A Robust Clustering Algorithm Based on Competitive Agglomeration and Soft Rejection of Outliers

Hichem Frigui
Department of Computer Engineering
and Computer Science
University of Missouri
Columbia, MO 65211
frigui@sun1.ece.missouri.edu

Raghu Krishnapuram
Department of Computer Engineering
and Computer Science
University of Missouri
Columbia, MO 65211
raghu@sun1.ece.missouri.edu

Abstract

We present a new clustering algorithm that addresses two major issues associated with conventional partitional clustering: the difficulty in determining the number of clusters, and the sensitivity to noise and outliers. The proposed algorithm determines the number of clusters by a process of competitive agglomeration. Noise immunity is achieved by integrating concepts from robust statistics into the algorithm. The proposed approach can incorporate different distance measures in the objective function to find an unknown number of clusters of various types including lines, planes and surfaces.

1 Introduction

Most traditional clustering algorithms fall into one of the following two categories [1]: hierarchical clustering and partitional clustering. In hierarchical clustering, the number of clusters need not be specified *a priori*, and problems due to initialization and local minima do not arise. Hierarchical clustering procedures consider only local neighbors and generate a nested sequence of partitions. In contrast, partitional clustering procedures generate a single partition of the data, and can incorporate knowledge about the global shape or size of clusters by using appropriate prototypes and distance measures in the objective function [5, 3, 4]. Prototype-based fuzzy clustering methods have the advantage of being dynamic in the sense that points can move from one cluster to another. They can also handle the case of overlapping clusters. The major drawbacks of the partitional approach are the difficulty in determining the number of clusters, and the sensitivity to initialization, noise and outliers.

The traditional approach to determining the "optimal" number of clusters is to pick the value of C that optimizes a global validity measure of the partition [1, 5, 6]. An alternative approach to this problem is to perform progressive clustering [4, 7, 8]. In this approach, clusters are progressively extracted from the data set, either one at a time or few at a time. In any case, these approaches are computationally expensive, and rely on validity measures (global or individual) which can be difficult to devise since one needs to take into account the variability in cluster shape, density, and size.

The proposed solutions to overcome the problem of noise and outliers can be divided into two classes. In the first class are algorithms that are derived by modifying the Fuzzy C-Means (FCM) [5] objective function [11, 12]. These algorithms are still sensitive to initialization and other parameters used by the algorithms. The second class of algorithms are based on robust estimators [13, 14, 16] which can tolerate up to 50% noise. The standard robust methods can only be used to find a single cluster in a noisy data set. Recently some of these algorithms have been extended to find multiple clusters by extracting one cluster at a time [9, 10, 15]. Since the set of outliers with respect to a cluster includes points from other clusters, the proportion of outliers can be very high. Therefore, even a breakdown point of 50% is not sufficient to make an algorithm highly robust. This problem can be overcome by considering either the validity of the data points included in the set of inliers [9, 10], or the probability that the points come from a known outlier distribution (such as uniform distribution) [15]. Thus, the validity criterion or the assumed distribution for the outliers plays a crucial role in ensuring high breakdown. However, in order to guarantee a global minimum, these algorithms may have to use many random initializations. Moreover, if the clusters overlap heavily, the idea of extracting them in a serial fashion will not work, since removing one cluster may partially destroy the structure of other clusters. Finally, all the current algorithms use hard rejection, which performs poorly when there are points in the "region of doubt", i. e., the region between inliers and outliers. Hard rejection also means that we need to start with a good initialization, because some of the points are completely ignored from the start. Thus, both from a cluster overlap point of view and outlier rejection point of view, a fuzzy approach has advantages.

