Field Test Results of an Automated Image Retrieval System

Kenneth W. Tobin, Thomas P. Karnowski, and Lloyd F. Arrowood
Image Science & Machine Vision Group
*Oak Ridge National Laboratory**
Oak Ridge, Tennessee.
USA
{tobinkwjr, karnowskitp, lfa}@ornl.gov

Fred Lakhani
Yield Management Tools Program
International SeMaTech
Austin, Texas
USA
fred.lakhani@sematech.orgu

Abstract

The rapid identification of yield detracting mechanisms through integrated yield management is the primary goal of defect sourcing and yield learning. At future technology nodes, yield learning must proceed at an accelerated rate to maintain current defect sourcing cycle times despite the growth in circuit complexity and the amount of data acquired on a given wafer lot [1]. As integrated circuit fabrication processes increase in complexity, it has been determined that data collection, retention, and retrieval rates will continue to increase at an alarming rate. Oak Ridge National Laboratory (ORNL) has been working with International SEMATECH (ISMT) to develop methods for managing the large volumes of image data that are being generated to monitor the status of the manufacturing process [2, 3]. This data contains an historical record that can be used to assist the yield engineer in the rapid resolution of manufacturing problems. To date there are no efficient methods of sorting and analyzing the vast repositories of imagery collected by off-line review tools for failure analysis, particle monitoring, line width control, and overlay metrology. In this paper we will describe a new method for organizing, searching, and retrieving defect imagery based on visual similarity. The results of an industry field test of the ORNL image management system at two independent manufacturing sites will also be described.

Keywords

yield management, yield learning, image management, datamining, content-based image retrieval, automated image retrieval, approximate nearest-neighbors searching, visual similarity

1 Introduction

The ability to manage large image databases has been a topic of growing research. Imagery is being generated and maintained for a large variety of applications including remote sensing, architectural and engineering design, geographic information systems, and weather forecasting. Content-based image retrieval (CBIR) is a technology that is being developed to address these needs [4]. CBIR refers to techniques used to index and retrieve images from databases based on their pictorial content. Pictorial content is typically defined by a set of features extracted from an image that describe the color, texture and/or shape of the entire image or of specific image regions. This feature description is used in CBIR to index a database through various means such as distance-based techniques, approximate nearest-neighbor searching, rule-based decision-making, and fuzzy inferencing [4, 5].

CBIR addresses a problem created by the growing proliferation of automated microscopy inspection in semiconductor manufacturing applications, i.e., the management and reuse of the large amounts of image data collected during defect inspection and review. For semiconductor yield management applications we have denoted CBIR technology as Automated Image Retrieval (AIR) [6, 7]. Digital imagery for failure analysis is generated between process steps from optical microscopy and laser scattering systems and from optical and confocal microscopy, scanning electron microscopy (SEM), atomic force microscopy (AFM), and focused ion beam (FIB) imaging modalities. This data is maintained in a data management system (DMS) and used by fabrication engineers to diagnose and isolate manufacturing problems. The semiconductor industry currently has no direct means of searching the DMS using image-based queries, even though 20,000 images are collected on average at a typical fabrication (fab)

*Prepared by the OAK RIDGE NATIONAL LABORATORY, Oak Ridge, Tennessee, 37831-6285, operated by UT-BATTELLE, LLC for the U.S. DEPARTMENT OF ENERGY under contract DE-AC05-00OR22725.

facility every week [8]. Current abilities to query the fabrication process are based primarily on product ID, lot number, wafer ID, time/date, process layer, engineer classification, or automatic defect classification (ADC), etc. Although this approach can be useful, it limits the user's ability to quickly locate historical examples of visually similar imagery, especially for data that was placed in the database over one or two weeks prior. Data much older than this is nearly irretrievable since retrieval is dependent on human memory and experience. Without the addition of datamining capabilities such as AIR, this large image repository will remain virtually untapped as a resource for rapidly resolving manufacturing problems.

