

NASA-CR-191975

GRANT

IN-61-CR

Fuzzy Set Methods for Object Recognition in Space Applications Fourth and Final Quarter Report

James M. Keller, ed.

University of Missouri-Columbia

4/1/92 - 6/30/92

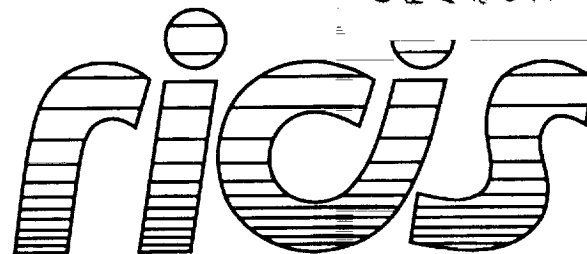
**Cooperative Agreement NCC 9-16
Research Activity No. SE.42**

**NASA Johnson Space Center
Information Systems Directorate
Information Technology Division**

N93-18224
--THRU--
N93-18228
Unclass

G3/61 0143399

(NASA-CR-191975) FUZZY SET METHODS
FOR OBJECT RECOGNITION IN SPACE
APPLICATIONS Quarterly Report No. 4
and Final, 1 Apr. - 30 Jun. 1992
(Research Inst. for Computing and
Information Systems) 79 p



*Research Institute for Computing and Information Systems
University of Houston-Clear Lake*

TECHNICAL REPORT

The RICIS Concept

The University of Houston-Clear Lake established the Research Institute for Computing and Information Systems (RICIS) in 1986 to encourage the NASA Johnson Space Center (JSC) and local industry to actively support research in the computing and information sciences. As part of this endeavor, UHCL proposed a partnership with JSC to jointly define and manage an integrated program of research in advanced data processing technology needed for JSC's main missions, including administrative, engineering and science responsibilities. JSC agreed and entered into a continuing cooperative agreement with UHCL beginning in May 1986, to jointly plan and execute such research through RICIS. Additionally, under Cooperative Agreement NCC 9-16, computing and educational facilities are shared by the two institutions to conduct the research.

The UHCL/RICIS mission is to conduct, coordinate, and disseminate research and professional level education in computing and information systems to serve the needs of the government, industry, community and academia. RICIS combines resources of UHCL and its gateway affiliates to research and develop materials, prototypes and publications on topics of mutual interest to its sponsors and researchers. Within UHCL, the mission is being implemented through interdisciplinary involvement of faculty and students from each of the four schools: Business and Public Administration, Education, Human Sciences and Humanities, and Natural and Applied Sciences. RICIS also collaborates with industry in a companion program. This program is focused on serving the research and advanced development needs of industry.

Moreover, UHCL established relationships with other universities and research organizations, having common research interests, to provide additional sources of expertise to conduct needed research. For example, UHCL has entered into a special partnership with Texas A&M University to help oversee RICIS research and education programs, while other research organizations are involved via the "gateway" concept.

A major role of RICIS then is to find the best match of sponsors, researchers and research objectives to advance knowledge in the computing and information sciences. RICIS, working jointly with its sponsors, advises on research needs, recommends principals for conducting the research, provides technical and administrative support to coordinate the research and integrates technical results into the goals of UHCL, NASA/JSC and industry.

***Fuzzy Set Methods for Object
Recognition in Space Applications
Fourth and Final Quarter Report***

THE UNIVERSITY OF CHICAGO PRESS

CHICAGO, ILLINOIS 60607

TEL: 773/936-3700

FAX: 773/936-3701

WWW.CHICAGO.PRESS.EDU

CHICAGO.PRESS.EDU

CHICAGO.PRESS.EDU

CHICAGO.PRESS.EDU

CHICAGO.PRESS.EDU

CHICAGO.PRESS.EDU

CHICAGO.PRESS.EDU

CHICAGO.PRESS.EDU

CHICAGO.PRESS.EDU

CHICAGO.PRESS.EDU

CHICAGO.PRESS.EDU

CHICAGO.PRESS.EDU

primary

RICIS Preface

This research was conducted under auspices of the Research Institute for Computing and Information Systems by James M. Keller of the University of Missouri-Columbia. Dr. Terry Feagin was the initial RICIS research coordinator for this activity. Dr. A. Glen Houston, Director of RICIS and Assistant Professor of Computer Science, later assumed the research coordinator role.

Funding was provided by the Information Technology Division, Information Systems Directorate, NASA/JSC through Cooperative Agreement NCC 9-16 between the NASA Johnson Space Center and the University of Houston-Clear Lake. The NASA technical monitor for this activity was Robert N. Lea, of the Software Technology Branch, Information Technology Division, Information Systems Directorate, NASA/JSC.

The views and conclusions contained in this report are those of the author and should not be interpreted as representative of the official policies, either express or implied, of UHCL, RICIS, NASA or the United States Government.



Fourth and Final Quarter Report: April 1, 1992 to June 30, 1992

Fuzzy Set Methods For Object Recognition In Space Applications

Fixed-Price Subcontract NO. 088

Under Cooperative Agreement No. NCC 9-16

Project No. SE. 42

To:

UNIVERSITY OF HOUSTON-CLEAR LAKE
2700 Bay Area Boulevard
Houston, TX 77058

UHCL Technical Representative:

Dr. Terry Feagin

Principal Investigator:

James M. Keller
Electrical and Computer Engineering Department
University of Missouri-Columbia
Columbia, Mo. 65211
(314) 882-7339
eckeler@umcvmb.bitnet

Introduction

For the fourth and final quarter of this research contract, we are going to report progress on the following four Tasks (as described in the contract):

1. Fuzzy Set Based Decision Methodologies
2. Membership Calculation;
3. Clustering Methods (including derivation of pose estimation parameters);
4. Acquisition of images and testing of algorithms.

The report, as has done in the past, consists of "stand alone" sections describing the activities in each task. It does not duplicate the material contained in the previous quarterly reports. For details of the earlier work done under this contract, please refer to the first three quarterly reports.

Fuzzy Set Based Decision Methodologies

In this section, we report on two new fuzzy set based techniques that we developed for decision making. These include:

1. A method to generate fuzzy decision rules automatically for image analysis.
2. A decision making algorithm based on possibility expectation.

The following pages contain the details of these two pieces of work.

A Method to Generate Decision Rules Automatically for Image Analysis

In this report, we propose a method to generate rules automatically for image analysis such as segmentation. The method used for segmentation is best described by the following paper submitted to the North American Fuzzy Information Proceeding Society (NAFIPS '92). For this report, slight modifications are made where only the experimental example differs from the original paper.

N 93-18225

Automatic Rule Generation for High-Level Vision

Frank Chung-Hoon Rhee and Raghu Krishnapuram
Department of Electrical and Computer Engineering
University of Missouri, Columbia, MO 65211

ABSTRACT

Many high-level vision systems use rule-based approaches to solve problems such as autonomous navigation and image understanding. The rules are usually elaborated by experts. However, this procedure may be rather tedious. In this paper, we propose a method to generate such rules automatically from training data. The proposed method is also capable of filtering out irrelevant features and criteria from the rules.

1. Introduction

High-level computer vision involves complex tasks such as image understanding and scene interpretation. In domains where the models of the objects in the image can be precisely defined, (such as the blocks world, or even the world of generalized cylinders) existing techniques for description and interpretation perform quite well. However, when this is not the case (such as the case of outdoor scenes or extra-terrestrial environments), traditional techniques do not work well. For this reason, we believe that the greatest contribution of fuzzy set theory to computer vision will

be in the area of high-level vision. Unfortunately, very little work has been done in this highly promising area. Fuzzy set theoretic approaches to high-level vision have the following advantages over traditional techniques: i) they can easily deal with imprecise and vague properties, descriptions, and rules, ii) they degrade more gracefully when the input information is incomplete, iii) a given task can be achieved with a more compact set of rules, iv) the inferencing and the uncertainty (belief) maintenance can both be done in one consistent framework, v) they are sufficiently flexible to accommodate several types of rules other than just IF-THEN rules. Some examples of the types of rules that can be represented in a fuzzy framework are [1] possibility rules ("The more X is A , the more possible that B is the range for Y "), certainty rules ("The more X is A , the more certain Y lies in B "), gradual rules ("The more X is A , the more Y is B "), unless rules [2] ("if X is A , then Y is B unless Z is C ").

The determination of properties and attributes of image regions and spatial relationships among regions is critical for higher level vision processes involved in tasks such as autonomous navigation, medical image analysis and scene interpretation. Many high-level systems have been designed using a rule-based approach [3,4]. In these systems, common-sense knowledge about the world is represented in terms of rules, and the rules are then used in an inference mechanism to arrive at a meaningful interpretation of the contents of the image. In a rule-based system to interpret outdoor scenes, typical rules may be

**IF a REGION is RATHER THIN AND SOMEWHAT STRAIGHT
THEN it is a ROAD**

**IF a REGION is RATHER GREEN AND HIGHLY TEXTURED AND
IF the REGION is BELOW a SKY REGION
THEN it is TREES**

Attributes such as "THIN" and "NARROW", and properties such as "BRIGHT" and "TEXTURED" defy precise definitions, and they are best modeled by fuzzy sets. Similarly, spatial relationships such as "LEFT OF", "ABOVE" and "BELOW" are difficult to model using the all-or-nothing traditional techniques [5]. We may interpret the attributes, properties and relationships

as "criteria". Therefore, we believe that a fuzzy approach to high-level vision will yield more realistic results.

In most rule-based systems, the rules are usually enumerated by experts, although they may also be generated by a learning process. Several techniques have been suggested in the literature to generate rules for control problems [6-9], some of which use neural net methods to model the control system [7-12]. These systems convert a given set of inputs to an output by fuzzifying the inputs, performing fuzzy logic, and then finally defuzzifying the result of the inference to generate a crisp output [13]. Some of the methods also "tune" the membership functions that define the levels (such as "LOW", "MEDIUM" and "HIGH") of the input variables [10]. While these methods have been shown to be very effective in solving control problems, they cannot be directly used in high-level vision applications. For example, in control systems, the fuzzy rules have consequents which are usually a desired level of a control signal whereas in high-level vision, the consequent clauses are usually fuzzy labels. Also, it is desirable that membership functions for levels of fuzzy attributes such as "THIN", and "NARROW", and properties such as "BRIGHT" be related to how humans perceive such attributes or properties. Hence they have very little to do with the decision making or reasoning process in which they are employed. In many reasoning systems for high-level vision, confidence (or importance) factors are associated with every rule since the confidence in the labeling may depend on the confidence of the rule itself. In this paper, we propose a new method to generate rules for high-level vision applications automatically. The rules so obtained may be combined with the rules given by the experts to complete the rule base.

In Section 2, we describe several fuzzy aggregation operators which can be used in hierarchical (multi-layer) aggregation networks for multi-criteria decision making. In Section 3, we describe how these aggregation networks can be used to filter out irrelevant attributes, properties, and relationships and at the same time generate a compact set of fuzzy rules (with associated confidence factors) that describes the decision making process. In Section 4 we present some

experimental results on automatic rule generation. Finally Section 5 contains the summary and conclusions.

2. Fuzzy Aggregation Operators

Fuzzy set theory provides a host of very attractive aggregation connectives for integrating membership values representing uncertain and subjective information [14]. These connectives can be categorized into the following three classes based on their aggregation behavior: i) union connectives, ii) intersection connectives, and iii) compensative connectives. Union connectives produce a high output whenever any one of the input values representing different features or criteria is high. Intersection connectives produce a high output only when all of the inputs have high values. Compensative connectives are used when one might be willing to sacrifice a little on one factor, provided the loss is compensated by gain in another factor. Compensative connectives can be further classified into mean operators and hybrid operators. Mean operators are monotonic operators that satisfy the condition: $\min(a,b) \leq \text{mean}(a,b) \leq \max(a,b)$. The generalized mean operator [15] as given below is one of such operator.

$$g(x_1, \dots, x_n; p, w_1, \dots, w_n) = \left(\sum_{i=1}^n w_i x_i^p \right)^{1/p}, \text{ where } \sum_{i=1}^n w_i = 1. \quad (1)$$

The w_i 's can be thought of as the relative importance factors for the different criteria. The generalized mean has several attractive properties. For example, the mean value always increases with an increase in p [15]. Thus, by varying the value of p between $-\infty$ and $+\infty$, we can obtain all values between min and max. Therefore, in the extreme cases, this operator can be used as union or intersection. The γ -model devised by Zimmermann and Zysno [16] is an example of hybrid operators, and it is defined by

$$y = \left(\prod_{i=1}^n x_i^{\delta_i} \right)^{1-\gamma} \left(1 - \prod_{i=1}^n (1 - x_i)^{\delta_i} \right)^{\gamma}, \text{ where } \sum_{i=1}^n \delta_i = n \text{ and } 0 \leq \gamma \leq 1. \quad (2)$$

In general, hybrid operators are defined as the weighted arithmetic or geometric mean of a pair of fuzzy union and intersection operators as follows.

$$A \oplus_{\gamma} B = (1 - \gamma)(A \cap B) + \gamma(A \cup B) \quad (3)$$

$$A \otimes_{\gamma} B = (A \cap B)^{(1 - \gamma)}(A \cup B)^{\gamma} \quad (4)$$

The parameter γ in (3) and (4) controls the degree of compensation. The γ -model in (2) is a hybrid operator of the type in (4). The compensative connectives are very powerful and flexible in that by choosing correct parameters, one can not only control the nature (e. g. conjunctive, disjunctive and compensative), but also the attitude (e. g. pessimistic and optimistic) of the aggregation.

One can formulate the problem of multicriteria decision making as follows. The support for a decision may depend on supports for (or degrees of satisfaction of) several different criteria, and the degree of satisfaction of each criterion may in turn depend on degrees of satisfaction of other sub-criteria, and so on. Thus, the decision process can be viewed as a hierarchical network, where each node in the network "aggregates" the degree of satisfaction of a particular criterion from the observed support. The inputs to each node are the degrees of satisfaction of each of the sub-criteria, and the output is the aggregated degree of satisfaction of the criterion. Thus, the decision making problem reduces to i) selecting robust and useful criteria for the problem on hand, ii) finding ways to generate memberships (degrees of satisfaction of criteria) based on values of features (criteria) selected, and iii) determining the structure of the network and the nature of the connectives at each node of the network. This includes discarding irrelevant criteria to make the network simple and robust.

In our previous research, we have investigated the properties of several union and intersection operators, the generalized mean, and the γ -model [14,17]. We have shown that optimization procedures based on gradient descent and random search can be used to determine the proper type of aggregation connective and parameters at each node, given only an approximate structure of the network and given a set of training data that represent the inputs at the bottom-most level and the desired outputs at the top-most level [14,17]. In this paper, we extend this idea to the detection of irrelevant attributes and automatic rule generation.

3. Redundancy Analysis and Rule Generation

In the approach we propose, we first fuzzily partition the range of values that each criterion (property or an attribute or a relation) can take into several linguistic intervals such as LOW, MEDIUM and HIGH. The set of properties or an attributes or a relations which are used are the ones that may appear in the antecedent clause of a rule. As explained in Section 1, the membership function for each level needs to be determined according to how humans perceive such attributes, properties or relations. The membership values for an observed attribute, property or relationship value in each of the levels is calculated using such membership functions. (Methods to generate degrees of satisfaction of relationships such as "LEFT OF" may be found in [18]). The memberships are then aggregated in a fuzzy aggregation network of the type shown in Figure 1. The top nodes of the network represent the labels that may appear in the consequents of the rules. A suitable structure for the network, and suitable fuzzy aggregation operators for each node are chosen. The network is then trained with typical attribute, property or relationship data with the corresponding desired output values for the various labels to learn the aggregation connectives and connections that would best describe in input-output relationships. The learning may be implemented using a gradient descent approach similar to the backpropagation algorithm [14,17]. It is to be noted that there is a constraint on the weights.

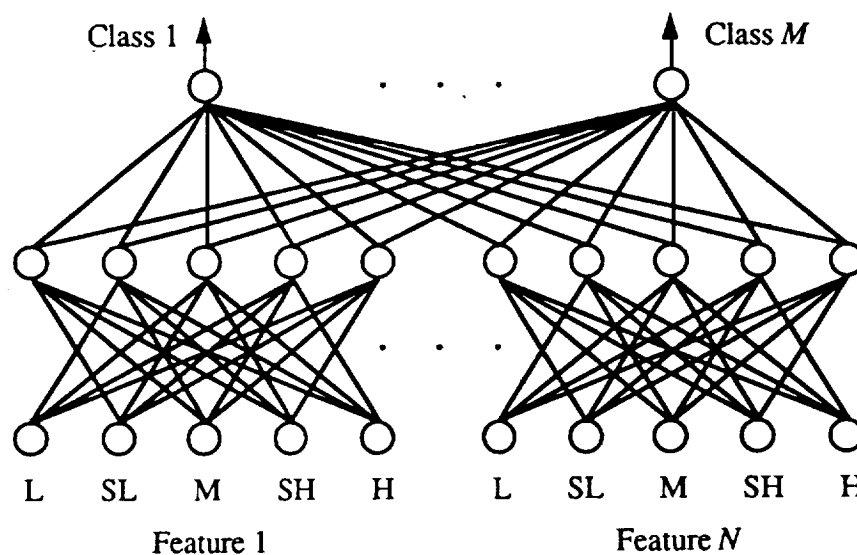


Figure 1 : Network for generating fuzzy rules.

Our experiments indicate that the choice of the network is not very critical. Also any compensative aggregation operator seems to yield good results. In all the results shown in this paper, we used the generalized mean operator as the aggregation operator. As indicated in Section 2, the generalized mean can closely approximate a union (intersection) operator for a large positive (negative) value of p . We start the training with the generalized mean aggregation function with $p=1$. If the training data is better described by a union (intersection) operator, then the value of p will keep increasing (decreasing) as the training proceeds, until the training is terminated when the error becomes acceptable. Also, the weights w_i in (1) may be interpreted as the relative importance factors for the different criteria. Initially we start the training with all the weights associated with a node being equal. As the training proceeds the weights automatically adjust so that the overall error decreases. Some of the weights eventually become very small. Thus, the training procedure has the ability to detect certain types of redundancies in the network. In general, there are three types of redundancies (irrelevant criteria) that are encountered in decision making [17]. These correspond to uninformative, unreliable and superfluous criteria.

Uninformative Criteria: These are criteria whose degrees of satisfaction are always approximately the same, regardless of the situation. Therefore, these criteria do not provide any information about the situation, thus contributing little to the decision-making process. For example, low texture content is a criterion that is always satisfied for both clear skies and roads, and hence it would be a uninformative criterion if one needs to distinguish between these two labels. Uninformative criteria do not contribute to the robustness of the decision making process, and therefore it is desirable that they be eliminated.

Unreliable Criteria: These correspond to criteria whose degrees of satisfaction do not affect the final decision. In other words, the final decision is the same for a wide range of degrees of satisfaction. For example, color would be an unreliable criterion for distinguishing a rose from a hibiscus because they both come in similar colors. Unreliable criteria do not contribute to the robustness of the decision making process, and therefore it is desirable that they be eliminated.

Superfluous Criteria: These are criteria which are strictly speaking not required to make the decision. The decisions made without considering such criteria may be as accurate or as reliable. For example, one may want to differentiate planar surfaces from spherical surfaces using Gaussian and mean curvatures, but the criteria are superfluous because either one of them is sufficient to distinguish between planar and spherical surfaces. However, redundancies of this type are not entirely without utility, since such redundancies make the decision making process more robust. If one criterion fails for some reason, we may still be able to arrive at the correct decision using the other. Hence such redundancies may be desirable to increase the robustness of the decision-making process.

Redundancy Detection and Estimation of Confidence Factors: A connection is considered redundant if the weight associated with it gradually approaches to zero (or a small threshold value) as the learning proceeds. A node (associated with a criterion) is considered redundant if all the connections from the output of this node to other nodes become redundant. Our simulations show that both in the case of uninformative criteria and unreliable criteria, the weights corresponding to all the output connections go to zero. Therefore such nodes (criteria) are eliminated from the structure. The examples in Section 4 illustrate this idea.

