Robust Clustering with Applications in Computer Vision

Jean-Michel Jolion, Peter Meer, and Samira Bataouche

Abstract—**A novel clustering algorithm based on the minimum volume ellipsoid (MVE) robust estimator recently introduced in statistics is proposed. The MVE estimator identifies the least volume region containing h percent of the data points. The clustering algorithm iteratively partitions the space into clusters without *a priori* information about their number. At each iteration, the MVE estimator is applied several times with values of h decreasing from 0.5. A cluster is hypothesised for each ellipsoid. The shapes of these clusters are compared with shapes corresponding to a known unimodal distribution by the Kolmogorov-Smirnov test. The best fitting cluster is then removed from the space, and a new iteration starts. Constrained random sampling keeps the amount of computation low. The clustering algorithm was successfully applied to several computer vision problems formulated in the feature space paradigm: multithresholding of gray level images, analysis of the Hough space, range image segmentation.**

Index Terms— **Clustering, feature space, Hough transform, multithresholding, range image segmentation, robust estimation.**

I. Introduction

Several computer-vision tasks can be formulated as the following pattern-recognition problem: Given a set of points in a p-dimensional space, find the best partition of the space into significant clusters. The points in the space correspond to feature vectors extracted from the image, and the space is called feature space. Each of the delineated clusters then represents a subset of the image. Different transformations of the input image yield different feature spaces and allow discrimination of different image subsets.

The simplest feature is the gray level of a pixel. The corresponding feature space is the histogram of the image $p = 1$. Clustering in this space yields a decomposition of the histogram into a few nonoverlapping intervals, and labeling of the clusters results in multithresholding of the image. A different feature vector is defined by the parameters of a line connecting two significant edge pixels. The feature space is a variant of Hough space ($p = 2$) and the delineated clusters correspond to the most significant lines in the image. Color images can be characterized by three-dimensional histograms, and thus, we have a three-dimensional feature space in which clusters correspond to pixels with similar color. To impose spatial contiguity, the coordinates must also be mapped into the feature space, thus increasing its dimension to five. Haralick and Watson [8] introduced the facet model describing images by a piecewise polynomial surface structure. When the locally estimated facet parameters are mapped into a feature space, the space is three-dimensional for planar and six-dimensional for biquadratic surfaces.

The books of Duda and Hart [4] and Jain and Dubes [13] give complete treatments of clustering techniques with applications in pattern recognition and computer vision. Most of the techniques are iterative and sensitive to the choice of k, which is the number of clusters in the feature space. If this choice is very incorrect, the final partition of the space may be incorrect as well. This sensitivity is an important limitation in computer vision, where only rarely is the number of significant feature properties known *a priori*.

Computational considerations often limit the resolution of the feature space, that is, the feature space is represented as a discrete array, and whenever the distance between two feature vectors is less than the quantization step, the points are allocated to the same bin. A discrete feature space can represent the image without loss only if the components of the feature vector are discrete variables. Histograms are an example. However, when the feature vector components can take any value, i.e., they are continuous variables, coarse quantization of the feature space can yield artifacts. For example, spurious lines may be derived from the Hough space if the size of the accumulator bins is too large.

Nevertheless, approaching computer vision problems as feature space clustering has numerous advantages. The feature space can integrate multiple sources of information about the same image region, allowing a more complete analysis. Feature vectors corrupted with nonGaussian and/or correlated noise can be handled as long as they give rise to clusters.

In this paper, we propose a novel approach to clustering in feature space based on a recently introduced robust statistical technique. The robust method reduces the interference between adjacent clusters and allows their reliable discrimination without any *a priori* knowledge of their number. The algorithm does not require quantization of the continuous feature vectors and therefore eliminates the above-mentioned artifacts.

The algorithm is described in Section II. In Section III, we apply it to multithresholding of gray images and in Section IV to analysis of the continuous Hough space. Range image segmentation by clustering in the facet space is presented in Section V. Mapping into and from the feature space is specific

Manuscript received December 2, 1990; revised April 15, 1991. This work was supported by the Air Force Office of Scientific Research under grant AFOSR–86–0092.

J-. M. Jolion and S. Bataouche are with the Laboratoire d'Informatique Université, Villeurbanne Cedex, France.