The ORNL AIR system represents a unique application of CBIR technologies to the manufacturing environment. In Section 2 we will provide an overview of the AIR software system. In Section 3 we will describe the method of image analysis, feature indexing and image retrieval. In Section 4 we provide a brief comparison of AIR and ADC technologies. In Section 5 we will present recent results obtained from field-testing of our image retrieval system at two semiconductor fabrication sites during the Fall of 2000.

2 Overview of the AIR System

Image retrieval technologies have been under development since the early 1990's but very few applications have evolved for solving specific, real-world problems such as those in the manufacturing environment. Researchers at ORNL developed the capability for a flexible image retrieval technology for industrial applications that independently takes into account details regarding the product defectivity, substrate (i.e., the background structure on which the defect resides), and imaging modality characteristics [6]. The fundamental premise of the ORNL AIR method and technology is that *a similar process or phenomena likely generates images that are visually similar*. This implies that statistical process information that is associated with retrieved images can be used to identify and isolate errant process tools and equipment. Therefore, in our AIR system, process data associated with the inspected product is included with the defect imagery in a relational database for subsequent statistical analysis to provide the yield engineer with defect sourcing information.

The basic component of the AIR system is the indexing and retrieval engine, a dynamic link library (DLL), that generates the defect and substrate image feature descriptions and the indexing structure used for efficient storage and retrieval of images from the database. In addition to the core AIR DLL, the system includes an ORACLE database, a set of interface DLLs and executables, and graphical user interfaces. For our current semiconductor application, the fab DMS system generates an ASCII data file on a daily basis that provides process and image data in a format suitable for inclusion into the AIR database. A Windows NT service executable periodically checks for output from the DMS system, and when detected, the service adds the imagery and associated process data to the ORACLE database and builds the indexing structure necessary for efficient image retrieval.

3 Image Indexing and Retrieval Method

In this section, we will describe the methodology associated with AIR processing. In overview, this begins with the generation and/or use of the defect detection mask, which localizes the defect in the image. Next a series of image features are extracted from the defect and substrate regions of the images. The features become the entire representation of the image and are indexed for rapid retrieval from the database. Finally, the image data is associated with the manufacturing process within a relational database for subsequent statistical process analysis.

3.1 Defect Masks and Feature Analysis

Figure 1 shows a SEM image containing a particle defect in (a) and the associated defect mask in (b).

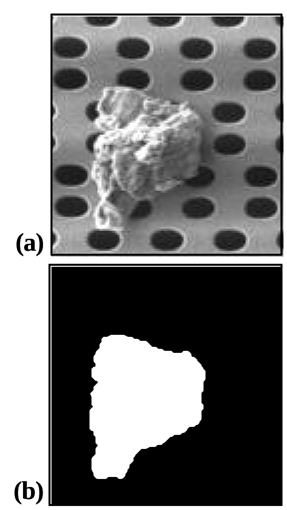

Figure 1 – (a) SEM image of a particle defect. (b) Associated defect mask generated by the inspection tool during automatic re-detection.

The defect mask is typically a binary representation that localizes the defect boundaries in the field of view. This mask can be generated in the form of a filled region, as shown in the figure, or as a perimeter composed of boundary pixels. Every defect detection tool in the semiconductor industry today that performs automated defect detection or re-detection generates a defect mask during the process. The defect mask is used to generate descriptive features regarding the defect such as its size and location, or more extensive information useful for ADC, such as color, texture, and shape features.

The defect mask is used in AIR to generate an extensive description of the defect region and the substrate region. There are currently 60 numerical features measured for the substrate that describe the color, texture, and structure. The defect is decomposed into 51 numerical features that describe the color, texture, and shape. The user has the ability to select various feature attributes when formulating a query so that, for example, a search can be accomplished to locate one defect shape on another product substrate by ignoring color attributes, which are likely to be highly variable from one process layer or product to the next. The user could also enable or disable other descriptive groups such as texture or shape as required.

This independent description of defect and substrate facilitates a wide variety of queries such as "find this defect on a different substrate", or "find this defect on any substrate". This flexibility allows a single AIR system to be used by a broad population of users with widely varying needs while still providing focused and specific image retrieval searches.

