

CONF-980117--

# ADAPTATION OF THE FUZZY K-NEAREST NEIGHBOR CLASSIFIER FOR MANUFACTURING AUTOMATION

Kenneth W. Tobin<sup>a</sup>, Shaun S. Gleason, Thomas P. Karnowski  
Oak Ridge National Laboratory<sup>b</sup>, P.O.Box 2008, Oak Ridge, Tennessee 37831-6011

RECEIVED

FFR 25 1998

OSTI

## ABSTRACT

The use of supervised pattern recognition technologies for automation in the manufacturing environment requires the development of systems that are easy to train and use. In general, these systems attempt to emulate an inspection or measurement function typically performed by a manufacturing engineer or technician. This paper describes a self-optimizing classification system for automatic decision making in the manufacturing environment. This classification system identifies and labels unique distributions of product defects denoted as "signatures". The technique relies on encapsulating human experience through a teaching method to emulate the human response to various manufacturing situations. This has been successfully accomplished through the adaptation and extension of a feature-based, fuzzy k-nearest neighbor (k-NN) classifier that has been implemented in a pair-wise fashion. The classifier works with pair-wise combinations of the user-defined classes so that a significant reduction in feature space and problem complexity can be achieved. This k-NN implementation makes extensive use of hold-one-out results and fuzzy ambiguity information to optimize its performance. A semiconductor manufacturing case study will be presented. The technique uses data collected from in-line optical inspection tools to interpret and rapidly identify characteristic signatures that are uniquely associated with the manufacturing process. The system then alerts engineers to probable yield-limiting conditions that require attention.

**Keywords:** fuzzy classifier, fuzzy ambiguity, k-nearest neighbor, parametric optimization, industrial automation, semiconductor

## 1. INTRODUCTION

Image processing technologies are being applied more frequently in industrial applications today than ever before. The availability of high-speed, robust, and relatively low cost computing platforms and dedicated hardware devices makes computer vision technology an attractive automation approach to many applications. Once a computer vision system begins generating data for a manufactured product, the available options for data analysis, reduction, and decision making are generally limited to those tools developed in-house. In this paper we will describe a general classification approach to analyze this data based on the fuzzy k-nearest neighbor (k-NN) classifier<sup>1</sup> implemented in a pair-wise fashion.

Pattern recognition methods are playing an increasingly important role in automation by reducing large amounts of data to provide a useful decision for process control and/or product finishing. This system was developed for automating the analysis and interpretation of wafer defect and electrical-test data collected in a semiconductor-manufacturing environment.<sup>2,3,4</sup> Although the system has been developed specifically for the automated analysis of semiconductor wafer product data, the application of the approach to other feature-based data measurements will be described as well.

### 1.1. Non-parametric Approach

As mentioned above, the fuzzy k-NN classifier system is feature-based. In the manufacturing environment, image data is collected from in-line inspection tools and defect signatures are segmented for analysis. For the native semiconductor application, a signature object is defined as a unique pattern of individual optical defects or electrical bin codes that were generated by an errant process. Features are extracted from the segmented object and are sent to the classifier where a user-defined class label is assigned to the result. The user-defined class result then indicates the specific tool or process that must be corrected<sup>5</sup>.

<sup>a</sup> K.W.T. (Correspondence): Email: tobinkwjr@ornl.gov; WWW: <http://www-ismv.ic.ornl.gov>; Telephone: 423-574-8521; Fax: 423-574-6663

<sup>b</sup> Work Performed for SEMATECH, Austin, Texas, under Contract No. ERD-95-1340 and prepared by OAK RIDGE NATIONAL LABORATORY, Oak Ridge, Tennessee, 37831-6285, managed by LOCKHEED MARTIN ENERGY RESEARCH CORP. for the U.S. DEPARTMENT OF ENERGY under contract DE-AC05-96OR22464.

DTIC QUALITY INSPECTED 3

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED



MASTER

19980402 068

## **DISCLAIMER**

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Simple morphological image processing tends to either join too many unrelated defects together or not enough together. Expert semiconductor fabrication engineers have demonstrated that they can easily group clusters of defects from a common manufacturing problem source into a single signature. Capturing this thought process is ideally suited for fuzzy logic. A system of rules was developed to join disconnected clusters based on properties such as elongation, orientation, and distance. The clusters are evaluated on a pair-wise basis using the fuzzy rules and are joined or not joined based on a defuzzification and threshold. The system continuously re-evaluates the clusters under consideration as their fuzzy memberships change with each joining action. The fuzzy membership functions for each pair-wise feature, the techniques used to measure the features, and methods for improving the speed of the system are all developed. Examples of the process are shown using real-world semiconductor wafer maps obtained from chip manufacturers. The superior clustering offered through fuzzy logic enables more accurate feature measurements and process characterization for the SSA classification process.