P. Meer is with the Department of Electrical and Computer Engineering, Rutgers University, Piscataway, NJ 08855-0909 and the Center for Automation Research, University of Maryland, College Park, MD 20742.

IEEE Log Number 9101440.

for an application and, therefore, is described in the respective sections. The place of our method among other clustering procedures is discussed in Section VI.

II. The Clustering Algorithm

Let $\mathbf{X}$ be a set of n distinct data points (feature vectors) in a p-dimensional feature space:

$$\mathbf{X} = \{\mathbf{x}_i; i = 1, \ldots, n\} \quad \mathbf{x}_i = \begin{pmatrix} x_1^i \\ \vdots \\ x_p^i \end{pmatrix}. \tag{1}$$

Every point has associated with it a scalar positive weight q_i. The weights describe the importance of the points in $\boldsymbol{X}$; their definition is application dependent. In general, q_i should not be an integer; however, for the applications discussed in this paper, we can restrict ourselves to integer-valued weights. In particular, q_i can represent the multiplicity of $\boldsymbol{x}_i$, i.e., how many local regions in the image are mapped into the same point in the feature space. The quantity $Q = \sum_{i=1}^{n} q_i$, representing the number of feature vectors extracted from the input image, will be of importance. It can be regarded as a measure of the "mass" present in the feature space.

Significant subsets are extracted from the image by partitioning the feature space into k p-dimensional clusters. The number of these clusters k is not known *a priori*. The clustering algorithm proposed in this paper is based on the robust minimum volume ellipsoid estimator (MVE) of Rousseeuw and Leroy [24].

The algorithm has an iterative nature. Denote by $\boldsymbol{X}_l$ the set of data points contained in the feature space at the l-th iteration. From this set, the best cluster (to be characterized below) is delineated and removed yielding the new set $\boldsymbol{X}_{l+1}$. The process stops whenever the number of remaining points becomes less than the assumed minimum cluster size or the number of detected clusters exceeds an upper bound. To extract a cluster, the space $\boldsymbol{X}_l$ is analyzed at different "resolutions" characterized by a step size h, $h \leq 0.5$. For a given value of h, we seek the minimum volume ellipsoid containing fraction h of the mass of $\boldsymbol{X}_l$, $Q_l = \sum_{X_l q_i}$. A cluster is delineated based on this ellipsoid, and its shape is compared with the shape of an ideal cluster generated by a Gaussian density. This assumption is not crucial and can be relaxed to any unimodal elliptical distribution [25]. The Kolmogorov-Smirnov test of fit is employed for the comparison, and the returned significance level characterizes the cluster delineated at resolution h. The cluster yielding the smallest significance level over all h values is the best cluster in $\boldsymbol{X}_l$. We call the algorithm the generalized minimum volume ellipsoid (GMVE) clustering since it employs several values of h, whereas the original MVE estimator has $h = 0.5$ and all $q_i = 1$. We now describe the algorithm in detail.

A. Finding the Minimum Volume Ellipsoid for a Given h

The number of data points n is usually very large. To keep the amount of computation at a feasible level, a random sampling procedure must be applied. Let $J = \{i_1, \ldots i_{p+1}\}$ define a sample of $p+1$ points drawn from $\boldsymbol{X}_l$. These points must not lie in the same $(p-1)$-dimensional hyperplane. We return to the problem of random sampling in Section II-C.

For this sample, the mean vector $\overline{\mathbf{x}}_J$ and the covariance matrix $\boldsymbol{C}_J$ are computed by using the least squares estimates

$$\overline{\mathbf{x}}_J = \frac{1}{q_J} \sum_{i \in J} q_i \mathbf{x}_i$$

$$\mathbf{C}_\mathrm{J} = \frac{1}{q_\mathrm{J} - 1} \sum_{i \in J} q_i (\mathbf{x}_i - \overline{\mathbf{x}}_J)\,(\mathbf{x}_i - \overline{\mathbf{x}}_J)^t \tag{2}$$

where the superscript t means transpose, and $q_J = \sum_{i \in J} q_i$. The quantities (2) define a cluster centered on $\overline{\mathbf{x}}_J$ with its shape determined by $\boldsymbol{C}_J$. The quadratic forms

$$d_i^2 = (\mathbf{x}_i - \overline{\mathbf{x}}_J)^t C_J^{-1} (\mathbf{x}_i - \overline{\mathbf{x}}_J) \quad i = 1, \ldots, n \tag{3}$$

