Robust Clustering Methods: A Unified View

Rajesh N. Davé and Raghu Krishnapuram, *Senior Member, IEEE*

***Abstract*—Clustering methods need to be robust if they are to be useful in practice. In this paper, we analyze several popular robust clustering methods and show that they have much in common. We also establish a connection between fuzzy set theory and robust statistics and point out the similarities between robust clustering methods and statistical methods such as the weighted least-squares (LS) technique, the M estimator, the minimum volume ellipsoid (MVE) algorithm, cooperative robust estimation (CRE), minimization of probability of randomness (MINPRAN), and the epsilon contamination model. By gleaning the common principles upon which the methods proposed in the literature are based, we arrive at a unified view of robust clustering methods. We define several general concepts that are useful in robust clustering, state the robust clustering problem in terms of the defined concepts, and propose generic algorithms and guidelines for clustering noisy data. We also discuss why the generalized Hough transform is a suboptimal solution to the robust clustering problem.**

***Index Terms*— Clustering validity, fuzzy clustering, robust methods.**

I. Introduction

IT is indeed true that algorithms utilized in engineering and scientific applications need to be robust. By robustness, we mean that the performance of an algorithm should not be affected significantly by small deviations from the assumed model and it should not deteriorate drastically due to noise and outliers. Techniques that have the ability to tolerate noise and outliers in the data have become very popular—for example, the Hough transform. The focus of this paper is on clustering techniques, which are an important part of many engineering and scientific applications. They have been used extensively in pattern recognition, computer vision, and control tasks. More recently, novel clustering techniques have been developed for the detection of clusters of various shapes such as lines, planes, circles, ellipses, curves, and curved surfaces [6], [9], [11], [17], [29], [30], [33]. In real applications, the data is bound to have noise and outliers, and the assumed models such as Gaussian distributions are only approximations to reality. Thus, it follows that clustering techniques need to be robust if they are to be a useful tool. This has led to the development of several new techniques [7], [8], [12], [26], [31], [36], [47] that claim to be robust to varying degrees.

Manuscript received July 30, 1995; revised August 2, 1996. This work was supported in part by the Office of Naval Research Grant N00014-96-1-0439. R. Davé was supported in part by the ECE Department, University of Missouri, Columbia, during his Sabbatical visit. R. Krishnapuram was supported in part by the Office of Naval Research Grant N00014-96-1-0439.

R. Davé is with the Department of Mechanical Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA.

R. Krishnapuram is with the Department of Computer Engineering and Computer Science, University of Missouri, Columbia, MO 65203 USA.

Publisher Item Identifier S 1063-6706(97)00596-1.

Robust statistics and fuzzy set theory are two disciplines that have evolved independently in the last couple of decades. However, as we show in this paper, they have much in common and the concept of weight functions in robust statistics can be related to the concept of membership functions in fuzzy set theory or possibility distributions in possibility theory. This fact might explain the claim made by the proponents of fuzzy set theory that a fuzzy approach is more tolerant to variations and noise in the input data when compared with a crisp approach.

At this juncture, it may be worthwhile to look into the concept of robustness from a robust statistical point of view. According to Huber [25], a robust procedure can be characterized by the following: 1) it should have a reasonably good efficiency (accuracy) at the assumed model; 2) small deviations from the model assumptions should impair the performance only by a small amount; and 3) larger deviations from the model assumptions should not cause a catastrophe. The first requirement is of great importance in statistics and it is taken for granted in engineering because systems have to be designed such that their performance falls within the specified tolerance. In other words, when the data is "clean" and follows the assumed model, a robust procedure must yield accurate estimates. However, in engineering practice, one also needs to pay much attention to the second requirement, namely, small deviations should not have a significant negative effect on the performance. The third property is related to the concept of breakdown point. The finite-sample breakdown point is defined as follows [19]. Let Z be a sample of N data points, and T be an estimator (e.g., regression or location). Now consider all possible corrupt samples Z' obtained by replacing any M of the original data points by arbitrary values. Let bias $(M; T, Z)$ denote the maximum bias in the estimate caused by such a contamination, i.e.,

$$\text{bias}(M; T, Z) = \sup_{Z'} \|[T(Z') - T(Z)]\| \tag{1}$$

where the supremum is over all the possible contaminated samples Z'. If this bias is infinite, the M outliers have an arbitrarily large effect on T and, thus, the estimator *breaks down*. Therefore, the breakdown point is defined as

$$\varepsilon_N^*(T, Z) = \min\left\{\frac{M}{N}\colon \text{bias}(M; T, Z)\text{is infinite}\right\}. \tag{2}$$

In plain English, it is the smallest fraction of contamination that in the worst case, can cause the estimator to break down completely. Obviously, one would like to have a procedure that has a very high breakdown point. In theory, however, the highest breakdown point one can achieve is 0.5 (or 50%) because for any higher contamination one is not *guaranteed*

to be able to distinguish the good points from the bad. To illustrate this point, let us assume that one is looking for a straight line fit of a given data set. Then, if the contaminating points "conspire" to form another straight line, and if the contamination level is larger than 50%, one cannot distinguish between the true line and the line represented by the contamination.

The definition of breakdown point involves arbitrarily large errors in the estimates, since it allows arbitrarily large degrees of corruption in the data set. This is neither acceptable nor possible in most engineering applications. We would rather that the errors in the estimates be bounded by the required accuracy. Moreover, feature values cannot be arbitrarily large, and the upper and lower bounds for the values of each feature are generally known in advance. Therefore, a more appropriate definition of breakdown point might be

$$\varepsilon_N^*(T, Z) = \min\left\{\frac{M}{N}\colon \text{bias}(M; T, Z) > \zeta\right\} \qquad (3)$$

where ζ denotes the acceptable accuracy for the application. While computing the bias using (1), the supremum is taken over the (contaminated) samples that lie within the known feature bounds.

Apart from noise and outliers, another issue that clustering algorithms need to deal with is the determination of the number of clusters. This problem is closely intertwined with the problem of robustness. Not knowing the number of clusters in a data set complicates the task of separating the good points from the noise points, and conversely, the presence of noise makes it harder to determine the number of clusters. Although several methods have been proposed to tackle the problem of unknown number of clusters [2], [10], [11], [16], [26]–[28], [30], [47], only a few of them deal with robustness aspects. Despite some of the recent advances, the solution to the general problem of robust clustering when the number of clusters is unknown remains elusive, and each of the techniques proposed in the literature has its share of limitations.

The purpose of this paper is not only to establish a connection between fuzzy set theory and robust statistics, but also to discuss and compare several popular clustering methods from the point of view of robustness. The main objectives of this paper are: 1) to show the similarities between prototype-based robust clustering algorithms (such as the Ohashi algorithm [12], [36], the noise clustering (NC) method [7], [8], the possibilistic clustering (PC) method [31]), and several popular clustering algorithms such as the potential function approach [44], the mountain method [4], [45], the deterministic annealing/least biased fuzzy clustering method [2], [39], [40], and the iteratively reweighted least-squares (LS) approach [1], [21], [34], [42], [46]; 2) to view the above approaches from the point of view of robust statistics to establish a theoretical basis for their robust behavior as well as to show the correspondence between concepts in robust statistics and fuzzy set theory; 3) to relate the above approaches to methods based on robust statistics such as the minimum volume ellipsoid (MVE) [42], cooperative robust estimation (CRE) [5], minimization of probability of randomness (MINPRAN) [43], and the epsilon contamination model [25], [47]; 4) to define and discuss general concepts pertaining to robust clustering and recommend generic algorithms based on the common principles that underlie most robust clustering algorithms proposed in the literature; and 5) to relate robust clustering to the generalized Hough transform and show that it falls short of being a good solution to the robust clustering problem.

The organization of this paper is as follows. In Section II, we briefly review the prototype-based approach to fuzzy clustering [3], [22], [32] and its two main limitations, i.e., the problem of noise and the problem of unknown number of clusters. In Section III, we analyze three prototype-based algorithms that have been developed to cope with the problem of noise and show that they are identical under certain conditions. The algorithms discussed in this section include the Ohashi algorithm [12], [36], the noise clustering (NC) algorithm due to Davé [7], [8], and the possibilistic C means (PCM) algorithm due to Krishnapuram and Keller [31]. We refer to the approach resulting from "unifying" these three methods as the N/PC1 approach.[1] In Section IV, we discuss the potential function approach and a derivative of this method called the mountain method, and relate this approach to the N/PC1. In Sections V and VI, we relate a least-biased fuzzy clustering method based on deterministic annealing and the weighted LS technique to the N/PC1 approach. In Section VII, we briefly review recent developments in robust statistics, and show that the "weight function" and "scale" in robust statistics correspond to the "membership function" and "resolution" in (robust) fuzzy clustering. These correspondences (as well as others) explain the robustness of fuzzy approaches in general and the N/PC1 approach in particular. In Section VIII, we discuss the MVE method due to Rousseeuw and Leroy [42], and its modification called the generalized minimum volume ellipsoid (GMVE) algorithm [26]. We draw parallels between the MVE/GMVE technique and the N/PC1 approach. In Section IX, we relate two recently proposed approaches in the computer vision literature, i.e., CRE [5] and MINPRAN [43], to the N/PC1 and GMVE approaches. In Section X, we compare the N/PC1 approach with another clustering technique called the Gaussian mixture density decomposition (GMDD) algorithm [47], which is based on the epsilon contamination model [25]. In Section XI, we define several concepts that are useful in robust clustering as applied to engineering problems. We also propose generic algorithms for clustering when the data contains noise and outliers. Using the concepts defined in this section, we show that the generalized Hough transform represents a suboptimal solution to the robust clustering problem. Finally, in Section XII, we summarize the strengths and shortcomings of the algorithms reviewed in this paper and point out some unsolved issues.

II. Prototype-Based Fuzzy Clustering

Let $X = \{\boldsymbol{x}_j | j = 1 \cdots N\}$ be a finite subset of an n-dimensional vector space over the reals. Here, X denotes the set of feature vectors. Let C denote the number of clusters. A $C \times N$ matrix $U = [u_{ij}]$ is called a constrained fuzzy C

[1] This acronym will be explained later.

partition of X if the entries of U satisfy [3], [22]

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

$$\text{and} \sum_{j=1}^{C} u_{ij} = 1 \qquad \text{for all } j. \tag{4}$$

In the above, u_{ij} denotes the grade of membership (belonging) of $\boldsymbol{x}_j$ in the ith fuzzy subset of X. Prototype-based fuzzy algorithms partition the data set by minimizing the following squared-error criterion:

$$J(B, U; X) = \sum_{i=1}^{C} \sum_{j=1}^{N} (u_{ij})^m \, d^2(\boldsymbol{x}_j, \beta_i). \tag{5}$$

In (5), $m \in (1, \infty)$ is a weighting exponent, $d^2(\boldsymbol{x}_j, \beta_i)$ is the distance from a feature point $\boldsymbol{x}_j$ to the prototype β_i, and $B = (\beta_1, \cdots, \beta_C)$ is a C tuple of prototypes, each of which characterizes one of the C clusters. Each prototype β_i consists of a set of parameters. In the basic fuzzy C-means (FCM) algorithm [3], which is one of the most frequently used (or even misused) clustering algorithms, it is simply the center of the cluster. Minimization of the objective function with respect to U subject to the constraints in (4) gives us [3]

$$\left.\begin{aligned} u_{ij} &= \frac{1}{\sum_{k=1}^{C} \left[\frac{d^2(\boldsymbol{x}_j, \beta_i)}{d^2(\boldsymbol{x}_j, \beta_k)} \right]^{1/(m-1)}}, && \text{if } I_j = \emptyset \\ u_{ij} &= 0 \quad i \notin I_j \\ \sum_{i \in Ij} u_{ij} &= 1 \quad i \in I_j \end{aligned}\right\}, \qquad \text{if } I_j \neq \emptyset \tag{6}$$

where $I_j = \{i | 1 \leq i \leq C, d^2(\boldsymbol{x}_j, \beta_i) = 0\}$. Prototype-based FCM's share the problem of high sensitivity to noise with all least-squares (LS) approaches. The influence of noise points can be reduced if the memberships associated with them are small in all clusters. However, as can be seen from (6), the memberships u_{ij} generated by the third constraint in (4) are relative numbers. This means that noise points and outliers will also have significantly high membership values and they can severely affect the prototype parameter estimates.

FCM-type algorithms usually use a fixed-point iteration scheme to find the solution of the minimization problem. This particular scheme makes the FCM-type algorithms susceptible to local minima. The issue of global convergence may be addressed by the use of multiple initializations at the cost of increased computations. However, noise in the data set can exacerbate the situation by creating many spurious local minima. Noise can also drastically distort the solution corresponding to the global minimum, and there is no way to handle this problem in the FCM formulation. This drawback has motivated researchers to seek alternative formulations. Before discussing these alternative approaches in more detail, we briefly address the other major problem with prototype-based clustering, i.e., the problem of unknown number of clusters.

The problem of unknown number of clusters is related to cluster validity. Cluster validity measures evaluate the "goodness" of a partition generated by a clustering algorithm. In general, clusters that are compact and dense are preferable. Ideally, the objective function used in the clustering algorithm should incorporate the desired validity measure, since the global optimum of the objective function would then correspond to the most "valid" solution. However, many validity measures proposed in the literature are too complex to optimize. The sum of intracluster distances is one of the most commonly used validity measures because of its analytical tractability. However, it favors spherical clusters of equal size unless distance measures other than the Euclidean distance are used [32]. Moreover, it does not capture the notion of density. Therefore, many researchers have suggested that the partition be evaluated on the basis of other validity measures to verify the goodness of the solution obtained by minimizing the sum of intracluster distances.

What happens when the number of clusters C is not known? The classical solution to this problem has been to apply a given clustering algorithm for a range of C values, and to evaluate the validity of the resulting partition in each case [3], [16], [28], [29]. The partition exhibiting the optimal validity is chosen as the true partition. There are several problems with this approach. First, the clustering must be performed for every value of C over a range, resulting in a large number of computations. Second, for each value of C, there is no guarantee that the resulting partition represents the global minimum. Third, the cluster validity tests are not reliable, particularly if noise is present. This has resulted in several other techniques which are based on the idea of cluster removal and/or merging [10], [11], [27], [28], [30], [33]. In progressive clustering [11], [27], [28], [30], [33], the number of clusters is overspecified, and "good" clusters found at any iteration are progressively removed. In compatible cluster merging, a cluster merging step is also used [10], [11], [27], [28], [30], [33]. Another variation of the progressive clustering technique is to seek one cluster at a time until no more "good" clusters can be found. This technique is used in many algorithms discussed in this paper [26], [27], [43], [47]. These techniques are more efficient than the classical method, but they are quite sensitive to the validity or "goodness of fit" measure used to evaluate individual clusters. For example, some of the algorithms discussed in this paper [26], [47] assume that the cluster distribution is Gaussian and use a "normality test" to determine the validity. It is to be noted that progressive clustering methods require validity measures for individual clusters as opposed to validity measures for the partition as a whole. These two types of validity measures can be quite different in nature.

As mentioned in Section I, the problem of unknown number of clusters is strongly linked to robust clustering. It is important to define the validity measure in such a way that its global optimum does correspond to the desired solution even when the data contains noise and outliers. This means that we should somehow exclude noise points while computing validity. In Section XI, we revisit the problem of validity and elaborate on the critical role it needs to play in robust clustering.

III. Prototype-Based Robust Clustering Methods

In an unpublished presentation, Ohashi [36] (as cited in [12]) made an attempt to handle the problem of noise sensitivity in the FCM-type algorithms. He introduced the idea of a class of outliers, and modified the objective function in (5) as

$$J(B, U; X) = \alpha \sum_{i=1}^{C} \sum_{j=1}^{N} (u_{ij})^m d^2(\boldsymbol{x}_j, \beta_i) + (1-\alpha) \sum_{j=1}^{N} (u_{*j})^m \tag{7}$$

where u_{*j} is the membership of point $\boldsymbol{x}_j$ in the class of outliers and the parameter α is prespecified. Although Ohashi may have been the first to "robustify" the FCM algorithm, unfortunately his work seems to be virtually unknown to the clustering community.