## 2. BACKGROUND

Fuzzy logic has been used to solve complex problems in control, pattern recognition, and image processing<sup>3</sup>. The strength of a fuzzy logic system is its ability to quantify a complex set of relationships where a "conventional" system of equations cannot be written. An example used to convey this notion is the parking of a car. Driving students can easily understand instructions like "Pull up to the forward car, turn the wheel back to the curb hard, ease in and straighten the car out as you enter the space, then turn the wheel back hard the other way and ease forward." However, it is very difficult to quantify these concepts as a set of equations suitable to a computerized control system. Even if the system of equations could be obtained, it would be complex and would also not match the reality of the problem well, since equations are designed to match exact situations that are not needed in this case and complicate our goal of parking the car.

Several good references<sup>3,4</sup> exist that describe fuzzy logic and fuzzy reasoning in much more detail than described here. We simply wish to introduce these concepts. The process of designing a fuzzy system consists of several steps described below.

1. Determine the suitability of fuzzy logic to the system. In this step, we are examining the problem objectively and determining the nature of the problem. If the solution can be described using linguistic rules, it is likely a good application for fuzzy logic.
2. Determine the linguistic rules that apply to the problem. In this step we examine the problem and empirically determine the governing rules that dictate a solution to the problem. Expert interrogation and examination of the actual features that can be measured relating to the problem are key components to this step.
3. Determine the fuzzy membership functions that quantify the different conditions and conclusions to the rules. In this step (as in step 2) it is very important that human feedback be applied. For example, if we say "this object is large" or "this object is small", we must determine a membership function that maps a measured characteristic (such as area) to a linguistic value. If the membership function does not reflect our intuitive feelings about what makes an object large, medium, or small it must be modified or the linguistic rules we determined are useless. The goal here is to find a membership function that captures our human labels adequately.
4. Determine the methods used to implement the fuzzy logic. The first method of issue is inference, where we determine the degree of truth for each linguistic rule in our system and how to combine the results of the rules. The two most widely used inference techniques are min-max and product-sum. In min-max the output degree of truth is limited to the degree of truth of the smallest premise, then we take the maximum over all fuzzy subset values for the output variable. In product-sum inferencing the output value is scaled by the degree of truth, then we sum all the fuzzy subset values assigned to the output variable. The last method of issue is the defuzzification, where we take the fuzzy memberships assigned to the output variable and come up with a final, "hard" or "crisp" number that defines our solution. The most commonly used method is a centroid or center-of-gravity technique, where the output memberships are combined and the center of the membership function values are found. The choice of these methods may rely on the implementation, ease-of-use, or heuristic determination<sup>3</sup>.

It is important to note that fuzzy logic was designed to deal with ambiguity in human intuition. Fuzzy logic is not imprecise or vague in its implementation. Characteristics of an image such as area can be easily measured and assigned a numerical

value. The problem arises when we want to put a label on the numerical value and identify it as very large, large, medium, etc. The ambiguity is thus not in the actual feature but in our human interpretation of the feature, especially in the context of rules that use the object size to make a decision.

### 3. FUZZY RULES AND FEATURE MEASUREMENTS

In the case of connecting clusters that were likely caused by the same errant manufacturing process, we examined hundreds of different images and wrote rules that applied to the clusters we thought belonged to common events. Some examples of these images are shown in Figure 1. The rules were determined largely through an intuitive process of writing down our thoughts on which clusters should be connected and why. For example, in Figure 1a (a collection of small and large scratches) the extreme proximity of the clusters dictates a high connectivity. In Figure 1b the different orientation of the large clusters on the right side of the image and the wafer outline near the image center leads us to believe they are actually caused by different events. Finally, Figure 1c shows several spun-on-glass streak event that should be joined together by combinations of proximity, orientation of the clusters, and the elongation of the clusters. Fifteen rules were determined. These rules are shown in Figure 2.

