

Field-test results of an image retrieval system for semiconductor yield learning

T.P. Karnowski^a, K.W. Tobin Jr.^a, L.F. Arrowood^a, R.K. Ferrell^a, J.S. Goddard, Jr.^a, and F. Lakhani^b

^aOak Ridge National Laboratory, Oak Ridge TN 37831

^bInternational SEMATECH, Austin TX 78741

ABSTRACT

Images of semiconductor defects are maintained in semiconductor yield management systems to diagnose problems that arise during the manufacturing process. A semiconductor -specific content-based image retrieval system was developed by Oak Ridge National Laboratory (ORNL) under the auspices of International SEMATECH (ISMT) during 1998-1999. The system uses commercial databases to store image information and uses a customized indexing technology to rapidly retrieve similar images. Additional defect information (position, wafer ID, lot, etc) has now been incorporated into the system through the use of additional database tables. During Fall 2000, the system was deployed in two ISMT member company fabs to demonstrate the utility of this approach in managing large databases of images and to show causal relationships between image appearance and wafer information such as processing layer, wafer lot, analysis dates, etc. This paper summarizes the results of these field tests and shows the utility of this approach through data analysis conducted on approximately one month of historical defect data.

Keywords: content-based image retrieval, semiconductor, yield enhancement, defect detection, automated image retrieval

1. INTRODUCTION

Images of semiconductor defects are maintained in semiconductor yield management systems to help diagnose problems that arise during the manufacturing process. Engineers frequently turn to these image repositories to examine historical data for trends. In concert with this effort, these images are organized into databases and are used, for example, to help train automatic defect classification systems. Anticipated advancements in semiconductor manufacturing will require faster manufacturing-process analysis to maintain economic feasibility despite the growth in circuit complexity and the amount of data acquired [1]. Since many facilities collect on the order of thousands of images each week, with some state-of-the-art facilities ranging up to 25 000, image management, usage, and cataloging is becoming a source of concern for data management systems [2].

A semiconductor-specific content-based image retrieval (SS-CBIR) system was developed by ORNL under the auspices of International SEMATECH during 1998-1999. The system uses commercial databases to store image information and uses a customized indexing technology to rapidly retrieve similar images. The system leverages advances in the field of content-based image retrieval (CBIR). CBIR is a technology that is being developed to address the needs of a wide variety of fields, including remote sensing, art galleries, architectural and engineering design, geographic information systems, weather forecasting, medical diagnostics, and law enforcement [3]. CBIR techniques are used to index and retrieve images from databases based on their pictorial content [4], typically defined by a set of features extracted from an image that describe the color [5, 6], texture [7, 8] and/or shape [9, 10, 11] of the entire image or of specific objects in the image. This feature description is used to index a database through various means such as distance-based techniques, rule-based decision making, and fuzzy inferencing [12, 13, 14]. The applications of CBIR to semiconductor manufacturing are discussed in [15, 16].

Additional defect information (position, wafer ID, wafer lot number, etc) has been incorporated into the system described in [14,15] through the use of foreign keys and additional database tables. During Fall 2000, the system was deployed in two ISMT member company fabs to demonstrate the utility of this approach in managing large databases of images and to show causal relationships between image appearance and wafer information such as layer, lot, dates, etc. This paper summarizes the results of these field tests and shows the utility of this approach through data analysis conducted on approximately one month of historical defect data.

This paper describes the overall technical approach, including the addition of wafer-related defect data intended to extend the context of the system beyond just image management. Our experimental section discusses the data acquisition and presents

results for the system, including overall database statistics and experimental methods and results. We conclude with a discussion of the effectiveness of the system and future directions in SS-CBIR development work.

2. ARCHITECTURE AND IMPLEMENTATION

Since past papers [15] have discussed the system architecture in detail, we will discuss the additional tables for the defect data and the implementation software for the field test. Although the SS-CBIR field test software was not designed to be a complete defect management system, it was necessary to include some defect management system (DMS)-type functionality to reach our project goals. To this end, we envisioned the submission of images to our system as the results of a defect detection instrument inspection. Table 1 shows the information submitted with each image from the manufacturer's DMS.

Value	Description
PATH	Name of image file
DEFECT INDEX	Unique defect ID value
X Coordinate	Defect X coordinate on wafer die
Y Coordinate	Defect Y coordinate on wafer die
X Die	Die coordinate in the X direction
Y Die	Die coordinate in the Y direction
X Size	Size of the defect bounding box in the X direction
Y Size	Size of the defect bounding box in the Y direction
Area	Approximate area of the defect measured by bounding box
"D" Size	Alternate defect size measurement; largest of X size, Y size, and square root of area
In Cluster	Boolean flag: 1 if defect is a member of a cluster
Height	Measure of the height of the defect, if applicable
Cluster ID	ID number of cluster where defect has membership.
ID On Wafer	ID number of defect on wafer itself (index number)
Date / Time	Date and time associated with inspection of wafer - i.e., 05271999 162840 for May 27 th , 1999, at 16:28.40 (24-hour clock)
Tool	Tool performing inspection
Layer Step ID	The layer, or step ID, associated with the inspection
Recipe	The recipe for the inspection
X Origin (Die)	The X origin of the coordinate system on the wafer (to relate the defect coordinates to actual physical values)
Y Origin (Die)	Y origin of the coordinate system
Wafer Orientation	Orientation of the wafer
Wafer Slot	Boat slot where wafer is located
Lot ID	ID value for the wafer lot
Wafer ID	ID value for the wafer
Device	The device type for the wafer lot
X Pitch	The distance between die streets in the X direction
Y Pitch	The distance between the die streets in the Y direction
SEM Class	The classification as assigned by ADC from an SEM inspection instrument, if applicable, of the defect image
Optical Class	The defect classification as assigned by ADC from an optical inspection instrument, if applicable
Engineer Class	The engineer-assigned defect classification, if applicable
Cluster Class	The classification, if applicable, of the cluster where the defect has membership
Modality	The modality of the image - 0 for optical, 1 for SEM