In [7], Davé independently proposed the idea of a *noise* cluster to deal with noisy data. In this approach referred to as the NC approach, the *noise* is considered to be a separate class and is represented by a prototype that has a constant distance, δ, from all the data points. The membership u_{*j} of a point $\boldsymbol{x}_j$ in the noise cluster is defined to be

$$u_{*j} = 1 - \sum_{i=1}^{C} u_{ij}. \tag{8}$$

Thus, the membership constraint for the good clusters is effectively relaxed to

$$\sum_{i=1}^{C} u_{ij} < 1.$$

This allows noise points to have arbitrarily small membership values in good clusters. Thus, the objective function in the NC approach is

$$J(B, U; X) = \sum_{i=1}^{C} \sum_{j=1}^{N} (u_{ij})^m d^2(\boldsymbol{x}_j, \beta_i) + \sum_{j=1}^{N} \delta^2 \left(1 - \sum_{i=1}^{C} u_{ij}\right)^m. \tag{9}$$

It was shown in [7], [8], and [10] that this approach is quite successful in improving the robustness of a variety of prototype-based clustering algorithms, and that it is applicable to many practical situations. The following membership update equation for this formulation can be derived by differentiating (9) with respect to u_{ij}:

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left[\frac{d^2(\boldsymbol{x}_j, \beta_i)}{d^2(\boldsymbol{x}_j, \beta_k)}\right]^{1/(m-1)} + \left[\frac{d^2(\boldsymbol{x}_j, \beta_i)}{\delta^2}\right]^{1/(m-1)}}. \tag{10}$$

The second term in the denominator of (10) becomes quite large for outliers, resulting in small membership values in all the good clusters for outliers. It is easily seen that the formulations in (7) and (9) can be considered equivalent if α in (7) is chosen such that

$$\alpha = \frac{1}{1+\delta^2}. \tag{11}$$

The formulation in (9) is easier to interpret than the one in (7), since the second term on the right-hand side of (9) corresponds to the weighted sum of distances to the *noise* cluster. The NC approach was introduced primarily to make the FCM-type algorithms less sensitive to noise and outliers by relaxing the constraint on the memberships so that the sum of memberships of a noise point in all the good clusters is not forced to be equal to one. The major advantage of this approach is that it is a robustified version of the FCM algorithm and can be easily used instead of the FCM algorithm provided a suitable value for δ can be found.

Another prototype-based approach, called the PCM algorithm was introduced in [31] to overcome the relative membership problem of the FCM. The primary objective of the possibilistic approach is to achieve membership values that are possibilistic, i.e., the membership value of a point in a class represents the typicality of the point in the class or the possibility of the point belonging to the class. This was done by modifying the objective function in (5) as follows:

$$J(B, U; X) = \sum_{i=1}^{C} \sum_{j=1}^{N} (u_{ij})^m d^2(\boldsymbol{x}_j, \beta_i) + \sum_{i=1}^{C} \eta_i \sum_{j=1}^{N} (1-u_{ij})^m. \tag{12}$$

In (12), the η_i are suitable positive numbers, and there are no constraints on the memberships u_{ij} other than the requirement that they should be in [0, 1]. The second term forces u_{ij} to be as large as possible, thus avoiding the trivial solution. The membership update equation in the PCM can be shown to be

$$u_{ij} = \frac{1}{1 + \left[\frac{d^2(\boldsymbol{x}_j, \beta_i)}{\eta_i}\right]^{1/(m-1)}}. \tag{13}$$

It may be noted that the noise distance δ^2 in the NC approach and the weights η_i in the PCM are similar, but the PCM has the advantage that the weight is different for each cluster. This, however, can be also interpreted as one noise class per good cluster. Thus, there are really C-noise classes in the PCM while there is one in the NC. Second, the membership in the second term, which can be considered as the membership in the noise cluster, is handled differently in the two algorithms. In the NC approach, the membership in the noise cluster is equal to the complement of the sum of memberships in all the good clusters. In the PCM, since there is one noise class per cluster, the membership is simply the complement of the membership in the good class. Thus, the NC algorithm would behave like a robustified FCM algorithm, while the PCM algorithm would behave like a collection of C independent NC algorithms, each looking for a single cluster.

In the PCM, the prototypes are automatically attracted to dense regions in feature space as the iterations proceed, i.e., it is a mode-seeking algorithm. This can be shown as follows. Solving for $d^2(\boldsymbol{x}_j, \beta_i)$ in terms of u_{ij} from (13), and eliminating the $d^2(\boldsymbol{x}_j, \beta_i)$ term from the PCM objective

function in (12), we obtain

$$J(B, U; X) = \sum_{i=1}^{C} \eta_i \sum_{j=1}^{N} (1 - u_{ij})^{m-1}.$$

The above objective function represents the following C-independent objective functions:

$$J_i(\beta_i, U_i; X) = \eta_i \sum_{j=1}^{N} (1 - u_{ij})^{m-1}, \qquad i = 1, \cdots, C.$$

For a given value of η_i, each of the C-objective functions is minimized by maximizing

$$\begin{aligned} J_i'(\beta_i, U_i; X) &= \eta_i \sum_{j=1}^{N} [1 - (1 - u_{ij})^{m-1}] \\ &= \eta_i \sum_{j=1}^{N} u_{ij}' \end{aligned}$$

where $u_{ij}' = [1 - (1 - u_{ij})^{m-1}]$ can be interpreted as a modified membership. Since u_{ij}' is obtained from u_{ij} via a monotonic mapping, u_{ij} varies the same way as u_{ij}, i.e., $u_{ij} = 0 \Rightarrow u_{ij}' = 0$; $u_{ij} = 1 \Rightarrow u_{ij}' = 1$. Furthermore, for the special case of $m = 2$, the above subobjective function reduces to

$$J_i'(\beta_i, U_i; X) = \eta_i \sum_{n=1}^{N} u_{ij}. \tag{14}$$

Thus, we see that for a given value of η_i, each of the C subobjective functions is maximized by choosing the prototype location such that the sum of the (modified) memberships is maximized. This is achieved if the prototype is located in a dense region. If there are indeed C-dense regions in feature space (corresponding to C-distinct clusters), then with proper initialization, each prototype will converge to a dense region. This behavior is also exhibited by several other algorithms discussed in this paper (see for example [2], [5], and [47]).

It is to be noted that the PCM is a particular implementation of the possibilistic approach. For instance, we can choose an alternative second term to prevent the trivial solution as follows:

$$\begin{aligned} J(\beta_i, U_i; X) = &\sum_{i=1}^{C} \sum_{j=1}^{N} (u_{ij})\, d^2(\boldsymbol{x}_j, \beta_i) \\ &+ \sum_{i=1}^{C} \eta_i \sum_{j=1}^{N} (u_{ij} \log u_{ij} - u_{ij}). \end{aligned}$$

Note that $u_{ij} \log u_{ij} - u_{ij}$ is a monotonically decreasing function in [0, 1], similar to $(1 - u_{ij})^m$. By setting the derivative of the above objective function with respect to u_{ij} to zero, it can be easily shown that the update equation for u_{ij} is

$$u_{ij} = \exp\left\{-\frac{d^2(\boldsymbol{x}_j, \beta_i)}{\eta_i}\right\}. \tag{15}$$

Thus, the objective function can be written as

$$\begin{aligned} M(B, U; X) &= -\eta_i \sum_{i=1}^{C} \sum_{j=1}^{N} \exp\left\{-\frac{d^2(\boldsymbol{x}_j, \beta_i)}{\eta_i}\right\} \\ &= -\sum_{i=1}^{C} \eta_i \sum_{j=1}^{N} u_{ij}. \end{aligned} \tag{16}$$

If the prototype of each cluster can be represented by the cluster center $\boldsymbol{c}_i$, then it can be easily verified that the prototype update equation for this version of the PCM can be written as

$$\boldsymbol{c}_i = \frac{\sum_{j=1}^{N} u_{ij} \boldsymbol{x}_j}{\sum_{j=1}^{N} u_{ij}}. \tag{17}$$

Equations (15) and (17) can be used in an alternating fashion in an iterative algorithm to estimate the cluster centers $\boldsymbol{c}_i$. In Sections V and VII, we show that this version of the PCM is related to the potential function approach and deterministic annealing/least-biased fuzzy clustering approach. In general, for different choices of the second term in (12), we get different algorithms with different monotonically decreasing membership functions. We refer to the family of such algorithms as PC algorithms, and the general approach as the PC approach.

The objective function of the PC approach is a collection of C-independent functions. Hence, when the specified value of C is smaller than the actual value, the algorithm can still find the specified number of *good* clusters from a data set, as long as each η_i is estimated correctly and good initializations for the prototypes are provided. However, if the initializations are poor, it is quite possible that the PCM will converge to a "meaningless" solution where all clusters found are identical while other clusters go undetected. Similarly, when the specified value of C is greater than the actual value, then the algorithm can still find C-good clusters out of which some of them may be identical. One needs to choose a sufficiently large number of distinct initial prototypes to ensure that all the clusters in the data set are found. (This behavior is different from that of the FCM, which will arbitrarily split or merge real clusters in the data set to produce exactly the specified number clusters.) In other words, the specification of C for the PCM is somewhat arbitrary. In particular, if C is chosen to be one, which is a more meaningful way of using the PCM algorithm, then it will find one good cluster. Thus, it can be seen that the PCM approach has the potential for solving the other major problem with the FCM, i.e., the need to know the number of clusters. (See also [27], and CRE in Section IX.)

When the PCM is used in its naturally more meaningful sense, i.e., for $C = 1$, it is identical to the NC algorithm. This follows because for $C = 1$, the objective functions in (9) and (12) are identical if

$$\delta^2 = \eta. \tag{18}$$

In (18), the subscript on η has been dropped because there is only one cluster. We refer to this special version as the

noise/possibilistic clustering algorithm or the N/PC1 algorithm where the number one signifies that only one cluster is searched for at a time. Furthermore, for this special case, all the three methods described in this section become identical. Thus, the discussion regarding the more general PC approach where the memberships are arbitrary monotonically decreasing functions of the distance equally applies to the NC and Ohashi algorithms when $C = 1$. In the subsequent sections, we treat the terms PCM and N/PC1 as loosely interchangeable. In most cases, discussion about the PCM applies equally to the N/PC1, but comparisons are made with the PCM if the more general formulation of the PCM for arbitrary C is also comparable.

The central problem with the three algorithms discussed in this section is that they require us to specify what we call a resolution parameter. The Ohashi algorithm requires the specification of α, the NC requires the specification of δ, and the PCM requires the specification of η. In Section VII, we show that these parameters relate to the concept of "scale" in robust statistics. Hence, robust methods may be used to obtain better estimates for these parameters in noisy situations. The next serious problem is the initialization of cluster prototypes. In theory, this problem does have a solution since one can try all possible initializations. (In many instances, the total number of all possible initializations is finite. For example, when one cluster is sought at a time and cluster centers are used as prototypes, we could limit the total number of possible initializations to the number of feature vectors N where, in each initialization, one of the feature points is used as the center. This solution, however, is quite impractical for large N.) For a given initialization, if a good estimate of the resolution parameter is available, the clustering algorithm will converge to a local minimizer, which will hopefully correspond to a *good* cluster. When a good cluster is found, it can be removed from the data set, and the remaining points can be further processed. This, however, brings us to another elusive problem, i.e., how does one decide if a detected cluster is good? The answer to this question lies in the definition of validity or *goodness of fit* measures, which invariably implies that we make assumptions about the distribution of points in a cluster (see Section II). We will come back to this issue again in Section XI.

IV. The Potential Function Approach and the Mountain Method

The potential function approach is a classical approach that has been used for the design of automatic pattern classification systems [44]. In this approach, the feature points are likened to energy sources. The potential generated by each feature point $\boldsymbol{x}_k$ has a peak value at the location of the feature point and decreases rapidly at any point $\boldsymbol{x}$ away from the feature point. Typical functions used as potential functions include

$$P(\boldsymbol{x}, \boldsymbol{x}_k) = \exp\{-\alpha||(\boldsymbol{x} - \boldsymbol{x}_k)||^2\}$$

and

$$P(\boldsymbol{x}, \boldsymbol{x}_k) = \frac{1}{1 + \alpha||\boldsymbol{x} - \boldsymbol{x}_k||^2}.$$

It can be seen that the above potential functions are similar to the membership functions in (13) and (15). The total potential at any point $\boldsymbol{x}$ in feature space can be defined as the sum of the potentials contributed by each feature point $\boldsymbol{x}_k$ at $\boldsymbol{x}$. It is reasonable to assume that the total potential in the "dense" regions of feature space (i.e., areas where there are many feature points close to one another) will be high. Therefore, the peaks of the total potential function correspond to cluster prototypes and valleys correspond to decision boundaries between clusters. Although the potential function approach was not proposed as a clustering algorithm, this idea can be used to define an objective function to locate clusters. In the case of the exponential potential function given above, the objective function, which is equal to the total potential, is given by

$$P_t(B; X) = \sum_{k=1}^{N} \exp\{-\alpha||(\boldsymbol{x} - \boldsymbol{x}_k)||^2\}. \tag{19}$$

We see that, for a fixed value of α or η_i, (16) and (19) are identical if we seek one cluster at a time. In this light, the objective function for the potential function approach is the same as that of the N/PC1. Each peak (local maximum) in the objective function of (19) corresponds to one cluster.

Traditionally, the peaks in the potential function are found using a direct search. The "mountain method" proposed by Yager and Filev [45] is a specific implementation of this approach, even though this connection is not mentioned in [45]. Since the implementation of the mountain method requires setting up a grid in feature space in order to search for the peaks directly, it becomes impractical for higher dimensions. Chiu and Cheng [4] proposed an improved implementation through a simple but significant change. In their approach, rough estimates of the local population densities are used to determine cluster centers in a serial fashion. They compute the total potential $P(\boldsymbol{x}_i)$ at a data point $\boldsymbol{x}_i$ as

$$P_t(\boldsymbol{x}_i) = \sum_{j=1}^{N} \exp\{-\alpha||(\boldsymbol{x}_i - \boldsymbol{x}_j)||^2\} \tag{20}$$

where α represents a resolution parameter which is assumed to be known or specified by the user. Note that feature space is considered to be isotropic, since no covariance matrix is used in the exponent. The data point with the highest local density is considered to be the first cluster center. Let $\boldsymbol{x}_1^*$ denote the first cluster center and let $P_t(\boldsymbol{x}_1^*)$ denote the potential at $\boldsymbol{x}_1^*$. In the next step, this cluster is "removed" from the data set by discounting the contribution due to the cluster center to the total potential at every point according to

$$P_t(\boldsymbol{x}_i)^{\text{new}} = P_t(\boldsymbol{x}_i)^{\text{old}} - P_t(\boldsymbol{x}_1^*) \exp\{-\beta||(\boldsymbol{x}_i - \boldsymbol{x}_1^*)||^2\}$$

where β represents another resolution parameter which is assumed to be known or specified by the user. At this point, the next highest density location is found, and this process is repeated until a certain termination criterion is met. The value of β is chosen to be larger than that of α to avoid a result with cluster centers too close to one another. It is claimed that this method is very fast compared to the FCM approach, since no

iterations are required. This may be true for small values of N, but as N becomes large it is no longer the case, as can be seen by the following argument. Computing the potential at each of the N points according to (20) requires O(N) computations and, therefore, the computation of the total potential, (before removing the first cluster) is O(N^2). Removal of each cluster is again O(N). Therefore, for C clusters, the computational complexity of this algorithm is O($N^2 + CN$) where each computation involves the exponential function. For the FCM, the computational load is O(CNI) where I is the number of iterations and each computation involves calculating the distance and the memberships. Therefore, when $N \gg CI$, the improved mountain method will become computationally more expensive than the FCM. Besides, in practice, one may have to run this method for a series of α and β values.

Since the potential function is computed only at some points in feature space, the mountain method gives us only rough estimates of cluster centers. Although issues relating to noise are not addressed in [4] and [45], this method can be quite robust if the clusters are roughly spherical and if suitable values for α and β can be found. This is because noise points will not adversely affect the peak locations of the potential function. The main disadvantage of this approach is that the results will be quite sensitive to the values chosen for the resolution parameters α and β. A large value for α will produce a single peak and a small value will produce many small spurious peaks in the total potential function P_t. A fixed relation between the values of α and β may not work if the clusters have widely varying sizes and shapes. Moreover, since the values of α and β are not determined from the data set, and since no cluster validity is used, one has no way of knowing if the "cluster centers" obtained using this algorithm actually represent any clusters.

Rather than using a direct search, one can use the objective function in (19) to find one cluster center at a time as follows. Let $\boldsymbol{c}_i$ denote a cluster center. Since the cluster center corresponds to a local maximum of the objective function in (19), the necessary condition for $\boldsymbol{c}_i$ can be derived by differentiating (19) with respect to $\boldsymbol{x}$ and setting it equal to zero. This gives us

$$\boldsymbol{c}_i = \frac{\sum_{j=1}^{N} u_j \boldsymbol{x}_j}{\sum_{j=1}^{N} u_j} \tag{21}$$

where

$$u_j = \exp\{-\alpha \|(\boldsymbol{c}_i - \boldsymbol{x}_j)\|^2\}. \tag{22}$$

It is to be noted that (21) is an implicit equation, since u_j depends on $\boldsymbol{c}_i$. Therefore, (21) and (22) need to be used in an alternating fashion to solve for the centers. Comparing (21) and (22) with (17) and (15), we see that this iteration scheme is identical to the PC technique, where the cluster centers are sought independently of one another. In other words, the peaks in the potential function approach correspond to locally dense regions and, hence, to the convergence points of the PC technique.