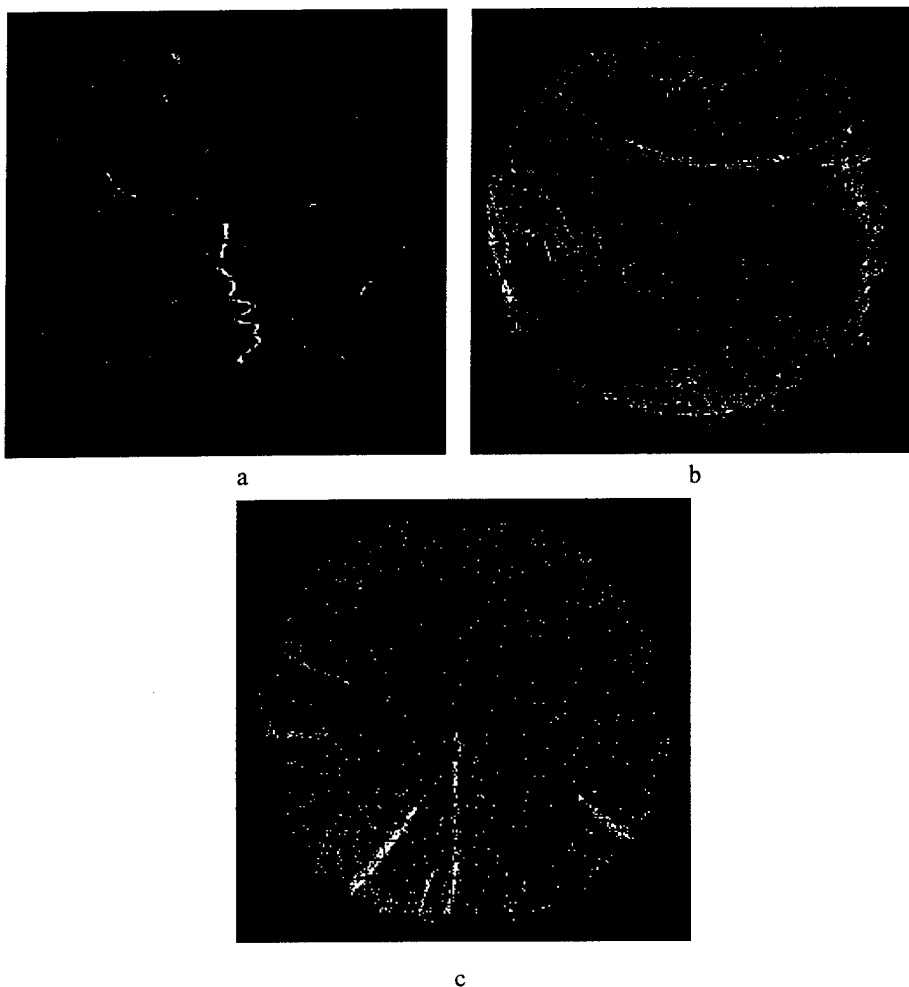


Figure 1 - Wafer maps showing clusters.

1. If Distance EXTRALOW then Connectivity HIGH
2. If Distance HIGH then Connectivity LOW
3. If Distance LOW and (Area\_I LOW or Area\_J LOW) then Connectivity HIGH
4. If Distance LOW and Area\_I HIGH and Area\_J HIGH and Orientation NOT LOW then Connectivity HIGH
5. If Distance MEDIUM and Area\_I HIGH and Area\_J HIGH and Elongation\_I HIGH and Elongation\_J HIGH and Orientation HIGH then Connectivity HIGH
6. If Distance MEDIUM and Area\_I MEDIUM and Area\_J MEDIUM and Elongation\_I HIGH and Elongation\_J HIGH and Orientation MEDIUM then Connectivity MEDIUM
7. If Distance MEDIUM and Area\_I MEDIUM and Area\_J MEDIUM and Elongation\_I NOT LOW and Elongation\_J NOT LOW and Orientation HIGH then Connectivity HIGH
8. If Distance MEDIUM and Orientation NOT HIGH then Connectivity LOW
9. If Distance MEDIUM and Area\_I HIGH and Area\_J HIGH and (Elongation\_I OR Elongation\_J LOW) then Connectivity MEDIUM
10. If Distance MEDIUM and Area\_I NOT LOW and Area\_J NOT LOW and Elongation\_I MEDIUM and Elongation\_J MEDIUM and Orientation NOT LOW then Connectivity MEDIUM
11. If Distance MEDIUM and Area\_I MEDIUM and Area\_J MEDIUM and Elongation\_I NOT LOW and Elongation\_J NOT LOW and Orientation HIGH then Connectivity HIGH
12. If Distance LOW and Area\_I NOT LOW and Area\_J NOT LOW and Elongation\_I NOT LOW and Elongation\_J NOT LOW and OR-I-J NOT LOW then Connectivity HIGH
13. If Distance LOW and Area\_I NOT LOW and Area\_J NOT LOW and Elongation\_I HIGH and Elongation\_J HIGH and Orientation LOW then Connectivity LOW
14. If Distance LOW and Area\_I NOT LOW and Area\_J NOT LOW and Elongation\_I LOW and Elongation\_J LOW then Connectivity HIGH
15. If Distance LOW and Area\_I MEDIUM and Area\_J MEDIUM and Elongation\_I HIGH and Elongation\_J HIGH and Orientation LOW then Connectivity MEDIUM