3.2 Indexing and Retrieval

The goal of indexing is to organize the image features in the database such that a ranked list of nearest neighbors can be retrieved without performing an exhaustive comparison with all the records in the database. For AIR this is achieved by generating a binary decision tree of the image features. A bin is defined as a bottom-level element in our tree structure, sometimes described as a "leaf" or terminal node, that contains a small list of images, e.g., a bottom-level bin may contain a list of image vectors $\{v_a, v_b, v_c, \ldots\}$. Under the AIR architecture, a query vector is compared at the top level to each of two sub-nodes and a decision is made as to which sub-tree to take. There are many ways to implement decision trees. For this work we have implemented an approximate nearest neighbor (ANN) indexing and search method that builds on *kd-tree* methods [9]. Whereas an exhaustive nearest-neighbor search of the *n* vectors (i.e., images) in the database would be of $O(n)$ computations, the *kd-tree* approach is of $O(log(n))$.

Figure 2 shows a simple example of a 2-dimensional feature space, (f_x, f_y), containing 18 image vector points partitioned into a *kd-tree* structure where each bin contains 3 points (i.e., image vectors). The *kd-tree* method allows for the rapid retrieval of the closest bin to the query point, *Q*, but the data in this bin are not necessarily the closest points and the nearest-neighbor result can be in error by an amount ε.

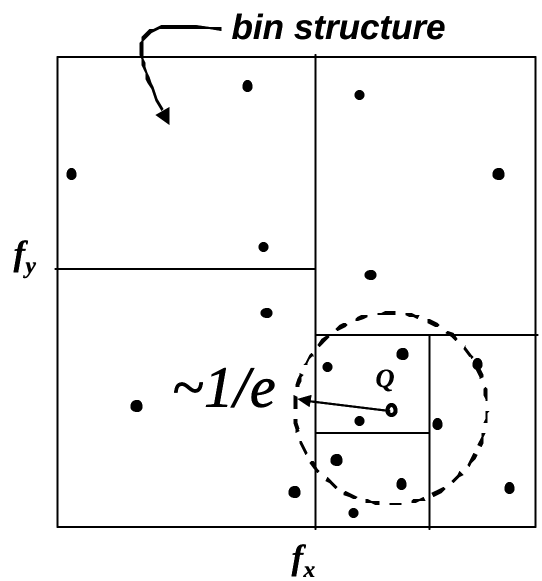

Figure 2- Example of a kd-tree bin structure showing the ANN search region about a query point, *Q*.

The ANN method incorporates a search window that results in the collection of neighboring bins about the query point. As this window increases in radius, the nearest neighbor error, ε, decreases, but the performance of the system also decreases to $O(n)$. The efficiency of the ANN method is proportional to $O((1/e)^{d/2} log(n))$, where d is the dimension of the feature space, n is the number of data points, and ε is the nearest neighbor error. The nearest-neighbor error is therefore inversely proportional to the size of the search window as shown in Fig. 2. As the radius of the search window increases, neighboring bins containing additional image vectors are included in the final nearest-neighbor search. As the radius continues to grow, the system approaches the complexity of an exhaustive nearest-neighbor search. Therefore, the accuracy of the AIR system is selectable as a trade-off between nearest neighbor performance and computational efficiency.

It should also be noted that the structure of the ANN method also facilitates the inclusion of new image data into the data set without necessarily requiring a rebuild of the database, or more specifically the indexing structure. The database is updated, i.e., rebuilt, on a periodic basis, e.g., once daily. During use, the system will be gathering image data that will be incorporated into the indexing structure during these periodic maintenance cycles. While images are being collected, they can be placed within the bin structure and retrieved during subsequent queries. As the number of image vectors in these bins increases, the efficiency of the ANN process will begin to decrease. When the periodic build is actuated, the image vectors will be re-distributed to result in the pre-defined minimum bin size required for optimal retrieval efficiency. The result is that this structure allows access to the latest image data by incorporating it into the database on the fly without immediate re-indexing.