The mountain method can be improved by performing a search over a range of resolution parameters α and β, which will make it essentially equivalent to Beni and Liu's algorithm based on deterministic annealing (see next section). Approaches similar to the mountain method for locating peaks in smoothed multidimensional histograms of feature points have been used successfully in color image segmentation [37]. The Parzen window method [13] to estimate probability density functions (PDF's) also uses a similar technique. The idea of identifying one peak at a time is similar to progressive clustering [11], [27], [28], [30].

V. Deterministic Annealing and Least-Biased Fuzzy Clustering (LBFC)

In [39] and [40], Rose *et al.* proposed a clustering algorithm based on deterministic annealing. It can be shown that the updating equations in this algorithm are a special case of the expectation maximization (EM) algorithm of Dempster *et al*. However, since the memberships are relative in this algorithm, from the point of view of robustness, it still suffers from the same drawbacks as the FCM algorithm. Another "least-biased fuzzy clustering (LBFC) method" based on the deterministic annealing approach in [39] and [40] has been proposed by Beni and Liu [2]. This algorithm tries to minimize the "clustering entropy" given by

$$S = -\sum_{j=1}^{N} p_j(\boldsymbol{x}_j, \boldsymbol{c}_i) \log p_j(\boldsymbol{x}_j, \boldsymbol{c}_i) \tag{23}$$

subject to the assumption that the centroids are "unbiased," i.e.,

$$\sum_{j=1}^{N} (\boldsymbol{x}_j - \boldsymbol{c}_i)\, p_j(\boldsymbol{x}_j, \boldsymbol{c}_i) = 0$$

where $p_j(\boldsymbol{x}_j, \boldsymbol{c}_i)$ represents the probability that centroid $\boldsymbol{c}_i$ will cluster point $\boldsymbol{x}_j$. Maximizing the clustering entropy leads to

$$p_j(\boldsymbol{x}_j, \boldsymbol{c}_i) = \frac{\exp\{-\beta D(\boldsymbol{x}_j, \boldsymbol{c}_i)\}}{\sum_{j=1}^{N} \exp\{-\beta D(\boldsymbol{x}_j, \boldsymbol{c}_i)\}} \tag{24}$$

where β is a constant ("resolution parameter") related to Lagrange multipliers in the maximization process and $D(\boldsymbol{x}_j, \boldsymbol{c}_i)$ is the city-block distance defined as

$$D(\boldsymbol{x}_j, \boldsymbol{c}_i) = \sum_{k=1}^{n} |x_{jk} - c_{ik}|.$$

In the above, n is the dimensionality of the data set. The centroid update equation is

$$\boldsymbol{c}_i = \sum_{j=1}^{N} p_j(\boldsymbol{x}_j, \boldsymbol{c}_i) \boldsymbol{x}_j. \tag{25}$$

The clustering algorithm consists of starting with one of the data points as the initial centroid and, then, alternately applying the update (24) and (25) until convergence. Since

the clusters are assumed to be independent of one another, only one centroid is chosen and updated at a time. It is to be noted that the normalization factor in (24) involves the memberships of all data points in a given cluster, and not the memberships of a particular point in all clusters (as in the case of the FCM algorithm). We can ignore the normalization factor in (24) and obtain equivalent results, provided we include the normalization factor in (25). With this change, the updating (25) and (24) are identical to (15) and (17) except for the distance measure. Thus, we see that this algorithm is equivalent to the PC approach.

Beni and Liu suggest that for a particular value of β, the algorithm be applied with every data point as the initial centroid. This will result in only a small number of different final centroids, since for many initializations, the final value of the centroid will be the same. As in the case of the PC and potential function approaches, the centers will correspond to locally dense regions in feature space. The number of different final centroids for a particular value of β is taken to be the "optimum" number of clusters at resolution β. This procedure is repeated over a range of β values. The minimum value of β is zero, and the maximum value denoted by $\beta_{\max}$ is related to the data accuracy. Beni and Liu suggest several ways to compute the value of $\beta_{\max}$. The concept of varying the resolution β in this approach is comparable to varying "δ" in the NC approach, "η" in the PCM approach, "h" in the GMVE approach (see Section VIII), or "t" in the Gaussian mixture density decomposition algorithm (see Section X). For a given value of this parameter, the clustering process results in a particular number of clusters. The true number of clusters is then decided based on the most frequently found number of clusters over the whole range of values of β.

Although interesting, this approach still has several limitations. First, it assumes that the feature space is isotropic (i.e., no covariance matrix is used in the Gaussian). Second, for a given resolution parameter β, one would generally find only clusters of a particular size. Thus, the "optimum" number of clusters obtained using the suggested procedure will not be reliable if the clusters vary widely in size. The level of computation required is also high for this algorithm. This is because a large value of N corresponds to a large number of initial cluster centroid candidates and, thus, many runs of the algorithm are required for each value of β. The noise sensitivity or robustness aspects of this method are not addressed by the authors in [2]. One may be tempted to conjecture that this algorithm is robust, since it is equivalent to the N/PC1 approach. However, the procedure to find the cluster centers and the number of clusters will break down when there is noise and outliers, because the algorithm will find many spurious centers corresponding to noise and outliers, in addition to the true centers.

VI. The Iteratively Reweighted Least Squares and the PCM

The iteratively reweighted least squares (IRLS) method is a popular technique in computer vision [21], [34], [46]. It is used to obtain a robust estimate of parameters starting from a rough initial estimate. The use of such a technique is also popular in robust statistics [1], [24], [42]. The weights are monotonically decreasing functions of the residuals. Here we show the connection between the weighted LS approach and the possibilistic and noise clustering methods.

The second term in the objective function of PCM in (12) was added so that we do not obtain a trivial solution when we minimize the objective function. However, the same effect can be achieved if we assume that the memberships are of the form given in (13) and substitute this expression for the memberships back in the objective function. This eliminates memberships from the objective function in (9) and after simplification we obtain

$$J(B; X) = \sum_{i=1}^{C} \sum_{j=1}^{N} \left(\frac{1}{1 + \left[\frac{d_{ij}^2}{\eta_i} \right]^{1/(m-1)}} \right)^{m-1} d_{ij}^2. \quad (26)$$

It is easily shown that the prototype parameter values that minimize (26) also minimize (12), and vice versa [22]. In other words, the PCM algorithm is equivalent to minimizing (26) with respect to the prototype parameters and then obtaining the membership values via (13) after the minimization has been done. If we are concerned only with the prototype parameter values, then the PCM algorithm is simply equivalent to minimizing (26). If we define

$$w_{ij}(d_{ij}^2; \boldsymbol{a}_i) = \left(\frac{1}{1 + \left[\frac{d_{ij}^2}{\eta_i} \right]^{1/(m-1)}} \right)^{m-1} \quad (27)$$

where $\boldsymbol{a}_i$ represents a vector of parameters ($\boldsymbol{a}_i = [\eta_i, m]^T$ in this case), then the objective function for the PCM can be written in the simple form

$$J(B; X) = \sum_{i=1}^{C} \sum_{j=1}^{N} w_{ij}\, d_{ij}^2. \quad (28)$$

It is to be noted that for $m > 1$, w_{ij} is a monotonically decreasing function of d_{ij}^2. This suggests that, in general, we could choose any monotonically decreasing function for w_{ij} to obtain variations of the PCM algorithm. Since (28) can be treated as C independent problems, it is obvious that this generalized formulation of the PCM algorithm is equivalent to the IRLS approach, which is considered to be robust in statistics [42]. In this general formulation, if we assume that the distance measure d_{ij}^2 is an inner-product norm-induced metric, then the update equation for the cluster center c_i can be derived by setting the derivative of (28) with respect to $\boldsymbol{c}_i$

to zero. This gives us

$$\boldsymbol{c}_i = \frac{\sum_{j=1}^{N} w_{ij}\boldsymbol{x}_j}{\sum_{j=1}^{N} w_{ij}}. \tag{29}$$

In obtaining (29), the weights w_{ij} are treated as constants as is commonly done in IRLS methods. The solution can be obtained iteratively, by updating the weights w_{ij} in each iteration using the new value of $\boldsymbol{c}_i$. Thus, we see that the weights in the IRLS technique play the role of memberships. Although the NC and Ohashi algorithms cannot be written in the form in (28) for arbitrary values of C, for $C = 1$ they do reduce to the weighted LS approach. This, in part, explains the robust behavior of these methods. We now look into robust statistics in more detail in order to obtain further insights into robust clustering.

VII. Relation Between Robust Statistics and (ROBUST) Fuzzy Clustering

The classical LS technique uses the solution corresponding to

$$\underset{\theta}{\text{Minimize}} \sum_{j=1}^{N} r_j^2$$

where r_j is the residual associated with observation $\boldsymbol{x}_j$, θ is a parameter vector to be estimated and N is the number of observations. For the purposes of the discussion in this section, we consider $\boldsymbol{x}_j$ and θ to be scalars and define the residual as

$$r_j = x_j - \theta.$$

It was realized that the LS technique is extremely sensitive to noise and outliers. Therefore, many robust methods were developed in statistics to overcome the noise sensitivity of the LS [18], [20], [25], [42]. Several different classes of robust methods (such as the M, R, L estimators, and the least-median of squares method) exist [18], [20], [25], [42]. Here we discuss the M estimator [25] in detail and show the connection between robust statistics and fuzzy clustering, in general, and the M estimator and the N/PC1 technique, in particular.

The M estimator uses a suitable symmetric positive–definite function (called the robust-loss function) $\rho(r)$ and forms the objective function by summing the loss over all points. Thus, in the M estimator approach, the objective function to be minimized can be written as

$$J(\theta) = \sum_{j=1}^{N} \rho\,(r_j). \tag{30}$$

If we let $\psi(r)$ be the derivative of $\rho(r)$, then a necessary condition for the minimum is obtained by setting the derivative of (30) with respect to the parameters θ to zero, i.e.,

$$\sum_{j=1}^{N} \psi(r_j)\,\frac{dr_j}{d\theta} = 0. \tag{31}$$

TABLE I
Examples of M-Estimators with their Associated Functions

Estimator with tuning constant	Loss Function,	ψ-Function	Weight Function, w	Range of r	Commonly used scale of r
Mean	$\frac{1}{2}r^2$	r	1	$-\infty$ to $+\infty$	none
Median	$\lvert r\rvert$	$\operatorname{sgn}(r)$	$\frac{\operatorname{sgn}(r)}{r}$	$-\infty$ to $+\infty$	none
Huber (k) $0<k$	$\frac{1}{2}r^2$ $k\,\lvert r\rvert - \frac{1}{2}k^2$	r $k\,\operatorname{sgn}(r)$	1 $\frac{k\,\operatorname{sgn}(r)}{r}$	$\lvert r\rvert \le k$ $\lvert r\rvert > k$	MAD
Cauchy (c) $0<c$	$\frac{c^2}{2}\log\left[1+\left(\frac{r}{c}\right)^2\right]$	$r\left[1+\left(\frac{r}{c}\right)^2\right]^{-1}$	$\left[1+\left(\frac{r}{c}\right)^2\right]^{-1}$	$-\infty$ to $+\infty$	none
Tukey's biweight(c) $0<c$	$\frac{1}{6}[1-(1-r^2)^3]$ $\frac{1}{6}$	$r\,(1-r^2)^2$ 0	$(1-r^2)^2$ 0	$\lvert r\rvert \le 1$ $\lvert r\rvert > 1$	$c\times MAD$
Hampel $0<a\le b\le c$ $a_1 = ab - \frac{1}{2}a^2$ $a_2 = (c-b)\frac{a}{2}$	$\frac{1}{2}r^2$ $a\lvert r\rvert - \frac{1}{2}r^2$ $a_1 + a_2\left[1-\left(\frac{c-\lvert r\rvert}{c-b}\right)^2\right]$ a_1+a_2	r $a\,\operatorname{sgn}(r)$ $\left(\frac{c-\lvert r\rvert}{c-b}\right)\operatorname{sgn}(r)$ 0	1 $\frac{a\,\operatorname{sgn}(r)}{r}$ $\left(\frac{c-\lvert r\rvert}{c-b}\right)\frac{\operatorname{sgn}(r)}{r}$ 0	$\lvert r\rvert \le a$ $a < \lvert r\rvert \le b$ $b < \lvert r\rvert \le c$ $\lvert r\rvert \ge c$	MAD
Andrews $0<c$	$[1/\pi^2](1-\cos\pi r)$ $2/\pi^2$	$\frac{1}{\pi}\sin\pi r$ 0	$\frac{1}{\pi r}\sin\pi r$ 0	$\lvert r\rvert \le 1$ $\lvert r\rvert > 1$	$c\times MAD$

In most cases, the solution of (30) is the same as that of (31). The problem defined by (31) [or (30)] is called the M estimator, and the implicit form in (31) allows one to select many different forms for the function $\psi()$, as will be discussed later. It is, in fact, customary to use (31) as the formulation of the M estimator. If θ simply consists of a single term, then $dr_j/d\theta$ in (31) is unity and can be removed, and the resulting M-estimate is the robust estimate of location of the sample. It can be easily shown that if the loss function is chosen to be $\rho(r) = 1/2r^2$ [i.e., if $\psi(r) = r$], then the resulting location estimate is the sample mean, and if $\rho(r) = |r|$ [i.e., if $\psi(r) = \operatorname{sgn}(r)$], the estimate is the sample median. Several ρ functions have been proposed (see Table I), which reduce the influence of large residuals on the estimated fit. Given a starting value of the parameter θ, Newton's method can be used to obtain a solution to (31) iteratively. Another way to solve this problem is to reformulate the M estimator and obtain the W estimator [18], [24], as explained below.

Let $w(r)$ be defined according to $rw(r) = \psi(r)$. Substituting for $\psi(r)$ in (31) for location estimation, we obtain

$$\sum_{j=1}^{N} (x_j - \theta)\, w(x_j - \theta) = 0.$$

Rearranging, we get

$$\theta = \frac{\sum_{j=1}^{N} w(x_j - \theta)x_j}{\sum_{j=1}^{N} w(x_j - \theta)}. \tag{32}$$

Thus, θ is a weighted mean of the x_j and can be solved for iteratively. In fact, this is one of the standard ways of

computing the M estimator (see [25, pp. 146–147], [14], and [24]). In the sense of the IRLS formulation, the N/PC1 algorithm is equivalent to the M estimator, and the PCM algorithm can be thought of as C-independent M estimators. From another view point, comparing (32) and (17), we see that $w(x_j - \theta)$ plays the role of the possibilistic membership or the membership in the good class as opposed to the noise class. The N/PC1 algorithm can be formulated as a W estimator by letting $C = 1$ in (9) or (12) and setting the derivative of the resulting function with respect to the parameter vector to zero

$$\sum_{j=1}^{N} (u_j)^m d_j \frac{dd_j}{d\beta} = 0. \tag{33}$$

When (33) is compared with (31), it is easy to see that $(u_j)^m d_j$ is equivalent to the ψ function of the M estimator and consequently, $(u_j)^m$ is the weight function if we set $r_j = d_j$. It is understood that the memberships u_j are computed using (10) or (13) for $C = 1$. Another way to derive (33) is to take the objective function in (26) for $C = 1$ (i.e., the weighted LS form), and then differentiate it with respect to β. This approach requires several simplifying steps before we can arrive at (33). From (33), the equivalent ρ, ψ, and w functions for the N/PC1 algorithm can be shown to be as follows:

$$\rho_j = \frac{1}{2}\left(\frac{1}{1+\left[\frac{d_j^2}{\eta}\right]^{1/(m-1)}}\right)^{m-1} d_j^2$$

$$\psi_j = \left(\frac{1}{1+\left[\frac{d_j^2}{\eta}\right]^{1/(m-1)}}\right)^{m} d_j$$

and

$$w_j = \left(\frac{1}{1+\left[\frac{d_j^2}{\eta}\right]^{1/(m-1)}}\right)^{m}. \tag{34}$$

In the above, $1 < m < \infty$. The ρ function in the above equation may be obtained by integrating the ψ function. On the other hand, since (33) can also be obtained by differentiating (26), it follows that $w_j d_j^2$ is the integral of ψ where w_j is defined in (27). For $m = 1$, we have hard memberships and we obtain the following:

$$\begin{cases} \rho_j = \dfrac{d_j^2}{2}; & \psi_j = d_j; \quad \text{and } w_j = 1, \quad \text{for } d_j^2 < \eta \\ \rho_j = \dfrac{\eta}{2}; & \psi_j = 0; \quad \text{and } w_j = 0, \quad \text{for } d_j^2 > \eta \end{cases}. \tag{35}$$

It is understood that η and δ^2 are interchangeable in (34) and (35).