In this paper, we introduce a new approach called Robust Competitive Agglomeration (RCA), that addresses the drawbacks of the above traditional methods. The RCA algorithm minimizes a fuzzy prototype-based objective function, so that it can be used to find clusters of various shapes. The objective function includes an agglomerative component in order to incorporate the advantages of hierarchical clustering, and a constraint on the memberships in order to introduce competition among clusters. The algorithm generates a sequence of partitions with a progressively diminishing number of clusters. When the algorithm terminates, the final partition is assumed to have the "optimal" number of clusters. However, in contrast to traditional hierarchical clustering, points can move from one cluster to another, and the sequence of partitions is not necessarily nested. Since the algorithm starts with a large number of small clusters, the final result is less sensitive to initialization and local minima. To achieve robustness, the RCA algorithm incorporates the concept of weight functions from robust statistics.

The organization of the rest of the paper is as follows. In section 2, we present the RCA algorithm. In section 3, we discuss how one can choose the weight functions to suit the given application. In section 4, we illustrate the performance of the RCA algorithm with examples. Finally, section 5 contains the summary conclusions.

2 The Robust Competitive Agglomeration Algorithm

Let $\mathcal{X} = \{\mathbf{x}_j \mid j = 1, \ldots, N\}$ be a set of N vectors in an n-dimensional feature space. Let $\mathbf{B} = (\beta_1, \ldots, \beta_c)$ represent a C-tuple of prototypes each of which characterizes one of the C clusters. Each β_i consists of a set of parameters that describe the cluster. The RCA algorithm minimizes the following objective function

$$J_R = \sum_{i=1}^{C}\sum_{j=1}^{N}(u_{ij})^2\rho_i(d_{ij}^2) - \alpha\sum_{i=1}^{C}\left[\sum_{j=1}^{N} w_{ij}u_{ij}\right]^2 \quad (1)$$

Subject to

$$\sum_{i=1}^{C} u_{ij} = 1, \qquad for\ j \in \{1, \cdots, N\}. \quad (2)$$

In (1), d_{ij}^2 represents the distance from feature vector $\mathbf{x}_j$ to the prototype β_i, u_{ij} represents the degree of membership of feature point $\mathbf{x}_j$ in cluster β_i, $U = [u_{ij}]$ is a $C \times N$ matrix called a constrained fuzzy C-partition matrix, and $\mathbf{B} = (\beta_1, ...\beta_c)$ is a C-tuple of prototypes. The objective function in (1) has two components. The first component can be viewed as a generalization of the M-estimator [16] to detect C (possibly overlapping) clusters *simultaneously*. The global minimum of this component is achieved when the number of clusters C is equal to the number of "good" sample points, i. e., each cluster contains a single good data point. The membership values, u_{ij}, which represent degrees of sharing, are used in this component to generate a fuzzy partition of this data set. This allows the clusters to share points so that points are not committed to clusters permaturely. In addition, these constrained memberships are needed to create a competitive environment, as will be explained later. The second component of J_R in (1) is used to maximize the number of "good" points in each cluster. The weights w_{ij} (to be defined) represent degrees of "goodness" or typicality of point $\mathbf{x}_j$ with respect to cluster i, and are used to generate robust estimates of the prototype parameters. The global minimum of this term (including the negative sign) is achieved when all points are lumped in one cluster, and all other clusters are empty. When both components are combined with a proper choice of the agglomeration parameter α, the final partition will minimize the sum of intra-cluster distances while partitioning the data set into the smallest possible number of clusters.

To minimize J_R with respect to the parameter vectors, $\mathbf{B}$, we fix $\mathbf{U}$ and set the gradient to zero

$$\sum_{j=1}^{N}(u_{ij})^2 w_{ij}\frac{\partial d_{ij}^2}{\partial \beta_i} = 0, \text{where } w_{ij} = \frac{\partial \rho_i(d_{ij}^2)}{\partial d_{ij}^2}. \quad (3)$$

Further simplification of this equation depends on the loss function $\rho()$ and the distance measure used, and will be discussed later in this section. It can be shown that the weight w in (3) is equivalent to the weight function of the W-estimator [17]. The choice of the weight function has a major influence on the robustness of our RCA algorithm, and will be discussed in section 3.