Rule Generation: The networks that finally result from this training process can be said to represent rules that may be used to make the decisions. If the final value of the parameter p at a given node is greater than one, the nature of the connective is disjunctive. If the value is less than one, it is conjunctive. Once the nature of the connective at each node is determined, we can easily construct the fuzzy rules that describe the input-output relations. In Section 4 we present some examples of this approach.

4. Experimental results

In this section, we present some typical experimental results involving real data to show the effectiveness of the proposed automatic rule generation method. The method is shown to generate decision rules that best describe the decision criteria for the classes in the experiment. Figure 1

shows the general 3 layer neural network used to generate the rules. The input layer consists of nN number of input nodes where N is the number of fuzzy features or criteria (such as properties and relationships) and n is the number of linguistic levels used to partition each feature. For the hidden layer, there are nN hidden nodes where each node is connected to all but one (i.e., it is connected to $n-1$) input nodes representing levels within each feature. The top layer fully connects the hidden layer. In the experimental results shown here, we used 5 fuzzy linguistic levels to represent each feature, therefore, each hidden node has 4 connections. Other types of network structures were also tried, however the one described above produced the best results. The target values in the training data were chosen to be 1.0 for the class from which the training data was extracted, and 0.0 for remaining classes. The feature values were always normalized so that they fall in the range $[0,1]$. Figure 2 depicts the trapezoidal fuzzy sets used to model the intuitive notions of the five linguistic levels **LOW (L)**, **SOMEWHAT LOW (SL)**, **MEDIUM (M)**, **SOMEWHAT HIGH (SH)**, and **HIGH (H)**.

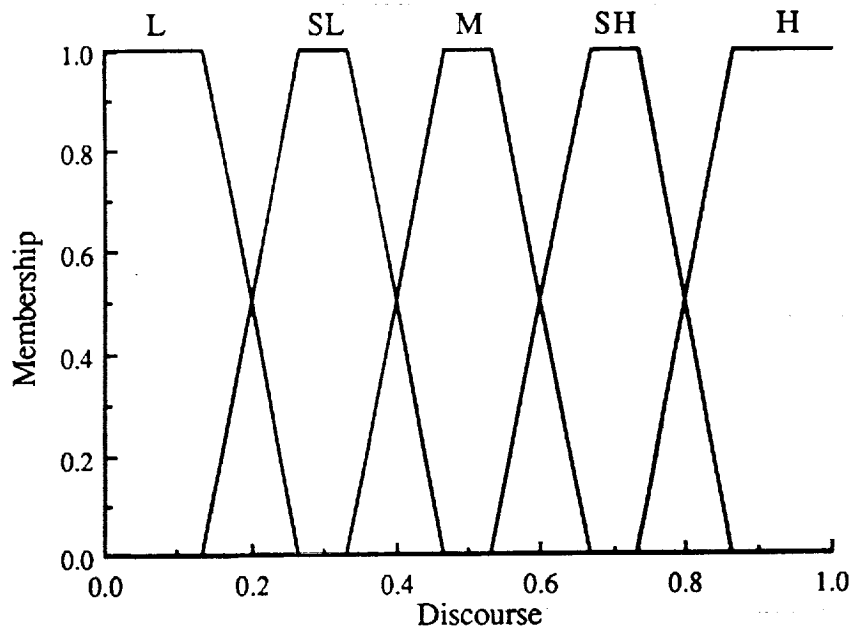


Figure 2 : Graphical representations of various fuzzy sets.

4.1 Example

Figure 3(a) shows a 200×200 image used for training in order to obtain rules that best describes the object (shuttle) and background. After examining a variety of possible features to be used, the two best features chosen were the difference entropy and contrast features. For definitions of the features, see report on membership generation methods. Figures 3(b) and 3(c) show images using these features. Figure 3(d) shows the scatter plot of the training samples extracted from two different regions (shuttle and background) in the image. We used 50 samples from each class. The membership values in each linguistic level for each sample is computed using the membership functions shown in Figure 2, and these with the corresponding desired targets are used as training data in the training algorithm described in Section 3. Figure 4 shows the reduced network after training. All the connections with weights below a value of 0.01 were considered redundant. Table 1 shows the final weights (which determine the confidence factors of the rules and criteria) and the p parameter values (which determine the conjunctive or disjunctive nature of the connective) for the specified nodes in Figure 4. Using the properties for the p values obtained, the following rules are generated, as discussed in Section 3.

$$\begin{aligned} \text{Class Shuttle} = & (\text{Difference Entropy M} \vee \text{Difference Entropy SH} \vee \text{Difference Entropy H}) \vee \\ & (\text{Contrast SL}). \end{aligned} \quad (5)$$

In other words, the rule may be summarized as

$$\begin{aligned} R_{\text{Shuttle}} : & \text{ IF Difference Entropy is M or SH or H or Contrast is SL} \\ & \text{ THEN the class is Shuttle.} \end{aligned}$$

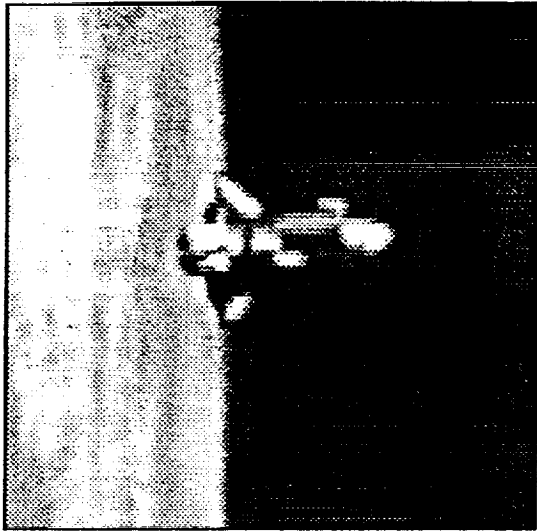
Similarly,

$$\begin{aligned} \text{Class Background} = & (\text{Difference Entropy SL} \vee \text{Difference Entropy SH}) \wedge \\ & (\text{Contrast L}) \end{aligned} \quad (6)$$

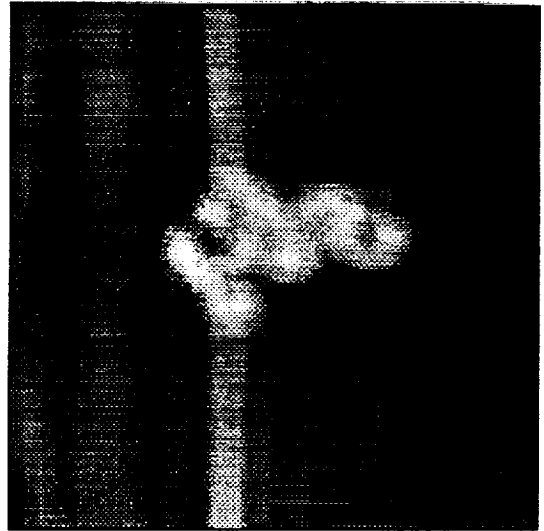
and

$$\begin{aligned} R_{\text{Background}} : & \text{ IF Difference Entropy is SL or SH and Contrast is L} \\ & \text{ THEN the class is Background.} \end{aligned}$$

These rules makes sense since by expanding (5) and (6), the expansions results in the appropriate cell locations where the training samples are located in Figure 3(d).



(a)



(b)



(c)

Figure 3(a) : image for training, (b) : difference entropy image, and (c) : contrast image.

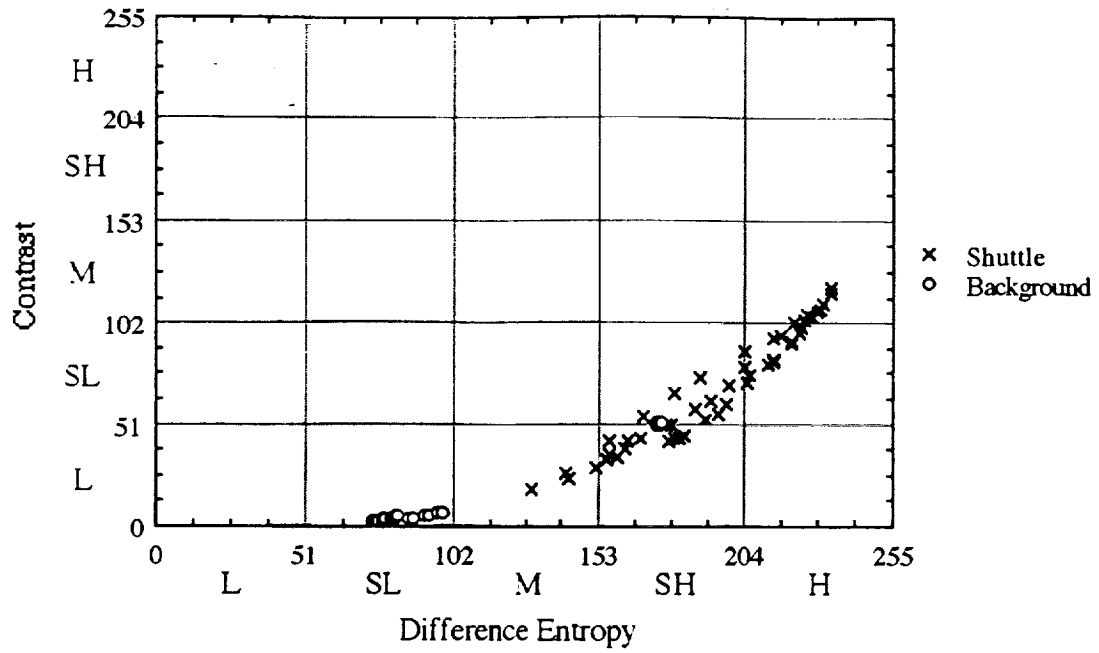


Figure 3(d) : Scatter plot of training samples for the classes shuttle and background.

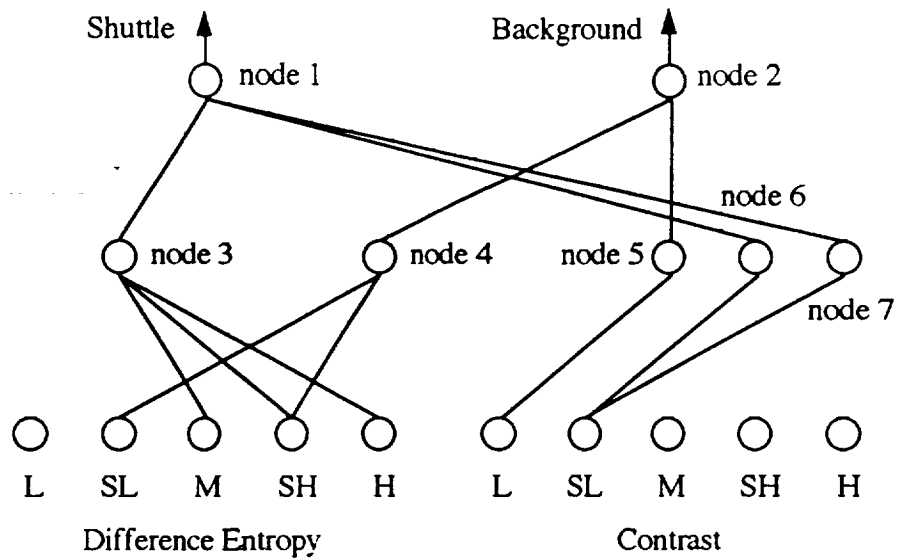


Figure 4 : Reduced network after training.

Table 1 : Values of weights and parameter p for the reduced network.

	weights	p
node 1	0.70 0.15 0.15	5.48
node 2	0.94 0.06	-0.21
node 3	0.49 0.01 0.50	7.04
node 4	0.94 0.06	4.00
node 5	1.0	0.78
node 6	1.0	1.88
node 7	1.0	1.88

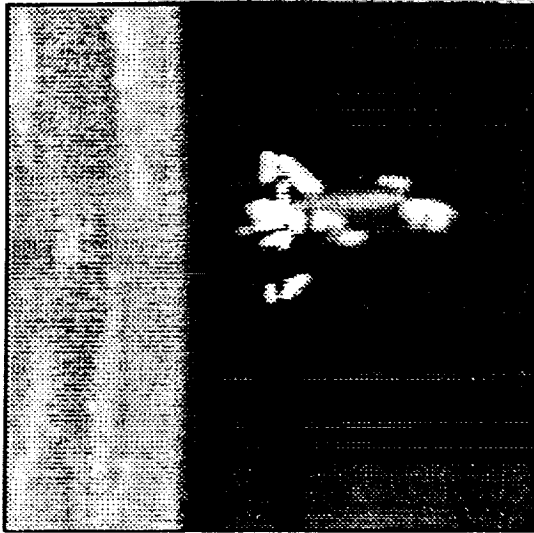
4.2 Segmentation

Figure 5(a) shows a 200×200 test image for segmentation using the reduced network after training shown in Figure 4. Figures 5(b) and 5(c) show images of the two features (difference entropy and contrast) that were chosen previously. After employing the shrink and expand algorithm to remove noise points, the resulting segmented image is shown in Figure 5(d) .

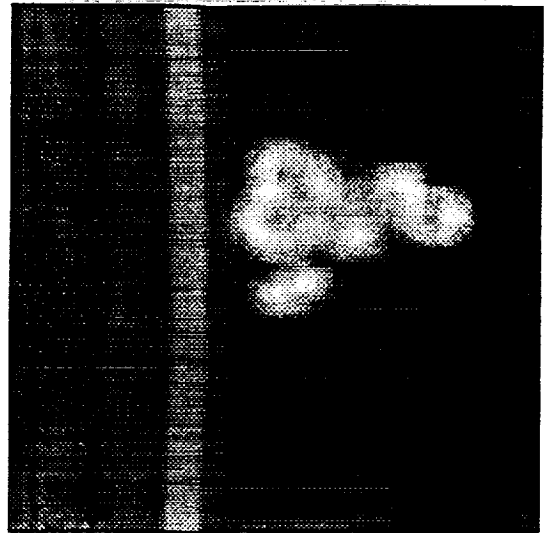
5. Summary and Conclusions

In this paper, we introduced a new method for automatically generating rules for high level vision. The range of each feature is fuzzily partitioned into several linguistic intervals such as LOW, MEDIUM and HIGH. The membership function for each level is determined, and the membership values for an observed feature value in each of the linguistic levels is calculated using these membership functions. The memberships are then aggregated in a fuzzy aggregation network. The networks are trained with typical data to learn the aggregation connectives and connections that would give rise to the desired decisions. The learning process can also be made to discard redundant features. The networks that finally result from this training process can be said to represent rules that may be used to make the decisions. Riseman et al used similar rules for

segmentation and labeling of outdoor scenes, but the weights used in the aggregation scheme were determined empirically [19]. The ability to generate rules that can be used in fuzzy logic and rule-based systems directly from training data is a novel aspect of our approach. One of the issues that requires investigation is the choice of the number of linguistic levels and its effect on the decision making process.



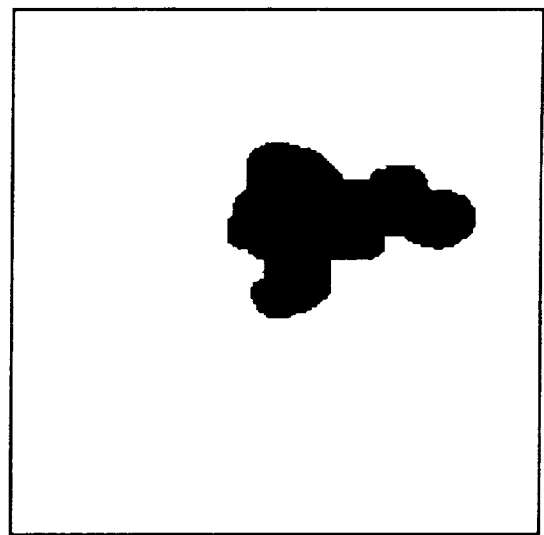
(a)



(b)



(c)



(d)

Figure 5(a) : image for testing, (b) : difference entropy image,
(c) : contrast image, and (d) : segmented image

6. References

1. D. Dubois and H. Prade, "Possibility Theory as a basis for preference propagation in automated reasoning", *Proceedings of the First IEEE Conference on Fuzzy Systems*, March 1992, San Diego, pp. 821-832.
2. D. Driankov, "Fuzzy logic with unless-rules", *Proceedings of the First IEEE Conference on Fuzzy Systems*, March 1992, San Diego, pp. 255-262.
3. T. O. Binford, "Survey of model-based image analysis systems", *International Journal of Robotics Research*, vol. 1, no. 1, Spring 1982, pp. 18-64.
4. D. M. McKeown, W. A. Harvey, and J. McDermott, "Rule-based interpretation of aerial imagery", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 7, no. 5, 1985, pp. 570-585.
5. J. Freeman, "The modeling of spatial relations", *Computer Graphics and Image Processing*, vol. 4, pp. 156-171, 1975.
6. B. Kosko, *Neural Networks and Fuzzy Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1992.
7. H. Takagi, "Fusion Technology of Fuzzy Theory and Neural Networks - Survey and Future Directions", *Proceedings of the International Conference on Fuzzy Theory and Neural Networks*, IIZUKA-90, pp. 13-17.
8. C. T. Lin, C. S. G. Lee, "Neural-network-based fuzzy logic control and decision system," *IEEE Trans. Computers*, vol. c-40, no. 12, Dec. 1991.
9. L. X. Wang, J. M. Mendel, "Fuzzy basis functions and orthogonal least squares learning," to appear in *IEEE Trans. Neural networks*, September 1992.
10. C. T. Lin, C. S. G. Lee, "Real-time Supervised Structure/Parameter Learning for fuzzy neural network," *Proceedings of the First IEEE International Conference on Fuzzy Systems*, San Diego, CA, March, 1992, pp. 1283-1291.

11. H. Nomura, I. Hayashi, and N. Wakami, "A learning method of fuzzy inference rules by descent method," *Proceedings of the First IEEE International Conference on Fuzzy Systems*, San Diego, CA, March, 1992, pp. 203-210.
12. D. G. Barkhart, P. P. Bonissone, "Automated fuzzy knowledge base generation and tuning," *Proceedings of the First IEEE International Conference on Fuzzy Systems*, San Diego, CA, March, 1992, pp. 179-188.
13. C. Lee, "Fuzzy logic in control systems: fuzzy logic controller - parts I & II," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-20, no. 2, pp. 404-435, 1990.
14. R. Krishnapuram and J. Lee, "Fuzzy-connective-based hierarchical aggregation networks for decision making," *Fuzzy Sets Syst.*, vol. 46, no. 1, pp. 11-27, Feb. 1992.
15. H. Dyckhoff and W. Pedrycz, "Generalized means as a model of compensation connectives," *Fuzzy Sets Syst.*, vol. 14, no. 2, pp. 143-154, 1984.
16. H.-J. Zimmermann and P. Zysno, "Decisions and evaluations by hierarchical aggregation of information," *Fuzzy Sets Syst.*, vol. 10, no. 3, pp. 243-260, 1983.
17. R. Krishnapuram and J. Lee, "Fuzzy-compensative-connective-based hierarchical networks and their application to computer vision," *The Journal of Neural Networks*, vol. 5, no. 2, March 1992.
18. R. Krishnapuram, J. M. Keller, and Y. Ma, "Quantitative Analysis of Properties and Spatial Relations of Fuzzy Image Regions", submitted to *NAFIPS-92*.
19. E. M. Riseman and A. R. Hanson, "A Methodology for the Development of General Knowledge-Based Vision Systems", in *Vision Systems and Cooperative Computation*, M. A. Arbib (Editor), MIT Press, Cambridge Ma, 1988.

Possibility Expectation and Its Decision Making Algorithm**James M. Keller and Bolin Yan**Electrical and Computer Engineering
University of Missouri
Columbia, MO 65211, USA

52-61

143 401

p-16

Abstract

The fuzzy integral has been shown to be an effective tool for the aggregation of evidence in decision making. Of primary importance in the development of a fuzzy integral pattern recognition algorithm is the choice (construction) of the measure which embodies the importance of subsets of sources of evidence. Sugeno fuzzy measures have received the most attention due to the recursive nature of the fabrication of the measure on nested sequences of subsets. Possibility measures exhibit an even simpler generation capability, but usually require that one of the sources of information possess complete credibility. In real applications, such normalization may not be possible, or even desirable. In this report both the theory and a decision making algorithm for a variation of the fuzzy integral are presented. This integral is based on a possibility measure where it is not required that the measure of the universe be unity. A training algorithm for the possibility densities in a pattern recognition application is also presented with the results demonstrated on the shuttle-earth-space training and testing images.