The M estimator represented by (33) and (34) is robust, as will be seen later in this section. For different choices of w in (32), different M estimators are obtained. There are a variety of M estimators reported in [18], [20], and [25]. Some of the common ones are listed in Table I. Most of the formulations listed in the table use a scaling factor for the residual r, i.e., the residual is scaled (normalized) before it is used in the loss function. The scaling factor is a robust estimate of dispersion, and helps in distinguishing between inliers and outliers. Many M estimators assign a weight of zero for points with scaled residuals ≥ 1 (see weight functions corresponding to Tukey and Andrews M estimators in Table I). The scaling factor used by Huber is the median of absolute deviations (MAD). Tukey uses $c \times MAD$, where c is an appropriate constant. The value of c is usually 9.0, if the "good" residuals are assumed to be Gaussian distributed [18].

By plotting the weight functions of the various M estimators, we can see how they limit the influence of outlying observations. The robustness of an estimator can be "measured" by several properties besides its breakdown point. For example, one can look at the gross-error sensitivity, efficiency, local-shift sensitivity, and location of the rejection point. A detailed discussion of these properties can be found in [20]. It is desirable that a robust estimator have a low gross-error sensitivity, an efficiency higher than that of the median under the assumed distribution, a low local-shift sensitivity, and a finite-rejection point. The mean estimator does not have most of these properties, mainly because its ρ function is not bounded, as can be deduced from Fig. 1(a). On the other hand, the rest of the estimators listed here do have a bounded influence of the outliers and, thus, exhibit robustness. Estimators such as Tukey's biweight estimator, Andews' sine estimator, and Hampel's estimator [20] have better robustness properties since they zero out the effect of high residuals (i.e., they have finite-rejection points). This can also be seen in Fig. 1. They are called redescending-type estimators.

It is clear from (34) that the N/PC1 approach is an M estimator. It is also essentially a redescending-type estimator, since for a small value of $m (m < 2)$, it more or less has a finite-rejection point and the good properties associated with it. This can be seen in Fig. 2, where the weight function for the N/PC1 algorithm is plotted for various choices of the "fuzzifier" m. From this figure and (33) it is clear that the N/PC1 algorithm is equivalent to a robust M estimator, and the PCM algorithm can be viewed as C independent robust M estimators. This connection clearly explains the robust behavior of the noise clustering and possibilistic clustering approaches reported in [7], [8], [10], [30], and [31]. It also provides a theoretical foundation for the N/PC1 and generalized PC approach from the point of view of robust statistics, and establishes a correspondence between the weight function in robust statistics, the membership function in fuzzy set theory, and the possibility distribution in possibility theory. From Fig. 2, we also see that the value of δ or η basically determines the boundary between the inliers and outliers. Thus, the correspondence between the resolution parameter in fuzzy clustering (δ or η) and scale in robust statistics is obvious. An immediate consequence of this correspondence is that, in

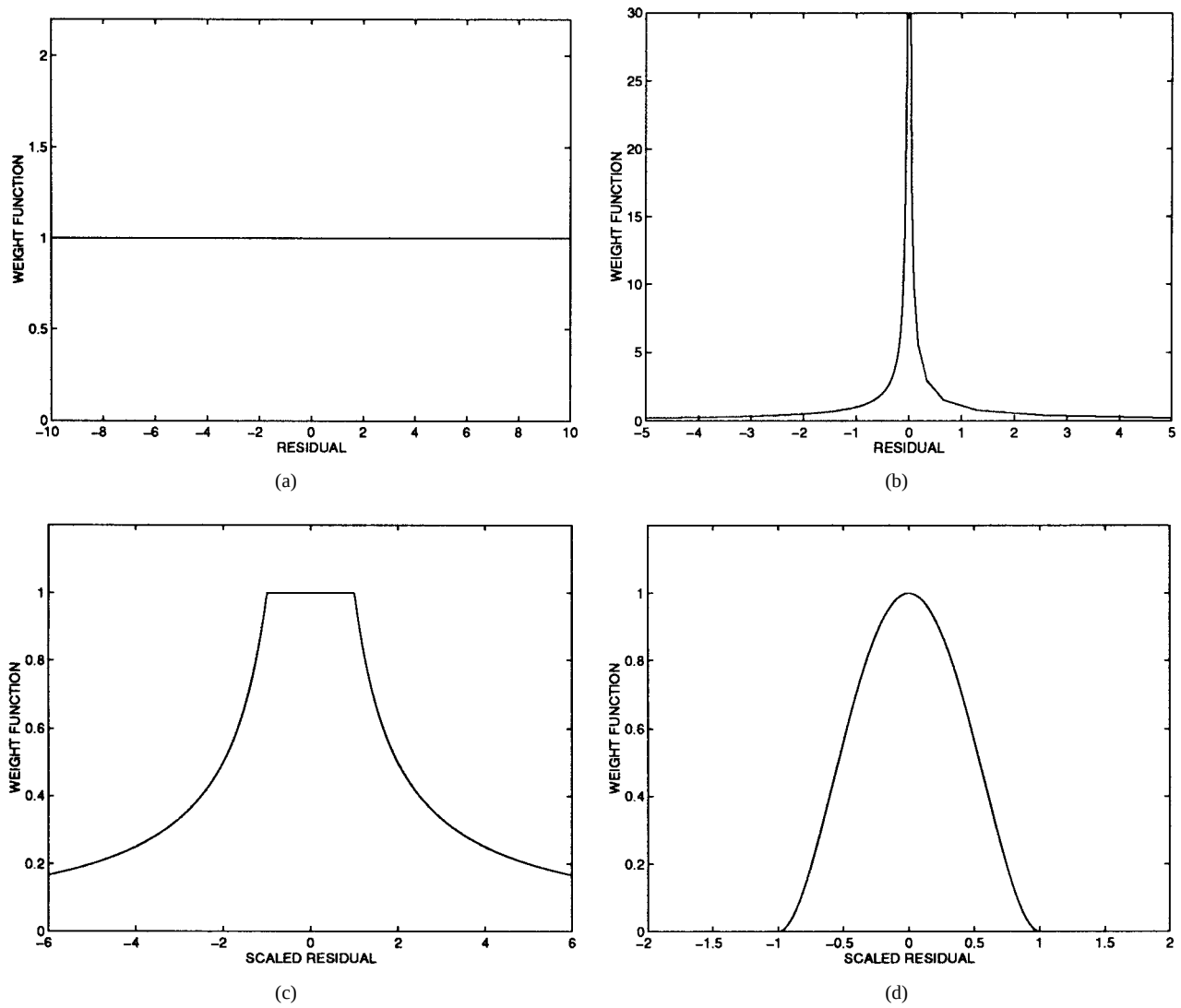


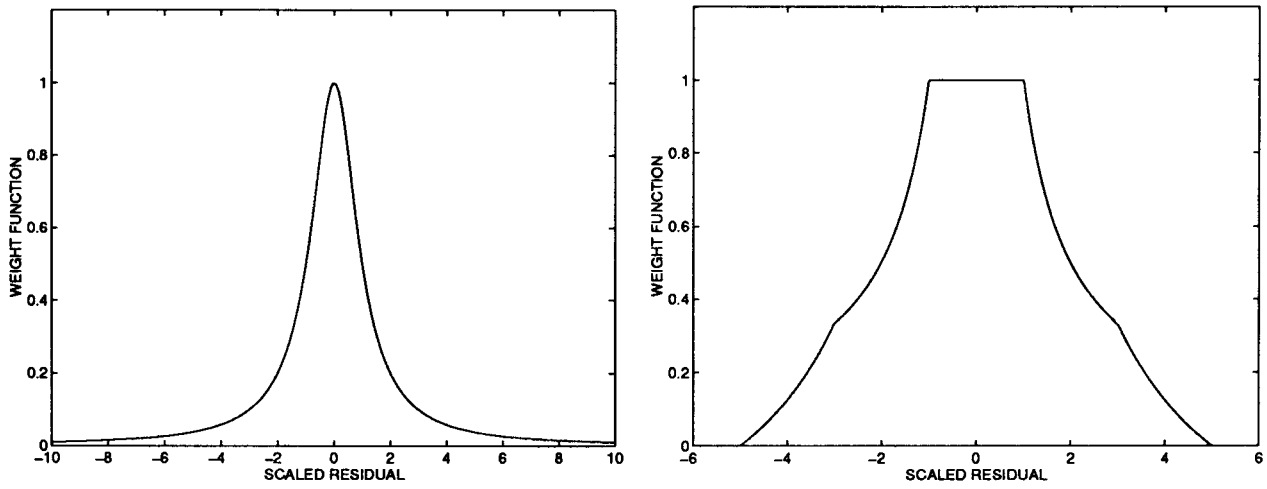
Fig. 1. Plot of weight functions for commonly used estimators: (a) Mean. (b) Median. (c) Huber. (d) Tukey's biweight.

noisy situations, a plethora of techniques from robust statistics can be used to estimate the resolution parameter in fuzzy clustering.

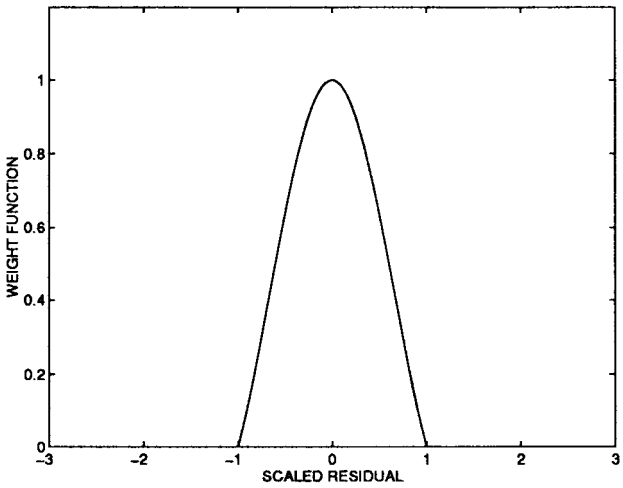
In fuzzy clustering, a value of $m = 1$ corresponds to hard (or crisp) memberships, and increasing values of m represent increasing fuzziness. This behavior is also seen in Fig. 2. Here, $m = 1$ corresponds to a hard rejection rule and is essentially the same as the "skipped means" rejection rule of Huber (see [20, pp. 64–65]). However, for $m > 1$, what one gets is "soft rejection" with a redescending ψ function. The problem with hard rejection is that although it may cope well with distant outliers, it cannot cope well with gross errors and other contamination that is close to the good data, i.e., the contamination in what is called the *region of doubt* (see [20, p. 69]). The efficiency of hard rejection is low in such cases. This is yet another interesting connection between robust statistics (more specifically M estimators) and fuzzy sets. To summarize, fuzzy memberships correspond to soft rejection of outliers, while hard memberships correspond to hard rejection of outliers. By the use of membership functions, fuzzy approaches can model the region of doubt better.

In the case of location estimation, all estimators in Table I other than the mean possess a high breakdown point of 0.5. However, their breakdown point decreases as the dimensionality of the data and/or parameters increases. It can be shown that the theoretical maximum of the breakdown point for M estimators is limited by a certain function of n, where n is the dimension of the data (or number of parameters in regression) [20], [25]. This upper limit is inversely proportional to n, and may not even be attained by some of the M estimators. This upper limit, even when attainable, is itself rather disappointing.

Although the breakdown aspects of the popular M estimators are not encouraging, one must keep in mind that the definition of breakdown allows for any and all types of contamination. In most practical applications where the outliers do not position themselves in such a "contrived" or "wicked" manner, the behavior of the estimator would be reliable. On the positive side, the M estimators are still much better than the usual LS methods whose breakdown point of $1/N$ (where N is number of data points) approaches zero for large N. This argument also applies to the FCM approach, whose breakdown point approaches zero in the limit,



(e) (f)



(g)

Fig. 1. (*Continued.*) Plot of weight functions for commonly used estimators: (e) Cauchy. (f) Hampel. (g) Andrews.

whereas the NC and PCM methods have better breakdown characteristics and, at least for finite-dimensional data, the breakdown point is also finite.

An interesting aspect of the breakdown analysis is that in many engineering applications, one needs methods that have a breakdown point even higher than the theoretical limit of 0.5. This is certainly true for clustering, because with respect to a given cluster, the points in all other clusters are outliers. In that case, a data set with say ten clusters would require a method having a breakdown point higher than 0.9. Since this seems theoretically impossible, does it mean that robust clustering is also impossible? We will come back to this question again in Section XI, after considering two other methods based on robust statistics.

VIII. The Minimum Volume Ellipsoid Method

Motivated in part by the relatively low breakdown point of the M estimators for regression in higher dimensions, Rousseeuw proposed the least-median squared error regression (based on an idea by Hampel [19]), and a similar approach for mutivariate location estimation, called the MVE estimator

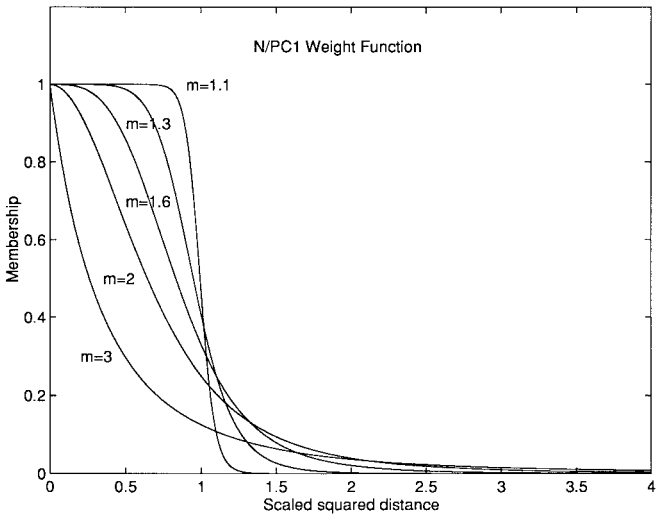


Fig. 2. Plot of the weight function corresponding to the noise/possibilistic clustering approach for various values of the fuzzifier m.

[41], [42]. In the MVE approach, one finds a minimum volume ellipsoid that covers (at least) h points of the set X. Usually, the value of h is taken to be equal to $[N/2] + 1$, where

N is the total number of points in X. The center of this ellipsoid is the estimate of the location, and the corresponding covariance estimation, which is a generalization of scale in higher dimensions, is obtained from the ellipsoid itself. The finite sample breakdown point of the MVE estimator can be shown to be $([N/2]-n+1)/N$ (where n is the dimensionality of the data points) which converges to 50% when $N \to \infty$ [42]. It may be obvious to the reader that there is no closed-form solution to the problem of finding the MVE. The solution is found by what may become an exhaustive search, although in practice (as Rousseeuw points out) only a limited search may be required [42]. In what follows, we first describe the procedure to find the MVE and then discuss the computational aspects.