To minimize (1) with respect to $\mathbf{U}$ subject to (2), we apply Lagrange multipliers and obtain

$$J = \sum_{i=1}^{C}\sum_{j=1}^{N}(u_{ij})^2\rho(d_{ij}^2) - \alpha\sum_{i=1}^{C}\left[\sum_{j=1}^{N} u_{ij}w_{ij}\right]^2 - \sum_{j=1}^{N}\lambda_j\Big(\sum_{i=1}^{C} u_{ij} - 1\Big). \quad (4)$$

We then fix $\mathbf{B}$ and solve

$$\frac{\partial J}{\partial u_{st}} = 2u_{st}\rho(d_{st}^2) - 2\alpha\sum_{j=1}^{N} u_{sj}w_{sj} - \lambda_t = 0 \quad (5)$$

to obtain an updating equation for the memberships u_{st}. The solution can be simplified considerably by assuming that the membership values do not change significantly from one iteration to the next, and by computing the term $\sum_{j=1}^{N} u_{sj}w_{sj}$ in (5) using the membership values from the previous iteration. With this assumption, (5) reduces to

$$u_{st} = \frac{2\alpha \times N_s + \lambda_t}{2\rho(d_{st}^2)}, \quad (6)$$

where

$$N_s = \sum_{j=1}^{N} u_{sj}w_{sj}, \quad (7)$$

is the robust cardinality of cluster s. Using (6) and (2), we can eliminate λ_t from (6) to obtain the following update equation for the membership of feature point $\mathbf{x}_t$ in cluster β_s

$$u_{st} = u_{st}^{\mathrm{RR}} + u_{st}^{\mathrm{Bias}}, \tag{8}$$

where

$$u_{st}^{\mathrm{RR}} = \frac{\frac{1}{\rho(d_{st}^2)}}{\sum_{k=1}^{C} \frac{1}{\rho(d_{kt}^2)}} \tag{9}$$

is the membership term which takes into account only the robust relative distances of the feature point to all clusters, and

$$u_{st}^{\mathrm{Bias}} = \frac{\alpha}{\rho(d_{st}^2)} \left(N_s - \overline{N}_t\right). \tag{10}$$

In (10),

$$\overline{N}_t = \frac{\sum_{k=1}^{C} \frac{1}{\rho(d_{kt}^2)} N_k}{\sum_{k=1}^{C} \frac{1}{\rho(d_{kt}^2)}}$$

is simply a weighted average of the cluster cardinalities. Thus, u_{st}^{Bias} is a signed bias term which is positive for clusters with cardinality higher than average, and negative for low cardinality clusters. It can be seen from (8) and (10) that the membership values of data points in spurious (low-cardinality) clusters are depreciated heavily when their distances to low-cardinality clusters are small. Moreover, when a feature point $\mathbf{x}_j$ is close to only one cluster (say cluster i), and far from other clusters, $u_{ij} \approx u_{ij}^{\mathrm{RR}}$, and no competition is involved. Thus, the algorithm encourages agglomeration of clusters with high cardinality and decay of clusters with low cardinality. When the cardinality of a cluster drops below a threshold, we discard the cluster, and decrement the number of clusters.

To ensure a balance between partitioning and agglomeration, the value of α should be chosen such that the two terms in (1) are of the same order of magnitude. Intuitively, it is better to start with a large value of α and decrease it slowly in each iteration to help the RCA algorithm seek a partition with a number of clusters that is close to the "optimum" within the first few iterations. Therefore an appropriate choice of α in iteration k is

$$\alpha(k) = \eta_0 e^{-k/\tau} \frac{\sum_{i=1}^{C} \sum_{j=1}^{N} (u_{ij})^2 \rho(d_{ij}^2)}{\sum_{i=1}^{C} \left[\sum_{j=1}^{N} u_{ij} w_{ij}\right]^2}. \tag{11}$$

where η_0 is the initial value and τ is the time constant.

It should be noted that depending on the value of α, the membership values u_{ij} may not be confined to the interval $[0, 1]$. therefore, they may need to be clipped. We now discuss the necessary conditions to minimize J_R with respect to the prototypes for the special cases of ellipsoidal and linear clusters.