1. Introduction

Decision making is a basic problem in science, engineering, and even in daily life. There are often conflicting requirements of low error rates and minimum computation time to reduce the cost. The purpose of this paper is to propose the concept of possibility expectation via the possibility integral as a decision making scheme, which can be used to construct optimal decision making algorithms. A possibility expectation is a value of nonlinear integration of two pieces of information, namely, an evidence function $h(x)$ and a possibility measure $\text{Pos}(\cdot)$. A possibility measure is a monotonic set function with the property that the measure of the universe X can be less than or equal to unity.

An example of possibility expectation is the following: In the court room, although the witnesses for both the defendant and plaintiff promise that they will tell the truth, the judge still needs to assign the grade of credibility (possibility densities) to each person to evaluate what the person says (evidence). The judge will integrate what each group of witnesses said with his belief in that group's credibility (possibility measure). Then the judge makes his decision

(possibility expectation).

In multicriteria decision making, as can be found in most pattern recognition problems, the value of each source of information (and thus all subsets of sources) toward each alternative can be different. For example, "greenness" may be a very important feature for recognizing certain types of trees in an image; whereas it may be quite unimportant as a feature for a roof of a building. This difference in the importance or credibility of subsets of information sources will be encoded in a possibility measure. The degree to which a given image region is green, to continue the example, is objective evidence supplied by the information source. After collecting all such objective information, it is the job of the decision making algorithm to fuse the objective evidence together with the worth of the sources. In our methodology, this will be accomplished by utilizing the possibility integral, a variation of the fuzzy integral [1].

The particular possibility measures which we describe generalize fuzzy measures in that it is not required that the measure of the entire domain of discourse be one. In a pattern recognition problem, it may not be possible, or may not be desirable to force one of the sources of information to have "perfect credibility". By relaxing this requirement, not only do we match real situations better, we also provide the opportunity to create better decision making algorithms, as we shall see later.

For a pattern recognition environment, a method to learn the possibility densities (values upon which the measure is generated) from training data is given. The results of the subsequent algorithm are used to segment a shuttle from the earth and space background.

2. Possibility Measures and Possibility Integral

Definition 2.1 A set function $\text{Pos}(\cdot): 2^X \rightarrow [0, 1]$ is called a possibility measure if it satisfies the following properties:

- (1) $\text{Pos}(\emptyset) = 0$, $\text{Pos}(X) \leq 1$.
- (2) If $A, B \in 2^X$ and $A \subset B$, then $\text{Pos}(A) \leq \text{Pos}(B)$.
- (3) $\text{Pos}(\bigcup_{j=1}^n A_j) = \sup_{\alpha \in [0,1]} [\text{Pos}(A_j)]$.

Note: If X is finite, a possibility measure is not a fuzzy measure when $\text{Pos}(X) < 1$; it is the same as fuzzy measure only when $\text{Pos}(X) = 1$. If X is infinite, a possibility measure is not a fuzzy measure in general [2]. Puri and Ralescu [3] give two counterexamples which show that, even in "nice" cases, a possibility measure is not a fuzzy measure in the infinite case.

Definition 2.2 Let $X = \{x_j \mid j = 1, \dots, n\}$ be a finite set and let **Pos** be a possibility measure on 2^X . The set $\{p^j = \text{Pos}(\{x_j\}) \mid j = 1, \dots, n\}$ is called the set of possibility densities for **Pos**.

By definition of the possibility measure, it is clear that the measure of any subset A of X can be generated by

$$\text{Pos}(A) = \max_{j \in A} \{p^j\},$$

and hence, a possibility measure is easily generated by its densities.

We note that possibility theory can be induced not only from the nested bodies of evidence within the Dempster-Shafer theory [4], but also from the fuzzy sets introduced by Zadeh [6]. A fuzzy set F is a set whose elements are characterized by the membership grade function $\mu_F(x): X \rightarrow [0, 1]$. A value of $\mu_F(x)$ expresses the grade of membership that an element $x \in X$ belongs to the fuzzy subset F of X . Let $\pi_F(x) = \mu_F(x)$ be a possibility distribution induced by a fuzzy set F . In general, a possibility distribution is thought of as an elastic restriction on the values within a domain of discourse which a fuzzy variable may assume [5]. The fuzzy set F provides the meaning of the restriction. A possibility measure is defined as $\text{Pos}(A) = \sup_{x \in A} [\pi_F(x)]$ for all $A \in 2^X$. This relationship holds also for non-normal fuzzy sets [6]. Although a fuzzy set and a possibility distribution have a common mathematical expression, the underlying concepts are different [5].

Our possibility measures are non-normalized generalizations of what are referred to as S-decomposable measures [7, 8], these being a class of fuzzy measures which are easily computable.

Definition 2.3 Let $h(x)$ be a function such that $h: X \rightarrow [0, 1]$, and let **Pos**(\cdot) be a possibility measure of 2^X . The possibility integral or the possibility expectation of $h(x)$ with respect to **Pos**(\cdot) is defined as

$$\int_X h(x) \circ \text{Pos}(\cdot) = \sup_{\alpha \in [0, 1]} [\alpha \wedge \text{Pos}(A_\alpha)], \quad \text{where } A_\alpha = \{x \mid h(x) \geq \alpha\}.$$

When $X = \{x_i \mid i = 1, \dots, n\}$ is finite, if we reorder X such that $h(x_1) \geq h(x_2) \geq \dots \geq h(x_n)$, then the possibility integral can be written as

$$\int_X h(x) \circ \text{Pos}(\cdot) = \bigvee_{j=1}^n [h(x_j) \wedge \text{Pos}(A_j)], \quad \text{where } A_j = \{x_1, x_2, \dots, x_j\}.$$

The rationale of the possibility expectation is to find the source within the universe where both the information value $h(x_j)$ and the possibility measure $\text{Pos}(A_j)$ are compatibly large, that is, where the feasibility of the data and the reliability of a subset of sources is jointly optimal.

The fuzzy integral developed by Sugeno [1] has the same formulation with the exception that a fuzzy measure is used in lieu of the possibility measure. One of the advantages of the possibility integral is that the measures $\text{Pos}(A_j)$ are easily calculated from the densities by the recursive relationship

$$\begin{aligned}\text{Pos}(A_1) &= \text{Pos}(\{x_1\}) = p^1; \\ \text{Pos}(A_j) &= \text{Pos}(A_{j-1} \cup \{x_j\}) = \text{Pos}(A_{j-1}) \vee p^j.\end{aligned}$$

In contrast, for Sugeno fuzzy measure g_λ with the fuzzy densities $\{g^1, \dots, g^n\}$, this recursive definition becomes

$$\begin{aligned}g_\lambda(A_1) &= g_\lambda(\{x_1\}) = g^1; \\ g_\lambda(A_j) &= g_\lambda(A_{j-1} \cup \{x_j\}) = g_\lambda(A_{j-1}) + g^j + \lambda g^j g_\lambda(A_{j-1}),\end{aligned}$$

where $\lambda \geq -1$ [1, 10, 11]. The value of λ must be calculated from the equation

$$\prod_{i=1}^n (1 + \lambda g^i) = 1 + \lambda, \quad [1].$$

If one is going to try to learn a measure (iteratively) from training data, the amount of computations necessary to learn a possibility measure, and then evaluate its possibility integral is considerably less than that required for a Sugeno fuzzy measure and its fuzzy integral.

For a multiclass pattern recognition problem (or any multicriteria decision making problem), the set \mathbf{X} represents sources of information (criteria). Each class (alternative) will have its own evidence function $h_i: \mathbf{X} \rightarrow [0, 1]$ to assess the feasibility that the decision is class i (alternative i) from the standpoint of each individual source, x_j . Also, each class will have its own possibility measure Pos_i which determines the worth of all subsets of sources in deciding that a particular object belongs to class i . Finally, the collection of possibility integrals

$$e_i = \int_{\mathbf{X}} h_i(x) \circ \text{Pos}_i(\cdot),$$

gives a class-individualized "fusion" of the direct evidence with the worth of that evidence. A final crisp decision can be made from the possibility expectations (integral values), for example, pick the class corresponding to the maximum possibility expectation. Alternately, these expectation values can be used as confidences for later processing.

3. Properties of The Possibility Integral

Several interesting properties of the possibility integral are proved in [11]. Of particular

interest to the algorithm presented in the next section are the following two results.

Theorem 3.1 $0 \leq \int_X \mathbf{h}(x) \circ \mathbf{Pos}(\cdot) \leq \mathbf{Pos}(X).$

Theorem 3.2 If $\mathbf{h}_1(x) \leq \mathbf{h}_2(x) \forall x$:

$$\begin{aligned} \int_X \mathbf{h}_1(x) \circ \mathbf{Pos}(\cdot) &\leq \int_X \mathbf{h}_2(x) \circ \mathbf{Pos}(\cdot), & \text{if } \mathbf{Pos}(X) > \mathbf{h}_1(x) \text{ for all } x, \\ \int_X \mathbf{h}_1(x) \circ \mathbf{Pos}(\cdot) &= \int_X \mathbf{h}_2(x) \circ \mathbf{Pos}(\cdot), & \text{if } \mathbf{Pos}(X) \leq \mathbf{h}_1(x) \text{ for all } x. \end{aligned}$$

4. Decision Rule and Training Algorithm

In the procedure given below, we consider a two class pattern recognition problem, or a two alternative decision process. The approach can be extended directly to multiple classes, but from the particular structure of the training mechanism, it would be more appropriate to view it as a series of two class problems, either as pairwise distinctions, or as each class against all of the remaining classes. Since the possibility integral algorithm does not create geometric decision boundaries in feature spaces (as, for example, Bayes Decision Theory), the second approach is reasonable and contains fewer subdecisions which need to be made to extend this to multiple classes.

The actual decision algorithm utilizes the nature of the possibility integral to split the input objects (as represented by the evidence function $\mathbf{h}(x)$) into four groups to reduce the computational load. The first two groups deal with the case where the strength of all objective evidence for one class outweighs that for the other. In most cases, this corresponds to the fact that, in a pattern recognition problem, a majority of the data are easily distinguished (being quite typical of their class). Decision rules 1 and 2 below are a consequent of Theorem 3.2 assuming that the possibility measures for both classes in this case are identical. Of course, there are problems where the objective evidence for one class can dominate that for the other class, and yet, the object belongs to the latter. This could happen if the worth of the source, i.e., the densities, are vastly different between classes. During training, this condition is monitored, and if the training data produce such outcomes, the first two rules are abandoned, forcing all training samples to be "conflict data".

The initial definition of "conflict" is an object where the evidence function for one class does not dominate that of the other. In this case, we split the training data (and also the unknown test objects) into two subgroups based on the class receiving the highest degree of support from any source. For each group, two possibility measures are formed which minimize the total misclassification of the training data. The purpose of partitioning the data in this

manner is to reduce the size of the training set since our initial training scheme is a complete search through a quantized set of all pairs of density functions. To reduce further the amount of computations, we note that the value of a possibility integral cannot be larger than the maximum of the function being integrated. This fact allows us to restrict the range of density values to be no larger than the maximum evidential support in the training set. (Reducing the training sets gives more opportunity to invoke this restriction). Optimal pairs of density functions (in term of minimal error rate on the training data) are formed and then used in the testing cycle. There are 4 possibility measures generated during training - one from each class in each of the two subgroups of conflict data.

The decision algorithm is summarized below.

BEGIN

FOR each feature data vector **DO** obtain $h_1(x_j)$ for all j and $h_2(x_j)$ for all j ;

(1) **IF** $h_1(x_j) > h_2(x_j)$ for all j , **THEN** the feature data vector belongs to class 1.

(2) **ELSE IF** $h_1(x_j) < h_2(x_j)$ for all j , **THEN** the feature data vector belongs to class 2.

(3) **ELSE**

If $\bigvee_{j=1}^n h_1(x_j) \geq \bigvee_{j=1}^n h_2(x_j)$, **Then**

$$e_1 = \bigvee_{j=1}^n [h_1(x_j) \wedge \text{Pos}_{11}(A_j)], \quad e_2 = \bigvee_{j=1}^n [h_2(x_j) \wedge \text{Pos}_{12}(A_j)]$$

Else

$$e_1 = \bigvee_{j=1}^n [h_1(x_j) \wedge \text{Pos}_{21}(A_j)], \quad e_2 = \bigvee_{j=1}^n [h_2(x_j) \wedge \text{Pos}_{22}(A_j)]$$

End If

If $e_1 > e_2$, **Then** the feature data vector belongs to class 1,

Else the the feature data vector belongs to class 2.

End If

END IF

END FOR

END .

5. Experimental Results

Two shuttle-earth-space intensity images were used in the experiment, in which all the data from the two images were treated as "conflict data" and hence only the third decision rule applies.

The training image is shown in Fig. 5.4 and the test image is shown in Fig. 5.5. Three texture feature images (contrast, difference, and the entropy) were derived from the training

and the test images respectively, i.e., three feature images for training and three feature images for testing (For the definition of these features, please see section on membership generation techniques in this report). The three feature images, used for training the possibility densities, are shown in Fig. 5.2. The three feature images used in testing are shown in Fig. 5.6.

The possibility distribution (or membership function) of each class in each feature, that used to generate the evidential function $h(x)$, is determined by using the possibilistic clustering algorithm on the histograms of each class in each feature, which is described in another section of this report.

While training, the possibility densities were determined with the "perceptron criterion" (i.e., minimize the decision error) from the feature images in Fig. 5.2. The segmentation result corresponding to the possibility measure(s) for the training image is shown in Fig. 5.3, in which the shuttle and its background are clearly segmented, except that the shuttle body seems disconnected. To improve the connection of the shuttle body, the possibility densities of the shuttle were raised slightly, from which the segmentation result in Fig. 5.4 and the result in Fig. 5.7 (for the test case) were obtained. These results can be improved quite easily with a shrink-expand operation.

6. Conclusion

In this paper, a decision making algorithm based on a variation of the fuzzy integral was proposed. The possibility integral has a particularly simple generation capability. The algorithm was run on the shuttle-earth-space images, reasonable good results were obtained.

7. References

- [1] M. Sugeno, "Theory of Fuzzy Integrals and its Applications", *Ph.D.Thesis*, Tokyo Institute of Technology, 1974.
- [2] A. Kandel, *Fuzzy Mathematical Techniques with Applications*, Addison-Wesley: Reading, MA, 1986.
- [3] M. Puri and D. Ralescu, "A Possibility Measure is Not a Fuzzy Measure", *Fuzzy Sets and Systems*, Vol. 9, No. 3, pp. 219-227.
- [4] G. Shafer, *A Mathematic Theory of Evidence*, Princeton University Press: Princeton, NJ, 1976 .
- [5] L. A. Zadeh, "Fuzzy Sets as a Basis for a Theory of Possibility", *Fuzzy Sets and Systems*, Vol. 1, No. 1, 1978, pp. 3-28.
- [6] D. Dubois and H. Prade, *Possibility Theory*, Plenum Press, 1986, pp. 9.
- [7] S. Weber, "Two Integrals and Some Modified Versions- Critical Remarks", *Fuzzy Sets and Systems*, Vol.20, 1986, pp 97-105.
- [8] H. Tahani, "Generalized Fuzzy Integral in computer vision", Ph.D. Thesis, University of Missouri-Columbia, 1991.
- [9] B. Yan and J. Keller, "Conditional Fuzzy Measure and Image Segmentation", *Proceedings of North American Fuzzy Information Processing Society* Columbia, Missouri, USA, 1991, pp. 32-36.
- [10] B. Yan, "Fuzzy Integral at $\lambda = -1$ ", *Proceedings of North American Fuzzy Information Processing Society*, Columbia,

Missouri, USA, 1991, pp. 18- 22.

- [11] B. Yan and J. Keller, "Possibility integral", *Academic Research Report* University of Missouri-Columbia, 1991.

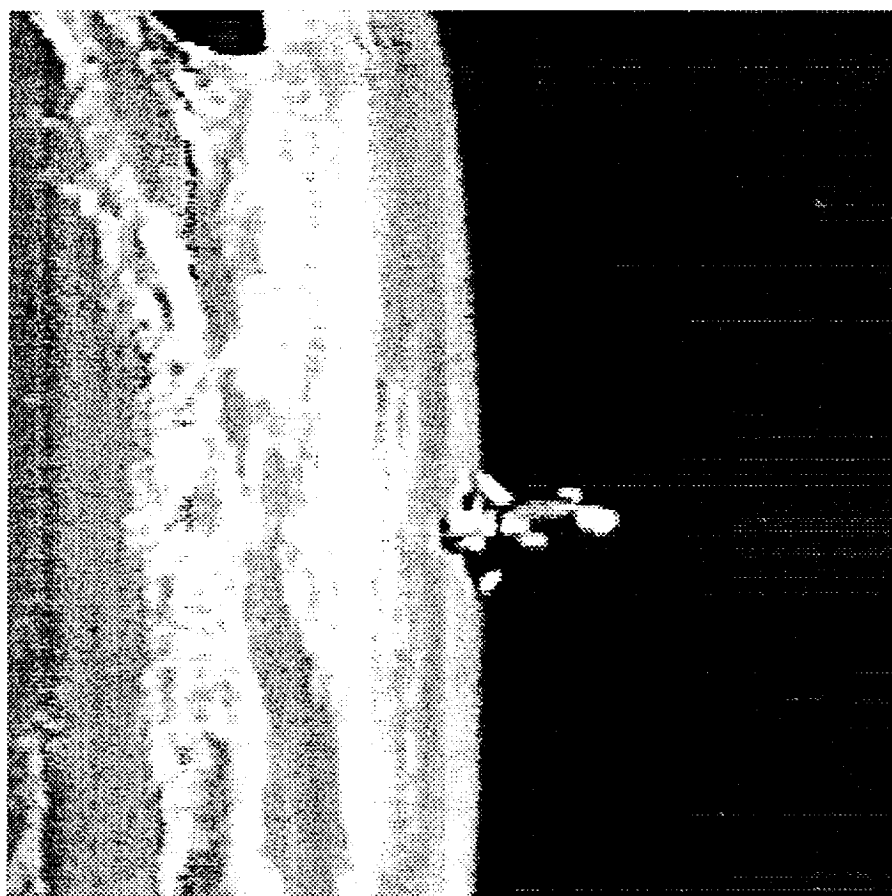


Fig 5.1 Intensity training image.