To find the MVE, one starts by drawing a subsample of size $(n+1)$ [i.e., $(n+1)$ observations] from the data set X, indexed by $K = \{k_1, \cdots, k_{n+1}\}$. For this subsample, the arithmetic mean $\overline{\boldsymbol{x}}_K$, and corresponding covariance matrix $\boldsymbol{C}_K$ are computed as follows:

$$\overline{\boldsymbol{x}}_K = \frac{1}{n+1} \sum_{i \in K} \boldsymbol{x}_i$$

and

$$\boldsymbol{C}_K = \frac{1}{n} \sum_{i \in K} (\boldsymbol{x}_i - \overline{\boldsymbol{x}}_K)(\boldsymbol{x}_i - \overline{\boldsymbol{x}}_K)^T. \tag{36}$$

Care is taken to select the observations so that $\boldsymbol{C}_K$ is nonsingular. The corresponding ellipsoid is then inflated or deflated to contain exactly $h = 50\%$ of the points by computing the correct magnification factor based on the median value of the squared Mahalanobis distance given by

$$m_K^2 = \underset{\boldsymbol{x}_i \in X}{\text{med}} (\boldsymbol{x}_i - \overline{\boldsymbol{x}}_K)^T \boldsymbol{C}_K^{-1} (\boldsymbol{x}_i - \overline{\boldsymbol{x}}_K). \tag{37}$$

The volume of the resulting ellipsoid, which is proportional to

$$[\det (m_K^2 \boldsymbol{C}_K)]^{1/2} = (\det \boldsymbol{C}_K)^{1/2} (m_K)^n \tag{38}$$

is noted down. The above procedure is repeated for many subsamples K, and the one with the lowest volume is considered the optimum solution to the MVE problem. The corresponding value of $\overline{\boldsymbol{x}}_K$ is the estimate of the location, and the estimate of the covariance matrix C is given by

$$\boldsymbol{C} = (\chi^2_{n,0.5})^{-1} m_K^2 \boldsymbol{C}_K \tag{39}$$

where $\chi^2_{n,0.5}$, which is the median of the chi-squared distribution with n degrees of freedom, is the correction factor. Although in theory one may need all possible subsamples of size n out of N, i.e., the combinatorial $\binom{N}{n}$, in practice one could get away with finite sampling. When N/n is large, Rousseeuw argues that one may need only m randomly selected subsamples so that the probability of finding at least one *good* subsample is *almost* one. The value of m is obtained from the following equation for the probability q of finding at least one good subsample when the fraction of contaminated points is ε

$$q = 1 - [1 - (1-\varepsilon)^n]^m.$$

It is interesting to note that the above expression is independent of the number of points N, but is a strong function of the dimension n. For $q = 0.95$, and $\varepsilon = 0.5$, one would need 11 subsamples for two-dimensional data, while 3067 subsamples are required for ten-dimensional data. If one would like the probability of finding a good subsample as high as 0.999, then for the same 50% contamination, one would require about 25 subsamples for $n = 2$, and 7070 subsamples for $n = 10$. This simple calculation shows that the number of required subsamples is not astronomical. The hidden part in the computations is that for each subsample, one must compute the median, which is O($N \log N$). For each subsample, one must also invert the covariance matrix. Thus, the computational load of this method can be very high, but not prohibitive for today's computers.

The resulting estimate is not very efficient if the actual contamination is much lower than the assumed contamination. This is related to the first property that Huber [25] likes to see in a robust procedure, as described in Section I. For that reason, it is recommended that a one-step reweighted LS regression be performed. Weights can be assigned based on a simple rule such as

$$w_i = \begin{cases} 1, & \text{if } (\boldsymbol{x}_i - \overline{\boldsymbol{x}}_K)^T \boldsymbol{C}_K^{-1} (\boldsymbol{x}_i - \overline{\boldsymbol{x}}_K) \le c \\ 0, & \text{otherwise} \end{cases} \tag{40}$$

where the cutoff c maybe taken to be $\chi^2_{n,0.975}$. Using these weights, a one-step reweighted estimates of location and covariance are computed as follows:

$$\overline{\boldsymbol{x}} = \frac{\sum_{i=1}^{N} w_i \boldsymbol{x}_i}{\sum_{i=1}^{N} w_i}$$

and

$$\boldsymbol{C} = \frac{\sum_{i=1}^{N} w_i (\boldsymbol{x}_i - \overline{\boldsymbol{x}}_K)(\boldsymbol{x}_i - \overline{\boldsymbol{x}}_K)^T}{\sum_{i=1}^{N} w_i - 1}.$$

The MVE algorithm can be compared to the N/PC1 approach in the following way. Let the distance measure be of the type $(\boldsymbol{x}_i - \boldsymbol{c}_i)^T \boldsymbol{C}_i^{-1} (\boldsymbol{x}_i - \boldsymbol{c}_i)$, where $[\det (\boldsymbol{C}_i)]^{1/2}$ is proportional to the scale factor δ (or η). The MVE tries to find the ellipsoid with the smallest volume that covers a certain fraction of the data points, usually about half. Thus, it tries to find the densest cluster containing a certain fraction of the points. The N/PC1 algorithm, on the other hand, will try to fit the most number of points in an ellipsoid, the volume of which is determined by the selected value of δ (or η). Thus, assuming that the N/PC1 does indeed find the global minimum of the objective function, it tries to find the densest cluster with a given volume. Therefore, these two approaches are similar but not the same. The MVE, by the very definition of its objective function, has a high breakdown point. The N/PC1 approach, on the other hand, does not directly guarantee a high breakdown point. However, if the correct value of the resolution parameter δ is chosen, then the global minimum of the N/PC1 corresponds to the densest ellipsoid of size δ. In particular, if we choose the value of δ to

be that of m_K in (37) and we use $m = 1$ (i.e., hard rejection), then it is clear that the ellipsoid corresponding to the global minimum of the N/PC1 objective function will include the same fraction h of the data points. Thus, the N/PC1 would have the same breakdown point as the MVE. Nevertheless, there are two problems with the current formulation of the N/PC1. First, the global minimum is not guaranteed by the iteration scheme and second, the value of δ is not available *a priori*. Still, when viewed in this light, the N/PC1 does in principle have the same breakdown point as the MVE. In Section IX, we will discuss the CRE technique which uses precisely this argument to achieve high robustness.

The complete MVE procedure may be compared to N/PC1 in another special manner. The first part of the MVE is like an initialization step to find a good estimate of location and scale, while the second part is like a single step of an M estimator cast as a W estimator. There is nothing unique about the manner in which Rousseeuw selects the weights either [see (40)]. Therefore, one could use any other W estimator, or one could use the N/PC1 algorithm. In this way, the first part of the MVE can be simply viewed as a robust initialization scheme for the N/PC1. Such a strategy would have the best combination of robustness and efficiency. We will look further into the comparison between the MVE approach and the N/PC1 approach after discussing an interesting version of the MVE that may be used to detect more than one cluster in the data set.

A modified version of the MVE, called the GMVE, was recently proposed by Jolion *et al.*, [26] for computer vision applications. In the GMVE, clusters are searched one at a time, i.e., one in each pass, through what in theory can amount to a nearly exhaustive search. Instead of fixing the value of h to be about half of the total number of points as in the MVE, in each pass several values of h (from $h_{\max}$ down to $h_{\min}$) are tried with a suitable step size. Equation (39) now is modified to take into account the fact that the current value of h may not correspond to the median in (37). For each value of h, the resulting MVE is compared against the shape of an equivalent multivariate Gaussian cluster using the Kolmogorov–Smirnov (K–S) normality test. The significance level of this test, returned by a standard program package, is compared against a predetermined cutoff level, and if it meets that criterion, a cluster is assumed to have been found. The points belonging to that cluster [i.e., the points that have weight of 1 per (40)] are removed from the data. Thus, each pass detects and removes one cluster, and the procedure is repeatedly applied until a certain number of clusters are found, or only a small number of points are left.

It is easy to recognize that in each pass of the GMVE one needs to perform a complete MVE search over all possible values of h to find a cluster, which is computationally expensive. Besides the computational cost, the other disadvantages of this method are the need to specify several threshold parameters (such as the step size of h), as well as the dependency of the algorithm on the K–S test. If the underlying distribution is non-Gaussian, and if the type of distribution is known in advance, then this method can still be used by simply replacing the K–S test by another appropriate test. Therefore, despite its shortcomings, this method does have some merit in that it is firmly rooted in robust statistics.

When the cluster shapes to be detected cannot be described by ellipsoids (as in the case of curves and surfaces [6], [9], [17], [29], [30], [33]), the GMVE approach would not work. However, a similar idea can be used in conjunction with the least-median of squares (LMS) or the least-trimmed squares (LTS) methods [41] to detect quadric surfaces from range images as done by Fu [15]. In the LTS approach, the largest squared errors are trimmed and the sum of the remaining squared errors is minimized. The trimming fraction h is varied from 0.5 to a certain preselected small value just as in the GMVE approach. The selection of the range of h and a good set of cluster validity criteria are required for this approach to work.

For the detection of multiple clusters, one cannot use the MVE, but must use the GMVE. Here, we point out the similarities between the GMVE and the approach based on the N/PC1 algorithm. When the number of clusters is not known *a priori*, the N/PC1 technique can be used to detect one cluster at a time as outlined in [10] or [27]. In essence, in the GMVE, the value of h becomes the resolution parameter, while in the N/PC1 technique, δ (or η) is the resolution parameter. The resolution parameter in either case must be varied in order to detect a good cluster. After a cluster is detected, one must use a validity test in either approach to validate and remove a good cluster before starting the next iteration. The GMVE approach has the advantage that for a given value of h one is basically guaranteed to find the smallest (and thus densest) ellipsoid containing h fraction of points. For the N/PC1 technique, there is no such guarantee, because the iteration scheme may converge to a local minimum. The N/PC1 scheme, however, may be modified to utilize several different random initializations (as was done in the LBFC approach described in Section V, or in CRE, MINPRAN, and GMDD algorithms to be discussed in Sections IX and X), in which case it would converge to a global minimum that would correspond to the densest cluster of size determined by the value of δ. This approach would be very much like the GMVE approach, which also requires picking many subsamples to find the MVE in each pass. One can make probabilistic arguments similar to Rousseeuw's regarding the number of different initializations necessary for the N/PC1. This would make the N/PC1 also highly computational, just as the GMVE. In Section X, we review a Gaussian mixture-density decomposition method that is based on precisely such a search. The theory behind this method indicates that the range to be searched for the "correct" value of δ or η is finite.

IX. Cooperative Robust Estimation and Minimization of Probability of Randomness

In this section, we discuss two robust approaches that have appeared in the recent computer vision literature. The first approach, called CRE, is due to Darrell and Pentland [5]. The second one, due to Stewart [43], is MINPRAN, which is based on MINimizing the probability of RANdomness.

The CRE technique is an attempt at "overcoming the (low) breakdown point of M estimators, by initializing multiple

hypotheses with different initial conditions and integrating information across multiple robust estimates." This technique models the data set (image) $\boldsymbol{d}$ as a sum of K masked individual components and an additive noise term η, where each component is associated with a support mask $s^{(k)}$ and a model $\boldsymbol{y}^{(k)}$ with parameters $\boldsymbol{x}^{(k)}$. Each model $\boldsymbol{y}^{(k)}$ is some function which can be applied to parameters $\boldsymbol{x}^{(k)}$ to generate (part of) the data set. For example, in range-image segmentation, the model could be a second-degree polynomial, and the $\boldsymbol{x}^{(k)}$ consist of the coefficients of the polynomial. The support masks play the role of membership functions or weight functions, and the support (membership or weight) for point j in model $\boldsymbol{y}^{(i)}$ may be denoted by $s_j^{(i)}$. Since we do not know the number of components K, the CRE technique starts by generating an initial set of hypotheses by sampling the space of possible models and parameters. For the CRE technique to work, "the general principle is that the initial set of hypotheses must have at least one hypothesis in rough correspondence with each real object (cluster) in the scene (data set)." In other words, each hypothesis represents one possible prototype. The algorithm further assumes that the "inlier bound" or scale (which relates to the parameter θ in their paper) is known for each hypothesis. The residuals corresponding to each hypothesis are evaluated, and hard rejection (based on the known scale θ) is used in an IRLS technique (see Section VI) to obtain a robust estimate of the parameters for the hypothesis. A subset of the initial hypotheses is selected based on the minimum description length (MDL) criterion [38]. Thus, the selected subset of hypotheses is such that they "explain" the data "seen" by all the hypotheses with the least encoding cost. The authors use a gradient descent technique to minimize the MDL criterion. After the "optimal" subset of hypotheses is selected, the support functions are updated using the cooperative support update rule, which basically assigns each good data point to the hypothesis with the smallest residual (i.e., creates a hard partition of the good data points). Data points which have a residual greater than the scale θ in all hypotheses are considered outliers.

In the light of the discussion in Section VI, we see that the CRE technique is essentially a variation of the PCM where η is assumed to be known and $m = 1$. The algorithm starts by an overspecified number of clusters, and the prototype of each cluster is updated in each iteration by using only those points that lie within a distance η. A subset of the hypotheses is then picked using the MDL criterion to eliminate spurious clusters, coincidental clusters, and clusters with large overlap.

The CRE technique can also be compared with the GMVE in the following manner. In the GMVE, the location and scale of each cluster is estimated by drawing a subsample and inflating the covariance matrix of the subsample to cover a fraction h of the points. In the CRE, since the scale is assumed to be known, one simply inflates the covariance matrix to match the scale. Rather than selecting the ellipsoid with the smallest volume for each h, CRE keeps all ellipsoids and selects a subset of them later using the MDL criterion. A major difference between the GMVE and CRE is that unlike in the GMVE, the CRE updates the prototypes using an iterative procedure just like the PCM. Therefore, when the scale is known, this approach will yield more efficient estimates.

If CRE is to be used for clustering, the most serious problem is that the scale needs to be known. This could be particularly difficult if the clusters are expected to be of various shapes and sizes. The assumption that the scale is known also plays an important role in eliminating certain points while estimating the prototype parameters and, thus, is critical to increasing the breakdown point. The number of initial hypotheses need to be very large not only to guarantee a low breakdown point, but also to ensure that the minimization of the MDL criterion gives us a good subset of the hypotheses. This is because the criterion function picks the optimal subset based on only those points that are "seen" by (i.e., have nonzero supports in at least one of) the hypotheses. (The PCM has a similar problem.) If the number of initial hypotheses is small, one can completely miss some of the clusters. Also, the particular formulation of the MDL criterion used by the authors discourages overlaps and, thus, is not suitable if clusters are not well separated.

MINPRAN [43] is a robust estimator for finding good fits in noisy data sets. It is not formulated as a clustering algorithm, but could be easily modified to find clusters. The algorithm assumes that the outliers are randomly distributed within the dynamic range of the sensor, and the noise (outlier) distribution is known. In [43] the author analyzes the case when the noise is assumed to be uniformly distributed, and indicates how it could be generalized for other kinds of distributions. Assuming that p points are required to completely instantiate a fit, MINPRAN chooses S-distinct subsets of p points from the data set containing N points. Each subset leads to a hypothesized fit, and the hypothesized fits are denoted by $\phi_1, \phi_2, \cdots, \phi_S$. The residuals associated with the N data points for each ϕ_i are arranged in ascending order in the form of a matrix. Let r_{ij} denote the residual associated with point $\boldsymbol{x}_j$ with respect to fit ϕ_i. Let $\mathcal{F}(r, k, N)$ denote the probability that at least k points fall within a "distance" r of a fit, given the known noise distribution and the total number of points N. It is easy to show that [43], for a given k, the minimum probability $\mathcal{F}(r_{ik}, k, N)$ occurs for the mth hypothesized fit ϕ_m, where m satisfies $r_{mk} = \min_i r_{ik}$ (i.e., the fit that accommodates k points within the narrowest band is least likely to be random). If we denote the smallest kth residual across all S fits by r_k^*, it follows that r_k^* is simply the minimum among the residuals in the kth column of the residual matrix. This procedure is repeated for all columns to generate $r_1^*, r_2^*, \cdots, r_N^*$. The corresponding probabilities $\mathcal{F}(r_k^*, k, N)$ are computed. Let $\mathcal{F}(r_m^*, m, N)$ denote the minimum of these probabilities. Then r_m^* is taken to be the true inlier bound (i.e., scale), provided $\mathcal{F}(r_m^*, m, N)$ is less than a randomness threshold $\mathcal{F}_0$. The corresponding hypothesized fit is taken to be the best fit. This algorithm can be easily modified to find clusters by interpreting the residuals as distances form hypothesized prototypes to the data points.

MINPRAN, as formulated above, finds only one curve or surface in a data set. If the data is expected to have multiple curves, in [43] it is recommended that one curve/surface be sought at a time and the points within the inlier bound of each curve/surface found be removed from the data set. This idea is similar to the progressive clustering idea that has been

used in several other algorithms [26], [45], and [47]. The randomness threshold $\mathcal{F}_0$ is estimated based on the assumed noise distribution, the number of samples S, and a specified probability P_0 that $\mathcal{F}(r, k, N)$ corresponding to MINPRAN's best fit to N noise points is less than F_0. The number of samples S is in turn estimated from several user-specified parameters such as the approximate number of curves/surfaces n_f and the estimated maximum fraction of outliers x_0. Thus, the randomness threshold indirectly depends on the estimated maximum fraction of outliers, and a fixed value of $\mathcal{F}_0$ for a variety of noise conditions may not work, especially if there is a wide variation in the number of curves/surfaces.