2.1 Ellipsoidal cloud clusters

To detect ellipsoidal clusters, we use the following distance measure proposed in [18]:

$$d_{Cij}^2 = \left|\mathbf{C}_i\right|^{1/n} (\mathbf{x}_j - \mathbf{c}_i)^T \mathbf{C}_i^{-1} (\mathbf{x}_j - \mathbf{c}_i). \tag{12}$$

In (12), $\mathbf{c}_i$ is the center of cluster β_i, and $\mathbf{C}_i$ is its covariance matrix. Using (3), It can be shown that the update equation for the centers is

$$\mathbf{c}_i = \frac{\sum_{j=1}^{N} (u_{ij})^2 w_{ij} \mathbf{x}_j}{\sum_{j=1}^{N} (u_{ij})^2 w_{ij}}, \tag{13}$$

and the update equation for the covariance matrices is

$$\mathbf{C}_i = \frac{\sum_{j=1}^{N} (u_{ij})^2 w_{ij} \left(\mathbf{x}_j - \mathbf{c}_i\right)\left(\mathbf{x}_j - \mathbf{c}_i\right)^T}{\sum_{j=1}^{N} (u_{ij})^2 w_{ij}}. \tag{14}$$

It is easily verified that, if we assume $\mathbf{C}_i = \sigma^2 \mathbf{I}_n$, then (12) reduces to the Euclidean distance. This simplified version can be used when the clusters are expected to be spherical.

2.2 Linear clusters

To detect clusters that resemble lines or planes, we use a generalization of the distance measure proposed in [5, 2]. This distance is given by

$$d_{Lij}^2 = \sum_{k=1}^{n} \nu_{ik} \left((\mathbf{x}_j - \mathbf{c}_i) \bullet \mathbf{e}_{ik}\right)^2, \tag{15}$$

where $\mathbf{e}_{ik}$ is the k^{th} unit eigenvector of the covariance matrix $\mathbf{C}_i$. The eigenvectors are assumed to be arranged in ascending order of the corresponding eigenvalues. The value of ν_{ik} in (15) is chosen dynamically, i.e.,

$$\nu_{ik} = \frac{\lambda_{i1}}{\lambda_{ik}},$$

where ν_{ik} is the k^{th} eigenvalue of the covariance matrix $\mathbf{C}_i$. It can be shown that for the distance measure in (15), the update equations for the centers and the covariance matrices are given by (13) and (14) respectively.

3 Choosing the weight function

The choice of the weight function depends on the type of clusters expected in the data set. For regular clusters representing point clouds, we suggest a weight function that satisfies the following properties:

- $\mathcal{P}_1$: $w(d^2)$ is monotonically non-increasing.
- $\mathcal{P}_2$: $w(d^2) = 0$ for $d^2 > T + \alpha S$, where T and S are robust estimates of the average squared distance and its standard deviation respectively, and α is a constant.
- $\mathcal{P}_3$: $w(0) = 1$, $w(T) = 0.5$, and $w\prime(0) = 0$.

Each cluster will have a different estimate of T and S, which are estimated as follows

$$T_i = k_1 \times \underset{x_j \in \mathcal{X}_i}{Med} (d_{ij}^2) \quad \text{for } 1 \le i \le C \tag{16}$$

$$S_i = k_2 \times \underset{x_j \in \mathcal{X}_i}{MAD} (d_{ij}^2) \quad \text{for } 1 \le i \le C \tag{17}$$

where

$$\mathcal{X}_i = \left\{ \mathbf{x}_j | d_{ij}^2 \le d_{kj}^2 \quad \forall k \neq i \right\}$$

and k_1 and k_2 are suitable constants. A weight function that satisfies the previous properties is defined by

$$w_i(d^2) = \begin{cases} 1 - \frac{d^4}{2T_i^2} & 0 \le d^2 \le T_i, \\ \frac{\left[d^2 - (T_i + \delta S_i)\right]^2}{2\delta^2 S_i^2} & T_i < d^2 \le T_i + \delta S_i, \\ 0 & d^2 > T_i + \delta S_i. \end{cases} \tag{18}$$

