1998 International Computer Symposium
Workshop on Image Processing and Character Recognition
December 17-19, 1998, N.C.K.U., Tainan, Taiwan, R.O.C.

AN IMAGE PROCESSING APPROACH FOR WAFER SCRATCH IDENTIFICATION*

Chua-Chin Wang, Chenn-Jung Huang, and Sheng-Hua Chen

Department of Electrical Engineering,
National Sun Yat-Sen University, Kaohsiung, Taiwan R.O.C
E-Mail: ccwang@ee.nsysu.edu.tw

Abstract

During an electrical testing stage, each die on a wafer must be tested in order to determine whether its functionality is the same as designed. If there is a scratch on the wafer, it is possible that some bad dies in the scratch area can not be detected by the tester. To avoid the bad dies being proceeded to final assembly, a testing factory has to assign some workers to check the wafers and mark the bad dies in the scratch or close to scratch area by hand. In this paper, we propose an automatic wafer-scale scratch identifier using a median filter and clustering approach to detect the scratch and mark all the bad dies. The results verify that our algorithm indeed detects the scratches, and it can be used in chip probing stage to automate the testing of wafers.

1. Introduction

During an electrical testing stage, each die on a wafer has to be tested to determine whether its functions work appropriately as designed. This test is performed by actually connecting the exposed pads on each circuit to a "probe" machine that runs testing programs from a computer throughout the CUT(Circuit-under-test). If it performs as expected the die is forwarded for assembly. The bad dies are automatically marked with a colored dot, i.e., "inking", so they can be separated from the good dies after the wafer is cut.

When there is a scratch on a wafer, all the dies in the scratch should also be marked as bad dies and thrown away at next stage. So far this marking work can not be done automatically because not all the dies in the scratch fail in the given electrical test. The testing factory has to arrange some workers to spend much more time and efforts to detect the scratches by eyes and mark the bad dies by hand. In this paper, we attempt to apply some image processing techniques [1], [2] to detect the scratch on a wafer and run programs on tens of wafers. The results show that our algorithm can detect the scratches and it can be used at the testing stage to help the testing process to be fully automated.

2. Scratch detecting algorithm

2.1 Basic Data Structure

The data format used by the prober is defined in a standard format called SECS (Semiconductor Equipment Communication Standard) developed by SEMI (Semiconductor Equipment and Materials Institute). The prober's wafer map [3] includes the formatting information defined by SEMI (lists, single and multi-byte integers, etc) as well as the map and header data. Our program reads in the wafer map file generated by the prober and transforms it into a bound matrix [4] with m by n entries. Here m denotes the total number of columns on the wafer, and n denotes the total number of rows on the wafer. To simplify the processing, only binary values are allowed for each entry in the matrix. We use 0 to represent a good die, and 1 for a bad die.

2.2 Median filter

After reading the wafer map file, we apply the median filter [1], [2] on the matrix to strip off the isolated bad dies on the wafer.

Recall that the median *m* of a set of values is that the values of a half of the elements in the set are less than *m* and those of the other half are greater than *m*. A median filter is defined as the median of all the pixels within a local region of an image. It can be operated on an image to remove long tailed noise such as negative exponential and salt-and-pepper type noise from an image with a minimum blurring of the image. The median filter can easily remove outlier noise from images that contain less than 50% of its pixels as outliers.

In order to perform median filtering in a neighborhood of a pixel, we first sort the values of the pixel and its neighbors, then determine the median, and finally assign this value to the pixel. For example, in a *3* x *3* neighborhood, the median is the *5*th largest value; in a *5* x *5* neighborhood, the *13*th largest value, and so on. Suppose that a *3* x *3* neighborhood has values (*40, 10, 20, 25, 15, 20, 35, 30, 20*), as shown in Fig. 1. These values are sorted as (*10, 15, 20, 20, 20, 25, 30, 35, 40*), which results in a median of *20*.