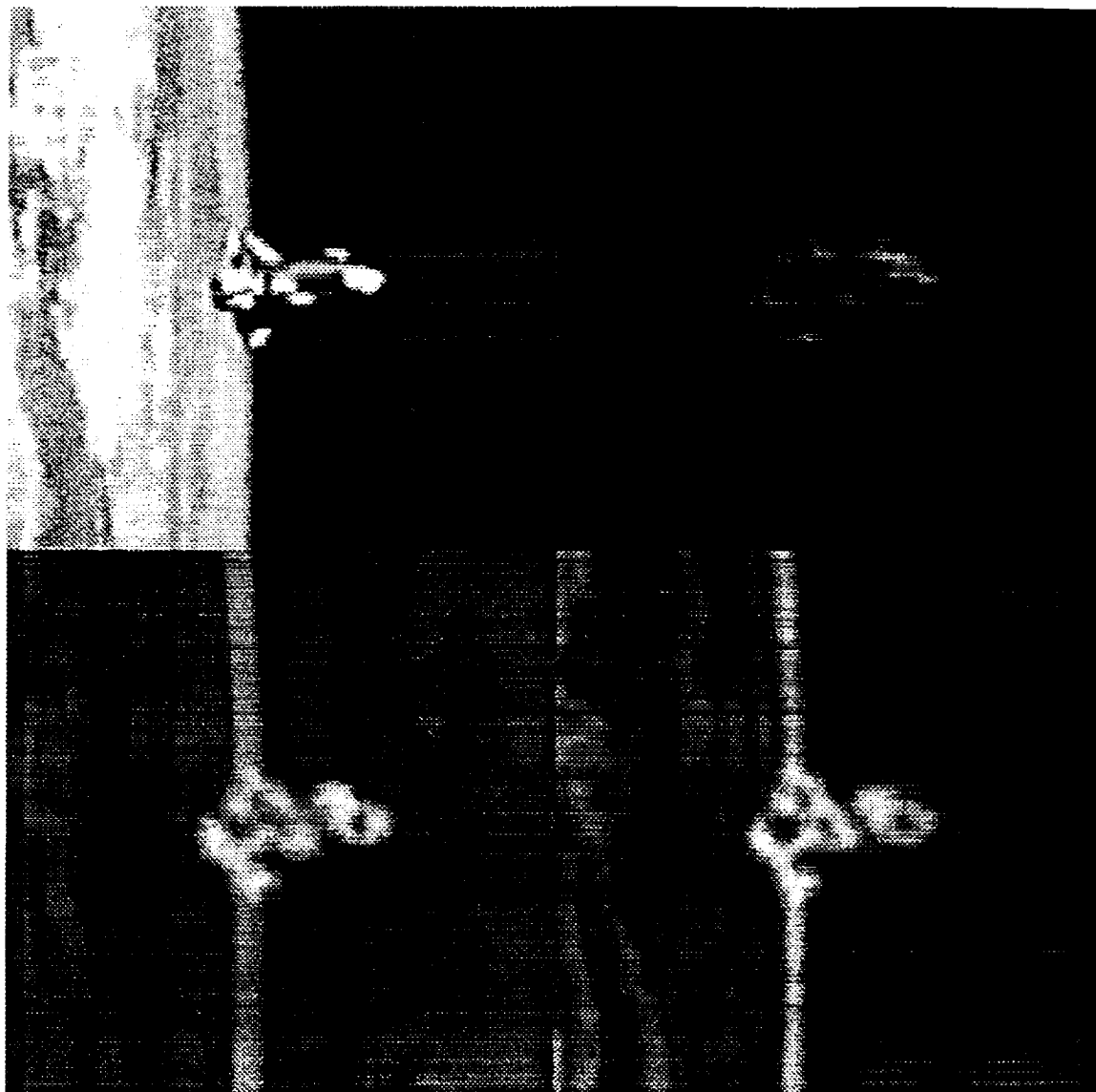


Fig 5.2 (top left) Intensity training image.
(top right) Contrast feature image.
(bottom left) Difference feature image.
(bottom right) Entropy feature image.

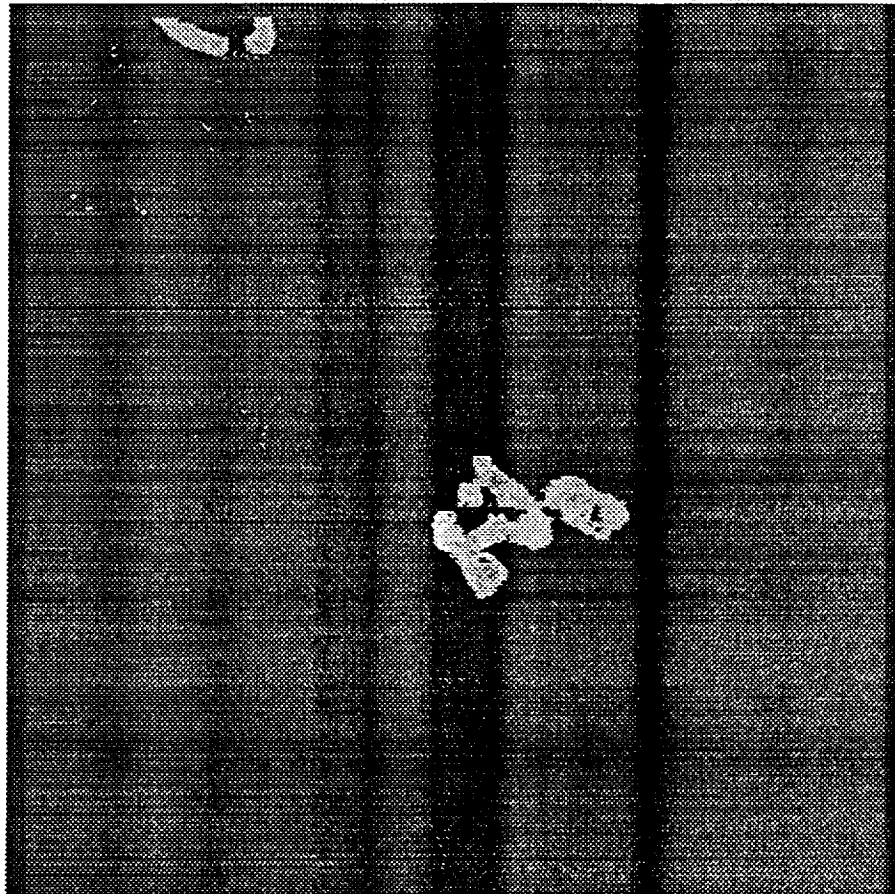


Fig 5.3 Segmented image1 using the possibility integral algorithm.

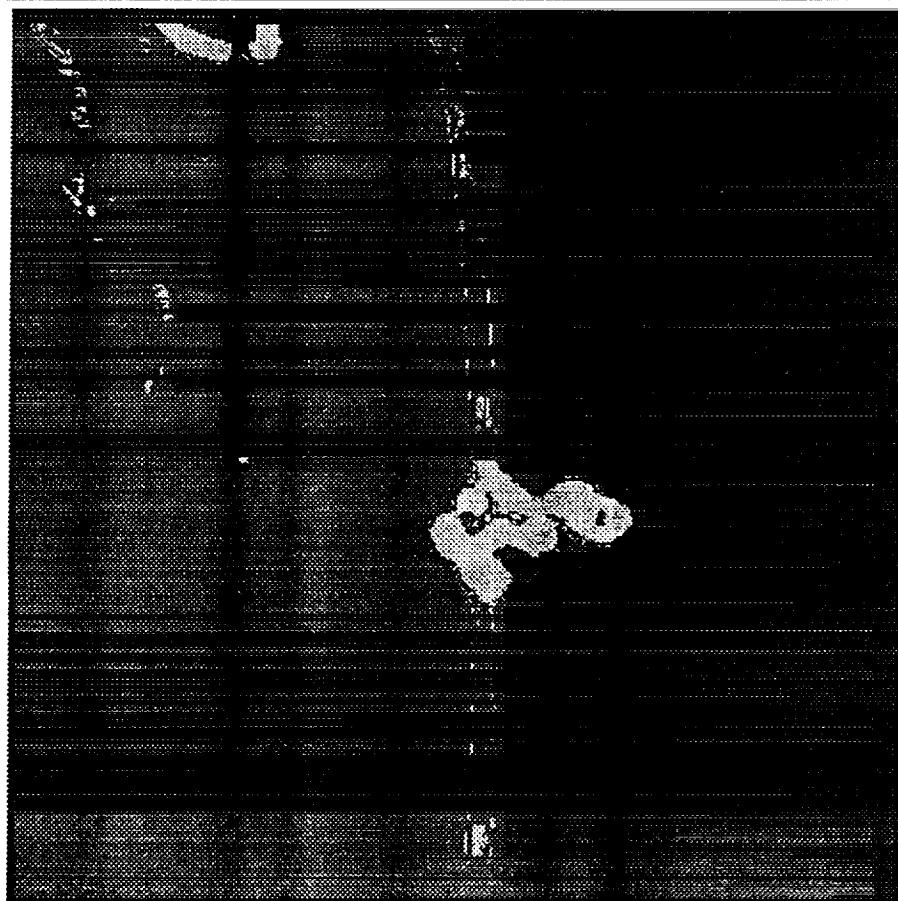


Fig 5.4 Segmented image2 using the possibility integral algorithm.

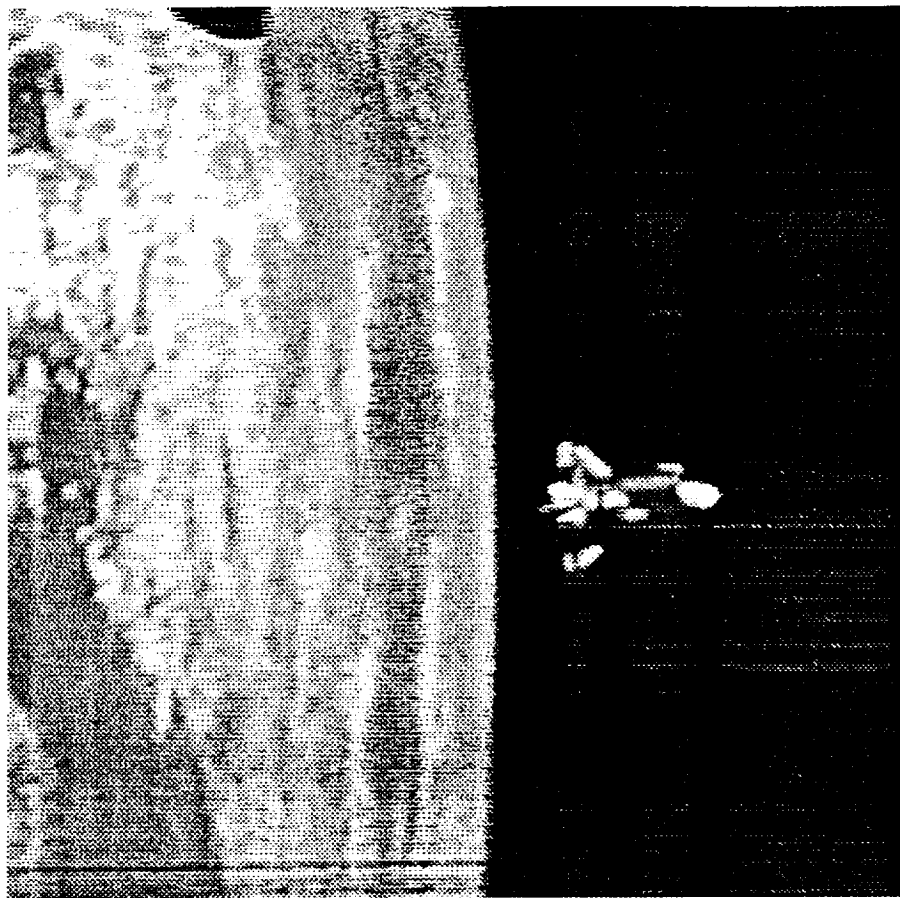


Fig 5.5 Intensity testing image.

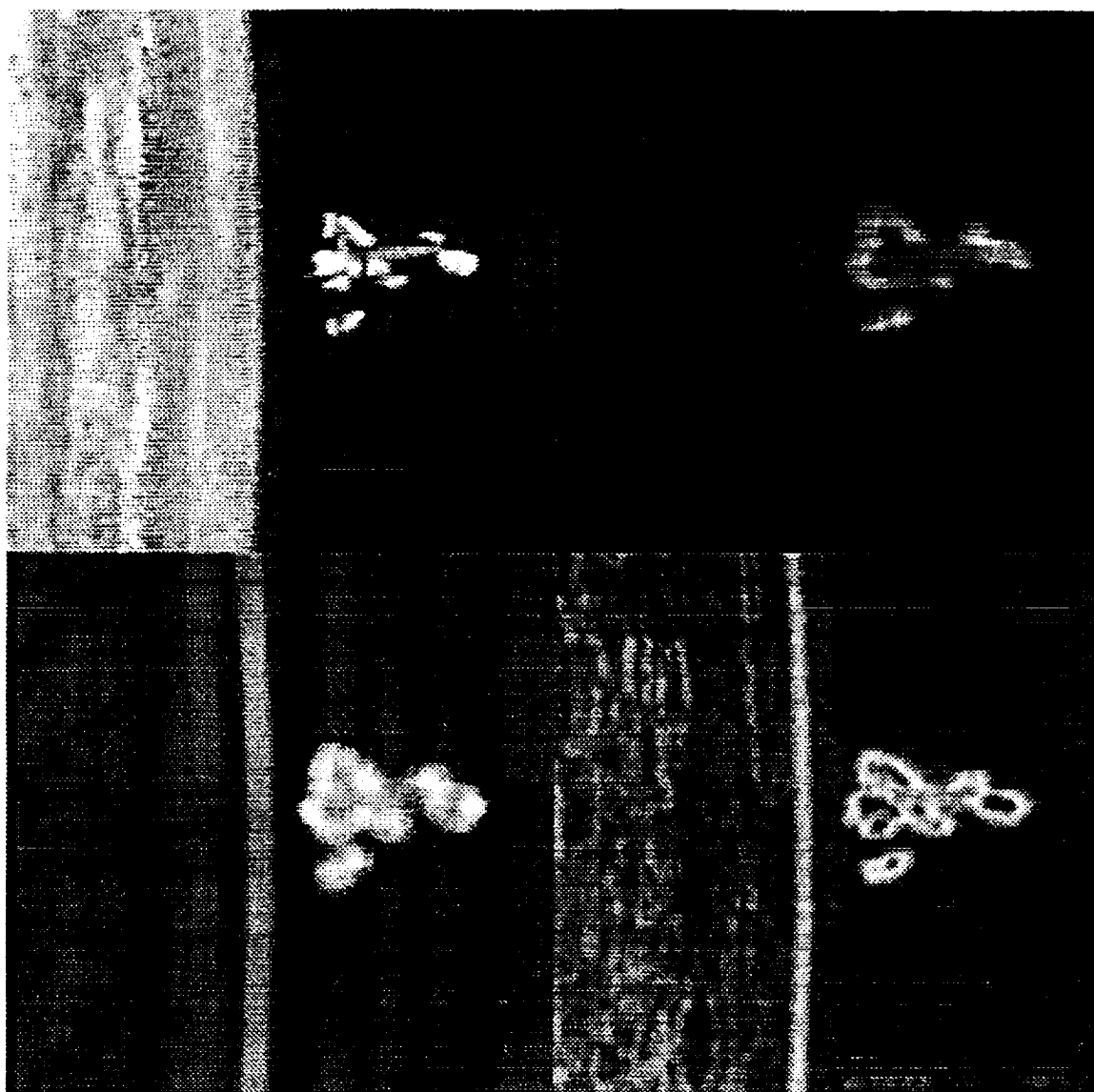


Fig 5.6 (top left) Intensity testing image.
(top right) Contrast feature image.
(bottom left) Difference feature image.
(bottom right) Entropy feature image.

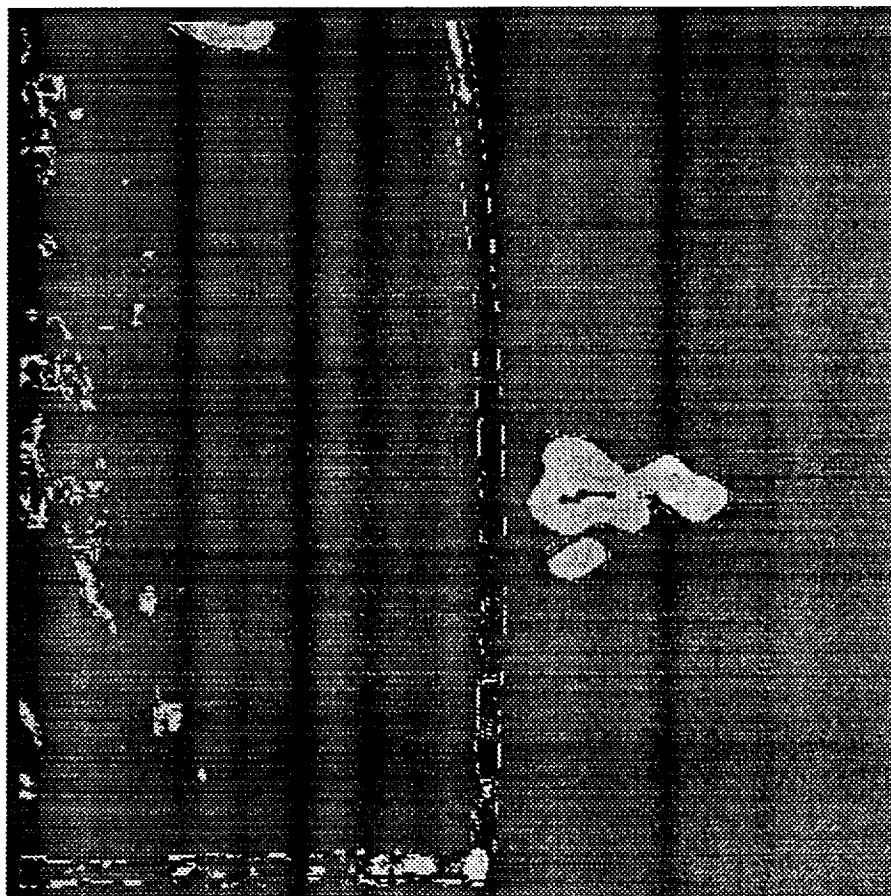


Fig 5.7 Segmented testing image using the possibility integral algorithm.

Calculation of Membership Functions

Our work in this area has progressed nicely. We have designed and implemented a new algorithm to generate membership values from a set of training data using a multi-layer neural network. This is in addition to the progress we made in the transformation of "probability density functions" into possibility distributions for use in assigning membership values to individual points as reported in the third quarter report.

Membership Generation Using Multilayer Neural Network

Jaeseok Kim
University of Missouri
Columbia, Mo 65211

53-61

143402

p. 12

There has been intensive research in neural network applications to pattern recognition problems. Particularly, the back-propagation network has attracted many researchers because of its outstanding performance in pattern recognition applications. In this section, we describe a new method to generate membership functions from training data using a multilayer neural network. The basic idea behind the approach is as follows. The output values of a sigmoid activation function of a neuron bear remarkable resemblance to membership values. Therefore, we can regard the sigmoid activation values as the membership values in fuzzy set theory. Thus, in order to generate class membership values, we first train a suitable multilayer network using a training algorithm such as the back-propagation algorithm. After the training procedure converges, the resulting network can be treated as a membership generation network, where the inputs are feature values and the outputs are membership values in the different classes.

This method allows fairly complex membership functions to be generated because the network is highly nonlinear in general. Also, it is to be noted that the membership functions are generated from a classification point of view. For pattern recognition applications, this is highly desirable, although the membership values may not be indicative of the degree of typicality of a feature value in a particular class.

A. Typical Example

In this section we show an example of a membership network that can generate membership values for "shuttle" and "background". The network we used had one input unit, eight hidden units and two output units. Input data to the network were feature values and the observed activation values of the outputs after the network was trained with the back-propagation algorithm were considered as the degree of belonging to the particular classes. In this experiment, there were only two classes: object (shuttle) and background (space and earth). The training image is shown in Fig 1.

We generated membership functions corresponding to four texture features. These four feature images are shown in Fig 2. These features were contrast, difference, entropy, difference entropy, and homogeneity. They are defined by

$$\text{Contrast} = \sum_{n=0}^{N_g-1} n^2 \left\{ \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j) \right\}.$$

$$\text{Entropy} = - \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j) \log (p(i,j))$$

$$\text{Difference Entropy} = - \sum_{k=1}^{N_g} p_{x-y}(k) \log (p_{x-y}(k))$$

$$\text{Homogeneity} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} \frac{1}{1+(i-j)^2} p(i,j)$$

where $p(i,j)$ is the (i,j) -th entry in the spatial gray level dependence matrix, and N_g is the number of gray levels. Also, $p_{x-y}(k)$ is defined by

$$p_{x-y}(k) = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j) \text{ such that } |i-j|=k$$

(See [1,2] for details.)

All feature values were normalized to lie between 0 and 255. The training sets were formed by manually picking samples from the object and background regions of all four texture feature images. There were 100 samples for each class. After the network was trained, we fed gray values (0-255) to the input unit and collected the activation values of output units to generate the membership functions. Fig 3.1 and Fig 3.2 show the histograms of the features for the background and the object, and the corresponding membership values for all four features.