Table 1 Defect information submitted with each image for the SS-CBIR field test

During our design process, we conceived an entity relationship diagram for our database design. The entities include the Image, including the file name associated with the image along with the feature values that describe its content; the Defect, including the classification of the defect, its location on the wafer and die, etc; the Inspection, data describing a single act of taking one wafer and running it through an inspection on a defect detection instrument; the Wafer, an entity containing a set of die and possibly one or more defects; and associated tables of defect classifications and inspection tool types. The tables were embodied in a software object coded in the SS-CBIR Dynamic Link Library (DLL).

In addition to the core SS-CBIR DLL, the SS-CBIR system included an ORACLE database, a set of interface DLLs and executables, and graphical user interfaces. For our testing purposes, a batch process developed by the fab personnel was run to generate data in format suitable for inclusion into the SS-CBIR database. A Windows NT service executable periodically checked for output from the batch process. When new output was detected, the service added the image and associated defect data to the SS-CBIR database. Each image underwent a segmentation to separate the defect from the background on the image. Since reference images were not available for the images, and previously generated image masks were not available, our implementation was forced to use a technique called "statistical masking" to separate defects from their background. Simply stated, the statistical mask was created by averaging all masks from a database of over 10,000 images acquired from ISMT member companies during previous work. The statistical average of these masks proved to be basically an ellipse of a certain size centered in the image (figure 1). This mask is clearly a gross approximation of reality. This approach prevented our system from using several vital image description parameters, including defect shape and defect image area. In addition, this mask approach caused inaccuracy in other defect descriptors (texture and color) by possibly incorrectly identifying the area where these values should be measured. Of course, in some cases useful information could be measured about the defect from the statistical mask. A random sampling of 1024 images from each site revealed that approximately 20% of the images had defects that were fairly centered and covered by the statistical mask. Some suggested remedies to this situation are discussed in the conclusion.

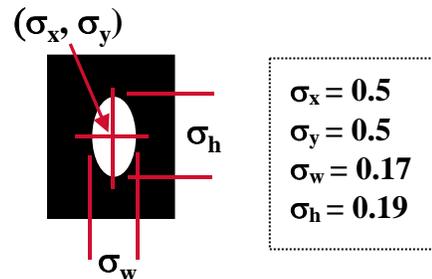
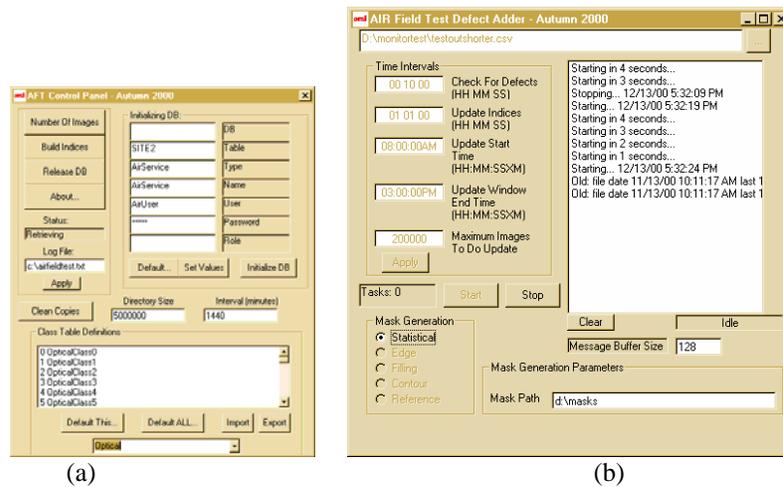


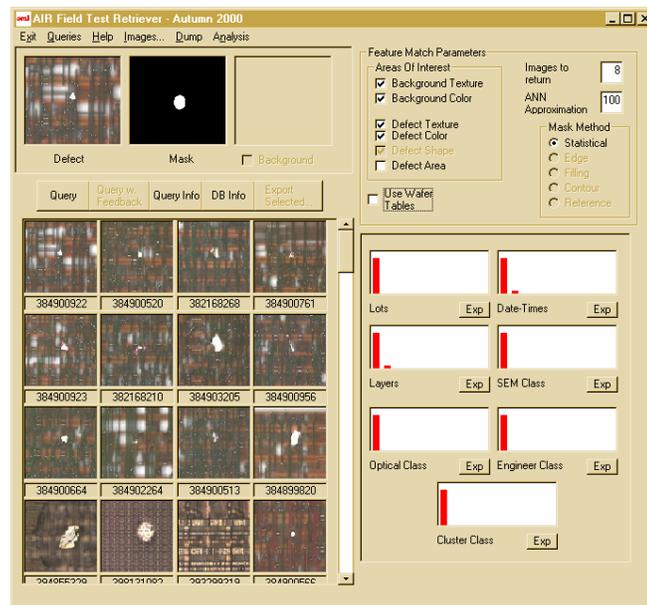
Figure 1 Statistical mask. Values derived from experimental database of over 10,000 images and given as fractions of image width and height.