(which is also known as squared Mahalanobis distances) characterize the distance of a point $\mathbf{x}_i$ from the cluster center. A constant squared Mahalanobis distance $m_{J,h}^2$ defines an ellipsoid in the p-dimensional space centered on $\overline{\mathbf{x}}_J$. The volume of the ellipsoid is proportional to the quantity

$$\gamma_{J,h} = (m_{J,h})^p \cdot |\boldsymbol{C}_J|^{1/2} \tag{4}$$

where $|\boldsymbol{C}_J|$ is the determinant of the covariance matrix. We are interested in the value $m_{J,h}$ for which $hQ_l (h \leq 0.5)$ of the mass in $\boldsymbol{X}_l$ is contained inside the ellipsoid. If the distances d_i (3) are ordered in ascending order, we have

$$m_{J,h} = d_{[j]} \tag{5}$$

where $d_{[j]}$ is the j-th element in the ordered list, and j is defined as the largest integer for which

$$\sum_{i=1}^{j} q_i \leq hQ_l. \tag{6}$$

The minimum volume ellipsoid containing hQ_l mass is then found by minimizing over the different samples J:

$$\gamma_h = \min_J \gamma_{J,h}. \tag{7}$$

The number of samples necessary to obtain a γ_h close to the global minimum will be discussed in Section II-C. Let J_h be the sample yielding the minimum in (7) and define

$$\mathbf{T}_\mathrm{h}^0 = \overline{\mathbf{x}}_{J_\mathrm{h}} \quad \mathbf{C}_\mathrm{h}^0 = \frac{1}{\chi^2_{\mathrm{p,h}}} \cdot m^2_{J_h,h} \cdot \mathbf{C}_{J_h} \tag{8}$$

where the superscript 0 indicates that the quantities are only initial estimates. The covariance matrix is weighted in (8) by the chi-square correction factor for consistency at multivariate normal data and by the Mahalanobis distance (5) for scaling. Rousseeuw and van Zomeren [25] recommend an additional correction factor for small data sizes, but in our applications, the factor's value is close to one. On-line computation of the values of the χ^2 distribution uses numerical approximation of the incomplete Gamma function [21]. Note that we made use of the usual assumption that the core of the cluster can be approximated by a p-dimensional Gaussian cluster. Clusters generated by other unimodal densities can also be considered.

In the latter case, however, the validity of chi-square values as thresholds and correction factors should be carefully examined. The quality of our experimental results based on real data does not justify the use of nonGaussian cluster models.

The effect of severe contaminations on the detected cluster from data belonging to adjacent clusters can be reduced if weights are defined for every data point $i = 1, \ldots, n$:

$$w_i = \begin{cases} 1 & \text{if } (\mathbf{x}_i - \mathbf{T}_\mathbf{h}^0)^t (\mathbf{C}_h^0)^{-1} (\mathbf{x}_i - \mathbf{T}_\mathrm{h}^0) \le \chi^2_{\mathrm{p},0.975} \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

The parameters of the minimum volume ellipsoid are then recomputed using the weighted estimators

$$\mathbf{T}_h = \frac{\sum_{i=1}^{n} w_i q_i \mathbf{x}_i}{\sum_{i=1}^{n} w_i q_i}$$

$$\mathbf{C}_h = \frac{\sum_{i=1}^{n} w_i q_i (\mathbf{x}_i - \mathbf{T}_h)(\mathbf{x}_i - \mathbf{T}_h)^t}{\sum_{i=1}^{n} w_i q_i - 1}. \tag{10}$$

Note that the estimators in (10) are optimum in least squares sense and thus improve the efficiency of the algorithm. The number of points in the cluster is $N_h = \sum_{i=1}^{n} w_i$, which in general will differ from j (6).