B. Discussion

Fig 3.1 (c) shows the membership functions of object and background for contrast feature. The membership functions are very steep because only one or two gray level values overlap between the histograms of the background and the object. On the other hand, Fig 3.2 shows broader membership functions because of a broader overlapping area between the histograms for the entropy and homogeneity features. An interesting observation is that when histograms of object and background overlap, the network sets the

crossover point at the middle of the overlapping area. This reveals the nice membership generation capability of the neural network.

C. Conclusion

This heuristic method of generating membership function has some merits compared to the probability-possibility transformation method described in our third quarterly report. The transformation method requires a precise estimation of a probability density function. In practice, this is difficult to achieve when the number of training samples is small. Also the resulting shape of the membership function is almost the same as the probability density function. In order words, membership functions generated by these methods seem to have a frequency interpretation of the data. Fig 4 and Fig 5 show examples of the transformation based membership functions obtained with 1,000 samples per feature per class. Even with this high number, the functions are rather noisy.

One short coming of this heuristic method is that the memberships do not represent "typicality". However, if the memberships are to be used subsequently in a pattern recognition algorithm then this method will provide better classification results.

REFERENCES

1. R. W. Connors and C. A. Harlow, "A Theoretical Comparison of Texture Algorithms", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 2, No. 3, May 1980, pp. 204-222.
2. R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural Features for Image Classification", IEEE Transactions on SMC, Vol. 3, No. 6, Nov. 1973, pp. 610-621.

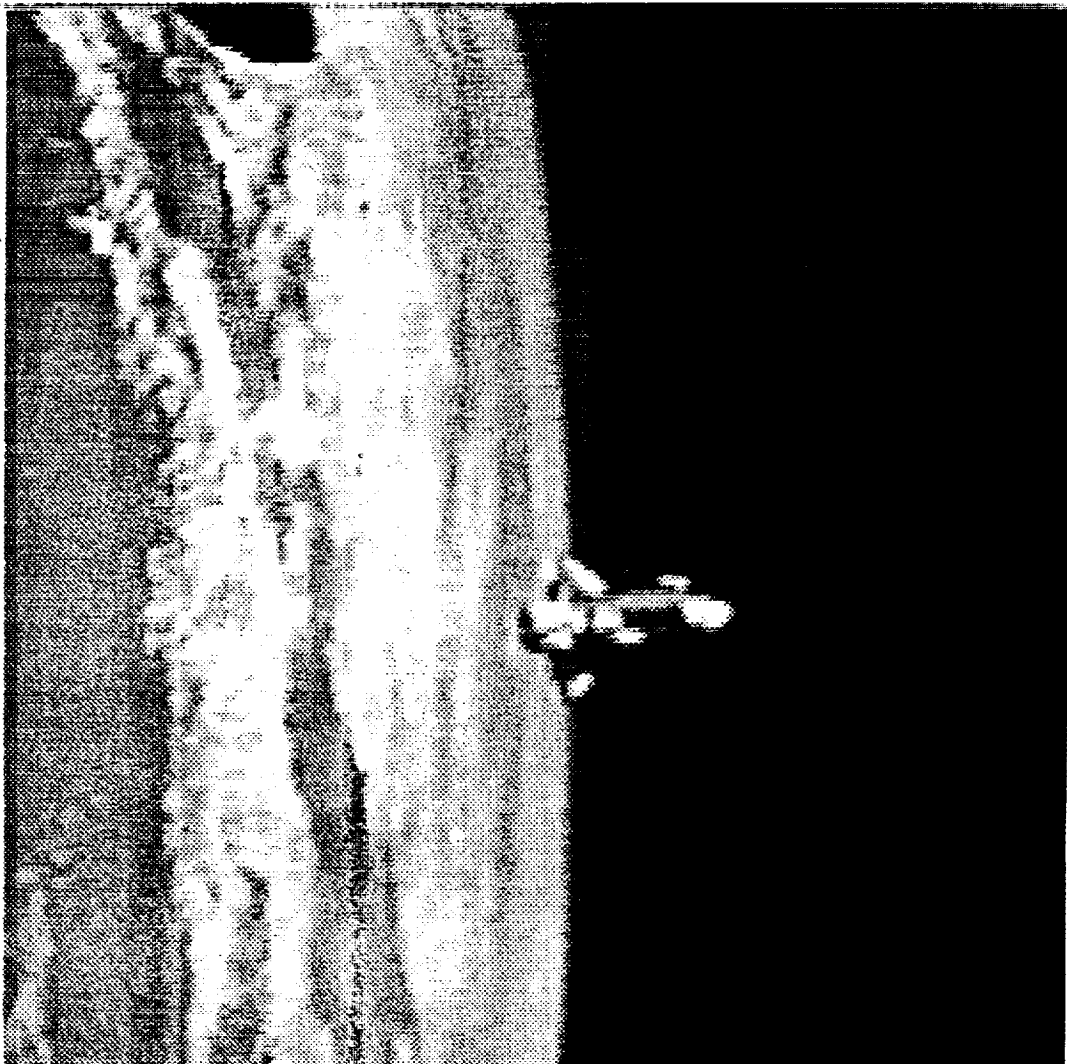


Fig.1 Space shuttle image

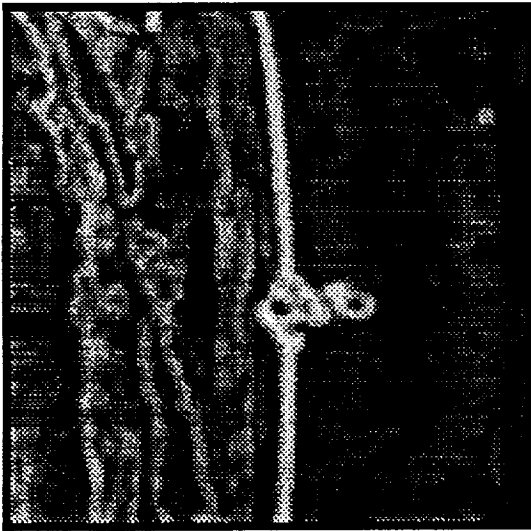
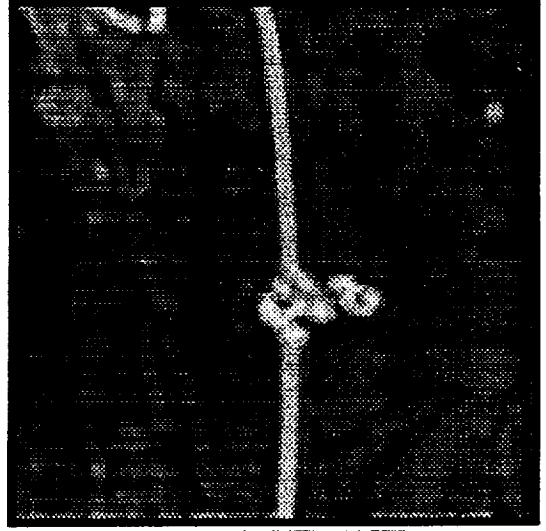
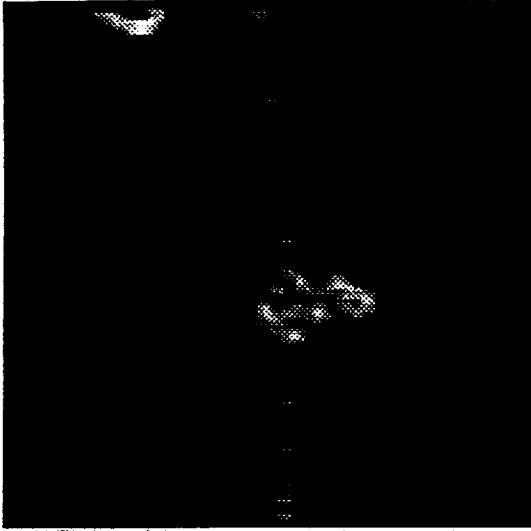
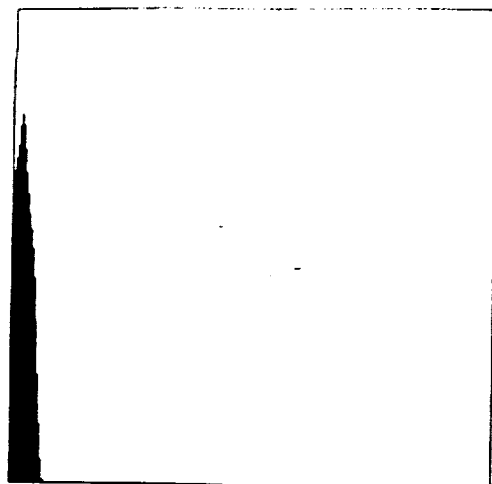
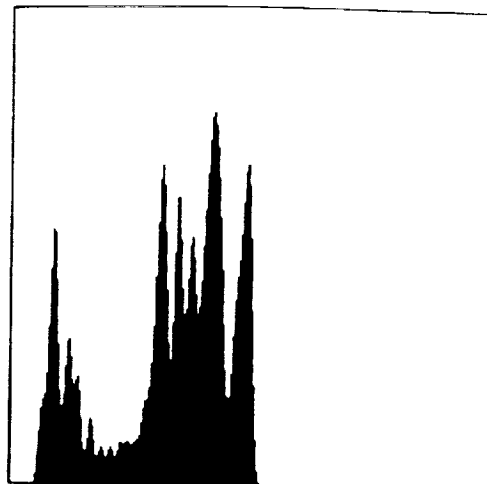


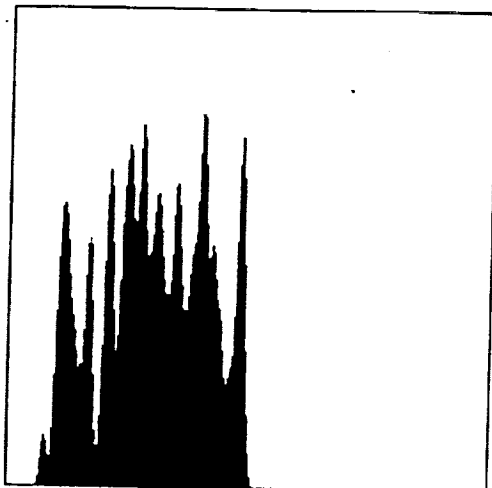
Fig 2. text features : in clockwise, contrast, difference entropy, entropy, and homogeneity.



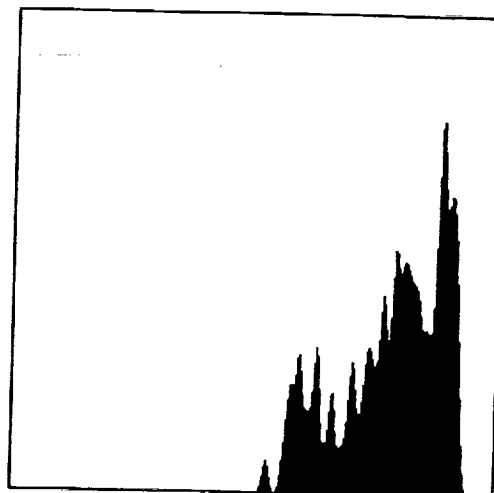
(a) hist. of background(contrast)



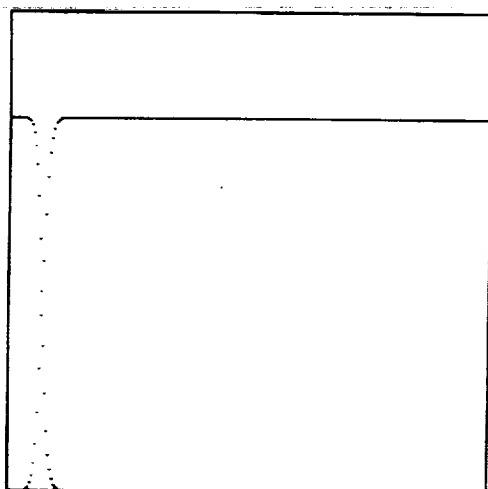
(d) hist. of background(diff. entropy)



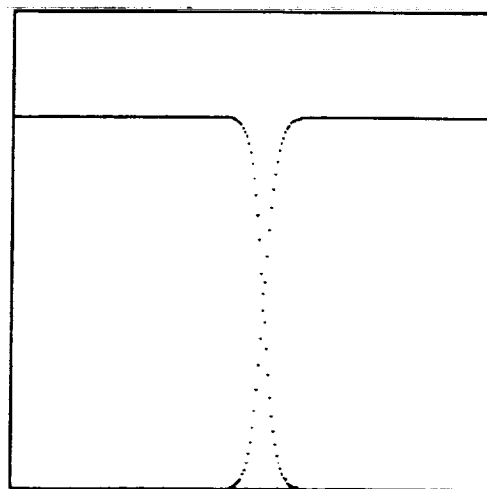
(b) hist. of object(contrast)



(e) hist. of object(diff. entropy)

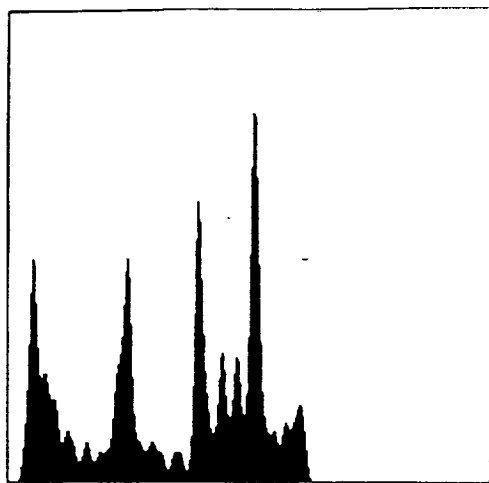


(c) membership fun(contrast)

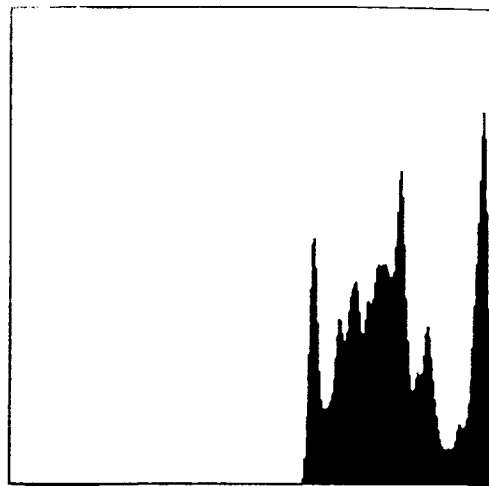


(f) membership fun(diff. entropy)

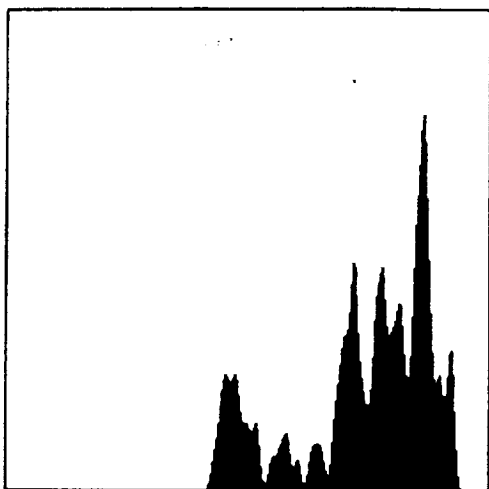
Fig 3.1 Histogram of background and object, and corresponding membership function.



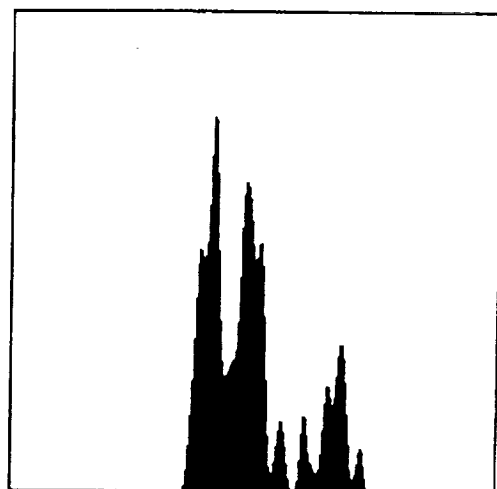
(a) hist. of background(entropy)



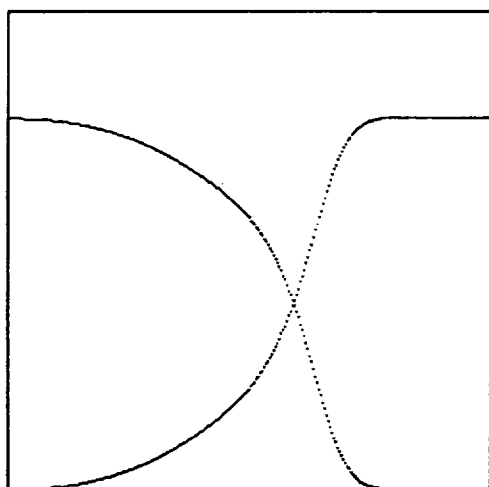
(d) hist. of background(homogeneity)



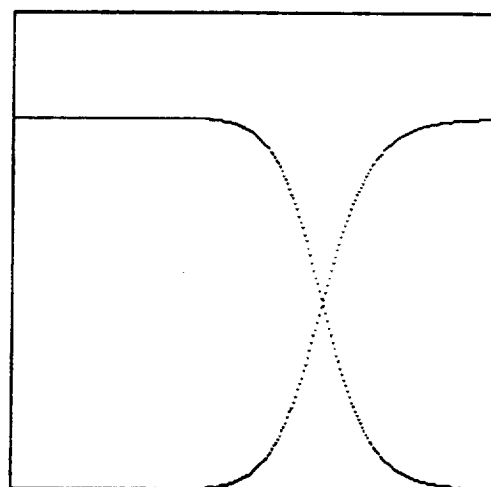
(b) hist. of object(entropy)



(e) hist. of object(homogeneity)

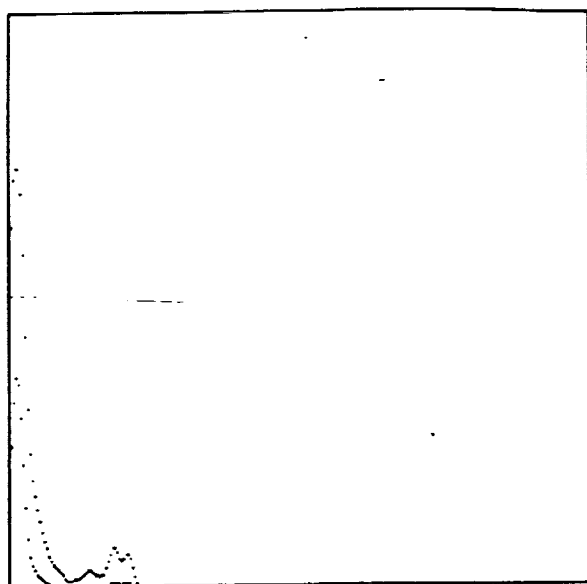


(c) membership fun(entropy)

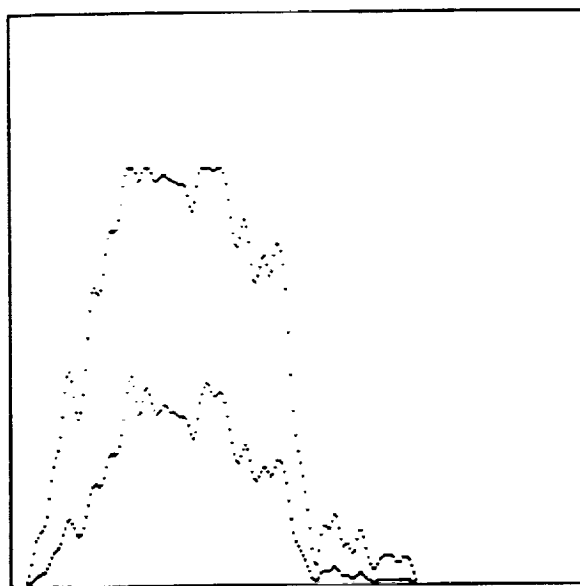


(f) membership fun(homogeneity)

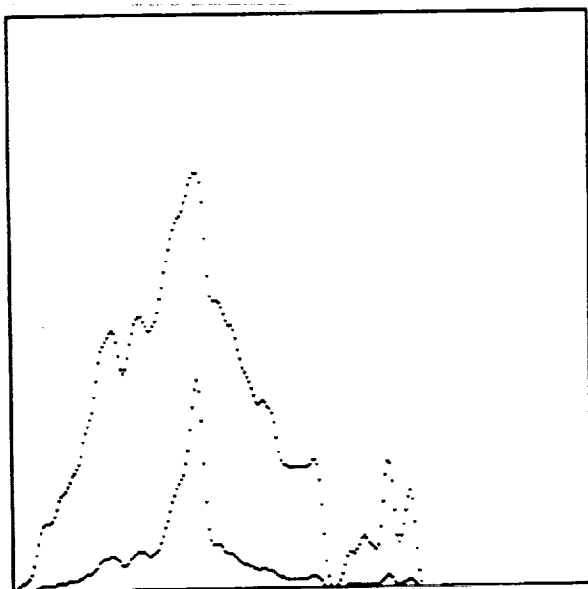
Fig 3.2 Histogram of background and object, and corresponding membership function.