40	10	20
25	15	20
35	30	20

Fig. 1 3 x 3 neighborhood

In our approach the median filter design can be even simpler because there are only binary values allowed for each entry of the matrix. We can count the total number of *1*'s appearing in the *3* x *3* neighborhood and set the value of the pixel to *1* if the number of *1*'s is greater than or equal to *5*. After the median filter is applied, all the *1*'s left in the matrix will be treated as bad dies in the scratch.

2.3 Clustering

After all the isolated bad dies have been removed, we proceed to the region-based segmentation step [5], which is used to group various regions in an image that have similar features. Clustering techniques [6] proposed in prior pattern-recognition literature have similar objectives, which can be applied for segmentation. The nearest neighbor method [6] is adopted to identify the bad dies of different scratches. The nearest neighbor method is implemented by defining the distance between two scratches to be the smallest distance between two bad dies such that one bad die is in each scratch. If S_i and S_j are scratches, the distance between them is defined as

$$Dist(S_i,S_j) = \min_{a\in S_i, b\in S_j} d(a,b) = 1 \qquad (1)$$

where *d(a,b)* denotes the distance between the bad dies *a* and *b*.

After the nearest neighbor algorithm is applied, the linked lists [7] are constructed to represent the members of different scratches on the wafer. The linked list consists of three fields; they are *x-die*, which denotes the matrix row index of the bad die, *y-die*, the matrix column index, and *next-die*, which points to the next element in the linked list. These data provide testing engineers information to either back trace or find out what might be the causes resulting in the scratches on the wafer.

2.4 Bad Dies Marking

As we mentioned earlier, some bad dies in the scratches might still pass the electrical test. Thus the approach that we propose is to go through all the elements in the linked lists and mark their 8-adjacent neighbors [6] as bad dies even they pass the electrical test. Two bad dies are said to be 8-adjacent if they share either a side or a corner, as shown in Fig. 1. The entry denoted by *15* in Fig. 1 is 8-connected to all of its 8-neighbors.

The last step in our approach is to mark isolated interior dies (a good die surrounded by 8 bad dies) as bad dies. After this is done, an output file is generated for next processing stage.

2.5 Image Processing Algorithm

The pseudo code of the algorithm is sketched as follows:

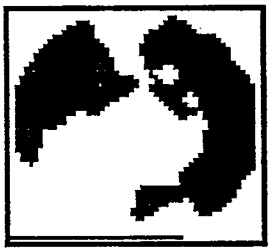
```
Step 1: Read in the wafer map file, and convert it to a binary matrix.
Step 2: Apply median filter on the binary matrix to remove the isolated entries.
Step 3: Perform the nearest neighbor method on the binary matrix and build the linked lists for the scratches.
Step 4: Go through the linked lists and mark all the 8-adjacent neighbors as bad dies.
Step 5: Fill isolated interior dies.
Step 6: Generate the output file for further processing.
```

3. Experimental Results

In our experiment, tens of wafer samples with scratches and without scratches are both collected for testing purpose. These samples are provided by Philips IC Testing Division of Building Elements Industries Ltd. Co., which are actual wafer samples. Fig. 2(a), Fig. 3(a), and Fig. 4(a) show three original sample map files. Note that it is easy to recognize the scratches on these wafers by human eyes, but testers without efficient software support can not identify these scratches. After median filter is applied, we obtain the binary matrices as shown in Fig. 2(b), Fig. 3(b), and Fig. 4(b). Finally we perform nearest neighbor method, mark the 8-adjacent neighbors, and fill the isolated interior dies as shown in Fig. 2(c), Fig. 3(c), and Fig. 4(c). Not only the bad dies and the doubtful dies are recognized, but also the dies located in scratch areas are labeled. In addition, the linked lists of scratches help R&D and testing engineers to find out why scratches occur.

(a) Original wafer map file

(b) Output of median filter



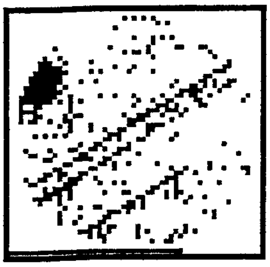
(c) The detected scratches

Fig. 2 Demonstration of scratch detection (I)

Fig. 5 shows a typical result when we run the same program on some wafer samples with no scratches on it. Notice that a cluster is treated as noise if the size of the cluster is too small, as shown in Fig. 5(b). Therefore our approach will not misinterpret the isolated bad dies as scratches.