B. Finding the Best Cluster

The minimum volume ellipsoid locates the region with the highest density of points in $\boldsymbol{X}_l$. We next allocate to these points a cluster quality measure. We compare the shape of the cluster inside the ellipsoid with the shape of the equivalent multivariate normal cluster using the Kolmogorov-Smirnov test. Let the number of points contained inside the ellipsoid

$$\Omega_h : (\mathbf{x}_i - \mathbf{T}_h)^t \mathbf{C}_h{}^{-1} (\mathbf{x}_i - \mathbf{T}_h) < \chi^2_{p,0.975} \tag{11}$$

be $i_{\max}$. The Mahalanobis distances of these points from the center (10) are then computed and put in ascending order $d_{[i]}$, $i = 1, \ldots, i_{\max}$. The measured (empirical) cumulative distribution function is defined as

$$F_{obs}(i) = \begin{cases} 0 & \text{if } i = 0 \\ \dfrac{\sum_{j=1}^{i} q_j}{\sum_{j=1}^{i_{\max}} q_j} & \text{if } 1 \le i \le i_{\max} \\ 1 & \text{if } i \ge i_{\max} \end{cases} \tag{12}$$

Under the Gaussian cluster assumption, we can calculate the theoretical cumulative distribution values $F_{thr}(i)$ from the relation

$$d^2_{[i]} = \chi^2_{p,F_{thr}(i)} \quad i = 1, \ldots, i_{\max}. \tag{13}$$

The Kolmogorov-Smirnov test compares the two cumulative distribution functions. The quantity

$$D_h = \max_i |F_{obs}(i) - F_{thr}(i)| \tag{14}$$

is used as a test criterion. The test is available in several program packages (e.g., p. 474 of [21]) and will not be described here. We denote the significance level returned by the program as I_h. This is our cluster quality measure, i.e., a scalar description of the compactness of the cluster delineated by the ellipsoid. Thus, compactness is measured by similarity to the multivariate normal cluster having the same covariance matrix. Note that the Kolmogorov-Smirnov test is more sensitive to deviations at the tails of the distributions than a χ^2-type test.

The value of h defining the mass of the minimum volume ellipsoid hQ_l was considered up to now as given. However, to find the best cluster at iteration l of the algorithm, several values of h are used in a decreasing order. The range of h is defined by

$$h_{\min} = \max \left[\frac{\max_i q_i}{Q_l}, \ \text{min\_size} \right]$$
$$h_{\max} = \min [0.5, \ \text{max\_size}] \tag{15}$$

where min_size and max_size are *a priori* given numbers limiting the size of any cluster in $\boldsymbol{X}$. If the value of h is smaller, the "resolution" we analyze the feature space $\boldsymbol{X}_l$ will be higher. Note that $h_{\max} = 0.5$ does not imply that clusters larger than half the data size are not detected. In such a case, the covariance matrix will yield unit weights (9) for most of the cluster. The value of h was decreased by a factor of 2 at each step:

$$h_j = \frac{1}{2} h_{j-1} \quad j = 1, \ldots, a \qquad h_0 = h_{\max}, h_a = h_{\min}. \tag{16}$$

By using a geometric series, we ensured that the number of steps per iteration was always less than 10.

When, for a value of h, the significance level of the delineated cluster satisfies

$$I_h \le 0.1 \tag{17}$$

and the analysis at higher resolutions (lower h) is no longer required. The shape of the cluster inside the ellipsoid is close to a multivariate normal one. The points $\mathbf{x}_i$ belonging to Ω_h are removed from $\boldsymbol{X}_l$, yielding the new set $\boldsymbol{X}_{l+1}$, and the $(l+1)th$ iteration starts again from $h = h_{\max}$. If (17) is not satisfied for any h defined by (16), the cluster corresponding to step size

$$j = \arg \min_h I_h \tag{18}$$

is the one removed from $\boldsymbol{X}_l$, that is, the most compact cluster found in the current iteration.

The iterations stop when either Q_{l+1} or the number of points remaining in $\boldsymbol{X}_{l+1}$ is less than a threshold or when the number of extracted clusters exceeds an *a priori* established upper bound. The number of clusters k detected in the feature space is equal to the index of the last iteration.