The loss function, $\rho_i()$, associated with this weight function can be obtained by integrating (18), i.e,

$$\rho_i(d^2) = \begin{cases} d^2 - \frac{d^6}{6T_i^3} & 0 \le d^2 \le T_i, \\ \frac{\left[d^2 - (T_i + \delta S_i)\right]^3}{6\delta^2 S_i^2} + \frac{5T_i + \delta S_i}{6} & T_i < d^2 \le T_i + \delta S_i, \\ \frac{5T_i + \delta S_i}{6} + K_i & d^2 > T_i + \delta S_i. \end{cases}$$

where the K_i are constants used to make all ρ_i functions reach the same maximum value. We choose

$$K_i = \max_{1 \le j \le C} \left\{ \frac{5T_j + \delta S_j}{6} \right\} - \frac{5T_i + \delta S_i}{6} \quad \text{for } 1 \le i \le C.$$

The constants K_i are added to force all functions to have the same maximum value, so that noise points will have the same membership in all clusters.

It can be seen that the above weight function uses "soft" or "fuzzy" rejection, which means that the algorithm does not blindly cut off points in the region of doubt. This makes the algorithm less susceptible to local minima. It also improves the efficiency of the estimate.

The RCA algorithm is summarized below

Fix the maximum number of clusters $C = C_{max}$;
Initialize the prototype parameters, and set $k = 0$;
Set w_{ij}=*1* $\forall$ i, j, *and compute* N_i *by using (7);*
Repeat
  Compute d_{ij}^2 *for* $1 \le i \le C$ *and* $1 \le j \le N$;
  Estimate T_i *and* S_i *by using (16) and (17);*
  Update the weights w_{ij} *by using (18);*
  Update $\alpha(k)$ *by using (11);*
  Update the partition matrix $\mathbf{U}^{(k)}$ *by using (8);*
  Compute the robust cardinality N_i *by using (7);*
  If ($N_i < \epsilon_1$) *Discard cluster* β_i;
  Update the number of clusters C;
  $k = k + 1$;
  Update the prototype parameters
Until $\Big($ *prototype parameters stabilize* $\Big)$;

---

4 Experimental Results

The examples used in this section consist of both synthetic and some real images. In all examples shown in this section, the RCA algorithm was initialized by running the Gustafson-Kessel algorithm [18] for 5 iterations when the distance in (12) is used, or by running the Adaptive Fuzzy Clustering algorithm [2] for 5 iterations if the distance in (15) is used. The initial value of η (η_0) was set to 1.5 and the time constant τ was set to 10. In all examples, we tried many different values for C_{max} ranging from 15 to 30, and in every case, the algorithm converged to the same final partition.

Fig. 1(a) shows a synthetic data set consisting of 4 Gaussian clusters of various sizes and orientations. Uniformly distributed noise was added to the data set so that the noise points constitute about 40% of the total points. Fig. 1(b) shows the initial 20 prototypes superimposed on the data set. The "+" signs indicate the cluster centers, and the ellipses shown in the figure enclose points having a Mahalanobis distance less than 9. After running the RCA algorithm for 2 iterations, 9 empty clusters are discarded. The prototypes of the remaining 11 clusters are shown in Fig. 1(c). The number of clusters is reduced to 6 after 3 iterations, and to 4 after 4 iterations. The final result after a total of 10 iterations is shown in Fig. 1(d).

Fig. 2(a) shows a synthetic fractal image containing a triangular region with two rectangular regions in the background. Fig. 2(b) shows the results of a fractal edge detector [19]. As can be seen in Fig. 2(b), the edges are ill defined, and there are many noise points. The prototypes found by the RCA algorithm, using the distance in (15), are shown in Fig. 2(c) superimposed on the edge image.

Fig. 3(a) shows a real 200×200 range image of a block obtained from ERIM. Fig. 3(b) depicts a cross-section of this image, to illustrate the degree of noise. The result of the RCA algorithm (using d_L in (15)) is displayed in Fig. 3(c), which consists of the correct number of planar surfaces. Each surface is displayed with a different gray value. Noise points (i.e. points having zero weight (w_{ij}) in all clusters) are shown as black pixels in the figure.