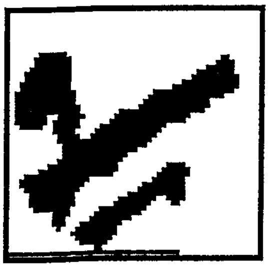
4. Conclusion

In this paper we have proposed an approach to identify the scratches on the wafer. The wafer map file is read in, and converted into a binary image. Next the median filter is applied to the binary image to remove the noise (isolated bad dies). Then nearest neighbor method is performed to build the linked lists for the scratches. Finally the bad dies of the scratches are marked to avoid unnecessary waste on the human cost and other resources for further processing. The test results show that our algorithm indeed detects all the scratches. The proposed approach has been successfully tested by IC Testing Division of Philips Electronic Building Elements Industries Ltd.



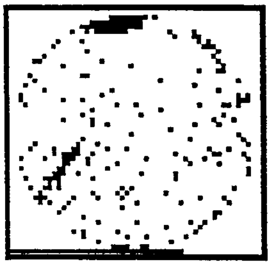
(a) Original wafer map file

(b) Output of median

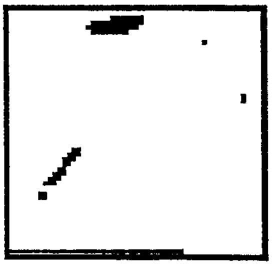


(c) The detected scratches

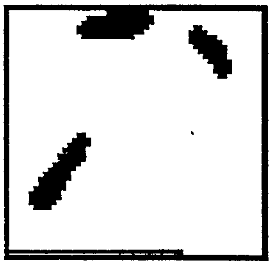
Fig. 3 Demonstration of scratch detection (II)



(a) Original wafer map file

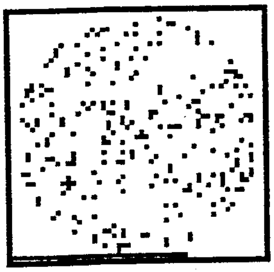


(b) Output of median filter

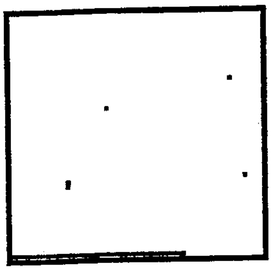


(c) The detected scratches

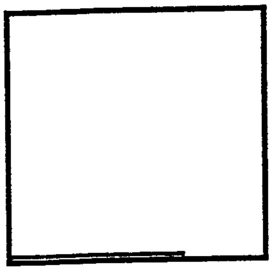
Fig. 4 Demonstration of scratch detection (III)



(a) Original wafer map file



(b) Output of median filter



(c) The detected scratches

Fig. 5 Demonstration of scratch detection (IV)

5. References

[1] JAHNE, B.: "Digital Image Processing: Concepts, Algorithms, and Scientific Applications," Springer-Verlag, 1997

[2] BAXES, G., "Digital Image Processing : Principles and Applications," Wiley, 1994

[3] BAUMGARTNER, T., "SECS II Wafer Map Format." Electroglas, 1996

[4] DOUGHERTY, E., and GIARDINA, C., "Matrix Structured Image Processing", Prentice-Hall, 1987

[5] PAL, N., and PAL, S.,"A Review On Image Segmentation Techniques," Pattern Recognition, vol.26, no.9, pp.1277-1294, 1993

[6] EARL, G., JOHNSONBAUGH, R., and JOST, S.: "Pattern Recognition and Image Analysis", Prentice Hall, 1996

[7] HOROWITZ, E., SAHNI S., and ANDERSON-FREED, S., "Fundamentals of Data Structures in C++', Computer Science Press, 1995