Screen shots of the user interfaces are shown below. These include an interface showing the overall state of the system (the "control panel"), the status of the submission of images and defects to the database (the "modifier"), and a GUI for retrieving images based on several different criteria including image content (the "retriever"). Very little user interaction is required for the control panel and modifier GUIs. The retriever, on the other hand, is the main mechanism for the user to test and experiment with the SS-CBIR system. Images can be imported as query images using a cut-and-paste operation or file open browser dialog boxes. Once an image is imported into the system, a mask is generated for the defect; several options were delimited, although our experiments at present were limited to the statistical mask. Queries are performed by simply selecting the image areas of interest (defect texture, defect color, background color, etc), and optionally a set of layers or lots which limits the query to images with these characteristics. Returned images are displayed in ranked order in a gallery. Clicking each returned image shows its lot, layer, file name, classifications, etc. The returned gallery can be exported to an HTML file. In addition, paretos (histograms with bins ordered by the highest response) are presented for the returned results. These paretos can be exported to comma-separated value files for use with other analysis tools.



(a)

(b)



(c)

Figure 2 Screen shots of the SS-CBIR GUI components for the field test. **a** - control panel; **b** - adder; **c** - retriever

3. EXPERIMENTAL RESULTS

The SS-CBIR field test software was installed on PCs at two member company sites and configured to run in batch mode. Once a day a batch process exported the results of the previous day's inspections from the fab defect management system (DMS) to a comma-separated value (CSV) file. This file contained essentially one line per inspection image. Each image corresponded to one defect; multiple images of the same defect were possible, however, and do occur in the data less than 5% of the time. The CSV line contained the image file name, wafer number, wafer lot, layer, etc as required by the database tables discussed above. Periodically the SS-CBIR software checked for the existence of this file. When the file was updated, each image was loaded into the database along with its associated data. Table 2 shows the database statistics for each site after approximately one effective month of data collection.

It is rather difficult to quantify the performance of the field test system, especially under the limited time frame of the testing, but we can separate our results into three areas: system performance, retrieval accuracy, and anecdotal support.

Value	SITE 1	SITE 2
Number of DEFECTS	59593	76653
Number of WAFERS	3856	3336
Number of LOTS	1375	1021
Number of STEP / LAYERS	99	164
Number of IMAGES	62594	78953
Oldest DATE	10-7-2000	9-14-2000
Latest DATE	11-6-2000	11-1-2000

Table 2 Database statistics for site 1 and 2

System Performance

We report results pertaining to system timing and robustness in this section. There are two times of interest in general to the fab user. First, the time to add images to the database is important because the SS-CBIR system should be basically invisible to the underlying defect detection and inspection activity.

Second, retrieval time is important because of usability issues and engineering response time. Histograms showing the distribution of times to add images for site 1 and 2 are shown in figure 3. These times were measured on a common machine using data collected and returned to ORNL. The machine in use was a 750 MHz Pentium III PC. Both histograms show a distinct bimodal appearance. This is due to basically both sites having two main types of images, optical and SEM, with different sizes. (all images were collected in jpg format).

Value	Site 1	Site 2
Addition Mean	0.834 seconds	0.476 seconds
Addition Median	0.765 seconds	0.328 seconds
Addition Maximum	6.0 seconds	3.813 seconds
Addition Minimum	0.312 seconds	0.016 seconds
Daily Rate	103598 images	181590 images
Retrieval Time (128 images)	7.5 seconds	7.25 seconds
Retrieval Time per database image	0.12 ms	0.09 ms

The median, mean, maximum and minimum times to add the images to the database are recorded in table 3, along with image retrieval time. The image retrieval time was measured by requesting 128 returned images and measuring the system response for each database. The time to load images from a network and display them is not included in this total. Both these sets of times show a very acceptable rate of performance, allowing an overall daily input of well over 100,000 images a day. The main difference between the timing for the sites is the image size; most images from site 2 were jpg images sized 320 x 240, while site 1 images were 640 x 480 or 480 x 480.

System robustness is another critical factor for fabrication facilities. Our testing was all done in a non-critical mode on our own machines. The sole impact on the fab infrastructure was the required daily batch dump of the defect data and network bandwidth to allow reading images. The systems ran effectively at both sites until testing was completed.