4 Differences Between AIR and ADC Technologies

AIR has been designed to allow for the management of large repositories of defect image data through one system. Since its inception as a yield management tool, there have been many questions regarding the differences between AIR and the more common ADC systems that have been proliferating throughout the industry over the past decade. To respond to this, it is necessary to view the two systems through the concept of a simple feature space as shown in Fig. 3. Each point shown in the graph in (a) and (b) corresponds to the feature description of an image. In the case of the classifier in (a) the goal is to classify the data point, *C*, whereas for image retrieval in (b) the goal is to retrieve other data points that are similar to the query, *Q*.

In more detail, Fig. 3a shows a representation of the ADC system whose function is to classify, or assign, an unknown data point, *C*, to a class that has been defined through a training procedure. The ADC system typically requires training with data that is specific to an inspection tool. Within that tool set, there is a requirement to train on specific layers or process steps, and for the various products that are being inspected. Training is a cumbersome and sometimes unwieldy process that has proved to be a limitation, especially in fabs that manufacture many different products [10]. The ADC system is typically trained on relatively few samples, e.g., ten examples per class, therefore resulting in a class representation that is limited to a small fraction of the universe of images that are generated by tools and inspection processes. This is represented in the figure by the small number of points shown in each class region. The training set also defines the boundary of the class region (e.g., the shaded areas in Fig. 3a), which can vary greatly depending on the training data and classifier method used. The ADC system has evolved to perform the function of associating defects with labels (e.g., tungsten particle, missing pattern, poly flake, etc.) and therefore has the potential to be correct or incorrect, and the classification process is only an intermediary step towards associating the label with an errant manufacturing process, i.e., defect sourcing. And finally, the ADC is defect-centric in that training and execution of the classification procedure focuses primarily on the defect itself, and largely ignores the substrate as an identifying characteristic of the image.

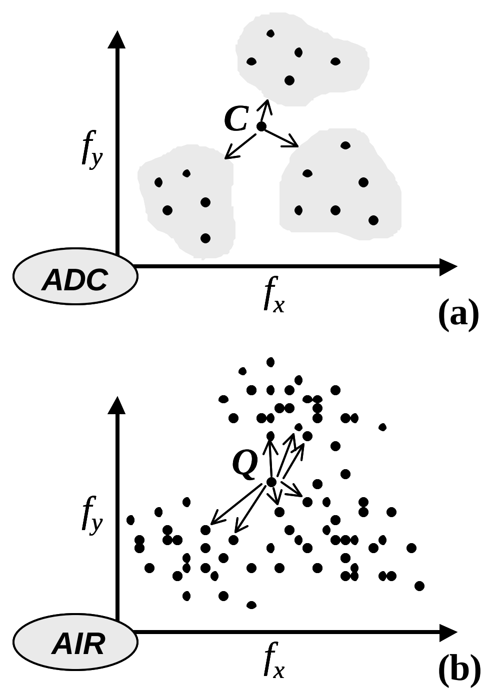

Figure 3 – A feature-space comparison of (a) ADC versus (b) AIR. In (a) the data point "C" is a point to be classified by the ADC system whereas in (b) the data point "Q" is a query upon which to perform an image retrieval.

Conversely, the AIR system, shown in Fig. 3b, performs the function of image retrieval based on a query point, *Q*. The AIR system organizes and maintains multiple sources of images in one system (e.g., optical and SEM, multiple layers, steps, tools, etc.) and an image-based query will retrieve a specified number of images from the database that are close to the query in the sense of visual similarity (e.g., based on a Minkowski distance in feature space). Therefore, the AIR system does not perform classification and does not assign a query point to a predefined label. When the database of images is coupled with the manufacturing data that describes the fabrication process - e.g., layer, step, lot, date, inspection tooling, EDX spectra, multiple modes of imaging such as optical, SEM, and Confocal - it becomes possible to associate the query image, *Q*, with visually similar historical images from the database therefore linking the query image directly to the process and potentially the source of the problem. And, since the AIR system focuses on both an extensive defect and substrate description, the association of defects with products, substrates, process steps, and layers is inherent in the analysis. An AIR system does not require training and its ability to comprehend a large population of images from multiple inspection tools and processes over a long period of time means that the limitations of ADC associated with focused training scenarios and frequent modifications to accommodate new products and process drift do not apply to AIR as they do with ADC.