Fig. 4 represents the results of segmentation of a color image obtained from the University of Massachusetts. In this example a sampling rate of 3 in the x and y directions was used to reduce the computations. Fig. 4(a) shows the original intensity image. Two color features, red-blue difference (=r-b) and excess green (=$2g$-r-b), were used in the segmentation. From the plot of the feature points in Fig. 4(b), we observe that the data set is noisy and the optimum number of clusters is not obvious. "Noise points" can also include feature vectors from miscellaneous tiny classes whose points are mostly lost while sampling the feature vectors. The RCA algorithm identified 5 clusters in the data set. Note that in Fig. 4(b), each point can represent several identical feature vectors. The prototype parameters of these clusters are shown in Fig. 4(b) superimposed on the original feature point data. Fig. 4(c) shows the segmentation result when each pixel is assigned to the class with the highest membership u_{ij}. Noise points are not assigned to any

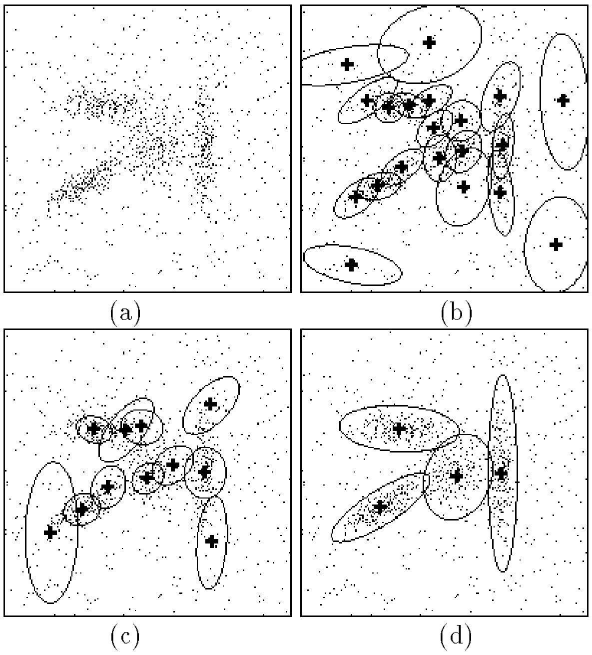

Figure 1: Intermediate results of the RCA algorithm (a) Original image, (b) prototypes used for initialization. Results at the end of (c) 2 iterations, and (d) 10 iterations(convergence).

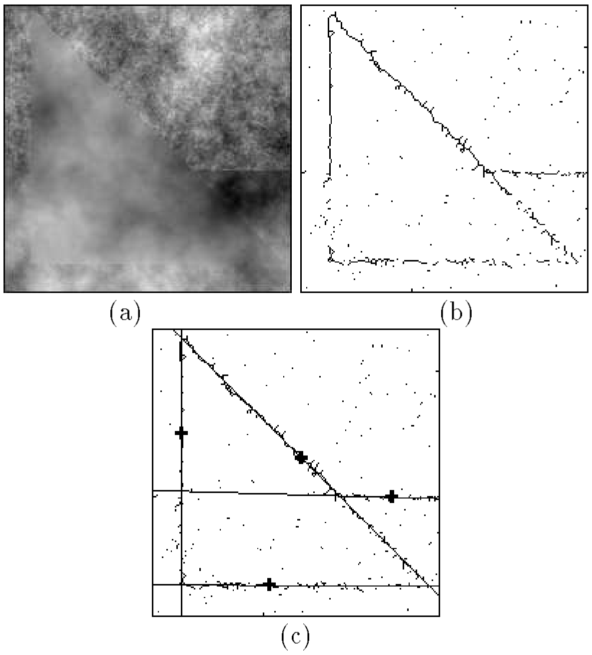

Figure 2: Results of the RCA algorithm on a noisy data set with linear clusters
(a) Image of a triangular fractal region, (b) edge image, (c) prototypes found by the RCA algorithm.

class and are shown as white pixels in the figure.