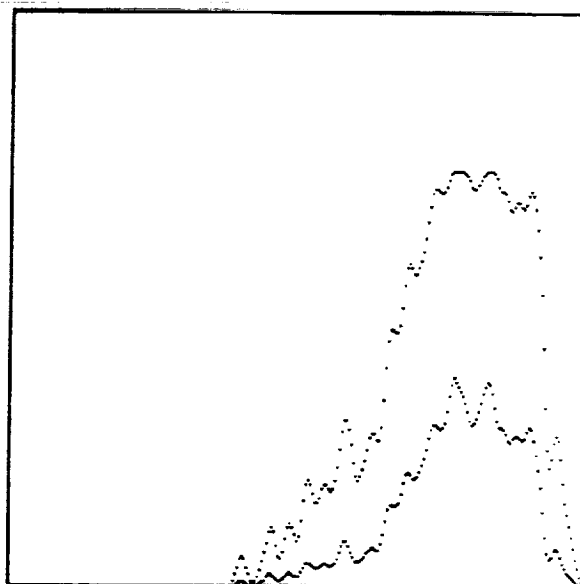
(a) background(contrast)



(b) object(contrast)

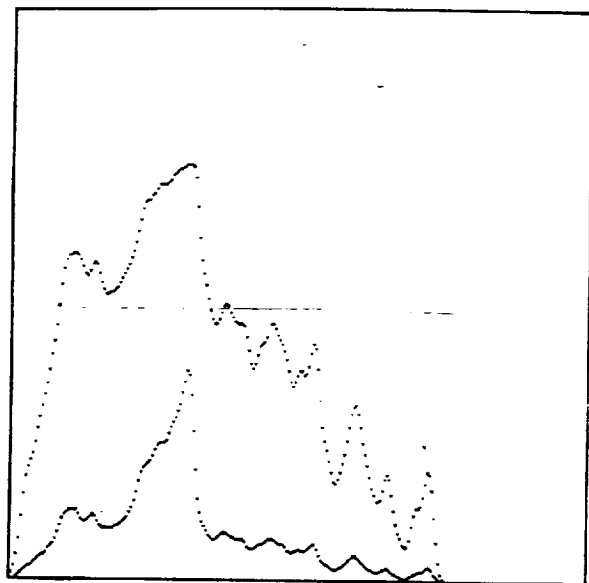


(c) background(diff. entropy)

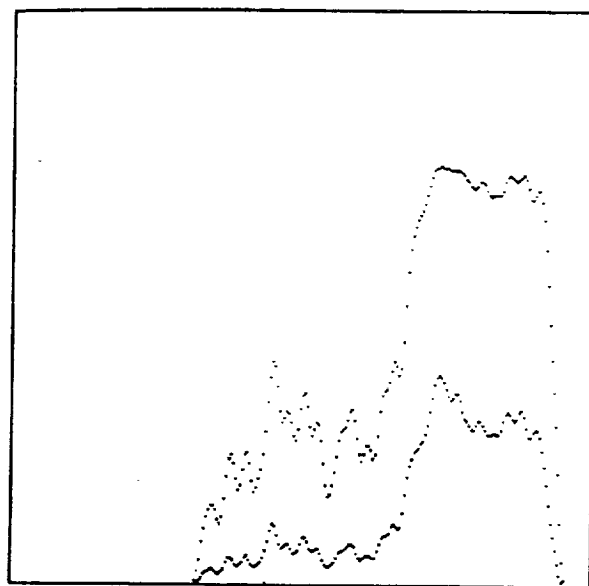


(d) object(diff. entropy)

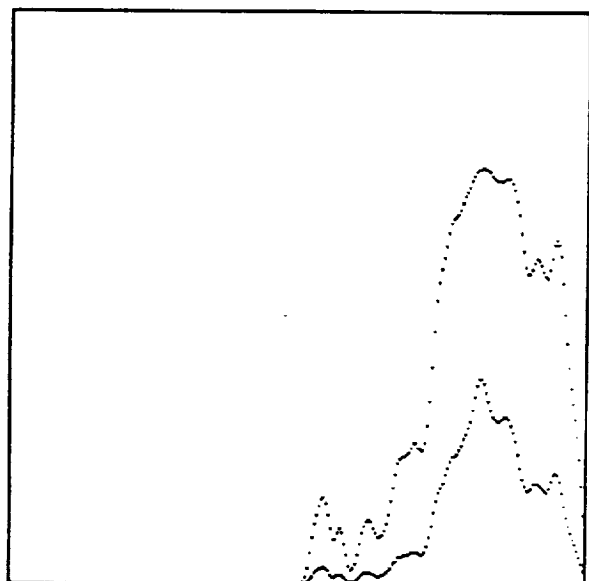
Fig 4.1 membership and p.d.f by Dubois and Prade :
 small graphe is p.d.f and big one is membership.



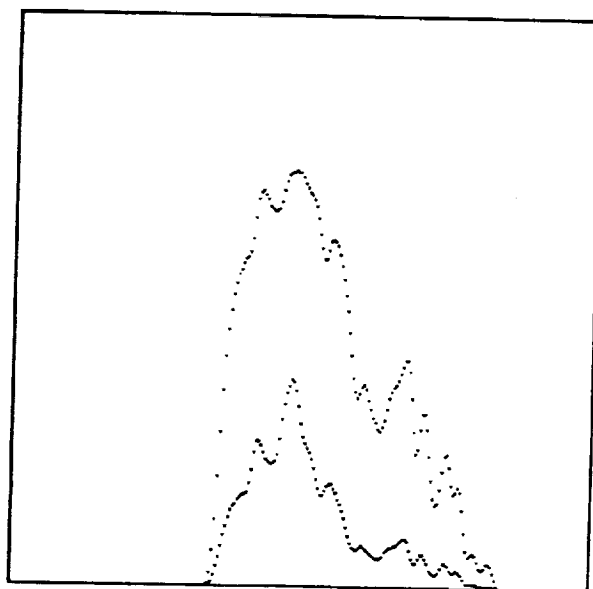
(a) background(entropy)



(b) object(entropy)

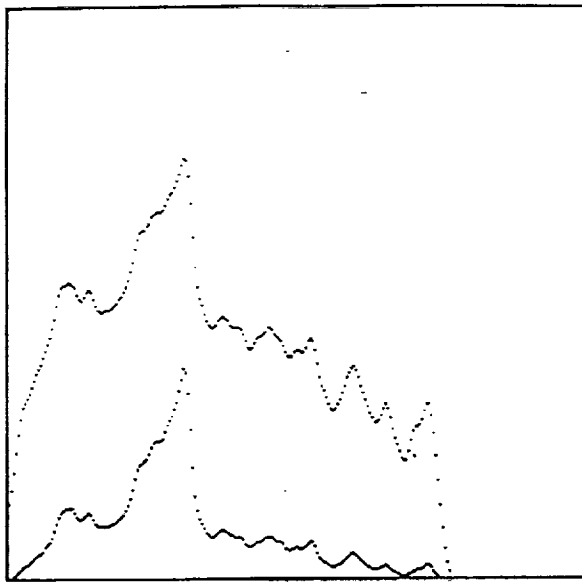


(c) background(homogeneity)

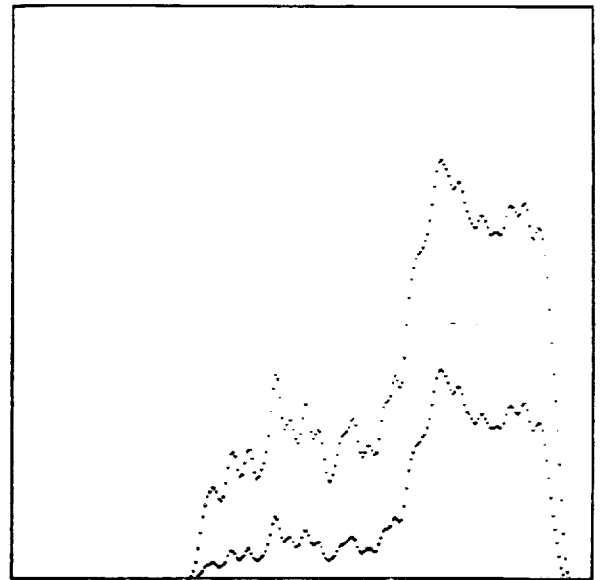


(d) object(homogeneity)

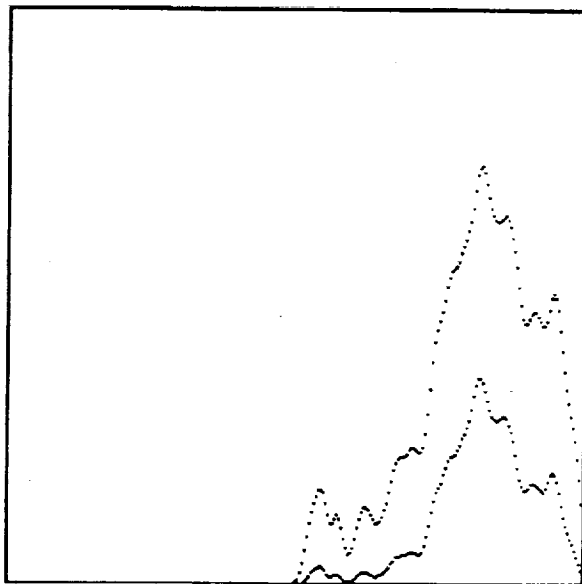
Fig 4.2 membership and p.d.f by Dubois and Prade :
small graphe is p.d.f and big one is membership.



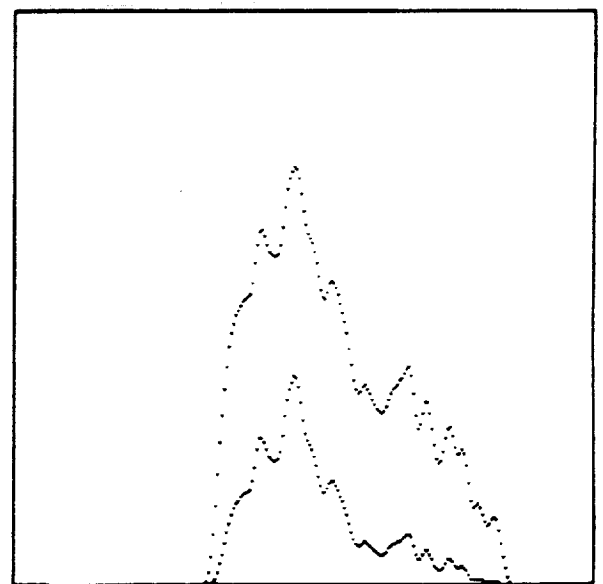
(a) background(entropy)



(b) object(entropy)

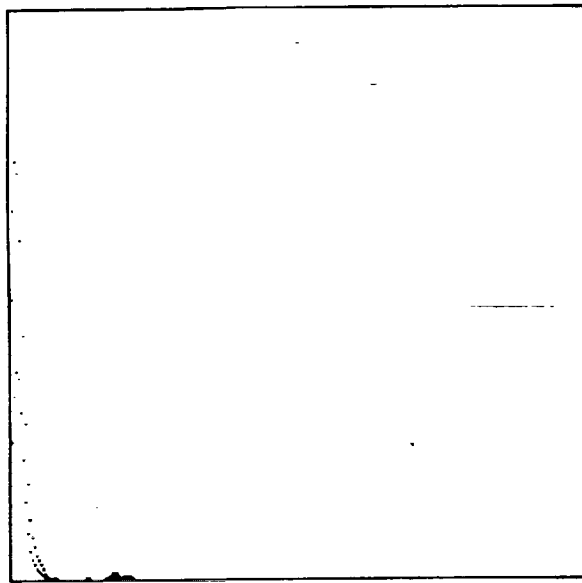


(c) background(homogeneity)

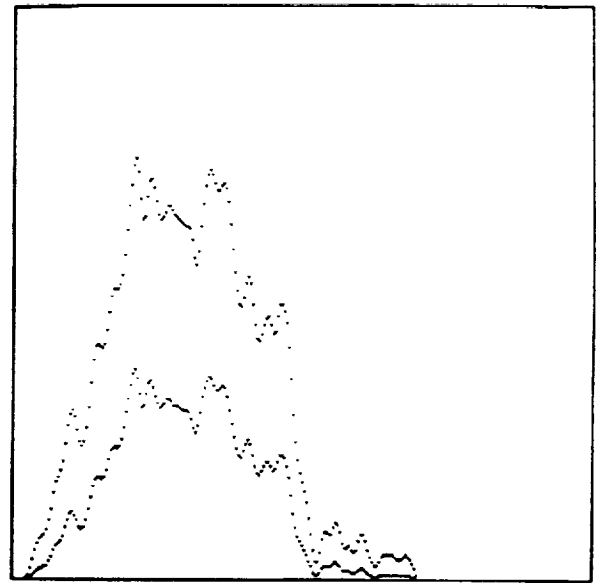


(d) object(homogeneity)

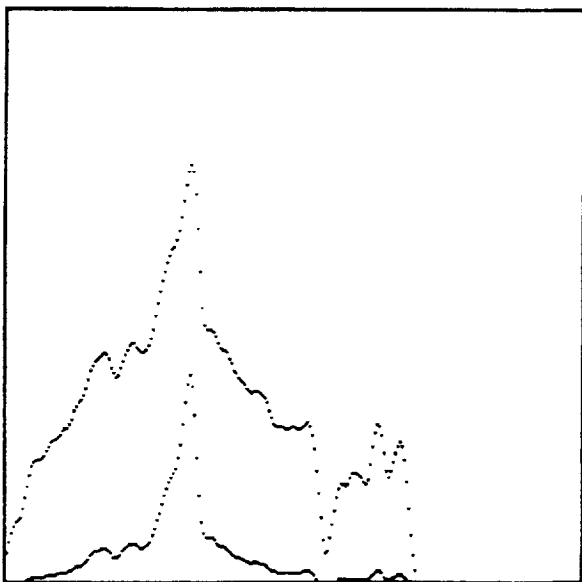
Fig 5.1 membership and p.d.f by Klir :
small graphe is p.d.f and big one is membership.



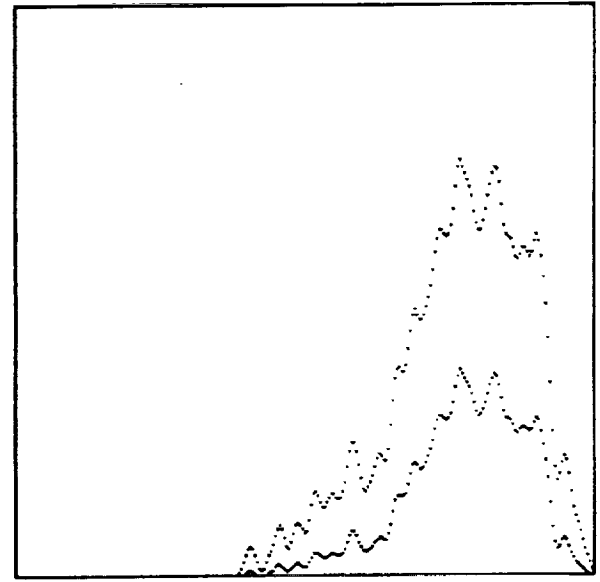
(a) background(contrast)



(b) object(contrast)



(c) background(diff. entropy)



(d) object(diff. entropy)

Fig 5.2 membership and p.d.f by Klir :
small graphe is p.d.f and big one is membership.

Clustering Methods

At the Third International Workshop on Neural Networks and Fuzzy Logic, we presented our new approach of possibilistic clustering applied to the recognition of Plano - Quadric clusters. In what follows, we present the paper which will appear in the proceedings of that Workshop, followed by other examples of the results of the algorithms. Several examples are of images of the shuttle.

N 93 - 18228

Possibilistic Clustering for Shape Recognition¹

James M. Keller and Raghu Krishnapuram
Department of Electrical and Computer Engineering
University of Missouri, Columbia, MO 65211

54-61
143403
p. 20

Abstract

Clustering methods have been used extensively in computer vision and pattern recognition. Fuzzy clustering has been shown to be advantageous over crisp (or traditional) clustering in that total commitment of a vector to a given class is not required at each iteration. Recently fuzzy clustering methods have shown spectacular ability to detect not only hypervolume clusters, but also clusters which are actually "thin shells", i.e., curves and surfaces. Most analytic fuzzy clustering approaches are derived from Bezdek's Fuzzy C-Means (FCM) algorithm. The FCM uses the probabilistic constraint that the memberships of a data point across classes sum to one. This constraint was used to generate the membership update equations for an iterative algorithm. Unfortunately, the memberships resulting from FCM and its derivatives do not correspond to the intuitive concept of degree of belonging, and moreover, the algorithms have considerable trouble in noisy environments. Recently, we cast the clustering problem into the framework of possibility theory. Our approach was radically different from the existing clustering methods in that the resulting partition of the data can be interpreted as a possibilistic partition, and the membership values may be interpreted as degrees of possibility of the points belonging to the classes. We constructed an appropriate objective function whose

minimum will characterize a good possibilistic partition of the data, and we derived the membership and prototype update equations from necessary conditions for minimization of our criterion function. In this paper, we show the ability of this approach to detect linear and quartic curves in the presence of considerable noise.

¹Research performed for NASA/JSC through a subcontract from the RICIS Center at the University of Houston - Clear Lake

I. Introduction

Clustering has long been a popular approach to unsupervised pattern recognition. It has become more attractive with the connection to neural networks, and with the increased attention to fuzzy clustering. In fact, recent advances in fuzzy clustering have shown spectacular ability to detect not only hypervolume clusters, but also clusters which are actually "thin shells", i.e., curves and surfaces [1-7]. One of the major factors that influences the determination of appropriate groups of points is the "distance measure" chosen for the problem at hand. Fuzzy clustering has been shown to be advantageous over crisp (or traditional) clustering in that total commitment of a vector to a given class is not required at each iteration.

Boundary detection and surface approximation are important components of intermediate-level vision. They are the first step in solving problems such as object recognition and orientation estimation. Recently, it has been shown that these problems can be viewed as clustering problems with appropriate distance measures and prototypes [1-7]. Dave's Fuzzy *C* Shells (FCS) algorithm [2] and the Fuzzy Adaptive *C*-Shells (FACS) algorithm [7] have proven to be successful in detecting clusters that can be described by circular arcs, or more generally by elliptical shapes. Unfortunately, these algorithms are computationally rather intensive since they involve the solution of coupled nonlinear equations for the shell (prototype) parameters. These algorithms also assume that the number of clusters are known. To overcome these drawbacks we recently proposed a computationally simpler Fuzzy *C* Spherical Shells (FCSS) algorithm [6] for clustering hyperspherical shells and suggested an efficient algorithm to determine the number of clusters when this is not known. We also proposed the Fuzzy *C* Quadric Shells (FCQS) algorithm [5] which can detect more general quadric shapes. One problem with the FCQS algorithm is that it uses the algebraic distance, which is highly nonlinear. This results in unsatisfactory performance when the data is not very "clean" [7]. Finally, none of the

algorithms can handle situations in which the clusters include lines/planes and there is much noise. In [8], we addressed those issues in a new approach called Plano-Quadric Clustering. In this paper, we show how that algorithm, coupled with our new possibilistic clustering, can accurately find linear and quadric curves in the presence of noise.