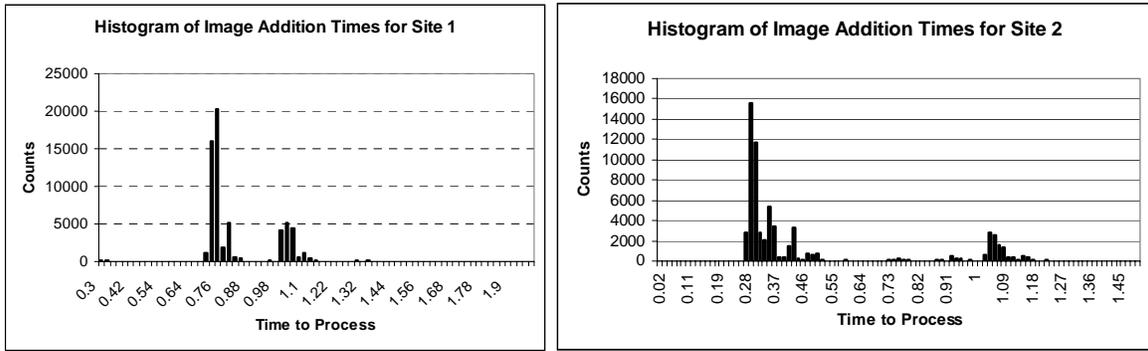


Figure 3 Histogram of image addition times for sites 1 and 2. Bimodal distributions are due to two fundamental imaging modalities (SEM and optical) at each site.

Retrieval Accuracy and Effectiveness

The retrieval accuracy for a large database of images is difficult to quantify. One approach taken in the past [15] involved assigning a grade to each retrieval result to determine overall system effectiveness. While this approach has value, we elected not to pursue such a subjective measure for this work. Instead, we will show illustrative examples of the types of images in the database, some effective retrieval results, then quantify our system based on how our retrievals match query images based on additional, non-visual information.

Image Examples

Figure 4 shows random image retrievals from site 1 and site 2. We see here that site 1 contains a relatively large number of SEM images, while site 2 contains relatively more optical images. Both have significant numbers of both types of imagery.

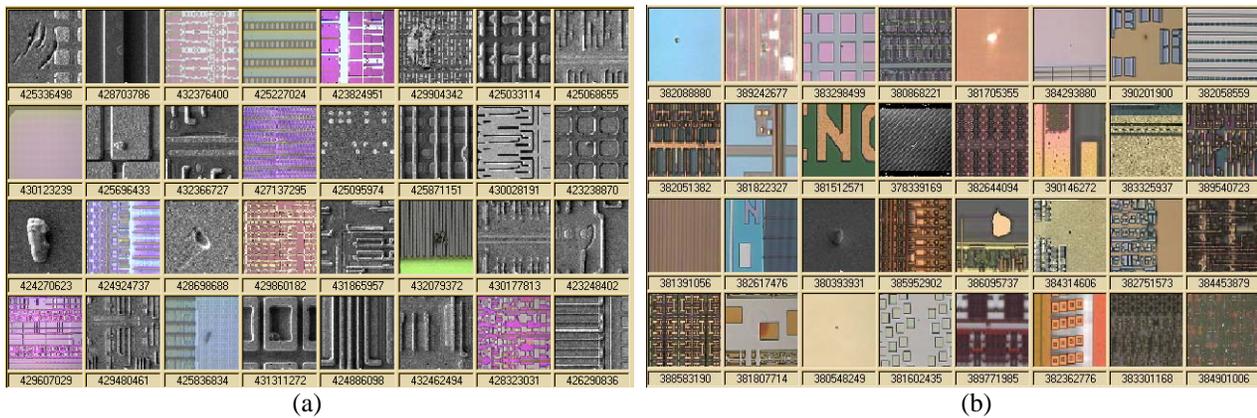
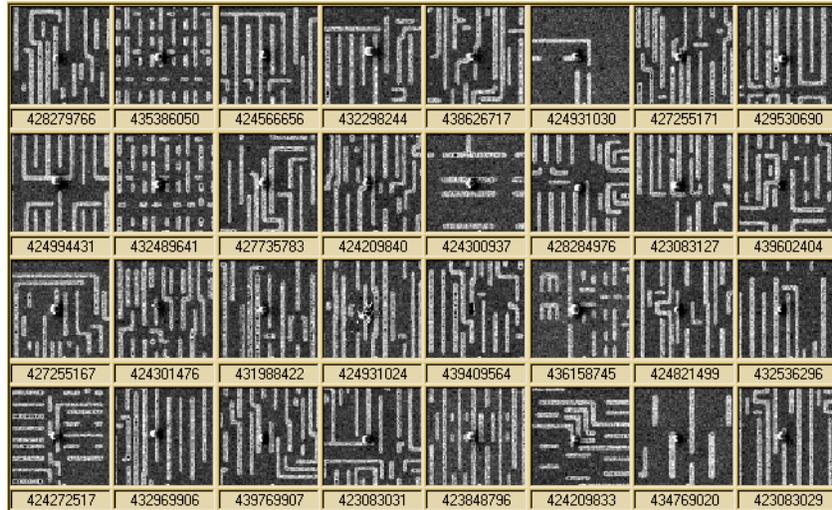
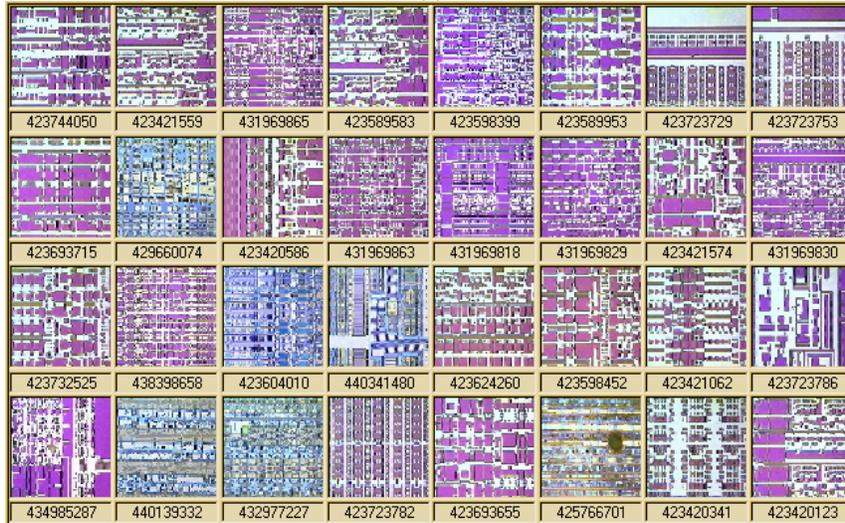