**Figure 2 - Fuzzy Rules for Semiconductor Cluster Connectivity.**

As the rules show, there are four different fuzzy input types (Distance, Area, Elongation, and Orientation) and one fuzzy output for our inference system. These variables are defined in Figure 3. Note that we denote two clusters that are candidates for connection as cluster "I" and cluster "J". Two fuzzy input types, Elongation and Area, are measured for each cluster in a pair. The other two input types, Distance and Orientation, are functions of both clusters. The fuzzy membership functions are shown in Figures 4 and 5. The membership functions were determined heuristically for our application by starting with a triangular function and using examples to determine suitable modifications. Random test images were used and a judgment was made about the characteristics of each object based solely on their appearance. We use the following notation to indicate the membership functions,

$$F_{Variable,SET}(v), \quad (1)$$

where *Variable* is the feature under consideration ( Distance, Orientation, Area\_I etc. ), *v* is its actual numerical value, and *SET* is the possible values it can assume ( EXTRALOW, LOW, MEDIUM, and HIGH for Distance ).

<b>Distance</b>	Smallest distance between the perimeter of two objects
<b>Orientation</b>	Measure of the similarity between the orientation of object I, J, and a line between the centroids of object I and J
<b>Area</b>	Size of each object I, J in pixels
<b>Elongation</b>	Measure of the object morphology; long narrow objects have high elongation; measured for each object I, J
<b>Connectivity</b>	Measure dictating whether two objects should be connected. The defuzzified value is thresholded to connect / not connect two objects.

**Figure 3. Fuzzy variable definitions.**

The features we measured to derive the fuzzy rule are fairly straightforward. The area was measured by simply counting the number of pixels in an object. The elongation was determined by using the formula<sup>5</sup>,

$$Elongation = \frac{MajorEllipse - MinorEllipse}{MajorEllipse + MinorEllipse}, \quad (2)$$

where *MajorEllipse* is the length of the major axis of the best-fitting ellipse to the object and *MinorEllipse* is the length of the minor axis of the best fitting ellipse to the object.

Two other input variables deserve special note. The Distance variable is the smallest length between the perimeters of two objects. This was regarded as a special feature because it was difficult and potentially time consuming to measure. However, a similar feature such as distance between centroids was deemed inadequate because it does not truly capture what an expert examines when determining whether two objects should be joined together. In the next section we will explain how we measured the distance between two objects.