Most analytic fuzzy clustering approaches are derived from Bezdek's Fuzzy C-Means (FCM) algorithm [9]. The FCM uses the probabilistic constraint that the memberships of a data point across classes must sum to one. This constraint came from generalizing a crisp C-Partition of a data set, and was used to generate the membership update equations for an iterative algorithm. These equations emerge as necessary conditions for a global minimum of a least-squares type of criterion function. Unfortunately, the resulting memberships do not represent one's intuitive notion of degrees of belonging, i. e., they do not represent degrees of "typicality" or "possibility".

There is another important motivation for using possibilistic memberships. Like all unsupervised techniques, clustering (crisp or fuzzy) suffers from the presence of noise in the data. Since most distance functions are geometric in nature, noise points, which are often quite distant from the primary clusters, can drastically influence the estimates of the class prototypes, and hence, the final clustering. Fuzzy methods ameliorate this problem when the number of classes is greater than one, since the noise points tend to have somewhat smaller membership values in all the classes. However, this difficulty still remains in the fuzzy case, since the memberships of unrepresentative (or noise) points can still be significantly high. In fact, if there is only one real cluster present in the data, there is essentially no difference between the crisp and fuzzy methods.

On the other hand, if a set of feature vectors is thought of as the domain of discourse for a collection of independent fuzzy subsets, then there should be no constraint on the sum of the memberships. The only real constraint is that the assignments do really represent fuzzy membership values, i.e., they must lie in the interval $[0,1]$. In [10], we cast

the clustering problem into the framework of possibility theory. We briefly review this approach, and show its superiority to recognize shapes from noisy and incomplete data.

II. Possibilistic Clustering Algorithms

The original FCM formulation minimizes the objective function given by

$$J(L, U) = \sum_{i=1}^C \sum_{j=1}^N (\mu_{ij})^m d_{ij}^2, \text{ subject to } \sum_{i=1}^C \mu_{ij} = 1 \text{ for all } j. \quad (1)$$

In (1), $L = (\lambda_1, \dots, \lambda_C)$ is a C -tuple of prototypes, d_{ij}^2 is the distance of feature point x_j to cluster λ_i , N is the total number of feature vectors, C is the number of classes, and $U = [\mu_{ij}]$ is a $C \times N$ matrix called the fuzzy C -partition matrix [9] satisfying the following conditions:

$$\begin{aligned} \mu_{ij} &\in [0, 1] \text{ for all } i \text{ and } j, & \sum_{i=1}^C \mu_{ij} &= 1 \text{ for all } j, \text{ and} \\ 0 &< \sum_{j=1}^N \mu_{ij} < N \text{ for all } i. \end{aligned}$$

Here, μ_{ij} is the grade of membership of the feature point x_j in cluster λ_i , and $m \in [1, \infty)$ is a weighting exponent called the fuzzifier. In what follows, λ_i will also be used to denote the i th cluster, since it contains all of the parameters that define the prototype of the cluster.

Simply relaxing the constraint in (1) produces the trivial solution, i. e., the criterion function is minimized by assigning all memberships to zero. Clearly, one would like the memberships for representative feature points to be as high as possible, while unrepresentative points should have low membership in all clusters. This is an approach consistent with possibility theory [11]. The objective function which satisfies our requirements may be formulated as:

$$J_m(L,U) = \sum_{i=1}^C \sum_{j=1}^N (\mu_{ij})^m d_{ij}^2 + \sum_{i=1}^C \eta_i \sum_{j=1}^N (1-\mu_{ij})^m. \quad (2)$$

where η_i are suitable positive numbers. The first term demands that the distances from the feature vectors to the prototypes be as low as possible, whereas the second term forces the μ_{ij} to be as large as possible, thus avoiding the trivial solution. The following theorem, proved in [9], gives necessary conditions for minimization, hence, providing the basis for an iterative algorithm.

Theorem:

Suppose that $X = \{x_1, x_2, \dots, x_N\}$ is a set of feature vectors, $L = (\lambda_1, \dots, \lambda_C)$ is a C -tuple of prototypes, d_{ij}^2 is the distance of feature point x_j to the cluster prototype λ_i , ($i = 1, \dots, C$; $j = 1, \dots, N$), and $U = [\mu_{ij}]$ is a $C \times N$ matrix of possibilistic membership values. Then U may be a global minimum for $J_m(L,U)$ only if $\mu_{ij} = \left[1 + \left(\frac{d_{ij}^2}{\eta_i} \right)^{\frac{1}{m-1}} \right]^{-1}$.

The necessary conditions on the prototypes are identical to the corresponding conditions in the FCM and its derivatives.

Thus, in each iteration, the updated value of μ_{ij} depends only on the distance of x_j from λ_i , which is an intuitively pleasing result. The membership of a point in a cluster should be determined solely by how far it is from the prototype of the class, and should not be coupled to its location with respect to other classes. The updating of the prototypes depends on the distance measure chosen, and will proceed exactly the same way as in the case of the FCM algorithm and its derivatives.

The value of η_i determines the distance at which the membership value of a point in a cluster becomes 0.5 (i. e., "the 3 dB point"). Thus, it needs to be chosen depending on the desired "bandwidth" of the possibility (membership) distribution for each cluster. This

value could be the same for all clusters, if all clusters are expected to be similar. In general, it is desirable that η_i relates to the overall size and shape of cluster λ_i . Also, it is to be noted that η_i determines the relative degree to which the second term in the objective function is important compared to the first. If the two terms are to be weighted roughly equally, then η_i should be of the order of d_{ij}^2 . In practice we find that the following definition works best.

$$\eta_i = \frac{\sum_{j=1}^N \mu_{ij}^m d_{ij}^2}{\sum_{j=1}^N \mu_{ij}^m} \quad (3)$$

This choice makes η_i the average fuzzy intra-cluster distance of cluster λ_i . The value of η_i can be fixed for all iterations, or it may be varied in each iteration. When η_i is varied in each iteration, care must be exercised, since it may lead to instabilities. Our experience shows that the final clustering is quite insensitive to large (an order of magnitude) variations in the values of η_i .

III. The Possibilistic C Plano-Quadric Shells Algorithm

Suppose that we are given a second degree curve λ_i characterized by a prototype vector

$$p_i^T = [pi1, pi2, \dots, pir]$$

to which it is desired to fit points x_j obtained through the application of some edge detection algorithm. If a point x has coordinates $[x_1, \dots, x_n]$, then let

$$q = [x_1^2, x_2^2, \dots, x_n^2, x_1x_2, \dots, x_{(n-1)}x_n, x_1, x_2, \dots, x_n, 1]^T.$$

When the exact (geometric) distance has no closed-form solution, one of the methods suggested in the literature is to use what is known as the "approximate distance" which is

the first-order approximation of the exact distance. It is easy to show [12] that the approximate distance of a point from a curve is given by

$$d_{Aij}^2 = d_{A^2}(x_j, \lambda_i) = \frac{\delta_{Qij}^2}{|\nabla d_{Qij}^2|^2} = \frac{d_{Qij}^2}{p_i^T D_j D_j^T p_i}, \quad (4)$$

where ∇d_{Qij}^2 is the gradient of the distance functional

$$p_i^T q = [p_{i1}, p_{i2}, \dots, p_{ir}] [x_1^2, x_2^2, \dots, x_n^2, x_1 x_2, \dots, x_{(n-1)} x_n, x_1, x_2, \dots, x_n, 1]^T \quad (5)$$

evaluated at x_j . In (4) the matrix D_j is simply the Jacobian of q evaluated at x_j .

One can easily reformulate the quadric shell clustering algorithm with d_{Aij}^2 as the underlying distance measure. It was shown in [8] that the solution to the parameter estimation problem is given by the generalized eigenvector problem

$$F_i p_i = l_i G_i p_i, \quad (6)$$

where

$$F_i = \sum_{j=1}^N (\mu_{ij})^m M_j,$$

$$M_j = q_j q_j^T, \text{ and}$$

$$G_i = \sum_{j=1}^N (\mu_{ij})^m D_j D_j^T,$$

which can be converted to the standard eigenvector problem if the matrix G_i is not rank-deficient. Unfortunately this is not the case. In fact, the last row of D_j is always $[0, \dots, 0]$. Equation (6) can still be solved using other techniques that use the modified Cholesky decomposition [13], and the solution is computationally quite inexpensive when the feature space is 2-D or 3-D. Another advantage of this constraint is that it can also fit lines and

planes in addition to quadrics. Our experimental results show that the resulting algorithm, which we call the Possibilistic C Plano-Quadric Shells (PCPQS) algorithm, is quite robust in the presence of poorly defined boundaries (i. e., when the edge points are somewhat scattered around the ideal boundary curve in the 2-D case and when the range values are not very accurate in the 3-D case). It is also very immune to impulse noise and outliers. Of course, if the type of curves required are restricted to a single type, e.g., lines, or circles, or ellipses, simpler algorithms can be used with possibilistic updates, as will be seen.

IV. Determination of Number of Clusters

The number of clusters C is not known *a priori* in some pattern recognition applications and most computer vision applications. When the number of clusters is unknown, one method to determine this number is to perform clustering for a range of C values, and pick the C value for which a suitable validity measure is minimized (or maximized) [14]. However this method is rather tedious, especially when the number of clusters is large. Also, in our experiments, we found that the C value obtained this way may not be optimum. This is because when C is large, the clustering algorithm sometimes converges to a local minimum of the objective function, and this may result in a bad value for the validity of the clustering, even though the value of C is correct. Moreover, when C is greater than the optimum number, the algorithm may split a single shell cluster into more than one cluster, and yet achieve a good value for the overall validity. To overcome these problems, we proposed in [8] an alternative Unsupervised C Shell Clustering algorithm which is computationally more efficient, since it does not perform the clustering for an entire range of C values.

Our proposed method progressively clusters the data starting with an overspecified number C_{max} of clusters. Initially, the FCPQS algorithm is run with $C=C_{max}$. After the algorithm converges, spurious clusters (with low validity) are eliminated; compatible

clusters are merged; and points assigned to clusters with good validity are temporarily removed from the data set to reduce computations. The FCPQS algorithm is invoked again with the remaining feature points. The above procedure is repeated until no more elimination, merging, or removing occurs, or until $C=1$.

V. Examples of Possibilistic Clustering for Shape Recognition

Figures 1 and 2 show the detection of a circular "fractal edge" from a synthetically generated image. Figure 1(a) is the original composite fractal image; figure 1(b) shows what a gray-scale edge operator finds (or doesn't find); figure 1(c) is the output of the horizontal fractal edge operator; with figure 1(d) giving the maximum overall response of the fractal operators in four directions. Figure 2(a) depicts the (noisy) thresholded and thinned result from figure 1(d). Figure 2(b) gives the final prototype found by the FPQCS (which, since there is only one cluster present, is the same as the crisp version). Note how the presence of noise distorts the final prototype. Figure 2(c) shows the possibilistic algorithm output, which is superimposed on the original image in figure 2(d). The results of the PPQCS algorithm are virtually unaffected by noise. Several examples comparing crisp, fuzzy and possibilistic versions of clustering can be found in [6,8,10].

Figure 3 depicts the algorithm applied to the image of a model of the Space Shuttle. Figure 3(a) is the original image. Figure 3(b) gives the output of a typical edge operator. Note that, due to the rather poor quality of the original image, the edges found both noisy and incomplete. This data was then input into the possibilistic plano-quadric clustering algorithm. Figure 3(c) gives the eight complete prototypes which were found after running the algorithm. Finally, figure 39(d) displays the prototype drawn only where sufficient edges points exist.

VI. Conclusions

In this paper, we demonstrated how our new possibilistic approach to objective-function-based clustering coupled with our plano - quadric shells algorithm can recognize first and second degree shapes from incomplete and noisy edge data. This approach is superior to both crisp and fuzzy clustering, as well as to traditional methods such as the Hough Transform. Extensions of this approach to other classes of shapes is currently underway.

Acknowledgment

We are grateful to our students Hichem Frigui and Olfa Nasraoui without whose suggestions and assistance the simulation experiments would not have been possible.

VII. References

1. R. N. Dave, "New measures for evaluating fuzzy partitions induced through C-shells clustering", *Proceedings of the SPIE Conference on Intelligent Robots and Computer Vision X: Algorithms and Techniques*, Boston, Nov. 1991, pp. 406-414.
2. R. N. Dave, "Fuzzy-shell clustering and applications to circle detection in digital images", *International Journal of General Systems*, vol. 16, 1990, pp. 343-355.
3. R. N. Dave, "Adaptive C-shells clustering", *Proceedings of the North American Fuzzy Information Processing Society Workshop*, Columbia, Missouri, 1991, pp. 195-199.
4. J. C. Bezdek and R. J. Hathaway, "Accelerating convergence of the Fuzzy C-Shells clustering algorithms", *Proceedings of the International Fuzzy Systems Association Congress*, Brussels, July 1991, Volume on *Mathematics*, pp. 12-15.
5. R. Krishnapuram, H. Frigui, and O. Nasraoui, "New fuzzy shell clustering algorithms for boundary detection and pattern recognition", *Proceedings of the SPIE Conference on Intelligent Robots and Computer Vision X: Algorithms and Techniques*, Boston, Nov. 1991, pp.458-465.
6. R. Krishnapuram, O. Nasraoui, and H. Frigui, "The fuzzy C spherical shells algorithm: A new approach", to appear in the *IEEE Transactions on Neural Networks*.
7. R. N. Dave and K. Bhaswan, "Adaptive fuzzy C-shells clustering and detection of ellipses", to appear in the *IEEE Transactions on Neural Networks*, vol. 3, no. 5, 1992.
8. R. Krishnapuram, H. Frigui, and O. Nasraoui, "A Fuzzy Clustering Algorithm to Detect Planar and Quadric shapes", submitted to Proceedings NAFIPS'92.

9. J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1981.

10. R. Krishnapuram and J. M. Keller, "A possibilistic approach to clustering", submitted to *IEEE Transactions on Fuzzy Systems*.

11. D. Dubois and H. Prade, *Possibility Theory: An Approach to Computerized Processing of Uncertainty*, Plenum Press, New York, 1988.

12. R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*, vol. I, Addison Wesley, Reading, MA, 1992, Chapter 11.

13. G. Taubin, "Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 11, Nov. 1991, pp. 1115-1138.

14. E. Backer and A. K. Jain, "A Clustering Performance Measure Based on Fuzzy Set Decomposition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 3, pp. 66, 1981.

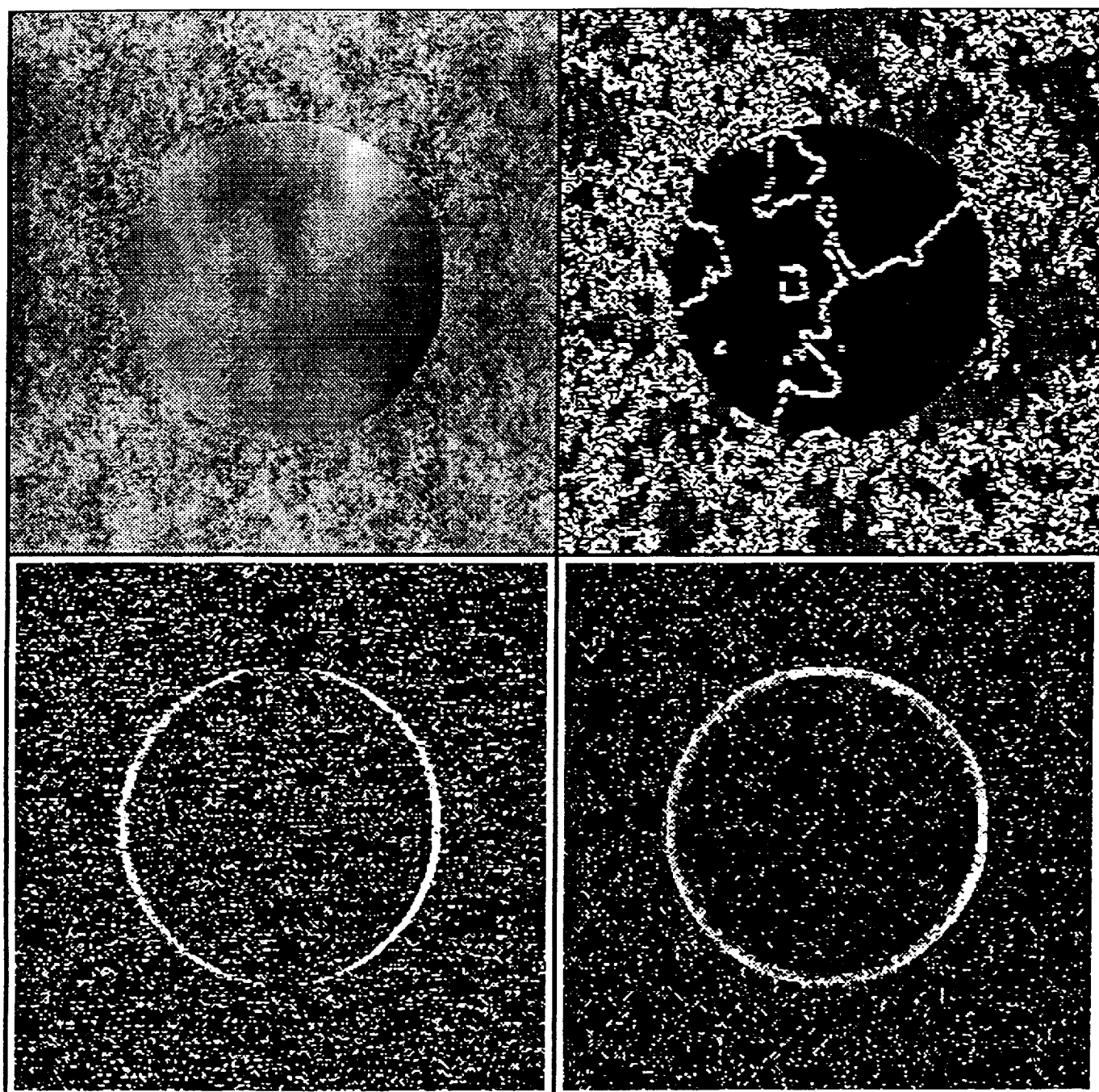


Figure 1. Detection of a fractal circular edge.

- (a) Upper Left. Original fractal composite image.
- (b) Upper Right. Output of gray scale edge operator.
- (c) Lower Left. Output of "horizontal" fractal edge operator.
- (d) Lower Right. Results of Maximum magnitude of outputs of four directions of fractal operators.