C. Implementational Issues

To reduce the amount of computation in an iteration of the algorithm (index l), for every value of h, we employed the

same set of samples J. Thus, at every value of h, the same Mahalanobis distances (3) are used, which can be computed and ordered at the beginning of the iteration.

The proposed clustering algorithm first detects the larger clusters present in the feature space. The parameter $h_{\min}$, which is the minimum step size of the analysis (highest resolution), is application dependent. When histograms are decomposed into clusters, $h_{\min} = 0$ since all the data should be accounted for. In other applications, such as analysis of the Hough space, only the most significant clusters should be discriminated, and $h_{\min} = 0.05$.

In our experiments, the feature spaces are contained around 1000 data points. The number of possible $(p+1)$-tuples of points is extremely large, and a complete scanning of the feature space is not possible. Random sampling is used to avoid excessive computation. An additional decrease in the number of samples is achieved if the random sampling is restricted to bounded subregions of the space, that is, after the first point is chosen at random, the remaining p points are chosen from a box centered on it. Since the clusters are compact regions in the feature space, under this restriction, we can expect similar performance of the algorithm using fewer samples. Indeed, for the examples discussed in this paper, the constrained random-sampling technique allowed us to decrease the number of samples per iteration to 25.

Since, usually, no information about the bounding distances is available, they must be determined adaptively from an analysis of the feature space. For every iteration, the global covariance matrix $\boldsymbol{C}_l$ is computed using (2) for the entire feature space $\boldsymbol{X}_l$. The squared Mahalanobis distance between the point chosen first $\mathbf{x}_0$ and another point $\mathbf{x}_i$ is

$$d^2_{0,j} = (\mathbf{x}_i - \mathbf{x}_0)^t \mathbf{C}_l^{-1} (\mathbf{x}_i - \mathbf{x}_0). \tag{19}$$

The point is accepted in the sample J if the distance (19) is smaller than a threshold. The χ^2-type threshold does not depend on the application; it depends only on the dimension of the feature space p. A larger p requires a smaller allowed region in the constrained random sampling to avoid the artifacts of the higher dimensional spaces. For $p < 3$, we choose the threshold to be 0.25, whereas for $p \geq 3$, we choose it to be 0.05. It must be emphasized that the performance of the algorithm is not sensitive to the precise threshold values. Whenever *a priori* information about cluster sizes is available, it can be used instead the generic threshold values.

In some applications, closely located clusters should be fused before other clusters are detected. Let $(\boldsymbol{T}_i, \boldsymbol{C}_i, N_i)$ and $(\boldsymbol{T}_l, \boldsymbol{C}_l, N_l)$ be the parameters of two clusters. The Mahalanobis distances of each cluster center from the other cluster can be computed as

$$\begin{aligned} d^2_{i,l} &= (\mathbf{T}_i - \mathbf{T}_l)^t \mathbf{C}_l^{-1} (\mathbf{T}_i - \mathbf{T}_l) \\ d^2_{l,i} &= (\mathbf{T}_l - \mathbf{T}_i)^t \mathbf{C}_i^{-1} (\mathbf{T}_l - \mathbf{T}_i). \end{aligned} \tag{20}$$

Note that the values of these distances differ since the two clusters have different covariance matrices. The clusters are fused if

$$\max\left[d^2_{i,l}, d^2_{l,i}\right] < \mathcal{X}^2_{p,0.975} \tag{21}$$

but other application-dependent fusion thresholds can also be employed. When, instead of the distance between a point and a cluster, a measure of the offset between two clusters is required, the Bhattacharyya distance [6] should be used.

In the next sections, we describe several applications of the GMVE clustering algorithm. All the examples are taken from real image data. The quality of the results shows the insensitivity of the algorithm to the Gaussian assumption employed in the cluster delineation process. The following computer vision problems are described: multithresholding of gray level images, analysis of the Hough space, and a facet model-based range image segmentation. It must be emphasized that no *a priori* information about the number of clusters is employed in the algorithm.