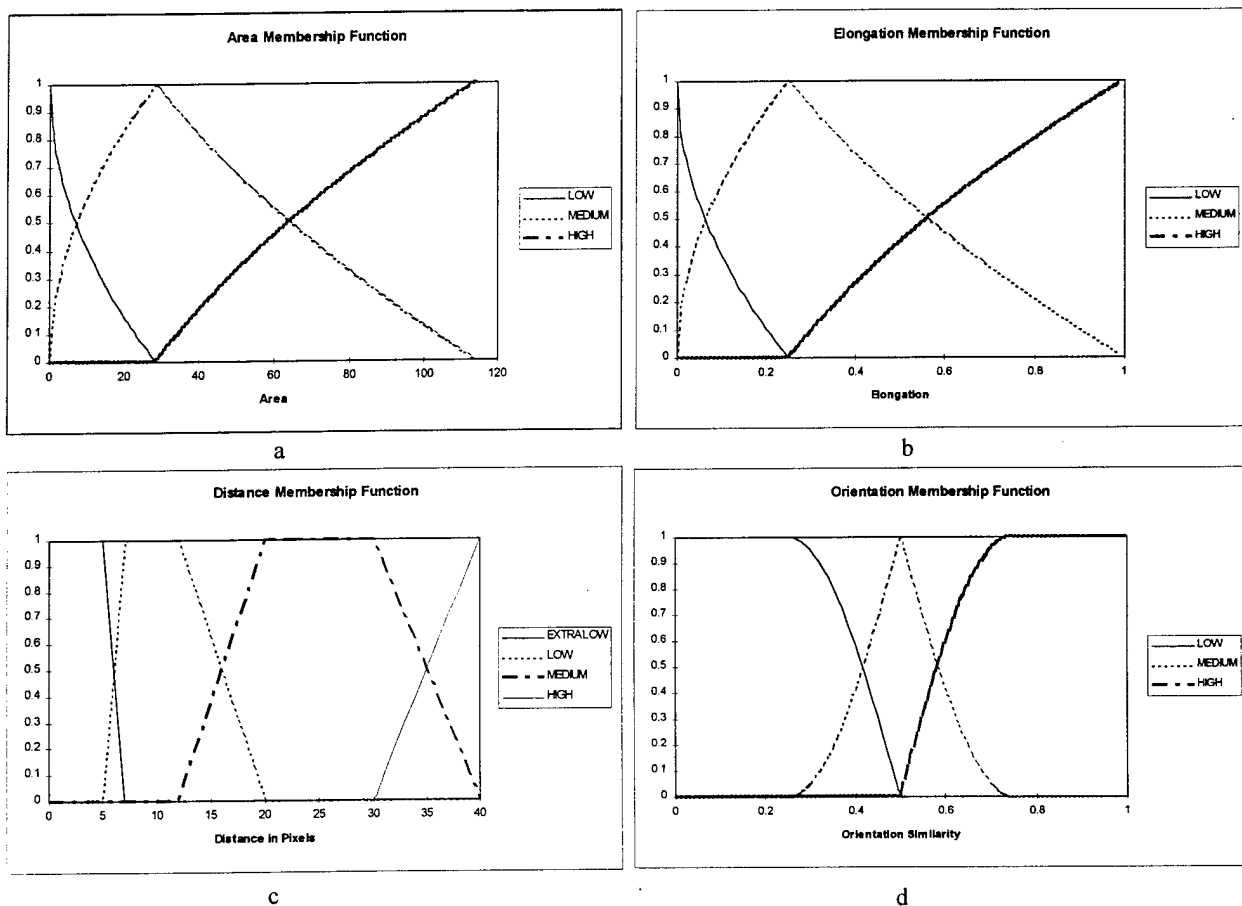
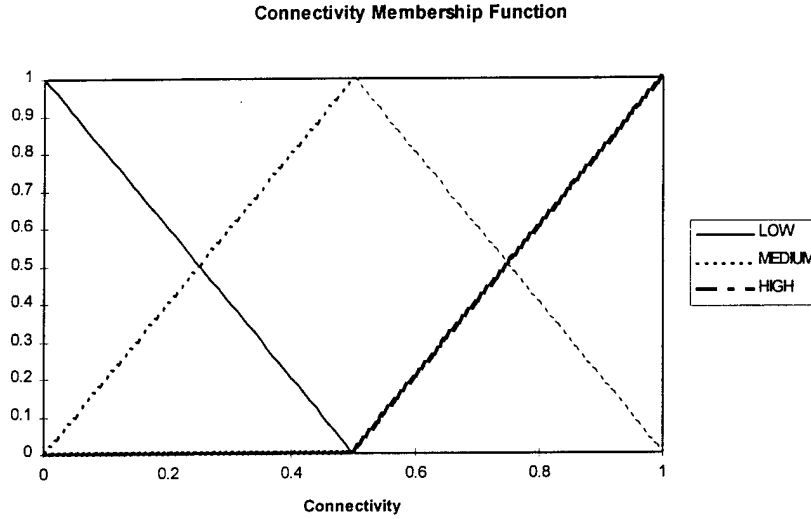


Figure 4 - Input variable membership functions. a) Area b) Elongation c) Distance d) Orientation.



**Figure 5 - Connectivity membership function.**

The Orientation variable is measured by the formula shown below.

$$Orientation = |\cos(O_I - O_J)| |\cos(O_I - O_L)| |\cos(O_L - O_J)|, \quad (3)$$

where  $O_I$  is the orientation of object I in radians,  $O_J$  is the orientation of object J in radians, and  $O_L$  is the orientation (in radians) of a line drawn between the centroids of object I and J

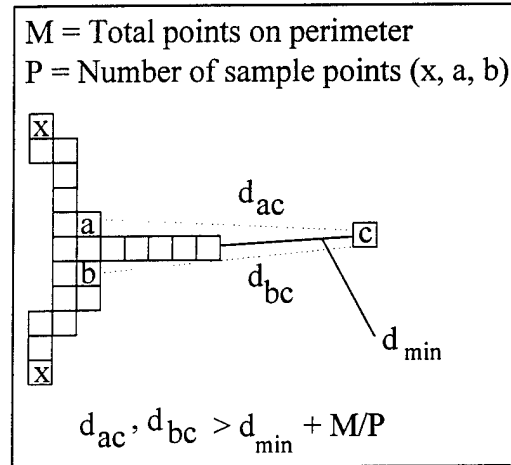
The orientation variable identifies objects that are collinear.

