N93-18228

Possibilistic Clustering for Shape Recognition[1]

S4-61
143403
p. 20

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

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.

[1] Research 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 it's superiority to recognize shapes from noisy and incomplete data.

II. Possibilistic Clustering Algorithms

The original FCM formulation minimizes the objective function given by

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

In (1), $L = (\lambda_1,...,\lambda_C)$ 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:

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

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 ith 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:

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

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 $X = \{x_1, x_2, ..., x_N\}$ is a set of feature vectors, $L = (\lambda_1,...,\lambda_C)$ is a C-tuple of prototypes, d_{ij}^2 is the distance of feature point $\mathbf{x}_j$ to the cluster prototype λ_i, (i = 1, ..., C; j = 1, ..., 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.

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

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

$$\mathrm{p}_i^{\mathrm{T}} = [p_{i1}, p_{i2}, \ldots, p_{ir}]$$

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

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

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

$$d^2_{\mathrm{A}ij} = d_{\mathrm{A}}{}^2(x_j,\lambda_i) = \frac{\delta^2_{\mathrm{Q}ij}}{|\nabla d^2_{\mathrm{Q}ij}|^2} = \frac{d^2_{\mathrm{Q}ij}}{p_i{}^{\mathrm{T}} D_j D_j{}^{\mathrm{T}} p_i} , \qquad (4)$$

where $\nabla d^2_{\mathrm{Q}ij}$ is the gradient of the distance functional

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

evaluated at $\boldsymbol{x_j}$. In (4) the matrix $\boldsymbol{D_j}$ is simply the Jacobian of $\boldsymbol{q}$ evaluated at $\boldsymbol{x_j}$.

One can easily reformulate the quadric shell clustering algorithm with $d^2_{\mathrm{A}ij}$ 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

$$\boldsymbol{F_i p_i} = l_i \boldsymbol{G_i p_i} , \qquad (6)$$

where

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

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

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

which can be converted to the standard eigenvector problem if the matrix $\boldsymbol{G_i}$ is not rank-deficient. Unfortunately this is not the case. In fact, the last row of $\boldsymbol{D_j}$ is always [0, . . . ,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.

