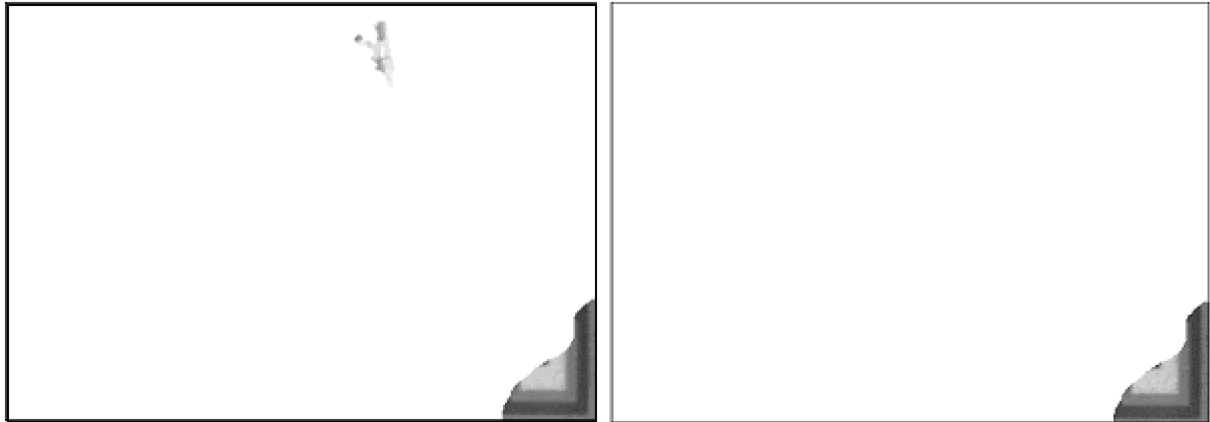


Figure 11: Image containing only scratch and chip out; Figure 12: Image containing only chip out defect

4. Summary

In this paper, we have described a vision system in terms of its hardware modules, as well as the image processing algorithms that it utilizes to scan the images and classify defect from the wafer surface. The algorithm for this defect detection process was developed in VC++ using multithreading. The reason for multithreading is for the fact that defect detection has to be fast and reliable. In addition to being tested in a laboratory environment, a prototype of this system was constructed and deployed to detect and classify the defects after wafer dicing. The system was evaluated under realistic conditions, and it was found that the vision system was able to successfully classify and detect non-uniformities.

The experimental results have proven to be good results statistically for several defect types and for several species of wafer die. We have begun to develop certain priority rules that are tailored to the individual types of defect. Because much of the refinement is related to region shape, the tools of mathematical morphology have been particularly effective in our experiments for the classification of defects. Our current results with chip out, bridging and scratch regions demonstrate impressive visual successes. These results need to be validated quantitatively however, in future work.

References

1. L. Breaux and B. Singh, "Automatic defect classification system for patterned semiconductor wafers," Int'l Symposium on Semiconductor Manufacturing, pp.68-73 (1995).
2. H Miyashita et al., "Particle Measurements in Vacuum Tools by In-Situ Particle Monitor," Journal of Vacuum Science and Technology, A-17, no. 3, pp.1066-1070 (1999).
3. Zhong, Z. W., and K. S. Goh, "Flip Chip on FR-4, Ceramics and Flex," Journal of Electronics Manufacturing, Vol. 10, No. 2, 2000, pp. 89-96.
4. Zhong, Z. W., and K. S. Goh, "Analysis and Experiments of Ball Deformation for Ultra-Fine-Pitch Wire Bonding," Journal of Electronics Manufacturing, Vol. 10, No. 4, 2000, pp. 211-217.

5. P.B. Chou, A.R. Rao, M.C. Sturzenbecker, F.Y. Wu, V.H. Brecher, “Automatic Defect Classification for Semiconductor Manufacturing”, Machine Vision and Applications, Vol. 9 (4), pp. 201-214, 1997.
6. A. K. Jain, R. Duin, and J. Mao, Statistical Pattern Recognition: A Review, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, pp. 4-37, Jan. 2000.
7. Don BE, Brecher V (1995) Recent advances in the automatic inspection of integrated circuits for pattern defects. Mach Vision Appl 8: 5-19.
8. Chin RT (1988) Survey Automated Visual Inspection: 1981 to 1987. Computer Vision Graphics Image Process 41: 346-381.
9. Chen CH, Cheng TH, Wu WT, Driscoll S (1998) Machine Vision Algorithms for Semiconductor Wafer Inspection: A Project with Inspex. Proc of SPIE 3521: 221-228
10. Babian F (1986) Optical defect detection limits in semiconductor wafers and masks. Ph.D. thesis, Stanford University, Stanford, California.
11. Newman TS, AK (1995) A survey of automated visual inspection. Comput Vision Graphics Image Process 61(2): 231-262.
12. Moganti M, Ercal F, Dagli CH, Tsunekawa S (1996) Automatic PCB inspection algorithms: a survey. Comput Vision Image Understanding 63(2): 287-313
13. Klaus Wiltschi, Axel Pinz, Tony Lindeberg, An automatic assessment scheme for steel quality inspection, Machine Vision and Applications (2000) 12: 113-128.
14. Dom BE, Brecher VH, Bonner R, Batcelder JS, Jaffe RS (1988) The P300: a system for automatic patterned wafer inspection. Mach Vision Appl 1(3): 205-221.
15. Khalaj BH, Aghajan HK, Kailath T, Digital image processing techniques for patterned wafer inspection. Proceedings of the SPIE - The International Society for Optical Engineering, vol.1926, pp.508-516, 1993.
16. Paulraj, R. Roy, and T. Kailath. Estimation of signal parameters via rotational invariance techniques - ESPRIT. In *Proc. 19th Asilomar Conf. on Circuits, Systems and Comp.*, pp. 83-89, Asilomar, Pacific Grove, CA, November 1985.
17. P. Xie and S.U. Guan, “A golden- template self-generating method for patterned wafer inspection”, Machine Vision and Applications (2000) 12: 149-156, May 2000.
18. N.G. Shankar, and Z.W. Zhong, Rule-based Inspection of Wafer Surface, The Fourth International Conference on Control and Automation (ICCA’03), pp. 752-755, Montreal, 2003.
19. N.G. Shankar, and Z.W. Zhong, Rule-Based Classification of Defects on Semiconductor Wafers, ICMAT, Symposium L, Singapore 2003.
20. Haralick, R.M., Sternberg, S.R., and Zhuang, X., "Image analysis using mathematical morphology," *IEEE Trans. Pattern Analysis and Machine Intelligence,* PAMI-9, 532-550 (1987).
21. Soille, *P., Morphological Image Analysis,* Springer-Verlag, Berlin, 1998.