#### 4. IMPLEMENTATION

Because of the nature of this problem - joining pairs of objects together based on their morphology and proximity - we implemented our rules in a pair-wise fashion by considering a pair of objects at a time. This was difficult in practice, however, because the number of comparisons increases by the square of the number of objects under consideration. We were able to introduce time-saving techniques that are described here.

We performed a uniform morphological dilation on the initial images to restrict the objects that could potentially be joined. In this step we ensured that sufficient dilations were performed so that objects that would have a Distance membership of NOT HIGH were considered. The dilation step served to reduce the number of objects under consideration considerably, resulting in time savings in the connectivity process.

The measurement of the distance between a pair of objects proved to be one of the most difficult and time consuming issues. We were able to accelerate the process by taking advantage of the fact that we did not need to measure a distance accurately if the fuzzification of the distance produced a membership of 1 in the HIGH subset as shown in Figure 4c. An algorithm was developed to determine the distance between to objects in an iterative fashion. Figure 6 serves to illustrate the algorithm. We sampled M points on an object with P total perimeter points. We compute two distances  $d_{ac}$  and  $d_{bc}$  as the distances between points a and b on one object and c on the other object. However, we have no more than M/P points between point a and point b, where M is the total number of border points for the object and P is the number of points we sample (of which a and b are two of these). In the scenario where this coarse measurement is most inaccurate, these M/P points form a kind of peninsula that reaches toward point c.



**Figure 6 - Sampling points on the object perimeter to determine distance error bounds.**

As Figure 6 shows, the closest that the points between a and b can be to c is

$$\min(d_{ac} - d_{bc}) - \frac{M}{P} . \quad (4)$$

Since we know we are not interested in exactly measuring large distances, we can make these coarse measurements, examine their results, and then continue with a finer search if necessary. This procedure continues until the exact distance (if required) is measured. In our SSA processing, we had access to a list of the coordinates of the border pixels of each object. This list was determined *a priori* by an earlier step of the SSA advanced clustering through a chain code process. We were able to use these lists to perform coarse distance measurements that were further refined as needed.

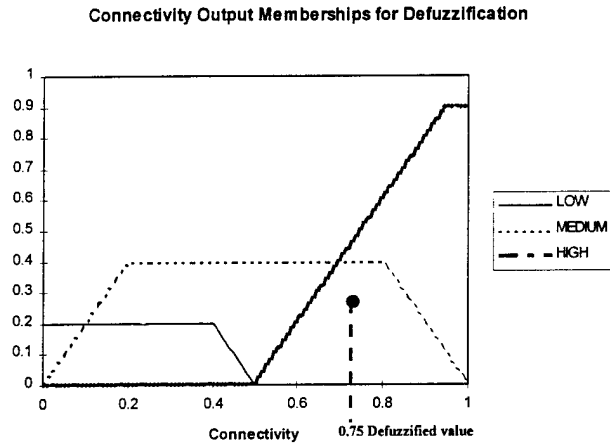
We also used another time-saving procedure. As the distances between two objects are computed, they are saved in a look-up table for future use. Whenever two objects I and J are connected, the table entries for all pairs involving I and J are replaced by the smallest of the distances between I and J. Thus the distance between any two objects need be computed only once at a cost of some memory space for the table.

Our inference system used the min-max technique for rule evaluation. In the min-max technique we step through the rules and determine the degree of truth for each condition in each fuzzy rule. For example, rule 8 from Figure 2 is "If Distance MEDIUM and Orientation NOT HIGH then Connectivity LOW". To determine the degree of truth for the first condition in this rule we measure the Distance and find the fuzzy membership for HIGH. We do a similar operation for Orientation. Using min-max inferencing, then, the degree of truth of the statement "Connectivity LOW" for this rule is simply

$$\min(F_{Distance, HIGH}(Distance), [1 - F_{Orientation, HIGH}(Orientation)]) \quad (5)$$

We apply similar operations across all the rules with conclusion "Connectivity LOW" and take the value of the rule with the maximum result (hence the "max" term in the min-max inferencing). After continuing for the other sets (MEDIUM and HIGH in this case) we have a membership function. Figure 7 shows the membership function result for LOW 0.2, MEDIUM 0.4, and HIGH 0.9. We compute the centroid of the regions for our defuzzification step. In our application, we threshold the defuzzified value to make a "connect / do not connect" decision.