In [43], it is stated that "MINPRAN is the first technique that reliably tolerates more than 50% outliers without assuming a known inlier bound." Although no comparison is made between MINPRAN and the GMVE in [43], MINPRAN can be interpreted as the "complement" of the GMVE in the following manner. The GMVE uses a validity test (the K–S test) assuming that the distribution of residuals of good points in known, whereas MINPRAN uses a test of randomness assuming that the distribution of (residuals of) noise points in known. Thus, the results will be similar if we assume that the validity of a pattern is automatically high if the pattern has a very small probability of coming from the assumed noise distribution. (From an alternative view point, MINPRAN chooses the fit that includes a given number of points within the smallest inlier bound. Thus, the basic idea behind MINPRAN and the GMVE are similar.) Indeed, in [43] it is shown that this argument is reasonable if 1) the noise is uniformly distributed within an interval of length Z_0; 2) the residuals due to good data have a Gaussian distribution with standard deviation σ; and 3) $\sigma \ll Z_0$. In such a case, MINPRAN finds nearly the correct inlier bound. The inherent assumption in MINPRAN is that the values of the parameters of the noise distribution are known. In contrast, the GMVE assumes that the parametric form of the distribution of the good points is known, and estimates the values of the parameters such as mean and covariance. To estimate the values of the parameters, the GMVE needs to perform a search over the fraction h of good points. A similar search over the parameters of the noise distribution would be required in MINPRAN if these parameters are not known (and consequently the randomness threshold cannot be estimated). For example, if the noise is uniformly distributed, but the length of the interval Z_0 is not known, there is no way to estimate the probability of occurrence of a particular set of residuals. In a sense, Z_0 plays the role of scale, since it determines whether the residuals within a given inlier bound could have come from noise or from good points. On the other hand, the significance of MINPRAN lies in the fact that it is fast [O($N^2 + SN \log N$], and can be used in situations where the assumptions on the noise distribution are reasonable.

It is pointed out in [43] that the performance of MINPRAN is not always good when there are multiple curves/surfaces, and the algorithm could give us "bridging fits," i.e., fits that include points from two different surfaces near a discontinuity. This is a consequence of selecting the best fit based on likelihood of nonrandomness rather than on validity. In Section XI, we return to this problem and discuss the crucial role of validity in robust clustering.

Ideally, one may want to accept the MINPRAN fit only if the residuals within the inlier bound are more likely to come from good points than from noise. This is easily verified using Bayes law if the distribution of the residuals due to good points is also known. The GMDD algorithm, to be discussed in the next section, takes this approach. However, since the GMDD does not assume that the noise distribution is known, it needs to perform a search for a parameter related to the scale parameter.

X. Gaussian Mixture Decomposition and the Epsilon Contamination Model

In yet another method based on robust statistics, an interesting variation of the epsilon contamination model of Huber [25] is considered by Zhuang *et al.* [47]. This approach views the clustering problem as a Gaussian mixture density decomposition problem, i.e., the problem of extracting each valid Gaussian component G_i in a given data set X. Each component G_i is characterized by $N(\boldsymbol{m}_i, \boldsymbol{C}_i)$, where $\boldsymbol{m}_i$ is the mean vector and $\boldsymbol{C}_i$ is the covariance matrix. If we assume that there are C components and that they are all independent of one another, then, with respect to a given component, the data belonging to the remaining components can be considered as outliers. Therefore, as in the case of possibilistic clustering, the decomposition problem can be viewed as C independent problems, where C may be unknown. Further, in the case of noisy data, it is assumed that each sample $\boldsymbol{x}_j$ in X is generated by an unknown Gaussian distribution $N(\boldsymbol{m}_i, \boldsymbol{C}_i)$ with probability $(1-\varepsilon)$ plus an unknown outlier distribution $h_i()$ with probability ε. The extent of contamination (i.e., the value of ε) is considered to be unknown. The probability density $f_i()$ of the samples is given by [47]

$$\begin{aligned} f_i(\boldsymbol{x}_j) = &(1-\varepsilon)(2\pi)^{-n/2}|\boldsymbol{C}_i|^{-1/2} \\ &\cdot \exp\{-\left(\tfrac{1}{2}\right)(\boldsymbol{x}_j - \boldsymbol{m}_i)^T \boldsymbol{C}_i^{-1}(\boldsymbol{x}_j - \boldsymbol{m}_i)\} \\ &+ \varepsilon\, h_i(\boldsymbol{x}_j). \end{aligned} \tag{41}$$

$f()$ is called a contaminated Gaussian density when $\varepsilon > 0$ [25]. A sample $\boldsymbol{x}_j$ should be classified as an inlier if it is realized from $N(\boldsymbol{m}_i, \boldsymbol{C}_i)$ or as an outlier otherwise. Let

$$\begin{aligned} g_i(\boldsymbol{x}_j) = &(2\pi)^{-n/2}|\boldsymbol{C}_i|^{-1/2} \\ &\cdot \exp\{-\left(\tfrac{1}{2}\right)(\boldsymbol{x}_j - \boldsymbol{m}_i)^T \boldsymbol{C}_i^{-1}(\boldsymbol{x}_j - \boldsymbol{m}_i)\}. \end{aligned} \tag{42}$$

Denoting the set of inliers by G_i and the set of outliers by B_i from the Bayes rule of classification, we have

$$G_i = \left\{(\boldsymbol{x}_j | g_i(\boldsymbol{x}_j) > \frac{\varepsilon h_i(\boldsymbol{x}_j)}{1-\varepsilon}\right\}$$

and

$$B_i = \left\{\boldsymbol{x}_j | g_i(\boldsymbol{x}_j) \leq \frac{\varepsilon h_i(\boldsymbol{x}_j)}{1-\varepsilon}\right\}.$$

Letting

$$a = \min\{g_i(\boldsymbol{x}_j) | \boldsymbol{x}_j \in G_i\}$$

and

$$b = \max\{g_i(\boldsymbol{x}_j) | \boldsymbol{x}_j \in B_i\}$$

ideally the likelihood of any inlier being generated by $N(\boldsymbol{m}_i, \boldsymbol{C}_i)$ is greater than the likelihood of any outlier being generated by $N(\boldsymbol{m}_i, \boldsymbol{C}_i)$. Hence, Zhuang *et al.*, argue that we may assume that $a > b$. Therefore, the Bayes classification becomes

$$G_i = \left\{ \boldsymbol{x}_j | g_i(\boldsymbol{x}_j) > \frac{\varepsilon\delta_i}{1-\varepsilon} \right\}$$

and

$$B_i = \left\{ \boldsymbol{x}_j | g_i(\boldsymbol{x}_j) \leq \frac{\varepsilon\delta_i}{1-\varepsilon} \right\} \tag{43}$$

where we can choose

$$\delta_i \in \left[\frac{(1-\varepsilon)b}{\varepsilon}, \frac{(1-\varepsilon)a}{\varepsilon} \right]. \tag{44}$$

The δ_i in the above equations should not be confused with the noise distance δ used by the NC algorithm, although they are related, as will be shown. Equations (43) and (44) suggest that if we assume that $h_i(\boldsymbol{x}_1) = h_i(\boldsymbol{x}_2) = \cdots = h_i(\boldsymbol{x}_N) = \delta_i$; we would get equivalent results. Using this assumption, (40) becomes

$$f_i(\boldsymbol{x}_j) = (1-\varepsilon) g_i(\boldsymbol{x}_j) + \varepsilon\delta_i.$$

The log-likelihood function Q of observing $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N$ conditioned on $\boldsymbol{m}_i, \boldsymbol{C}_i, \varepsilon$, and δ_i is

$$\begin{aligned} Q &= \log P(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N | \boldsymbol{m}_i, \boldsymbol{C}_i, \varepsilon, \delta_i) \\ &= \sum_j \log f_i(\boldsymbol{x}_j) \\ &= N \log(1-\varepsilon) + \sum_j \log \left[g_i(\boldsymbol{x}_j) + \frac{\varepsilon\delta_i}{1-\varepsilon} \right]. \end{aligned}$$

Therefore, the maximization of Q at δ_i with respect to $\boldsymbol{m}_i$ and $\boldsymbol{C}_i$ is equivalent to maximizing

$$q(\boldsymbol{m}_i, \boldsymbol{C}_i, t_i) = \sum_j \log [g_i(\boldsymbol{x}_j) + t_i] \tag{45}$$

at t_i with respect to $\boldsymbol{m}_i$ and $\boldsymbol{C}_i$, provided that $t_i = \varepsilon\delta_i/(1-\varepsilon)$.

If we assume that the clusters are generated by Gaussian distributions, the prototype of each cluster can be represented by $\beta_i = (\boldsymbol{m}_i, \boldsymbol{C}_i)$. We may define the distance of a feature vector $\boldsymbol{x}_j$ from the prototype β_i as the inverse of the probability given in (42), i.e.,

$$\begin{aligned} d^2(\boldsymbol{x}_j, \beta_i) &= d_{ij}^2 \\ &= \frac{1}{g_i(\boldsymbol{x}_j)}. \end{aligned} \tag{46}$$

Further, if w_{ij} is assumed to be of the form in (27), for $m = 2$, we have

$$w_{ij} = \frac{1}{1 + t_i g_i^{-1}(\boldsymbol{x}_j)} \tag{47}$$

where $t_i = 1/\eta_i$. Now consider the following objective function similar to the one in (28):

$$J_G(B; X) = \prod_{i=1}^{C} \prod_{j=1}^{N} w_{ij}\, d_{ij}^2$$

where the product rather than the sum is used to simplify the resulting expression. Substituting for d_{ij}^2 and w_{ij} from (46) and (47) and taking logs, we can write the above objective function as

$$J'_G(B; X) = \sum_{i=1}^{C} \sum_{j=1}^{N} \log \left[\frac{1}{(g_i(\boldsymbol{x}_j) + t_i)} \right].$$

We may equivalently maximize the C individual terms

$$J'(G_i)(B; X) = \sum_{j=1}^{N} \log [g_i(\boldsymbol{x}_j) + t_i]. \tag{48}$$

Comparing (48) with (45), we see that the possibilistic clustering problem with the chosen distance measure reduces to a Bayesian classification problem when the data is epsilon contaminated. This also establishes that probability density estimation and possibilistic clustering are basically the same problem and the bandwidth η_i (or noise distance δ used by the NC) is related to the parameter δ_i used in the modeling of the unknown outlier distribution by the following relation:

$$\begin{aligned} \eta_i &= \frac{1}{t_i} \\ &= \frac{1-\varepsilon}{\varepsilon\delta_i}. \end{aligned} \tag{49}$$

The algorithm suggested by Zhuang *et al.* involves finding a best-fitting normally distributed component for each possible value of t_i in a given range. To obtain reliable results, for each value of t_i, a Monte Carlo procedure is used to select the initial values of $\boldsymbol{m}_i$ and $\boldsymbol{C}_i$ in a fixed-point iteration scheme that finds the best Gaussian component. As in the case of the GMVE, each component so found is checked using the K–S test and, if it is good, then the component is removed from the data and the process is repeated. This method has some of the same disadvantages as the GMVE since the computation load is high, and there are similar problems associated with the use of the K–S test. When the clusters overlap heavily, the assumption "with respect to each cluster the points belonging to other clusters can be considered as outliers," as well as the assumption that $a > b$ [which leads to (43)], are difficult to justify. Removing one cluster can destroy the structure of other overlapping clusters, and they may no longer pass the K–S test. In any case, the role of the parameter t_i in this method is similar to the role of h in the GMVE, except that the derivation of the range for t_i in this method is based on probabilistic arguments, while the range of h in GMVE is based on heuristic arguments. The main advantage of either of these methods is that there is no need to know the exact number of clusters *a priori*. It is clear from (48) and the discussion above that t plays the same role in this algorithm as α, δ, or η play in the Ohashi, NC, and PCM algorithms, respectively. Since the range of t is finite, as indicated by (33) and (39), the ranges of α, δ, or η are also finite.

In the next section, we define several concepts that are useful in robust clustering. We then express the robust clustering problem in terms of the defined concepts and suggest generic algorithms to solve the robust clustering problem.

XI. General Concepts Related to Robust Clustering

The common theme among all the robust methods that we have reviewed in this paper is that we need to reject or ignore a subset of points in the given data set to achieve robustness. The rejection can be either hard or fuzzy; a fuzzy rejection is preferable because it handles the region of doubt in a better fashion. Most of the algorithms reviewed in this paper achieve this by minimizing a function of the form $\Sigma_i \Sigma_j u_{ij} d_{ij}^2$, where the membership (or weight) u_{ij} is a monotonically decreasing function of distance. The summation over i can be ignored if there is only one cluster, or if we treat the clusters to be independent, or if we seek only one cluster at a time. In the last two cases, each local minimum of the function corresponds to one cluster. The objective function in these cases can be written as

$$J = \sum_j u_j d_j^2 = \sum_j \rho(d_j^2). \tag{50}$$

When there are multiple clusters, and the number of clusters C is known, one can treat the set of C prototypes $\{\beta_i\}$ as a single complex prototype β. In this case, we can define the distance d_j^2 of a point $\boldsymbol{x}_j$ from the complex prototype as $d_j^2 = D(d_{1j}^2, d_{2j}^2, \cdots, d_{Cj}^2)$, where D is a suitable function. We can similarly define u_j to be the degree of "goodness" of point $\boldsymbol{x}_j$. Therefore, the objective function in this case can also be written as (50). For example, if $D(.) = [\Sigma_{i=1}^{C} d_{ij}^{2/(1-m)}]^{1-m}$; then, as a consequence of the reformulation theorem [23], the objective function in (50) becomes a robust version of the FCM algorithm and for $m \to 1$ a robust version of the K-means algorithm. In this section, we base our discussion on (50). However, we note that the summation over j can also be replaced by other operators (such as the median) to obtain other variations.

The memberships u_j, $j = 1, \cdots N$, define a fuzzy subset $\mathcal{C}$ of the set of feature vectors X whose membership function is given by

$$\mu_{\mathcal{C}} : X \to [0, 1], \qquad \text{where } \mu_{\mathcal{C}}(\boldsymbol{x}_j) = u_j.$$

Thus, each cluster is represented by a fuzzy set $\mathcal{C}$, and the estimate for the prototype parameters of a given cluster is obtained using the corresponding fuzzy set $\mathcal{C}$. [This is done by optimizing (50) with respect to the prototype parameters.] Conversely, each choice for the values of the prototype parameters β of a cluster induces a fuzzy subset with the memberships

$$u_j = w[d^2(\boldsymbol{x}_j, \beta); \boldsymbol{a}] = w(d_j^2; \boldsymbol{a}).$$

In the above equation, $\boldsymbol{a}$ is a vector of parameters that dictates the shape of the monotonically decreasing membership (weight) function $w()$. [For the N/PC1, $w()$ is given by (27).] We refer to such parameter-induced fuzzy subsets of X as components. The cardinality of a cluster is defined to be the cardinality of the corresponding fuzzy set $\mathcal{C}$. We denote this by Card($\mathcal{C}$). In other words,

$$\text{Card}(\mathcal{C}) = \sum_{j=1}^{N} u_j.$$

We refer to the fuzzy set $\mathcal{C}$ corresponding to a cluster found by an algorithm as the included component and the fuzzy set $\mathcal{R} = X - \mathcal{C}$ as the rejected component. We define the inclusion rate f associated with a particular result of robust clustering as

$$f = \frac{\text{Card}(\mathcal{C})}{N}.$$

The rejection rate r is simply 1-f.

The quantity $\Sigma_j u_j d_j^2$ can be viewed as the validity associated with cluster i or fuzzy set $\mathcal{C}$. We denote this by

$$\text{Val}(\mathcal{C}) = \sum_j u_j d_j^2.$$

The above validity measure represents the fuzzy (or weighted) sum of distances of the points to the cluster. As discussed in Section III, this validity measure, although analytically attractive, does not always give an intuitively correct result. Many other validity measures are possible. For instance, one could use the density criterion [16]

$$\text{Val}(\mathcal{C}) = \frac{\text{Card}(\mathcal{C})}{[\det(\boldsymbol{C})]^{1/2}}$$

where $\boldsymbol{C}$ is the covariance matrix associated with the cluster. One may also use other measures such as the K–S test. Several validity measures for shell clusters may be found in [10], [11], and [30].

One of the central problems in robust clustering is to find the correct weight functions, or equivalently, the membership functions. In Section I, we stated that in the case of a single cluster, the best achievable breakdown point is 50%. Therefore, it would seem that by ignoring 50% of the points, we will find a good estimate. However, this is not a good choice for the following reasons.

1) The choice of 50% only guarantees that our estimates do not have arbitrarily large errors. It does not guarantee that the errors in the estimates will be acceptable from an engineering point of view. In other words, the estimator could break down according to (3). This is because when the actual number of good points is greater than 50%, we will not be using all the good points in our estimate, which can lead to a poor estimate (or a poor regression fit). (It is well-known that the median is not very accurate and does not provide a good fit to the data [20], [25], [35].) A better strategy would be to use as many good points as possible.
2) When fraction of noise points is larger than 50%, it does not automatically mean that robust methods break down.