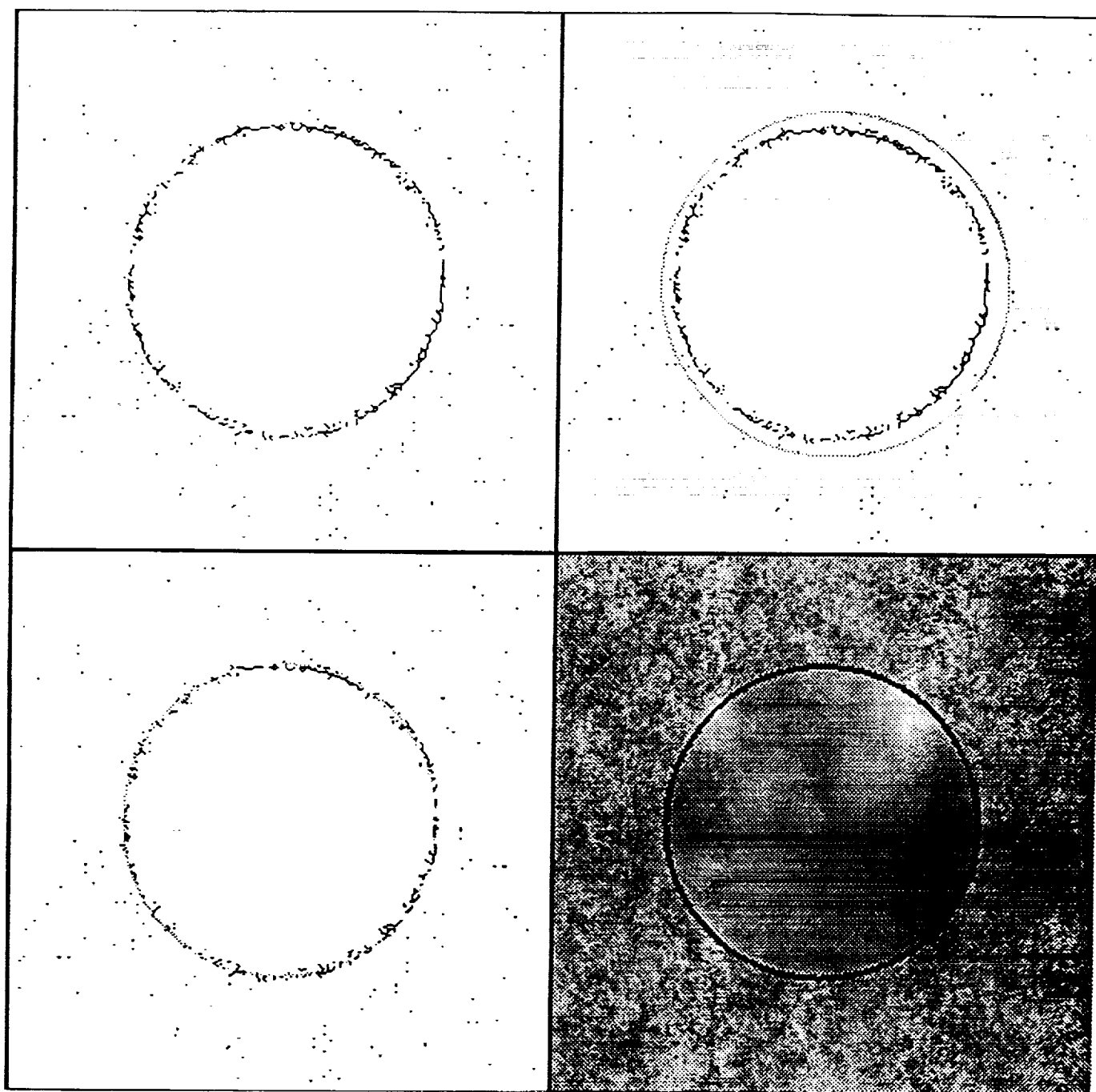


Figure 2. Recognition of circular boundary.

- (a) Upper Left. Figure 1(d) thresholded and thinned.
- (b) Upper Right. Circular prototype found by fuzzy (or crisp) clustering.
- (c) Lower Left. Circular prototype found by possibilistic clustering.
- (d) Lower Right. Possibilistic prototype superimposed on original image.

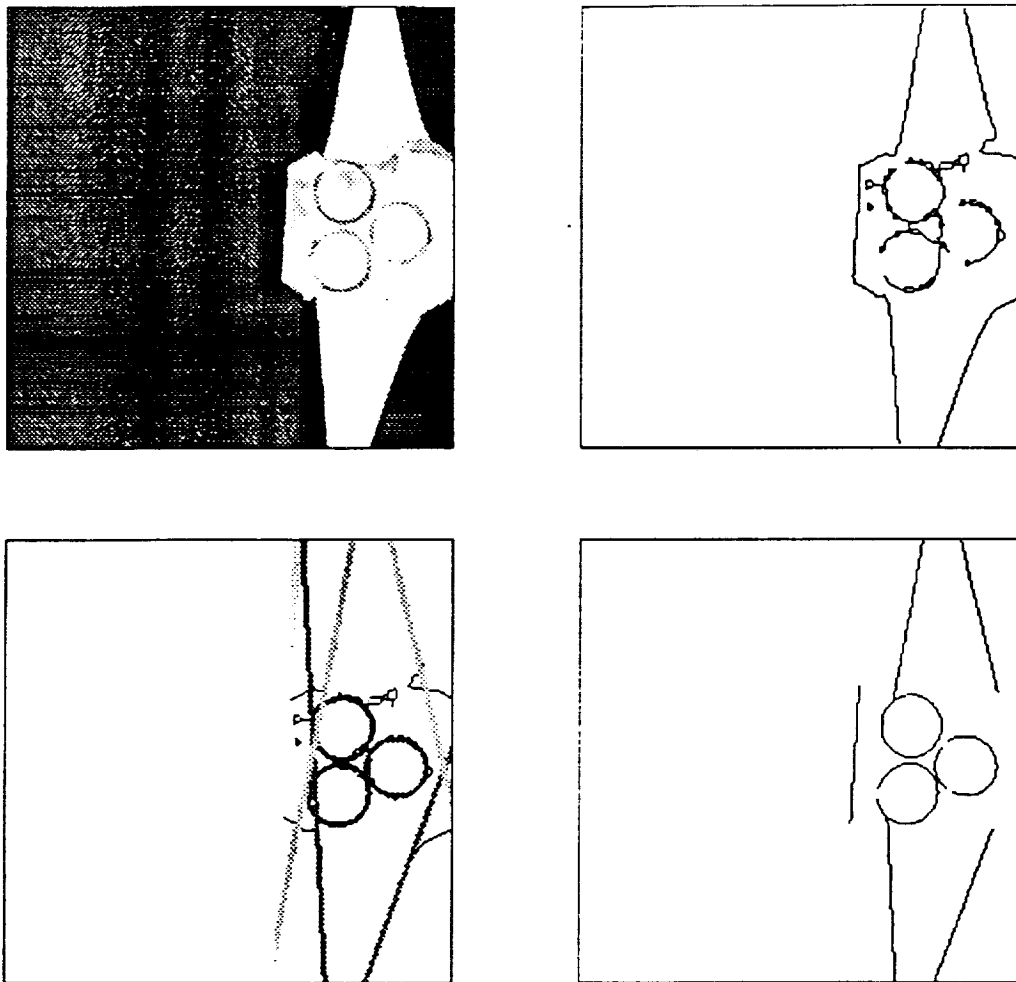


Figure 3. Recognition of Shuttle model boundaries.

- (a) Upper Left. Original Shuttle image.
- (b) Upper Right. Incomplete and noisy edges found by edge operator.
- (c) Lower Left. Prototypes found by Possibilistic Plano-Quadric clustering.
- (d) Lower Right. Possibilistic prototypes superimposed drawn where there is sufficient edge information.

Pose Estimation Using Possibilistic Clustering

In the Third Quarter report, we described how the Unsupervised C Quadric Shells (UCQS) algorithm could be used to estimate the pose of the shuttle. The shuttle's image is taken from the back so that the exhaust nozzles and the back edges of the three wings are apparent. Given an original unrotated image, the exhaust nozzles can be parametrized by three circles, and the three wings can be parametrized by three straight lines. These parameters are easily determined by the UCQS algorithm. As the shuttle rotates, the shape of the nozzles will change from circles to ellipses, so will the orientation of the straight lines representing the three wings. The UCQS algorithm is used in order to cluster this edge image and determine the parameters of the ellipses and lines. Finally, these parameters can be used to solve for the translation and rotation parameters, as long as the translation is made in the image plane. In fact, depth information can also be derived from the change in the size of the nozzles.

We also consider the case where only line information is available. Once again, our new possibilistic plano-quadric clustering approach is used to detect and recognize the linear segments. In what follows, derivation of pose parameters is given for both the case where three corresponding line segments have been identified, and where one circle and one line have been matched.

POSE ESTIMATION:

The 3-D object attitude in space can be determined from a single perspective image. Dhome et al [1] developed a method to solve for the three-dimensional attitude of an object based on the perspective projection of three image lines. Krishnapuram & Casasent [2] developed a method for determining two of the three rotation angles necessary to describe an object attitude in 3-D space from a single perspective projection of one circle.

I. Determination of The Attitude of One Object From Three Lines:

The perspective projection of a point $P_i = (X_i, Y_i, Z_i)$ on an image is the point $p_i = (x_i, y_i, z_i) = (X_i f/Z, Y_i f/Z, f)$. Let l_i be an image line characterized by a vector $v_i = (a_i, b_i, 0)$ and a point $p_i = (x_i, y_i, f)$. l_i is the perspective projection of a space line L_i . Therefore it lies in the "interpretation plane" containing the origin of the coordinate system O and the image line l_i . The normal N_i to this plane is perpendicular to v_i and the vector Op_i . Thus $N_i = v_i \times Op_i = (b_i f, -a_i f, d_i)^T$, where $d_i = a_i y_i - b_i x_i$ is the Euclidean distance between the center of the image and line l_i . If $V_i = (A_i, B_i, C_i)^T$ is the director vector of the space line L_i , then it must be orthogonal to N_i , hence $V_i \cdot N_i = 0$ implying that :

$$(A_i, B_i, C_i)^T \cdot (b_i, -a_i, d_i/f) = 0 \quad (1)$$

Consider three object lines in 3-D space L_{0i} , $i = 1, \dots, 3$ defined in a model reference frame (S_{om}). The director vector of L_{0i} is $V_{0i} = (A_{0i}, B_{0i}, C_{0i})^T$. When the object is rotated in 3-D space, the lines L_{0i} are rotated into lines L_{3i} . Therefore

$$(A_{3i}, B_{3i}, C_{3i})^T = R_{\alpha\beta\gamma} (A_{0i}, B_{0i}, C_{0i})^T \quad (2)$$

where $R_{\alpha\beta\gamma}$ is the rotation matrix.

The perspective projections of lines L_{3i} are the lines l_{0i} . Equation (1) becomes

$$(A_{3i}, B_{3i}, C_{3i})^T \cdot (b_{0i}, -a_{0i}, d_{0i}/f)^T = R_{\alpha\beta\gamma} (A_{0i}, B_{0i}, C_{0i})^T \cdot (b_{0i}, -a_{0i}, d_{0i}/f)^T = 0 \quad (3)$$

where $i = 1, \dots, 3$ and α , β , and γ are the unknown rotation angles about x, y, and z axes respectively. Solving this system of equations is too complicated. A specially defined model coordinate system (S_{1m}) and a corresponding viewer coordinate system (S_{1v}) can be used to simplify the problem [1]. With these coordinate systems, only two rotation angles α and β need to be determined, i.e. the system of equations (3) can be reduced to two equations and two unknowns. First, α is found by iteratively solving an 8th order equation. Then β is solved for by substitution. When the three lines are coplanar, or when they form a junction, the 8th order equation reduces to a 4th order equation.

II Determination of the Attitude of an Object From a Circle and a Line:

Given a circular curve on the x-y plane, and an $x' y'$ view of this curve in a different coordinate system $x' y' z'$. The two frames (x, y, z) and (x', y', z') are related by a homogenous transformation T , such that

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = T \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & t_{13} & 0 \\ t_{21} & t_{22} & t_{23} & 0 \\ t_{31} & t_{32} & t_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}$$

A circle of radius r on the xy plane is described by :

$$\begin{cases} x^2 + y^2 = r^2 \\ z = 0 \end{cases} \quad \begin{matrix} (4) \\ (5) \end{matrix}$$

In the (x', y', z') frame, equations (4) & (5) become

$$(t_{11}x' + t_{12}y' + t_{13}z')^2 + (t_{21}x' + t_{22}y' + t_{23}z')^2 = r^2 \quad (6)$$

$$t_{31}x' + t_{32}y' + t_{33}z' = 0 \quad (7)$$

Substituting z' in terms of x' and y' from equation (7) into equation (6) yields the equation for the 2-D projection of the 3-D circular curve onto an arbitrary $x' y'$ plane. Making use of the fact that the columns of T are mutually orthogonal unit vectors, we obtain

$$\left(1 + \frac{t_{31}^2}{t_{33}^2}\right) x'^2 + \left(1 + \frac{t_{32}^2}{t_{33}^2}\right) y'^2 + \left(\frac{2 t_{31} t_{32}}{t_{33}^2}\right) x'y' = r^2 \quad (8)$$

This is the equation of an ellipse in the (x', y', z') frame. If the parameters of this ellipse are known, equation (8) can be solved for the transformation parameters t_{31} , t_{32} , and t_{33} .

The transformation matrix T can be written as a function of the rotation angles α , β , and γ :

$$T = \begin{bmatrix} \cos\gamma \cos\beta & \cos\gamma \sin\beta \sin\alpha - \sin\gamma \cos\alpha & \cos\gamma \sin\beta \cos\alpha + \sin\gamma \sin\alpha & 0 \\ \sin\gamma \cos\beta & \sin\gamma \sin\beta \sin\alpha + \cos\gamma \cos\alpha & \sin\gamma \sin\beta \cos\alpha - \cos\gamma \sin\alpha & 0 \\ -\sin\beta & \cos\beta \sin\alpha & \cos\beta \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Having already solved for t_{31} , t_{32} , and t_{33} , a and b can be easily determined from the 3rd row of T .

In order to determine the 3rd angle γ , a line can be used in addition to the circle. In this case the two rotation angles α and β can be determined as discussed previously. Knowing these two angles, equation (3) with $i = 1$ (since we have only one line) becomes simple to solve, since the only unknown is γ .

References

1. M. Dhome, M. Richetin, J-T Lapesté, and G. Rives, " Determination of the attitude of 3-D objects from a single perspective view ", in IEEE trans. PAMI, vol. PAMI-11, pp 1265 - 1278, Dec 1989.

2. R.Krishnapuram and D. Casasent, " Hough Transform detection of 3-D curves and target trajectories", in Applied Optics, vol. 28, pp 3479 - 3486, Aug 89.

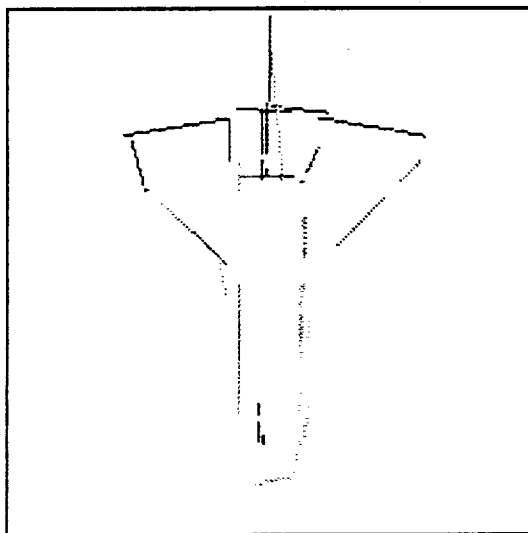
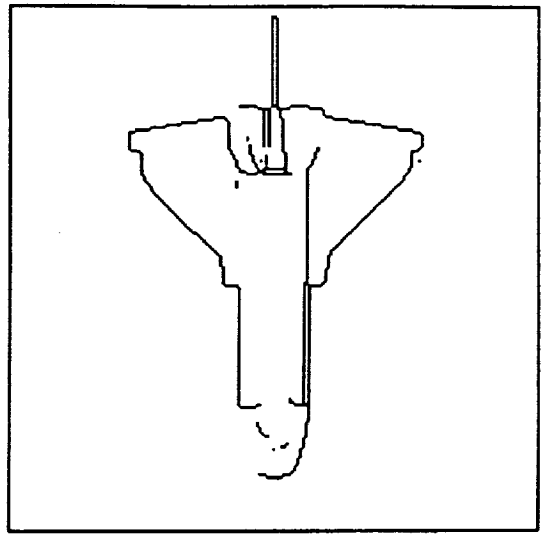
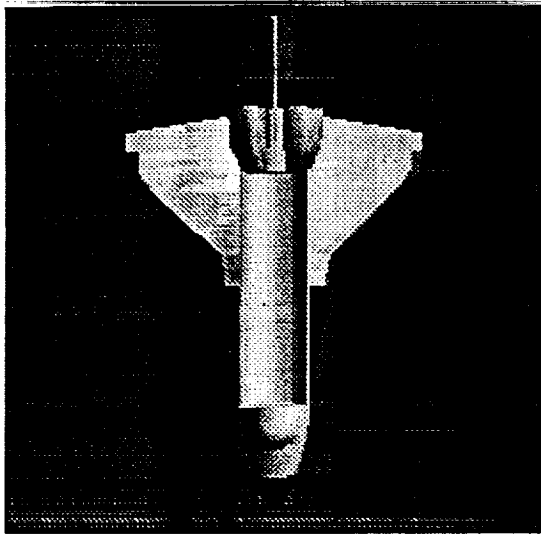
OMIT

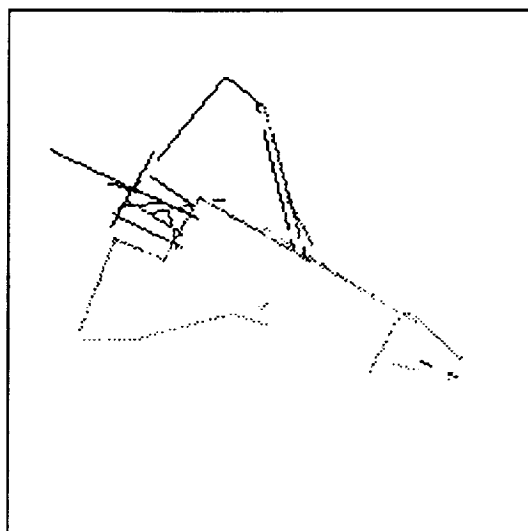
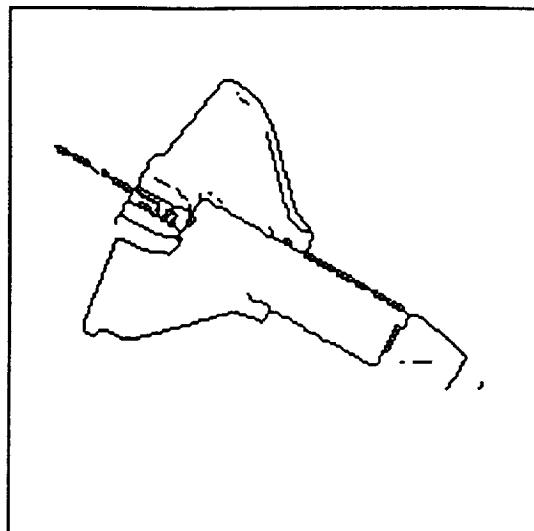
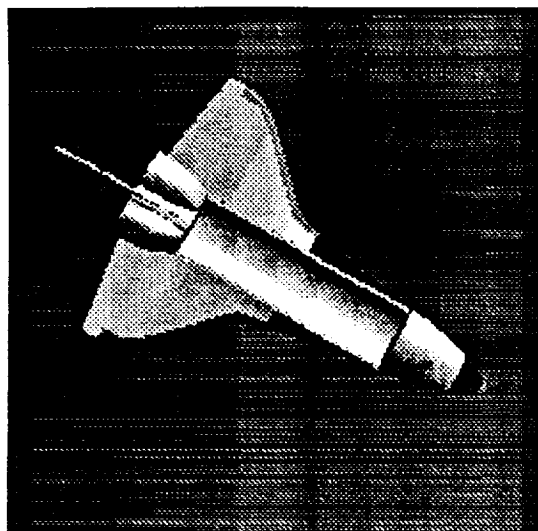
Testing the Algorithms

Besides the examples shown in the earlier reports, and the accompanying papers, we conclude this report with several examples of the results of our research.

Examples of Determination of Lines from Different Orientations of the Shuttle by Possibilistic Clustering

The following two pages show the use of the Possibilistic Plano - Quadric Clustering Algorithm to identify the lines on images of the Shuttle. The gray scale images were synthetically generated by Lincom. A simple edge detector was run on the images. Thresholded output of the edge data was then sent to the unsupervised clustering algorithm (which also determines the optimum number of clusters). The prototypes which were identified are then displayed. After matching is performed, the approach described above could be used to determine the rotation angles to specify the pose of the second image relative to the reference model (first image). A complete solution to this problem is being proposed for a second year effort.





Further examples of recognition of linear and quadric curves

The following images were shown at the NASA Workshop, although they were not included in the paper which is to appear in the proceedings. In each case, the original image was processed by an appropriate edge operator, the results were thresholded, and the resulting edge points were used as input to the possibilistic clustering algorithm.