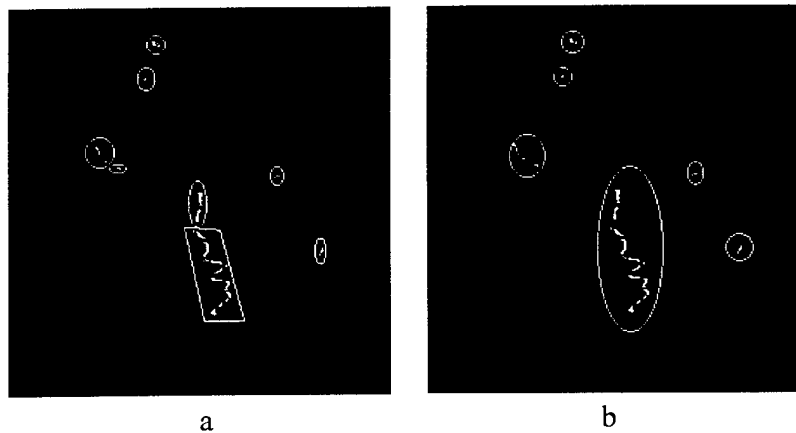




**Figure 7 - Example of connectivity membership function after rules evaluation.**

## 5. RESULTS

The clustering results are shown in Figures 8, 9, and 10 for the wafer maps of Figure 1. Evaluating these results is somewhat subjective because there is some ambiguity in the judgment of individuals for clustering. However, the results shown do point out the advantages of the fuzzy clustering method. In all these results a gray scale image of the wafer maps was created by dividing the wafer surface into pixels and counting the number of defects in each pixel. A threshold was then applied to produce an initial segmentation into clusters and non-clustered defects. The clusters were then dilated to produce the initial, “unfuzzy” clustering results shown in Figure 8a, 9a, and 10a. In Figure 8 we see a fairly satisfactory result with this clustering. Perhaps the biggest flaw is the failure to join the two halves of the large scratch in the middle of the image. In Figure 8b we applied the fuzzy clustering described here and successfully joined these halves to produce a more intuitively “correct” clustering.



**Figure 8 - Clustering results from Figure 1a.**

Figure 9 shows the clustering results from Figure 1b. The first image shows an excessive number of disconnected clusters using the simple proximity based method. The second image shows the result of performing additional dilations in an attempt to use the simple proximity method to join more clusters. In some cases this strategy of additional dilations will

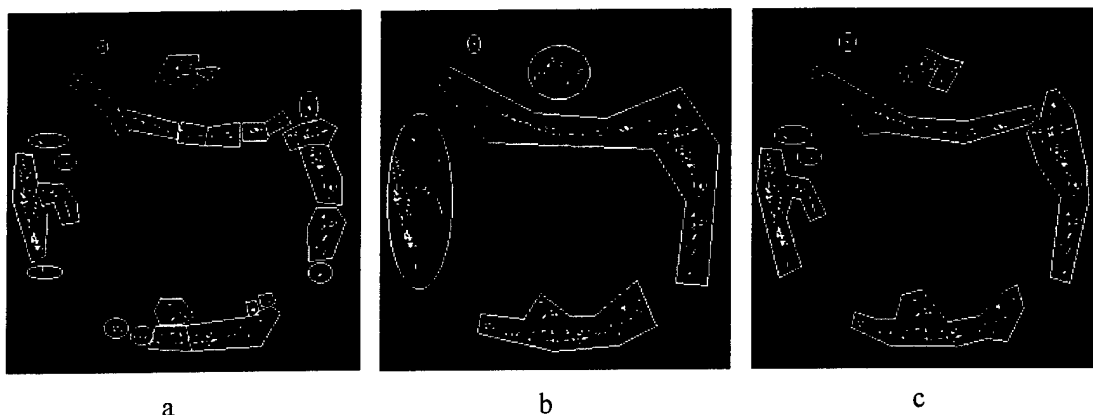


Figure 9 - Clustering results from Figure 1b.

produce good results. If this technique were applied to Figure 1a, for example, the result would be identical to the fuzzy connectivity clustering shown in Figure 8b. In Figure 9b, however, it definitely shows a major flaw; the large circular cluster in the middle of the map is joined to the cluster on the right side of the map. In this case we have joined two distinct events into one. The fuzzy clustering result in Figure 9c, on the other hand, maintains the separate identity of the clusters and allows more accurate post-processing to extract features and classify the segmented events.