As an example, consider the case of fitting a single line to a data set. Let the number of good points constitute a fraction f of the data points, even if $f < 0.5$, an algorithm that uses only a fraction f of the total points, can still find a good estimate for the line as long as the noise points do not "conspire" to form a straight line, which is unlikely. In contrast, an algorithm that uses 50% of the points can break down completely. Thus, an algorithm that explores all possible inclusion rates would perform better and give us good estimates even beyond the theoretical breakdown point almost all the time. The GMVE, MINPRAN, and GMDD use a similar idea to break the 50% theoretical limit. In this respect, the Hough transform (HT) can be considered more robust than the LMS algorithm. The HT can find accurate estimates of parameters for a line regardless of the percentage of contamination, as long as no subset of the noise points forms another line with a higher cardinality. We accept the result of the HT provided the strength of the peak exceeds an acceptable threshold. (Of course, this assumes that problems such as bin splitting do not occur in the HT.) We will return to the HT later in this section.

The above argument shows that to design an algorithm with the best possible performance even beyond the breakdown point, we need to consider all possible inclusion rates f and find the highest value of f that gives us an acceptable value of Val($\mathcal{C}$). In other words, cluster validity plays a crucial role, because without it we would have no way of selecting the best inclusion rate f and, hence, the best weight or membership function for the estimate. (Note that validity helps us break the 50% theoretical limit for the breakdown point only most of the time, i.e., there is no guarantee that one will find the correct solution all the time.)

As mentioned in the beginning of this section, robust estimators achieve their robustness by (fuzzily) rejecting a subset of the data set. As discussed above, the optimum rejection needs to be based on a validity measure, and ideally the same validity measure should also be used in the objective function. If a fuzzy set $\mathcal{C}$ that represents the points that are included in a given estimate has a validity greater than the acceptable threshold, i.e., if $\text{Val}(\mathcal{C}) \geq \varepsilon$, then we refer to such a fuzzy set as a valid component. (For the sake of this discussion, we assume that the higher the validity, the better the result. When the opposite is true, as in the case of the sum of intracluster distances, a valid component satisfies $\text{Val}(\mathcal{C}) \leq \varepsilon$.) We refer to a valid component that is not a subset of any other valid component as a maximal valid component (MVC) and denote it by $\mathcal{V}$. In other words, if $\mathcal{V}$ is a maximal valid component and $\mathcal{C} \supset \mathcal{V}$, then $\text{Val}(\mathcal{C}) < \varepsilon$. It is to be noted that there can be many distinct MVC's for a given data set. (We will see an example later in this section.)

We can now state the robust clustering problem when $C = 1$ as follows: Find the maximal valid component in the data set that has the largest cardinality. This formulation assumes that the solution that "explains" or "accounts for" as many data points as possible is the best one. We refer to the MVC which is the solution to this problem as the optimal MVC.

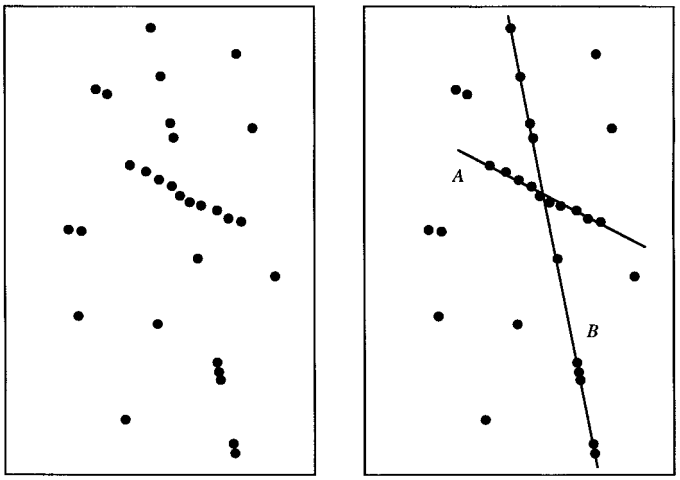
The above discussion suggests the following MVC algorithm for finding a robust estimate of the prototype parameters when a single cluster is assumed to be present in the data set.

The MVC Algorithm to Find a Robust Estimate of the Parameters of a Single Prototype:

fraction := 1.0; *max_validity_found* := 0;
valid_component_found := FALSE;
repeat {
  for (all possible components $\mathcal{C}$ of X with cardinality $N * fraction$) do {
    Find the parameters of the prototype for the chosen component;
    Compute *validity* of $\mathcal{C}$;
    If [(*validity* $\geq$ *acceptable_validity*) and (*validity* $>$ *max_validity_found*)] then {
      max_validity_found := *validity*;
      store the fuzzy set $\mathcal{C}$ and the estimated parameters of the prototype;
      valid_component_found := TRUE;
    }
  }
  fraction := *fraction* $-\Delta f$;
}
until [(*fraction* $\leq$ *min_fraction*) OR (*valid_component_found* = TRUE)].

In the above algorithm, *fraction* denotes the included fraction, and it is assumed that the chosen validity measure is always positive for any fuzzy set. It is to be noted that if the algorithm terminates with *valid_component_found* = FALSE, then it means that no solution exists that meets the desired validity threshold. The proposed algorithm can be viewed as a generalization of the MVE algorithm. When the components are crisp rather than fuzzy, and the validity criterion used is the volume, it reduces to the MVE algorithm. However, unlike the MVE, there is no need for a reweighting step because we use fuzzy subsets, and the fraction of good points is not assumed.

We would like to caution the reader that the above algorithm represents an exhaustive search. This could be simplified in several ways, depending on the application. For example, the algorithm assumes that there could be several MVC's with the same cardinality, in which case we need to pick the solution with the better validity. This is unlikely in practice, and therefore the *for loop* could be terminated when a valid component is found. We could also pick a random subset of all possible parameter-induced fuzzy subsets $\mathcal{C}$ with cardinality $N*fraction$. Another possibility is to choose a random subset of the data set in each iteration of the *for loop* to compute a set of prototype parameter values. These values can then be used to induce the component $\mathcal{C}$. Moreover, since the solution corresponds to one of the local extrema of the objective (validity) function, we can use the component $\mathcal{C}$ randomly generated at the beginning of the *for loop* as the initialization for a gradient descent or Picard iteration technique. This can reduce the total number of the *for-loop* iterations required for a reliable result. These modifications to speed up the algorithm



(a) (b)

Fig. 3. (a) Original noisy data set in which a linear cluster is to be found. (b) Solution A corresponds to a density criterion, whereas solution B corresponds to a sum of squared distances criterion.

are a combination of the techniques used in the LBFC, MVE, CRE, MINPRAN, and GMDD algorithms.

The acceptable value ε for the validity is application dependent. Furthermore, choosing the correct measure for the validity is very important. As an example, let us consider estimating the parameters of a line from the image containing edge pixels, shown in Fig. 3. Let us assume that each dot represents an edge pixel in this figure. Let us further assume that we use crisp components $\mathcal{C}$, and choose the average squared distance of the points to the line as the validity function, and set the acceptable validity to $4\Delta^2$, where Δ represents the distance between two adjacent pixels. For this situation, line B is the optimal solution because it is the MVC with the highest inclusion rate f. Note that there are many MVC's in this image. For instance, the subset of points touching line A is an MVC. There are also other ones with smaller cardinalities, many of which use just two points. On the other hand, if we choose the surface-density criterion [30] as the validity measure, there will be a lot fewer MVC's, and we might find that line A is the optimal solution. In many computer vision applications, we might prefer line A, even though the f value for line A is lower than that of line B. Thus, this example shows that the type of validity measure chosen is critical, because it can help reduce the number of candidate solutions as well as find an intuitively correct solution. It is to be noted that when an exhaustive (or random) search technique is used to find the solution of the robust clustering problem, the type of validity measure used in the objective function does not significantly influence the computational burden. However, if we use iterative techniques, the validity criterion can significantly increase the computational complexity if it is not analytically tractable.

The above discussion deals with a situation in which there is only one cluster. We now consider the case of multiple clusters. We first consider the case when the number of clusters C is known. As discussed above, this case can be reduced to a single cluster case with a complex prototype. Ideally, we would like the solution of the problem to be an MVC induced by a complex prototype. As in the single cluster case, for a given data set, there might be several MVC's for the multiple cluster case.

The robust clustering problem when the number of clusters is known can be stated as follows: find the MVC $\mathcal{V}$ induced by a complex prototype with the highest cardinality. We refer to the fuzzy set $\mathcal{C}$ corresponding to the solution of this problem for a given value of C as the optimal MVC, and denote it by $\mathcal{V}_{opt}$. We now outline a robust algorithm to find C clusters in a data set. When several MVC's have the same cardinality, the proposed algorithm picks the one whose validity is the best. Suitable definitions for the union of fuzzy subsets as well as for the validity measure for the union needs to be chosen. As before, we would like to caution the reader that this algorithm represents an exhaustive search. The methods suggested for speeding up the algorithm for the case of a single cluster apply to this algorithm as well.

The MVC Algorithm to Find Robust Estimates of the Parameters of C Prototypes:

fraction := 1.0; *max_validity_found* := 0;
valid_component_found := FALSE;
repeat {
  for (all possible components $\mathcal{C}$ of X with cardinality $N^* fraction$) do {
    for (all possible complex prototypes $\beta = \{\beta_i\}$ that induce $\mathcal{C}$ do {
      Compute the validity of $\mathcal{C}$ based on $\{\mathcal{C}_i\}$;
      If [(*validity* $\geq$ *acceptable_validity* and (*validity* > *max_validity_found*)]
      then {
        max_validity_found = *validity*;
        Find fuzzy sets $\mathcal{C}_i$ corresponding to $\{\beta_i\}$.
        (Note that $\cup_{i=1}^{C} \mathcal{C}_i = \mathcal{C}$)
        Store fuzzy sets $\mathcal{C}_i$ and the estimated parameters of the prototypes;
        valid_component_found := TRUE;
      }
    }
  }
  fraction := *fraction* $-\Delta f$;
}
until [(*fraction* $\leq$ *min_fraction*) OR (*valid_component_found* = TRUE)].

One may ask the question: what is the best breakdown point one can theoretically achieve in the case of multiple clusters? The answer to this question can be obtained by the following reasoning. Let the cardinality of cluster i be N_i. Let us further suppose that the total number of data

points in the data set $N = \Sigma N_i + N_o$, where N_o denotes the cardinality of the rejected component $\mathcal{R}$. Let cluster k with cardinality $N_k = N_{\min}$ be the smallest cluster. The definition of the breakdown point involves the smallest number of points that can cause an unacceptable error in the estimate. Since we are free to place the noise points anywhere in feature space, we can place them in such a way that when they are included in $\mathcal{C}$ instead of the smallest good cluster, the resulting validity of $\mathcal{C}$ is better. It follows that if the cluster formed by the noise points is located sufficiently far away from cluster k, and the cardinality of the fuzzy set representing the noise points is equal to $N_{\min}$, then the prototype estimates could be completely wrong if the algorithm includes the noise cluster in $\mathcal{C}$ instead of the legitimate one. Thus, the best possible breakdown one can achieve for multiple clusters is $N_{\min}/N$. Although this is a disappointing result, it is to be kept in mind that simply because the fraction of noise points exceeds $N_{\min}/N$ in a given data set, it does not mean that all robust algorithms will fail. When properly formulated, a robust clustering algorithm will continue to work as long as the noise points do not give rise to a higher validity than one of the legitimate clusters.

We now discuss the case when the number of clusters is not known. In general, we would like to have a solution such that the included component $\mathcal{C}$ is as large as possible. Therefore, the problem of robust clustering when the number of clusters is unknown can be stated as follows: among all the optimal maximal valid components $\mathcal{V}_{opt}$, find the one with the largest cardinality. (Note that each optimal maximal valid included component $\mathcal{V}_{opt}$ corresponds to a different value of $\mathcal{C}$.) We now outline two algorithms that may be used to solve this problem.

The first algorithm finds the solution by repeatedly running the generic algorithm to find robust estimates of C prototypes for increasing values of C until a solution is found. This represents the classical approach. The second algorithm is based on removing one cluster at a time. It uses the MVC algorithm to find a robust estimate of the parameters of a single prototype described above, and finds the largest MVC in each pass. It then updates the data set by subtracting the fuzzy set corresponding to the MVC from the data set, i.e., it sets $X = X - \mathcal{C}$. This process is repeated until no more valid components can be found. In general, since the clusters are fuzzy, when a cluster is removed from the original data set X, the updated data set becomes a fuzzy subset of the original data set. This means that some of the points are only "partially available" in the next run. This can introduce problems, particularly in the case of shell clusters, because some of the points may be shared fully between several clusters. (Note that there is no constraint on the sum of memberships of a point across clusters.) From this point of view, this may not be a good approach, especially when the clusters are expected to overlap heavily. However, it is considerably faster than the classical approach. The classical approach provides a better mixture model, and becomes more attractive when the number of clusters is known or can be estimated. We once again caution the reader that both these algorithms use an exhaustive search, and variations and simplifications are possible depending on the application.

The MVC Algorithm to Find Robust Estimates of the Parameters of an Unknown Number of Prototypes:

Set $K := 1$; *max_cardinality* := 0;
Repeat {
Run the Generic Algorithm to Find Robust Estimates of C Prototypes with $C = K$;
If (*valid_component_found* = TRUE) {
    Compute *cardinality* of the resulting maximal valid component $\mathcal{V}_{opt}$;
    If (*cardinality* > *max_cardinality*) {
        max_cardinality = *cardinality*;
        Store the estimated parameters of the prototypes as well as the value of K;
    }
    $K := K + 1$;
until ($K > C\_\max$).

The Progressive MVC Algorithm to Find Robust Estimates of the Parameters of an Unknown Number of Prototypes:

valid_component_found = TRUE;
list_of_found_clusters = EMPTY;
While (*valid_component_found* = TRUE) {
    Run the Generic Algorithm to Find a Robust Estimate of the Parameters of a Single Prototype;
/*This algorithm returns the optimal $\mathcal{C}$ if found.
Else resets *valid_component_found*
to FALSE */
    If (*valid_component_found* = TRUE) {
        Add the fuzzy set $\mathcal{C}$ and the estimated parameters of the prototype β to *list_of_found_clusters*;
        Form new data set $X = X - \mathcal{C}$
}.

What is the best breakdown point that one can achieve when the number of clusters is unknown? We have seen that robust methods break down when they cannot distinguish between a noise cluster and a good cluster. In the case when the number of clusters C is known, this happens when the validity of $\mathcal{C}$ is higher when $\mathcal{C}$ includes the noise cluster rather than the smallest legitimate cluster. This is because the algorithm is forced to pick only C clusters and, therefore, it must reject the good cluster in favor of the noise cluster. However, if we do not know how many clusters are present, then, if we can achieve a higher validity for $\mathcal{C}$ by including the noise points, we do have the choice to accept it as a legitimate solution. In other words, since our solution to robust clustering is based on cluster validity when the number of clusters is unknown, we must consider all MVC's to be legitimate because we have no basis on which to reject an MVC. From this view point, the concept of breakdown does not apply to the case when the number of clusters is unknown. This does not mean that breakdown is not an issue in such a situation. It simply means

that the system will not break down provided we accept the validity measure to be the determining test.

A comparison of the MVC approach with the generalized Hough transform (GHT) [21] gives us insights into the limitations of the GHT. In spite of its popularity, it is well known that the GHT suffers from several drawbacks. In the simplest form of the GHT, each point $\boldsymbol{x}_j$ in feature space "votes" for all possible combinations of prototype parameter values in parameter space (also known as the accumulator array) that can generate a curve passing through $\boldsymbol{x}_j$. Conversely, every "cell" in parameter space can be seen as inducing a crisp subset of the data set X. This crisp subset includes all points $\boldsymbol{x}_j$ that lie on the curve(s) generated by the set(s) of parameter values represented by the cell, and such points are said to be "accumulated" by the cell. Thus, the GHT and the MVC approaches are similar. However, there are important differences. The GHT considers crisp subsets of X, whereas the MVC approach considers fuzzy subsets. When the points do not exactly lie on a curve (i.e., when they are scattered), no crisp subset induced by a cell in parameter space can contain all the points belonging to the curve. This is true regardless of whether we quantize the parameter values coarsely or finely (see [10] for examples). Moreover, when the quantization is coarse, the accuracy of the estimates is poor. This means that any given cell in parameter space can accumulate only some of the points belonging to a curve, giving rise to the famous "bin splitting" problem. Another way of looking at this phenomenon is that no "peak" in the accumulator array passing the threshold test corresponds to the best fit, because each peak represents a valid component rather than a maximal valid component. This is a direct consequence of using only crisp subsets induced by quantized parameters rather than all fuzzy subsets. Another source of problems in the GHT is the validity measure. The GHT uses the total number of points as the validity measure, and completely ignores the gaps between the points. This can produce many false peaks when the data sets are noisy, as illustrated by Fig. 3. Curiously enough, the bin-splitting problem can be solved by applying a robust clustering algorithm to the GHT parameter space (see [26], for example). This solution is, however, computationally prohibitive in high dimensions, because both the GHT and robust clustering are highly computational in such cases. To summarize, the GHT technique is similar to the MVC approach. However, for the reasons cited above it is suboptimal. Its performance can be improved by considering fuzzy subsets and by choosing a better validity measure.