Figure 4 Examples of random images from (a) site 1 and (b) site2

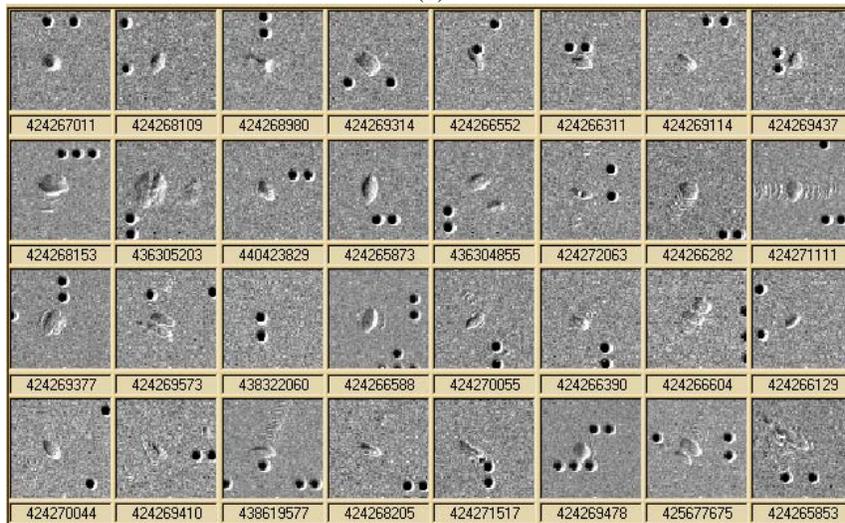
In figures 5 and 6 we show some examples of good queries from both image databases. In all images, the query image is the first image visible, in the upper left of the image. The first query (5a) shows a SEM image of a complex background of metal circuit lines with a small round particle well-centered in the field of view. The returned imagery match this query image very well. The next image is an optical microscopy image. The defect is difficult to detect in most of these images. This shows how the statistical mask can possibly confuse the system because the "defect" area masked out has a very definite texture and color, very similar to the background if not indistinguishable. Image 5c shows a nice SEM image of a pit or scratch; again, it is well centered and a fairly good match for the statistical mask. The results of site 2 all show optical images. Note that the query image 6a has found multiple images of the same defect for some of the results.



(a)

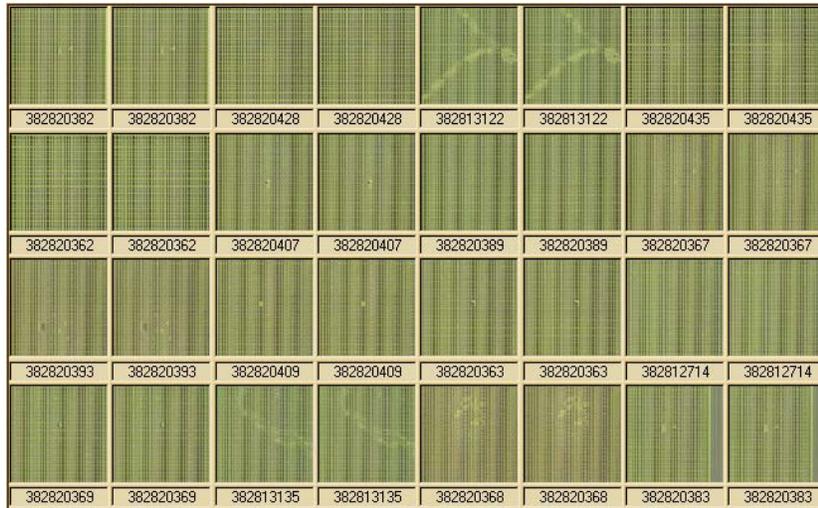


(b)

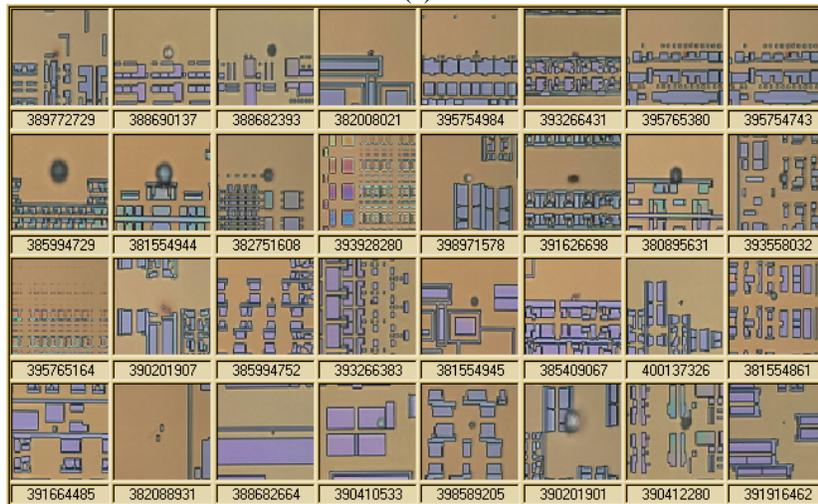


(c)