Finally, Figure 10 also shows a three image sequence of simple clustering, simple clustering with more dilation, and finally fuzzy clustering. In this case the fuzzy result is able to maintain the separate identity of the three streaks at the bottom of the image, that helps capture the true nature of the events far better than the other two clusterings.

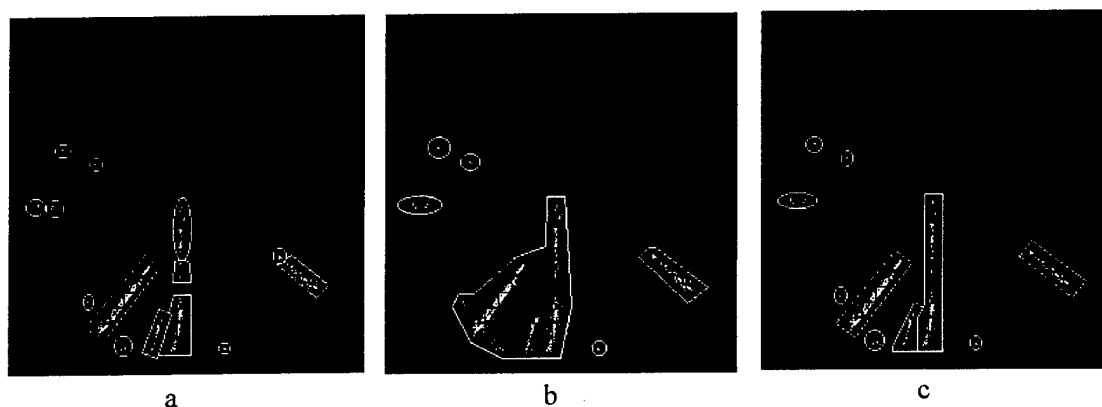


Figure 10 - Clustering results from Figure 1c.

## 6. CONCLUSION

In this paper we have shown the applicability of fuzzy logic to the problem of connecting clustered defects in semiconductor wafer maps. We offered some background on fuzzy logic and how it pertains to our solution. We showed the fuzzy rules and feature membership functions that were used in the SSA software package. Due to the nature of our problem, we required pair-wise rule evaluation and feature measurement that can be computationally expensive. We discussed some techniques to improve the method. Finally we showed some clustering results that point out the ability of our fuzzy inference system to connect clusters. While no single clustering method will satisfy every human observer, these results show the effectiveness of our fuzzy logic system in joining disconnected clusters while still maintaining the separate identity of elongated, separated events. The technique proved very effective in field tests for this semiconductor inspection application. We believe the technique could apply well to many other manufacturing environments where defects are

distributed in spatial patterns by errant manufacturing processes.

## 7. REFERENCES

1. K.W. Tobin, S.S. Gleason, T.P. Karnowski, and S.L. Cohen, "Feature Analysis and Classification of Manufacturing Signatures on Semiconductor Wafers", Machine Vision Applications in Industrial Inspection V, Proc. SPIE, Vol. 3029, p. 12-25, April 1997.
2. Gleason, S.S., and Tobin, K.W., "Directional-based Dilation for Connection of Piece-Wise Objects: A Semiconductor Manufacturing Case Study", International Conference on Image Processing, Lausanne, Switzerland, Sept. 16-19, 1996.
3. Terano, T., Asai, K, and Sugeno, M., "Applied Fuzzy Systems", Academic Press, Inc., 1994.
4. Yager, R.R., and Zadeh, L.A., "An Introduction to Fuzzy Logic Applications in Intelligent Systems", Kluwer Academic Publishers, 1991.
5. Jain, Anil K., "Fundamentals of Image Processing", Prentice Hall Information and System Sciences Series, 1989.

M98003174



Report Number (14) ORNL/CP-96062  
CONF-980117--  
\_\_\_\_\_  
\_\_\_\_\_

Publ. Date (11) 199801

Sponsor Code (18) DOE, XF

JC Category (19) UC-900, DOE/ER

DOE