5 Field Testing and Results

ORNL performed two field tests of the AIR software system during the Fall of 2000 for the purpose of verifying the fundamental premise that *a similar manufacturing process or phenomena likely generates images that are visually similar.* It was also desired that testing in a manufacturing environment be performed to determine system robustness, timing, capacity, usability, and what fab data was key to sourcing problems based on defect imagery. Additional defect information (e.g., defect position on the wafer, wafer ID, Lot No., etc.) had been incorporated into the ORNL AIR system through the use of foreign keys and additional database tables. The system was deployed at two semiconductor manufacturing sites 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 section summarizes the results of these field tests and demonstrates the utility of this approach through data analysis conducted on approximately one month of historical defect data at the two independent fabrication sites.

5.1 Architecture and Implementation

The AIR system architecture was described in Section 3 above. For field testing, a method for maintaining and associating process information with defect imagery was created. Although the AIR field test software was not designed to be a complete defect management system, it was necessary to include some DMS-type functionality to reach our project goals. Figure 4 shows the graphical user interface (GUI) for the field test system.

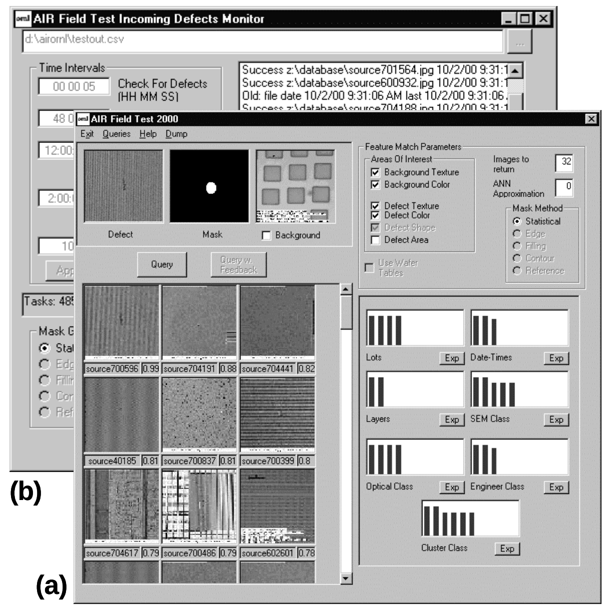

Figure 4 – (a) AIR Field Test software interface for controlling image retrieval. (b) Control panel for loading wafer data and images and monitoring progress.

Toward this end, we envisioned the submission of images to our system as the result of defect detection during inspection. During our design process, we developed database tables containing several relevant entities. These entities included the *Image*, that stores 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*, 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 AIR field test DLL.

5.2 Results

Table 1 shows the database statistics for each of the manufacturing sites after approximately one effective month of data collection. The table shows the total number of images associated with the various defects (i.e., there can be more than one image of each defect generated by different inspection tools), and the number of wafers, lots, and process steps.

Table 1 - Database statistics for Sites 1 and 2.

Value	SITE 1	SITE 2
Number of DEFECTS	59,593	76,653
Number of WAFERS	3,856	3,336
Number of LOTS	1,375	1,021
Number of STEP / LAYERS	99	164
Number of IMAGES	62,594	78,953
Oldest DATE	10-7-2000	9-14-2000
Latest DATE	11-6-2000	11-1-2000

Regarding system performance there are two times of interest in general to the semiconductor fab user. First, the time to add images to the database is important because the AIR system should basically be invisible to the underlying defect detection and inspection activity. Second, retrieval time is important because of usability issues and engineering response time.

Table 2 lists the timing statistics for data from the two test sites. For the purposes of comparison on a common platform, the data sets from each test site for the initial month of testing (i.e., 62,594 images from Site 1 and 78,953 images from Site 2) where loaded on a common machine using data that had been collected and returned to ORNL. The machine used was a 750 MHz Pentium PC. The median, mean, maximum and minimum time to add the images to the database are recorded in Table 2, along with image retrieval time. The image retrieval time was determined 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 sustained input of well over 100,000 images. The main difference between the timing for the sites is the image size; most images from Site 2 were JPEG, 320 x 240 images, while Site 1 images were JPEG, 640 x 480 or 480 x 480.