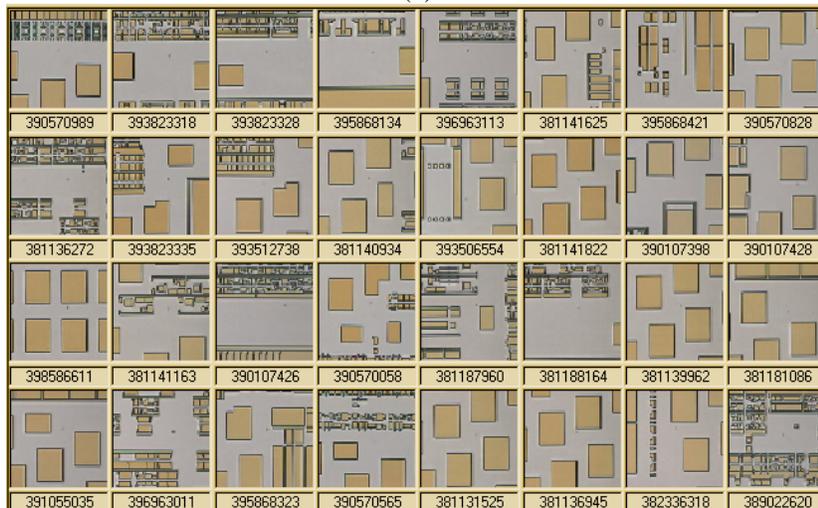
Figure 5 Examples of visually-good queries from site 1



(a)



(b)



(c)

Figure 6 Examples of visually-good queries from Site 2

Non-Visual Data Analysis

We have modeled the retrieval system as a sort of nearest-neighbor classifier for the step/layer, lot, and optical classifications (SEM classifications had too many unclassified results to make a similar comparison meaningful). The experiments were performed as follows. For each site, we sampled 1024 images and submitted them as query images returning 64 results. We then counted how many times the most common occurrence in the results matched the query image. For example, we determined the layer/step with the most common occurrence in the first 4, 8, 16, 32, and 64 returned images. If the most common occurrence matched our query image, the query was assigned a value of 1.0 (for success). Situations where no one value had a majority (ties) were assigned a value of 0.5 for unknown. We would like to note the following:

- The system did not use a very accurate defect - background segmentation mask; while the statistical mask is "better than nothing", its drawbacks are obvious.
- Queries which seek to find similar step/layers, optical classifications, etc. are very easy to implement in database query language, and basically get 100% performance 100% of the time. Our queries for these experiments are based solely on image content and have value based on how they can match images visually, which can supersede lots, layers, and other additional information traditionally used to source yield problems. Our goal is to show that visually similar defects and images have a definite relationship to non-image information, not to design a step/layer classifier.
- Due to the large number of images, the queries were conducted by a random sampling of 1024 images from both databases. In some cases, a "correct" retrieval was not possible because there were not many examples from that particular lot or layer in the database.
- For the optical classifications, we threw out results that contained "not classified" results.

The following charts show the results of this pseudo-classification test. Each chart contains weighted and unweighted results for sites 1 and 2, with classifiers using the first 4, 8, 16, 32, and 64 results. Unweighted results are computed by finding the number of correct classifications for a given layer/step, lot, or optical ADC class, then averaging these. This number considers all layer/steps, lots, and optical classes equally and does not depend on the number of occurrences of each class. Weighted results are computed by determining the number of correct answers and adding these, then dividing by the total number of queries. The performance drops as more neighbors are considered because most queries did not return a majority of the query class. Our analysis does not attempt to explain this phenomena, but one likely reason for this result is the variety of image content from a given layer or lot. (See figure 8 in the "Anecdotal Support" section for an example of this.) An additional obvious explanation is that images are ordered by their similarity to the query image, and the more dissimilar an image appears, the less likely it is that it has common non-image characteristics. The results average in the 60% at 4 nearest