III. Histogram Based Multithresholding of Images

A histogram is a discrete feature space representation of a digital image. The value of every pixel is an integer (gray level image) or a p-tuple of integers (colored image). The set of possible (p-tuples of) integers is the feature space $\boldsymbol{X}$. The weight q_i is the number of pixels mapped into the i-th point in the feature space, i.e., the content of the i-th bin in the histogram. In multithresholding, the histogram is decomposed into a few nonoverlapping clusters characterized by their centers $\mathbf{T}_i$. Each pixel in the image is then mapped into the value of the center of the cluster to which the pixel belongs. The result is a global segmentation of the image. Note that the number of thresholds used (i.e., the number of clusters) is not known *a priori* since different histograms require different decompositions.

We are concerned here only with gray-level images. In Jolion *et al.* [15], the case of color images is also treated. For gray-level images, the feature space has $p = 1$. Several methods are available to decompose a gray-level histogram into clusters (see [26] for a recent review). Most of these methods attempt either to maximize a criterion describing the separation between two adjacent clusters or to minimize the error relative to an assumed (usually Gaussian) mixture population.

Our gray-level histogram decomposition results are compared with the thresholding method proposed by Otsu [20]. In this method, the thresholds are chosen to maximize the between-cluster variances. The number of clusters, i.e., the desired decomposition of the histogram, must be known *a priori*. The Otsu method tries to partition the histogram into the most uniform clusters (small within-cluster variances) yielding the largest distance (large between-cluster variances) between those clusters. We prefer the Otsu method since it does not employ the assumption where the histogram is a Gaussian mixture. It must be emphasized that the employed images are not especially well suited for multithresholding. Nevertheless, they supply nonsynthetic histograms and allow comparison of different approaches to range image segmentation.

In Fig. 1(a), a range image of a coffee cup is shown. Most of the surfaces in the image are of second degree, and thus, this example is a difficult one for any segmentation algorithm seeking to delineate uniform regions. In Section V, the subject

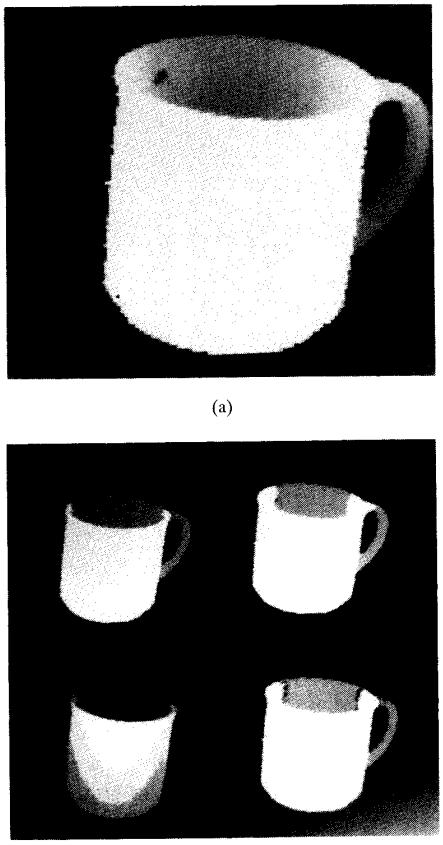


Fig. 1. Coffee-cup range image: (a) Original; (b) segmented. Top left: Otsu's method, three clusters. Top right: GMVE algorithm, three clusters. Bottom left: Otsu's method, six clusters. Bottom right: GMVE algorithm, six clusters.

of range image processing is discussed in more detail, and a more appropriate segmentation approach will be given. Our clustering algorithm detected six clusters having the following percentages of the image size: 60.87, 25.35, 8.2, 1.33, 0.67, and 0.48. The last three clusters are very small; therefore, we should also consider segmentation into three clusters only and allocate the remaining 2.5% of the pixels to the closest clusters. The images segmented by our algorithm into three and six clusters are shown in the top-right and bottom-right of Fig. 1(b), respectively. The regions are displayed with arbitrary gray levels to emphasize their shapes. Note that the small clusters lie on the borders of the large regions.

The results of applying Otsu's method to the same image are shown on the left in Fig. 1(b), where the three-cluster result is at the top, and the six-cluster result is at the bottom. None of the clusters is very small, and a completely different partition of the feature space is obtained when six clusters are assumed instead of three.