Table 2 - Timing statistics for image addition and retrieval.

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

Next we have modeled the retrieval system as a k-nearest neighbor (k-NN) classifier for the step/layer, lot, and optical classification categories. The experiments were performed as follows. For each site, we sampled 1024 images from each database and submitted them as query images returning 64 results. We then counted how many times the most common occurrence in the results matched the selected parameter in 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 (for success). Ties were assigned a value of 0.5 for unknown and if no matches were returned a value of 0 was assigned.

Figures 5 through 7 show the results of this k-NN test. Each figure contains weighted and un-weighted results for Sites 1 and 2, with classifiers using the first 4, 8, 16, 32, and 64 returned results. Un-weighted 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 in the data set. 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 images that are further down the list of retrieved images become visually dissimilar to the query and therefore are less likely to come from the same source. Note that in Fig. 5 there are 99 individual process steps represented in the Site 1 data and 164 in the Site 2 data. In Fig. 6 there are 1,375 lots and 1,021 lots representing Sites 1 and 2 respectively. In Fig. 7 there are 148 optical classes and 39 optical classes representing Sites 1 and 2 respectively. In all three of these k-NN comparisons the number of times the top four returned images matches the query for the indicated parameters averages around 70% - which supports our hypothesis regarding the link between visual similarity and manufacturing processes.

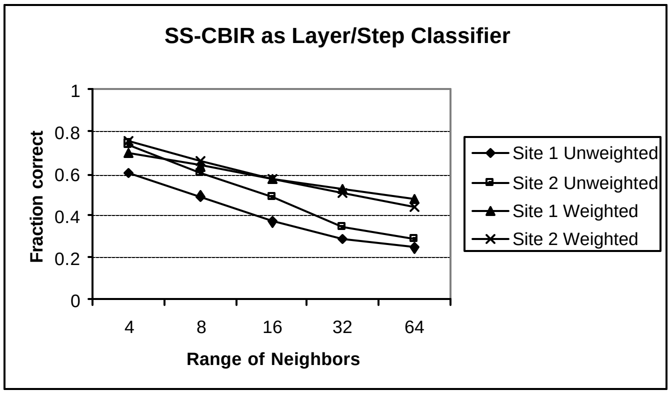

Figure 5 - Results using the AIR field test system as a Layer/Step classifier. Note that SS-CBIR refers to "semiconductor-specific CBIR".

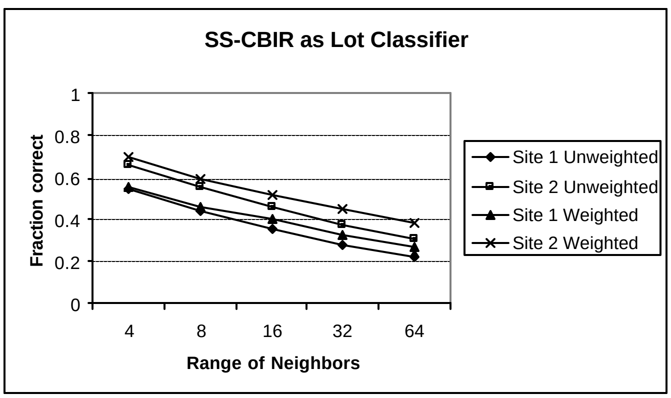

Figure 6 - Results using the AIR field test system as a Lot classifier.

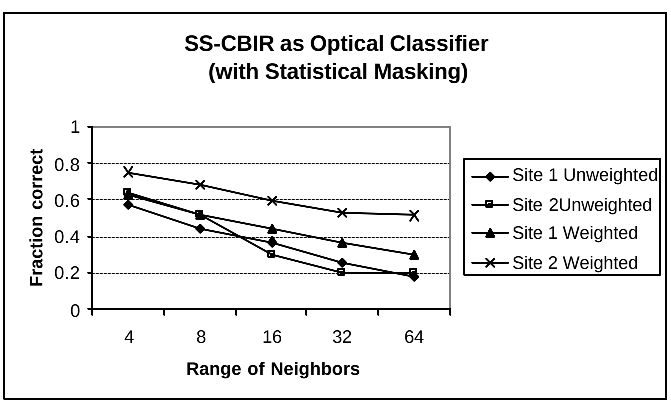

Figure 7 - Results using the AIR field test system as an Optical Defect classifier.