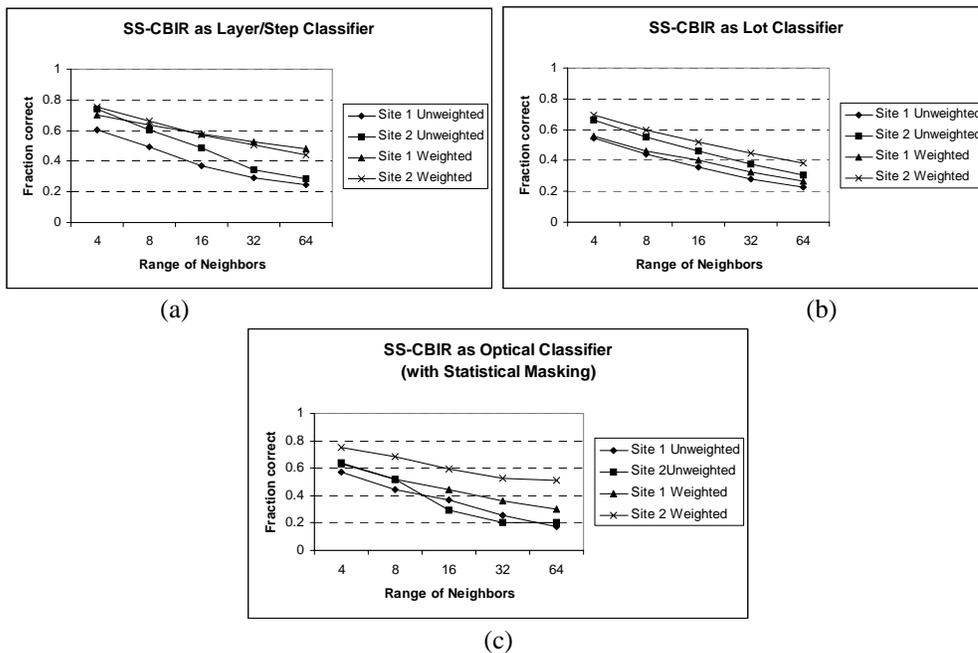


Figure 7 Results from SS-CBIR system as a classifier. (a) Layer/Step; (b) Lot; (c) Optical ADC

neighbors, then drop approximately 8% per classifier group to around 30-20% for the last group. The weighted results perform slightly better.

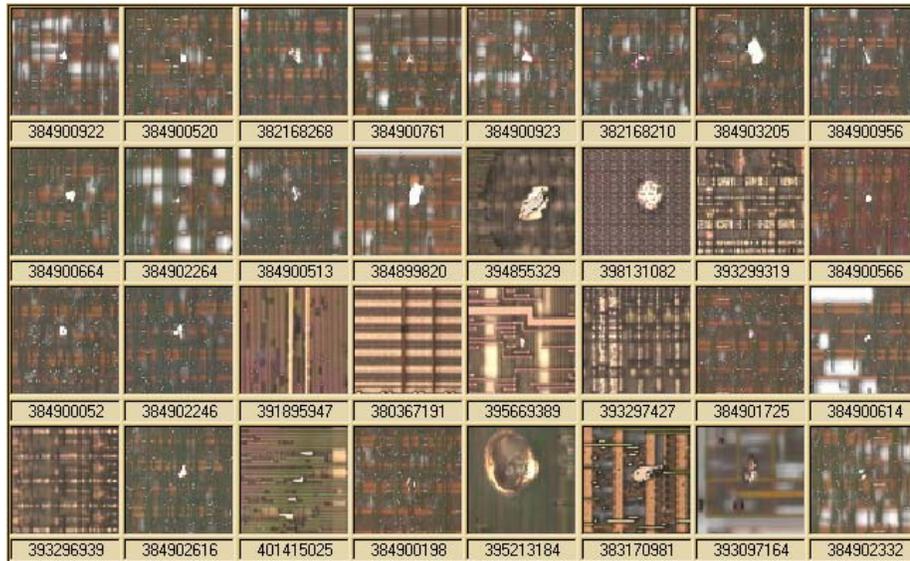
Anecdotal Support

Our feedback from the member company participants was encouraging. The main point of criticism was the lack of a better masking technique. Besides that, the users felt the system would be extremely valuable for multiple uses when integrated into a more fab-ready software package. Performance was perceived in a very positive manner with respect to speed.

In addition to simple image searching, technologists had several creative ideas for the use of this system in an integrated package. One application mentioned was using the system to find libraries and examples of defect images. An image could be submitted for query. When the image results were received, filtering them by layer or lot could be performed, and then the



(a)



(b)

Figure 8 (a) Randomly selected images from layer "X" (b) Result of query using desired image (upper left), from layer "X"

resulting gallery exported for use as an example page. Figure 8 shows such a case. The query results are shown in (b). Of the pictured 32 images, 27 are from the layer of interest; the remaining are from 9 other layers. Part (a) shows a random sampling of images selected solely on the basis of their layer. These random samples are from an overall set of 1692 images from layer X. This illustrates, albeit anecdotally, how SS-CBIR is an effective tool to locate similar-appearing images across layers as well as within a single layer due to its sensitivity to image content.

4. CONCLUSIONS

The results of the field test experiments showed the feasibility for image retrieval in the semiconductor fabrication environment for use with yield engineering practices. The system as designed was robust and operated in an expedient manner. The results of SS-CBIR queries are informative visually. The results of the "SS-CBIR as classifier" experiment are encouraging and show a definite relationship between image visual content and non-image (defect and wafer) data.