In Fig. 2(a), an outdoor scene is shown. Our algorithm extracted eight clusters, with the smallest one containing 4% of the image. The image segmented by our algorithm is shown

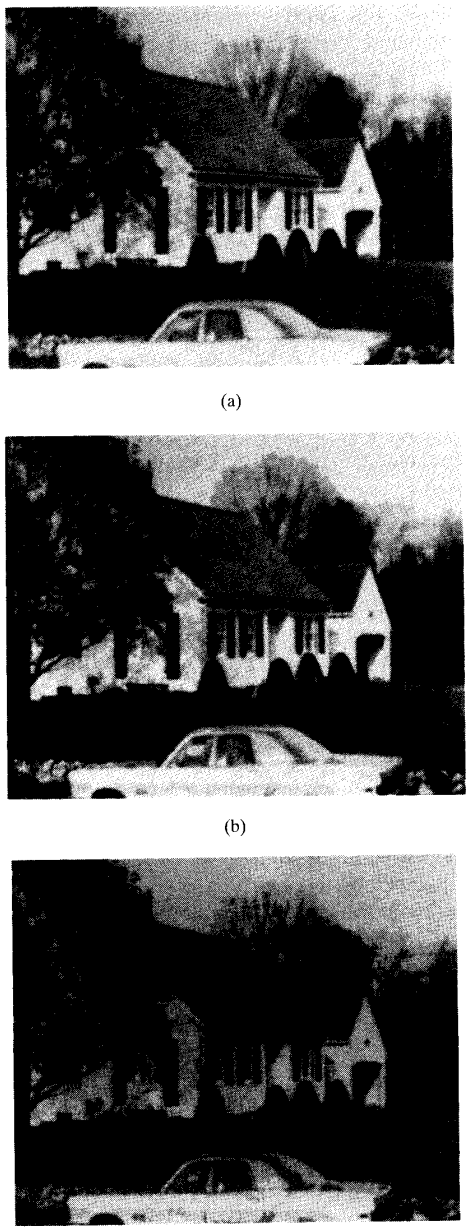


Fig. 2 Outdoor scene: (a) Original; (b) segmented with GMVE algorithm, eight clusters; (c) segmented with Otsu's method, eight clusters.

in Fig. 2(b) and by Otsu's method (eight clusters) in Fig. 2(c). Our algorithm is clearly superior in preserving the details of the image (see, for example, the roof of the house).

In the clustering algorithm, the feature space is analyzed by random sampling, and therefore, a small probability of error is always incorporated into a decision. We have investigated the sensitivity of the algorithm to random sampling by running it ten times on the histogram of the outdoor image. The his-

togram has 223 bins, i.e., the feature space contains 223 points. The constrained random sampling technique was employed with a threshold of 0.25 since $p < 3$. Twenty five samples per iteration were used.

The algorithm found between eight and 14 clusters. However, if we consider a cluster significant only if its size exceeds 1% of the data, the following results were obtained: eight clusters were detected six times, nine clusters once, and ten clusters three times. The coordinate of the centers and standard deviations of that coordinate over the ten trials, for the six clusters that were always detected, were (205.96, 1.25), (98.29, 5.32), (143.82, 4.76), (16.31, 1.1), (70.45, 7.22), and (31.6, 2.92). Almost the same clusters were extracted with a previous and different version of the random sampling program. Thus, in spite of the constrained random sampling, the analysis of the feature space is robust and yields unique results. The complete multithresholding algorithm took less than 30 s to run.

Fig. 3. Hough transform example: Top left—indoor scene; top right—corresponding edge image; bottom left—maxima-based Hough accumulator analysis; bottom right—GMVE clustering-based Hough space analysis.

IV. Analysis of the Hough Space

The Hough transform is a versatile method of detecting line segments in digital images (see [12] for a recent survey). In the original Hough transform, a point in the image generates a curve in the feature space. The feature space is quantized, and the value of each bin lying along the curve is incremented by 1. This discrete feature space is also known as the Hough accumulator array. The significant features (lines) then correspond to maxima in the accumulator.