5 Conclusion

In this paper, we have presented a new robust and computationally attractive clustering algorithm. Our algorithm minimizes a fuzzy prototype-based objective function, and produces a sequence of partitions with a decreasing number of clusters. The initial partition has an overspecified number of clusters, and the final one has the "optimal" number of clusters. We have formulated the RCA algorithm independently of the distance measure, which makes the proposed algorithm easily adaptable to different distance measures. Concepts from robust statistics have been incorporated into the RCA algorithm to make it insensitive to noise and outliers. The RCA algorithm uses two sets of memberships. Constrained fuzzy memberships are used to generate a fuzzy partition and to create a competitive environment which reduces the number of clusters from iteration to iteration. Possibilistic [12] or typicality based memberships are used to obtain robust estimates of the prototype parameters. The RCA algorithm is also relatively insensitive to initialization and local minima effects when compared with traditional objective-function based clustering algorithms. This is due to its agglomerative property, i.e., each cluster is initially approximated by many prototypes.

The RCA algorithm has a time complexity similar to the FCM algorithm [5] ($\mathcal{O}(NC)$) since the memberships u_{ij} and the prototype parameters are updated in a similar way. The weight function $w(d^2)$ requires computing the median, which can be computed iteratively in $\mathcal{O}(\log N)$ passes through the data set. In general, it is expected that the RCA algorithm will give robust estimates of all prototype parameters as long as the number of noise points does not exceed the number of points of the least dense cluster and all noise points do not "conspire" to form another cluster around the least dense cluster.

Acknowledgments

This work was partially supported by a grant from the Office of Naval Research (N00014-96-1-0439)

References

[1] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data,* Englewood Cliffs, NJ: Prentice Hall, 1988.

[2] R. N. Davé, "Use of the adaptive fuzzy clustering algorithm to detect lines in digital images," *Intelligent Robots and Computer Vision VIII*, vol. 1192, pp. 600-611, 1989.

[3] R. N. Davé and K. Bhaswan, "Adaptive Fuzzy C-Shells Clustering and detection of ellipses," *IEEE Trans. on Neural Networks*, Vol. 14, pp. 643-662, 1992.

[4] R. Krishnapuram, H. Frigui, and O. Nasraoui, "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, 1992.

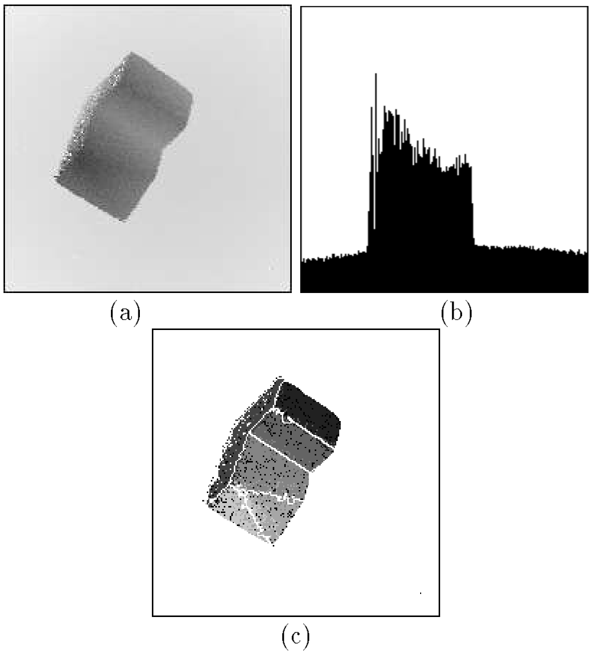

Figure 3: Results of the RCA algorithm on a noisy range image
(a) Original noisy block, (b) cross-section of the image, (c) planar surfaces obtained by the RCA algorithm.