XII. Conclusion

In this paper, we have reviewed several robust approaches for clustering. The noise clustering method and the possibilistic clustering method were discussed and compared with several other techniques in detail. These two methods were shown to be identical when only one cluster was sought at a time. Furthermore, it was shown that under certain conditions, they are either identical or very similar to Ohashi's method, the potential function approach, the mountain method, and the LBFC approach. We established a robust-statistical foundation for the noise/possibilistic clustering methods by showing that they are *robust* M estimators and that they are equivalent to the IRLS approach. We established a correspondence between typicality-based membership functions in fuzzy set theory and weight functions in robust statistics. This is a significant result, because it explains why fuzzy methods that use the typicality interpretation are robust. We also pointed out the similarities between the noise/possibilistic clustering approach and four other approaches based on robust statistics, i.e., the GMVE method, CRE, minimization of probability of randomness, and the GMDD method.

Almost all clustering techniques reviewed in this paper have the same underlying principles and hence have the same type of limitations. We have shown that except for the GMVE and MINPRAN, all the other methods use an objective function of the form $\Sigma_i \Sigma_j u_{ij} d_{ij}^2$, where the membership (or weight) u_{ij} is a monotonically decreasing function of distance. The summation over i can be ignored if we treat the clusters as independent, or if we seek only one cluster at a time, or if we seek C clusters simultaneously. This objective function can be viewed as the total fuzzy intracluster distance, where the memberships are possibilistic. In other words, the memberships represent degrees of typicality. Although the GMVE and MINPRAN do not use the same objective function, their solutions are essentially equivalent to the one found by minimizing $\Sigma_i \Sigma_j u_{ij} d_{ij}^2$ (see Sections XIII and IX).

In Section XI, we have defined many general concepts that are useful in robust clustering and presented a general perspective on robust clustering. The set of generic algorithms outlined in Section XI are meant only as general guidelines along which more computationally efficient algorithms can be developed for specific applications. These algorithms are all based on the idea of validity. We believe that cluster validity plays a pivotal role in robust clustering because without the concept of validity, we could neither separate the good points from the noise points and outliers nor verify that our solution is good. The solution to the robust clustering problem requires that we reject a fuzzy subset of the data set before we compute the parameter estimates. However, it is possible to optimize the objective function very trivially by excluding all points. Therefore, we need an additional constraint such as cluster validity to avoid the trivial solution. Hence, the solution to the general clustering problem appears to be inalienable from the notion of validity. Ideally, the objective function should be the same as the cluster validity. This is stating the obvious, because the objective function defines what one is looking for; in other words, one needs to describe precisely what one is looking for before one can find it. Unfortunately, the definition of cluster validity remains one of the most elusive problems, especially when the distribution is unknown. However, when it can be defined, it does provide a key to the solution of the robust clustering problem.

The classical approach to clustering based on variations of the K-means or the fuzzy C-means is not robust. The alternative formulations based on noise clustering or possibilistic clustering are robust in that they can be shown to be founded on robust statistics. Robustness is a very desirable property for techniques used in engineering applications. However,

theoretical ways to quantify robustness as proposed in the robust statistics literature (e.g., the breakdown point) may not be highly relevant to clustering. We believe that a better way to evaluate the robustness of a procedure is to follow the verbal guidelines suggested by Huber, as stated in Section I. In particular, the first two properties are of great importance. To reiterate, the performance of a method should deteriorate only slightly under small deviations, and it should have a good efficiency (accuracy) of estimation.

There are two main reasons why the need for a very high breakdown point is irrelevant. First, the definition of the finite-sample breakdown point allows for any and all sorts of contamination. This makes the definition rather extreme in that it requires the estimate to be arbitrarily far from the actual value. As mentioned in Section I, in engineering, when the estimate is off even by a small amount, that error may be already intolerable. On the other hand, the breakdown point takes into account the worst possible scenario, whereas in engineering, it is simply too costly to build a device that works even in absurd situations. A more practical design would consider bad scenarios which are probable rather than worry about hypothetical situations. The second main reason is that the theoretical maximum for the breakdown point is only 0.5 when there is one cluster, and the theoretical maximum deteriorates rapidly when there are several clusters in the data. This means that we cannot build any system at all if we worry about the breakdown point. As explained in Section XI, a more practical approach is to realize that even when the data is contaminated beyond the theoretical breakdown point, it does not automatically mean that we cannot design a system that *almost always* works.

Apart from the problem of variety of cluster shapes, the two most challenging issues in cluster analysis are noisy data and unknown number of clusters. Most clustering methods reported in the literature cannot cope well with either of these problems. Dealing with noise alone is not trivial, and the methods reviewed in this paper attempt to solve these problems with varying degrees of success. Even though some of the techniques described here can handle noise to a reasonable extent, we conclude that none of them is reliable *and* computationally efficient when the number of clusters is *unknown*. They all perform a potentially exhaustive search by varying the value of a resolution parameter. For example, the noise clustering approach requires varying δ, the possibilistic approach requires varying η, the LBFC method requires varying β, the GMVE approach requires varying the fraction h, and the Gaussian mixture density decomposition method requires varying t_i. The CRE and MINPRAN would also require a search if they did not assume that the inlier bound or the noise distribution is known. These methods, as well as the GHT, are all variations or specific implementations of the generic algorithms presented in Section XI. They differ in the type of distance measure used for d_{ij}^2, or the type of monotonically decreasing function used for u_{ij}, or the measure used for validity, or the technique used for optimization (fixed point iterations, gradient descent, random initializations, or random subsampling). A challenging problem is to find more computationally efficient implementations of the generic algorithms described in Section XII, especially for more general prototypes in high dimensions such as linear and nonlinear manifolds that involve a very large number of parameters. For such applications, more research is required on suitable validity measures as well as other optimization issues.

On the positive side, this review also shows that several robust methods are available in the literature that can be used for solving many engineering problems, especially in applications where cluster distributions or validity measures can be defined. Although some of these methods are computationally intensive they are still better than an exhaustive search, and by combining the good features of different methods, one can design good solutions to many problems. The connection between robust clustering and robust statistics allows one to use concepts from both disciplines in conjunction with Monte Carlo techniques, gradient descent, fixed point iteration, and genetic algorithms to avoid the exhaustive search required by the generic algorithms outlined in Section XI. Finally, it is hoped that the unified view of various clustering methods presented here from the perspective of robustness is useful for researchers interested in developing better methods or using them intelligently to solve challenging problems. The equivalence of many of these methods should also caution researchers against re-inventing the wheel.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers whose constructive comments greatly improved the presentation of the paper. The authors would also like to thank their students O. Nasraoui and H. Frigui for their critical comments and for the preparation of the plots.

REFERENCES

[1] A. E. Beaton and J. W. Tukey, "The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data," *Technometrics*, vol. 16, pp. 147–185, 1974.
[2] G. Beni and X. Liu, "A least biased fuzzy clustering method," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 954–960, Sept. 1994.
[3] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms.* New York: Plenum, 1981.
[4] S. Chiu and J. J. Cheng, "Automatic rule generation of fuzzy rule base for robot arm posture selection," in *Proc. NAFIPS Conf.*, San Antonio, TX, Dec. 1994, pp. 436–440.
[5] T. Darrell and A. P. Pentland, "Cooperative robust estimation using layers of support," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 474–487, May 1995.
[6] R. N. Davé, "Fuzzy-shell clustering and applications to circle detection in digital images," *Int. J. General Syst.*, vol. 16, pp. 343–355, 1990.
[7] ——, "Characterization and detection of noise in clustering," *Pattern Recognition Lett.*, vol. 12, no. 11, pp. 657–664, 1991.
[8] ——, "Robust fuzzy clustering algorithms," in *2nd IEEE Int. Conf. Fuzzy Syst.*, San Francisco, CA, Mar. 28–Apr. 1, 1993, pp. 1281–1286.
[9] R. N. Davé and K. Bhaswan, "Adaptive fuzzy C-shells clustering and detection of ellipses," *IEEE Trans. Neural Networks*, vol. 3, pp. 643–662, May 1992.
[10] R. N. Davé and T. Fu, "Robust shape detection using fuzzy clustering: Practical applications," *Fuzzy Sets Syst.*, vol. 65, pp. 161–185, Jan. 1995.
[11] R. N. Davé and K. J. Patel, "Progressive fuzzy clustering algorithms for characteristic shape recognition," in *Proc. NAFIPS 90: Quarter Century of Fuzziness*, I. B. Turksen, Ed., June 1990, vol. I, pp. 121–124.
[12] J. J. De Gruijter and A. B. McBratney, "A modified fuzzy K-means method for predictive classification," in *Classification and Related Methods of Data Analysis*, H. H. Bock, Ed. Amsterdam, The Netherlands: Elsevier, 1988.
[13] R. Duda and P. Hart, *Pattern Classification and Scene Analysis.* New York: Wiley, 1973, ch. 4.

[14] R. Dutter, "Numerical solution of robust regression problems: Computational aspects, a comparison," *J. Statist. Computat. Simulat.*, vol. 5, pp. 207–238, 1977.
[15] T. Fu, "Robust approach to object recognition through fuzzy clustering and hough transform based methods," Ph.D. dissertation, Dept. Mech. Eng., New Jersey Inst. Technol., Newark, 1995.
[16] I. Gath and G. Geva, "Unsupervised optimal fuzzy clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, pp. 773–781, July 1989.
[17] I. Gath and Y. Man, "Detection and separation of ring-shaped clusters using fuzzy clustering," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 855–861, Aug. 1994.
[18] C. Goodall, "M-estimator of location: An outline of the theory," in *Understanding Robust and Exploratory Data Analysis*, D. C. Hoaglin, F. Mosteller, and J. W. Tukey, Eds. New York: 1983, pp. 339–403.
[19] F. R. Hampel, "Beyond location parameters: Robust concepts and methods," in *Proc. 40th Int. Statist. Inst.*, 1975, vol. 46, pp. 375–382.
[20] F. R. Hampel, E. M. Ponchotti, P. J. Rousseeuw, and W. A. Stahel, *Robust Statistics: The Approach Based on Influence Functions.* New York: Wiley, 1986.
[21] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision.* Reading, MA: Addison-Wesley, 1992, vol. I, ch. 11.
[22] R. J. Hathaway and J. C. Bezdek, "Switching regression models and fuzzy clustering," *IEEE Trans. Fuzzy Syst.*, vol. 1, pp. 195–204, Aug. 1993.
[23] ———, "Optimization of clustering criteria by reformulation," *IEEE Trans. Fuzzy Syst.*, vol. 3, pp. 241–245, May 1995.
[24] P. W. Holland and R. E. Welsch, "Robust regression using iteratively reweighted least squares," *Communication Statistics—Theory and Methods*, vol. A6, no. 9, pp. 813–827, 1977.
[25] P. J. Huber, *Robust Statistics.* New York: Wiley, 1981.
[26] J.-M. Jolion, P. Meer, and S. Bataouche, "Robust clustering with applications in computer vision," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 791–802, Aug. 1991.
[27] R. Krishnapuram, "Generation of membership functions via possibilistic clustering," in *Proc. 3rd IEEE Conf. Fuzzy Syst.*, Orlando, FL, July 1994, pp. 902–908.
[28] R. Krishnapuram and C.-P. Freg, "Fitting an unknown number of lines and planes to image data through compatible cluster merging," *Pattern Recogn.*, vol. 25, no. 4, pp. 385–400, 1992.
[29] R. Krishnapuram, H. Frigui, and O. Nasraoui, "Quadric shell clustering algorithms and their applications," *Pattern Recogn. Lett.*, vol. 14, no. 7, pp. 545–552, July 1993.
[30] ———, "Fuzzy and possibilistic shell clustering algorithms and their application to boundary detection and surface approximation: Parts I and II," *IEEE Trans. Fuzzy Syst.*, vol. 3, pp. 29–60, Feb. 1995.
[31] R. Krishnapuram and J. M. Keller, "A possibilistic approach to clustering," *IEEE Trans. Fuzzy Syst.*, vol. 1, pp. 98–110, May 1993.
[32] ———, "Fuzzy and possibilistic clustering methods for computer vision," in *Neural Fuzzy Syst.*, S. Mitra, M. Gupta, and W. Kraske, Eds., SPIE Inst. Ser., 1994, vol. IS-12, pp. 133–159.
[33] R. Krishnapuram, O. Nasraoui, and H. Frigui, "Fuzzy C spherical shells algorithm: A new approach," *IEEE Trans. Neural Networks*, vol. 3, pp. 663–671, Sept. 1992.
[34] D. G. Lowe, "Fitting parametrized three-dimensional models to images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 441–450, May 1991.
[35] F. Mosteller and J. W. Tukey, *Data Analysis and Regression.* Reading, MA: Addison-Wesley, 1977.
[36] Y. Ohashi, "Fuzzy clustering and robust estimation," in *9th Meet. SAS Users Grp. Int.*, Hollywood Beach, FL, 1984.
[37] Y. Ohta, *Knowledge-Based Interpretation of Outdoor Natural Color Scenes.* Boston, MA: Pitman Adv. Publ., 1985.
[38] J. Rissanen, "A universal prior for integers and estimation by minimum description length," *Ann. Statistics*, vol. 11, no. 2, pp. 416–431, 1983.
[39] K. Rose, E. Gurewitz, and G. C. Fox, "A Deterministic annealing approach to clustering," *Pattern Recogn. Lett.*, vol. 11, pp. 589–594, Sept. 1990.
[40] ———, "Constrained clustering as an optimization method," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 785–794, Aug. 1993.
[41] P. J. Rousseeuw, "Least median of squares regression," *J. Amer. Statistics Assoc.*, vol. 79, pp. 871–880, 1984.
[42] P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection.* New York: Wiley, 1987.
[43] C. V. Stewart, "MINPRAN: A new robust estimator for computer vision," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, no. 10, pp. 925–938, Oct. 1995.
[44] J. T. Tou and R. C. Gonzales, *Pattern Recognition Principles.* Reading, MA: Addison-Wesley, 1974.
[45] R. R. Yager and D. P. Filev, "Approximate clustering via the mountain method," *IEEE Trans. Syst. Man, Cybern.*, vol. 24, pp. 1279–1284, Aug. 1994.
[46] P. Whaite and F. P. Ferrie, "From uncertainty to visual exploration," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 1038–1049, Oct. 1991.
[47] X. Zhuang, Y. Huang, K. Palaniappan, and Y. Zhao, "Gaussian mixture density modeling, decomposition and applications," *IEEE Trans. Image Processing*, vol. 5, pp. 1293–1302, Sept. 1996.

Rajesh N. Davé received the B.Tech. degree in mechanical engineering from Indian Institute of Technology, Bombay, in 1978, and the M.S. and Ph.D. degrees in mechanical engineering from Utah State University, Logan, in 1981 and 1983, respectively.

He is currently an Associate Professor in the Department of Mechanical Engineering at New Jersey Institute of Technology, Newark. His main research interests are in the areas of pattern recognition/image processing, granular flows, and computer applications in mechanical engineering. His recent research efforts include development of clustering algorithms based on fuzzy set theory, robust clustering methods, neuro-fuzzy controllers, image and motion analysis for experimental studies of granular flows, study of microstructure in shear-induced granular flows using fractals, and development of nonintrusive rigid body tracking technique for dry granular flows.

Dr. Davé is a member of the Phi Kappa Phi honor society, the American Society of Mechanical Engineers, and the North American Fuzzy Information Processing Society.

Raghu Krishnapuram (S'83–SM'87) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Bombay, in 1978, the M.S. degree in electrical engineering from Louisiana State University, Baton Rouge, in 1985, and the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, in 1987.

From 1978 to 1979, he was with Bush India, Bombay, where he participated in developing electronic audio entertainment equipment. From 1979 to 1982 he was a Deputy Engineer at Bharat Electronics Ltd., Bangalore, India, manufacturers of defense equipment. He is currently an Associate Professor in the Electrical and Computer Engineering Department, University of Missouri, Columbia. In 1993 he visited the European Laboratory for Intelligent Techniques Engineering (ELITE), Aachen, Germany, as a Humboldt Fellow. His research encompasses many aspects of computer vision and pattern recognition. His current interests include applications of fuzzy set theory and neural networks to pattern recognition and computer vision.