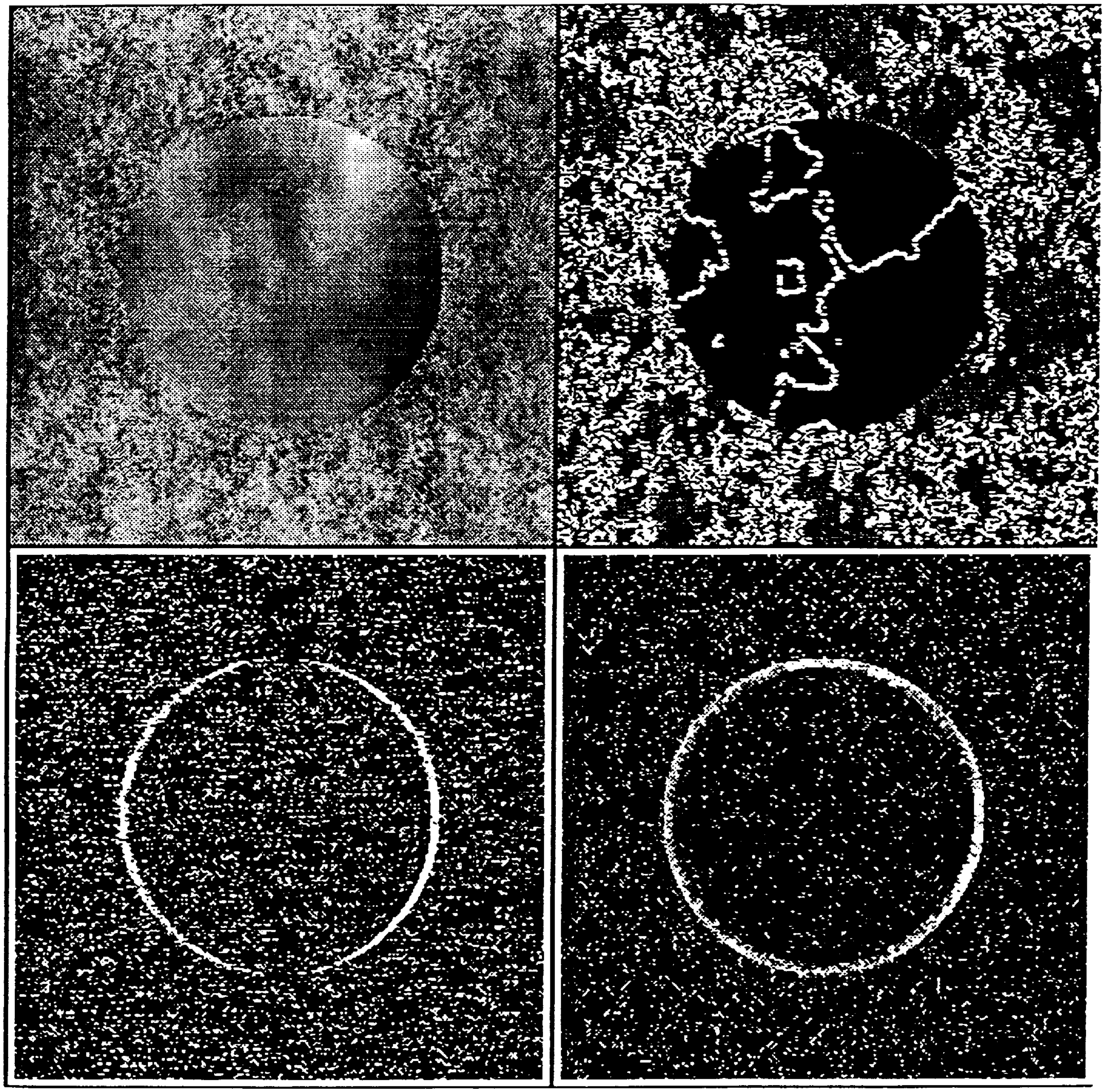

Figure 1. Detection of a fractal circular edge.

(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.

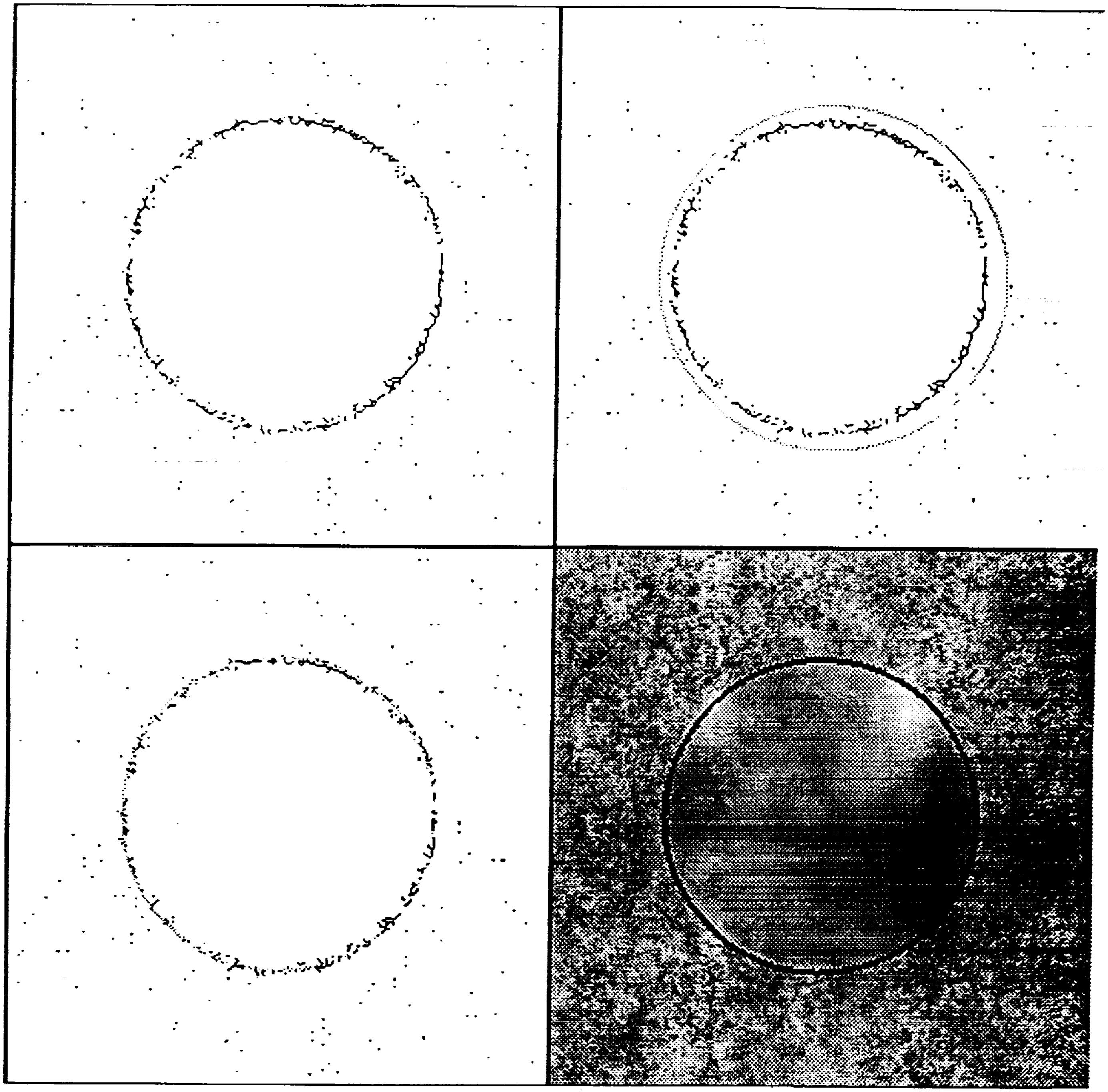

Figure 2. Recognition of circular boundary.