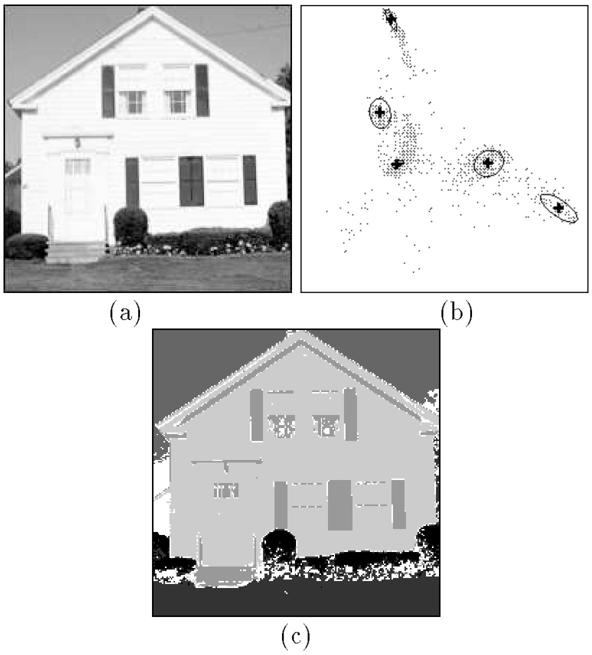

Figure 4: Segmentation of an outdoor scene.
(a) Intensity image, (b) clustered feature space, (c) segmented image.

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

[6] I. Gath and A. B. Geva, "Unsupervised optimal fuzzy clustering," *IEEE Trans. Patt. Anal. Mach. Intell.*, Vol. 11, pp. 773-781, 1989.

[7] R. N. Davé and K. J. Patel, "Progressive fuzzy clustering algorithms for characteristic shape recognition," *Proc. N. Am. Fuzzy Inf. Process. Soc.*, Toronto, pp. 121-124, 1990.

[8] R. Krishnapuram, O. Nasraoui, and H. Frigui, "The Fuzzy C Spherical Shells algorithms: a new approach," *IEEE Trans. Neural Net.*, Vol. 3 pp. 663-671, 1992.

[9] J. M. Jolion, P. Meer, and S. Bataouche, "Robust clustering with applications in computer vision," *IEEE Trans. Patt. Anal. Mach. Intell.*, Vol. 13, pp. 791-802, 1991.

[10] X. Zhuang, T. Wang, and P. Zhang, "A highly robust estimator through partially likelihood function modeling and its application in computer vision," *IEEE Trans. Patt. Anal. Mach. Intell.*, Vol. 14, pp. 19-34, 1992.

[11] R. N. Davé, "Characterization and detection of noise in clustering," *Pattern Recognition Letters*, Vol. 12, pp. 657-664, 1991.

[12] R. Krishnapuram and J. Keller, "A Possibilistic approach to clustering," *IEEE Trans. Fuzzy Syst.*, Vol. 1, pp. 98-110, 1993.

[13] P. Meer, D. Mintz, A. Rosenfeld, and D. Y. Kim, "Robust regression methods for computer vision: a review," *Intern. J. Comput. Vision*, Vol. 6, pp. 59-70, 1991.

[14] P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection,* John Wiliey & Sons, New York, 1987.

[15] C. V. Stewart, "MINPRAN: A New Robust Estimator for Computer Vision," *IEEE Trans. Patt. Anal. Mach. Intell.*, Vol. 17, pp. 925-938, 1995.

[16] P. J. Huber, *Robust Statistics,* John Wiliey & Sons, New York, 1981.

[17] P. W. Holland and R. E. Welsch, "Robust regression using iteratively reweighted least-squares," *Commun. Statist. Theor. Meth.*, Vol. A6, pp. 813-827, 1977.

[18] E. E. Gustafson and W. C. Kessel, "Fuzzy clustering with a fuzzy covariance matrix," *Proc. IEEE CDC*, San Diego, Calif., pp 761-766, 1979.

[19] B. Goldstein, *One Variable Fractal Analysis for Edge Detection,* Honors Project, Dept. of ECE, University of Missouri, Columbia, May 1992.