One of the largest issues in the field test experiments was the absence of definitive defect masks. This problem prevented fab engineers from effectively testing the system for usefulness. There is a potential solution to this problem. Masks are generally created by defect detection instruments. For background, defect inspection generally consists of two steps. First, the wafer is scanned to determine defect locations. Then a "redetection" phase is applied where detection hardware is driven to the previously determined site and a higher-resolution image of the defect is taken. This image can be submitted to an image archiving system. If the instrument is equipped with automatic defect classification, a reference image is acquired from a neighboring die and digitally subtracted from the defect image to separate the defect from the background. The resulting mask can also be saved along with the defect image. Using this information can be extremely useful for CBIR systems, as it effectively adds much information to the image. However, generally this mask image is not available. An instrument manufacturer may be able to access the mask or separate defect features generated during the course of ADC, but not sharing these masks leads to closed systems that are not in the best interests of the semiconductor manufacturing community. An additional complication is the existence of multiple files for each defect image. There are several techniques for circumventing these problems. These include free sharing of masks, storing masks in efficient manners (i.e., run-length encoding) in image file headers, encrypting mask information throughout the image itself using digital watermarking techniques, and probably the "holy grail", determining a method for robust non-referential defect detection.

A random sampling of 1024 images from each site revealed that approximately 20% of the images had defects that were fairly centered and covered by the statistical mask. Of course, the statistical mask does not allow us to measure anything about the defect's shape and area within the field of view. Furthermore, the statistical mask covering a defect can bias the measurements of defect color and texture. However, it is interesting to see that cases where the statistical mask coincided with the actual defect (for example, figure 5a) show good agreement both visually and with respect to layers and classifications. This type of result renders much enthusiasm for the success of SS-CBIR when the defect mask issue is resolved.

Despite this problem, both sites agreed that the SS-CBIR software shows tremendous potential for usefulness in the yield management environment. Applications for the technology are abundant and become quite clear when deployed in a semiconductor manufacturing facility.

Future experiments and work in this area include attempting to use SS-CBIR as an image manager, incorporating wafer process information, and using SS-CBIR as a tool for automatic defect classification library building. The image management idea stems from the concept that images with similar visual appearance are useful to a point, but become redundant at some juncture. Using SS-CBIR to detect these redundant images can deal with the glut of image data by giving confidence to purging techniques. Incorporating wafer processing information will give an extra dimension to the issue of linking visually similar defects to non-defect data and may point the way for further uses in yield management and preventive maintenance. Finally, the largest problem of ADC - training - can be circumvented if the SS-CBIR system is employed to seek commonality between images and examples of known classes.

-
1. Semiconductor Industry Association International Technology Roadmap for Semiconductors, 1999.
 2. Tobin, K.W., Karnowski, T.P., and Lakhani, F. "Managing Defect Image Databases for Semiconductor Yield Monitoring and Control", Global Semiconductor, Sterling Publications, London, England, February 2000.

-
3. V.N. Gudivada and V.V. Raghavan, "Content-Based Image Retrieval Systems", *IEEE Computer Magazine*, 0018-9162, September 1995, p. 18.
 4. M. De Marsicoi, L. Cinque, and S. Levialdi, "Indexing Documents by their Content: A Survey of Current Techniques", *Imaging and Vision Computing*, Vol. 15, 1997, p. 119.
 5. Y. Gong, C.H. Chuan, and G. Xiaoyi, "Image Indexing and Retrieval Based on Color Histograms", *Multimedia Tools and Applications*, Vol. 2, 1996, p. 133.
 6. V.E. Ogle and M. Stonebraker, "Chabot: Retrieval from a Relational Database of Images", *IEEE Computer Magazine*, 0018-9162, September 1995, p. 40.
 7. B.S. Manjunath and W.Y. Ma, "Texture Features for Browsing and Retrieval of Image Data", *IEEE Transactions of Pattern Analysis and Machine Intelligence*, Vol. 18, No. 8, August 1996.
 8. J. You, H. Shen, and H.A. Cohen, "An Efficient Parallel Texture Classification for Image Retrieval", *Journal of Visual Languages and Computing*, Vol. 8, 1997, p. 359.
 9. B.M. Mehtre, M.S. Kankanhalli, and W.F. Lee, "Shape Measures for Content Based Image Retrieval: A Comparison", *Information Processing and Management*, Vol. 33, No. 3, 1997, p. 319.
 10. J.E. Gary and R. Mehrotra, "Similar Shape Retrieval Using a Structural Feature Index", *Information Systems*, Vol. 18, No. 7, 1997, p. 525.
 11. R. Mehrotra and J.E. Gary, "Similar-Shape Retrieval in Shape Data Management", *IEEE Computer Magazine*, 0018-9162, September 1995, p. 57.
 12. C. Hsu, W.W. Chu, and R. K. Taira, "A Knowledge-Based Approach for Retrieving Images by Content", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 8, No. 4, August 1996.
 13. P.W. Huang and Y.R. Jean, "Design of Large Intelligent Image Database Systems", *International journal of Intelligent Systems*, Vol. 11, 1996, p. 347.
 14. B. Tao and B. Dickinson, "Image Retrieval and Pattern Recognition", *SPIE*, Vol. 2916, 1996, p. 130.
 15. Karnowski, T.P., Tobin, K.W., Ferrell, R.K., and Lakhani, F., "Content Based Image Retrieval for Semiconductor Manufacturing", *SPIE's 12th International Symposium on Electronic Imaging: Science and Technology*, San Jose Convention Center, January 2000.
 16. Tobin, K.W., Karnowski, T.P., Lakhani, F., "The Use of Historical Defect Imagery for Yield Learning", *The 11th Annual IEEE/SEMI Advanced Semiconductor Manufacturing Conference and Workshop*, Fairmont Copley Plaza Hotel, Boston, MA, September 12-14, 2000.