(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.

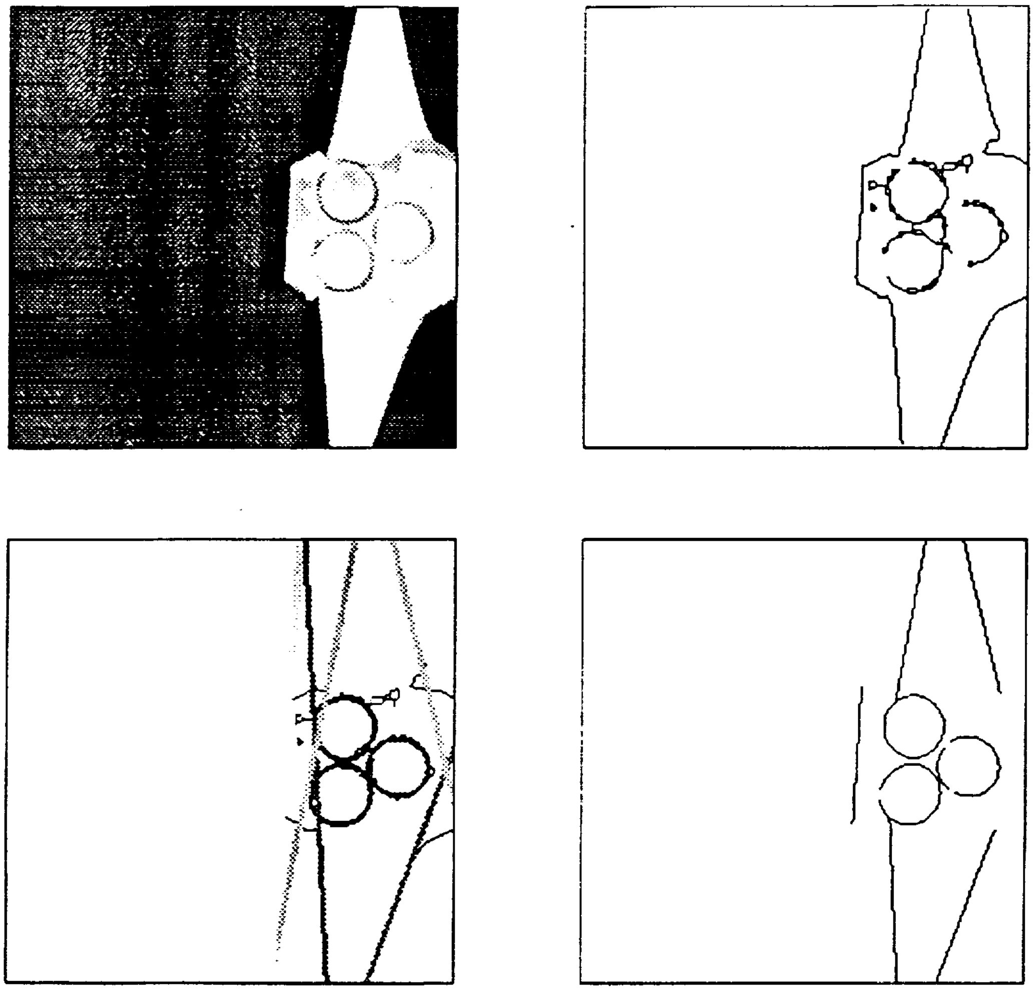

Figure 3. Recognition of Shuttle model boundaries.

(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 $\mathbf{Op_i}$. Thus $N_i = v_i$ ¥ $\mathbf{Op_i} = (b_i\ f, -a_i\ f, d_i)^T$, 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 $\mathbf{V_i} = (A_i, B_i, C_i)^T$ is the director vector of the space line L_i, then it must be orthogonal to N_i, hence $\mathbf{V_i}\ .\ \mathbf{Ni} = 0$ implying that :

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

Consider three object lines in 3-D space L_{0i}, i = 1, ..., 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

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

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

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

$$(A_{3i}, B_{3i}, C_{3i})^T . (b_{0i}, -a_{0i}, d_{0i}/f)^T =$$

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

where i = 1,..., 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 :

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

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

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

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

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

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

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.