6 Conclusions

In this paper we have described a novel content-based image retrieval and management system that has been applied to the semiconductor manufacturing environment. The manufacturing focus of the ORNL CBIR application takes advantage of the way in which defects are detected with standard industry inspection equipment by uniquely describing the defect and the background areas of the image independently in terms of color, texture, structure, and shape. Current image retrieval systems for semiconductor manufacturing depend on additional alphanumeric data to perform retrieval functions (e.g., lot number, time/date, wafer ID, etc.), which produces an inherent limitation to the process of locating historic imagery that may have been caused by a similar manufacturing process.

The AIR system has been installed in two semiconductor manufacturing sites to determine system performance and retrieval characteristics. The system was shown to perform exceptionally well in terms of storage capacity and the time required to add and retrieve images and process data through the system. Through these field tests we were able to demonstrate our fundamental premise that *a similar process or phenomena likely generates images that are visually similar* by performing a series of k-NN classification tests to associate queries with process parameters such as process step, lot number, or optical classification code. Without the addition of content-based image retrieval, this large image repository of semiconductor images will remain virtually untapped as a resource for rapidly resolving manufacturing problems. The application of the ORNL AIR technology to other manufacturing environments that generate large amounts of product imagery during defect inspection and quality control is inherent.

7 Acknowledgements

The authors wish to acknowledge the Member Companies of International SEMATECH who participate in the Yield Management Tools Program in Austin, TX, for their advice, fabrication data, direction, and field test support of our research into automated defect data reduction technologies and new strategies for yield learning.

8 References

[1] Semiconductor Industry Association, International Technology Roadmap for Semiconductors, 1999.

[2] K.W. Tobin, T.P. Karnowski, S.S. Gleason, D. Jensen, F. Lakhani, “Integrated Yield Management”, 196th Meeting of the Electrochemical Society, Inc., Oct. 17-22, Honolulu, Hawaii, 1999.

[3] K.W. Tobin, T.P. Karnowski, R.K. Ferrell, and F. Lakhani, “Managing Defect Image Databases for Semiconductor Yield Monitoring and Control”, Global Semiconductor, Sterling Publications, February 2000.

[4] V.N. Gudivada and V.V. Raghavan, “Content-Based Image Retrieval Systems”, IEEE Computer Magazine, 0018-9162, September 1995, p. 18.

[5] 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.

[6] K.W. Tobin, T.P. Karnowski, and R.K. Ferrell, “Image Retrieval in the Industrial Environment”, IS&T/SPIE’s 11th International Symposium on Electronic Imaging: Science and Technology, San Jose Convention Center, January 1999.

[7] T.P. Karnowski, K.W. Tobin, R.K. Ferrell, and F. Lakhani, “Content Based Image Retrieval for Semiconductor Manufacturing”, IS&T/SPIE’s 12th International Symposium on Electronic Imaging: Science and Technology, San Jose Convention Center, January 2000.

[8] K.W. Tobin, T.P. Karnowski, F. Lakhani, “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.

[9] S. Arya, D.M. Mount, N. S. Netanyahu, R. Silverman, and A.Y. Wu, “An Optimal Algorithm for Approximate Nearest Neighbor Searching in Fixed Dimensions”, Proc. Of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, 1994, pp. 573-582.

[10] K.W. Tobin, T.P. Karnowski, S.S. Gleason, D. Jensen, and F. Lakhani, “Using Historical Wafermap Data for Automated Yield Analysis”, 45th Annual International Symposium of the American Vacuum Society: Manufacturing Science and Technology, Nov. 2-6, Baltimore, Maryland, 1998.