Quantization of the Hough space yields serious problems ranging from loss of accuracy to detection of artifacts due to false alignments in the image. Not all the accumulator array can be accessed by the transform. Risse [22] has shown that for the parametrizations that are usually used, at most, 52% of the storage space in the array will be used. Risse also defined the concept of hi-fi quantization as the coarsest uniform rectangular Hough accumulator array in which all realizable lines are discriminable. To achieve hi-fi quantization, an $N \times N$ image requires an accumulator array size of $O(N^6)$. In practice, however, the array size is usually chosen equal to the size of the image N^2, and thus, similar lines may remain undiscriminated. The quantization effects can be reduced but not completely eliminated through dynamic quantization of the parameter space [23]. Kiryati and Bruckstein [16] treated the tradeoff between the resolution achievable in the image and in the accumulator as a nonbandlimited signal sampling problem.

Our clustering algorithm identifies the regions of highest density in the feature space by locating minimum volume ellipsoids that contain $h\%$ of the mass in the space. The algorithm does not require generation of a quantized version of the space as an intermediate step; the Hough space is thus analyzed in the continuous domain. The accuracy of the coordinates of a feature point is limited only by the finite word length of the computer. There is no redundancy in the memory since the feature vectors in the space are stored as an ordered list.

To isolate the pixels of interest in the input image, edge detection was first performed. We have used a 5×5 Canny operator ($\sigma = 0.5$) that returns the gradient magnitude and orientation at every pixel. Pixels having magnitudes less than 0.2 of the maximum magnitude in the edge image were discarded. Nonmaximum suppression was then performed orthogonal to the gradient orientation.

For every retained edge pixel, the parameters of the most probable lines containing that pixel were then determined. A line is parametrized by the slope angle of the normal θ and the distance to the origin ρ. In every 5×5 neighborhood centered on an edge pixel, the parameters were computed for the lines containing the center and two edge pixels located on the border of the neighborhood. In most cases, only one such line is detected. The two-dimensional Hough space is the ordered (but not quantized) set of (ρ, θ) pairs.

In Fig. 3 (top left), a 256×256 indoor scene is shown. The edge image (top right) has been processed using nonoverlapping windows of 32×32 pixels. The constrained random sampling procedure again used the threshold 0.25. Note that special care must be taken to avoid the artifacts created by the cyclic θ coordinate. Fig. 3 (bottom right) shows the output of the algorithm superposed on the input image. For comparison, an ordinary Hough accumulator technique was also applied. The accumulators were of the same size as the windows. The line segments detected as local maxima (with minimum bin amplitude 5) are shown in Fig. 3 (bottom left). Note the spurious lines at the lower edge of the computer terminal and around the desk. These artifacts are due to quantization of two nearly parallel lines.

Several issues should be emphasized in connection with the application of our algorithm to the Hough space. All the available data is made use of, and no artifacts can be introduced by the quantization of the space. There is no need for thresholding for detection of local maxima; the clusters are extracted based on the local density of the continuous feature space. The analysis of the space is by random sampling, i.e., a reduced subset of points suffices to arrive at the correct decisions. Recently, Bergen and Shvaytser [1] proved theoretically that the Hough accumulator can indeed be analyzed by probabilistic algorithms. Kiryati *et al.* [17] have shown that points can be randomly selected in the image domain without significant decrease in the performance of the

Hough transform. Xu *et al.* [27] also used random sampling to build the feature space in a method similar to the well-known RANSAC procedure of Fischler and Bolles [5]. Thus, a Hough transform technique incorporating random sampling to reduce the computational burden is a realistic approach.

V. Range Image Segmentation

In range images, the distance from the observer (depth information) is mapped into pixel values. Range images are important in computer vision, especially in industrial applications. Since range images represent depth, planar and higher order surfaces that cannot be approximated locally by constant patches are more frequent in range images than in images that represent reflectance information. Transitions between surfaces are often smooth, i.e., step edges are less common. Our range images are defined on a regular lattice, that is, if the data from the sensor were obtained in a nonCartesian coordinate system and/or are sparse, preprocessing is assumed to have already taken place. The interpolation of sparse data can make use of robust techniques described elsewhere [19].

Haralick and Watson [8] introduced the facet model in which images are described by a piecewise polynomial structure. Besl and Jain [2] employed, for segmentation of range images, eight fundamental surface types defined in differential geometry by combinations of the signs of the mean and Gaussian curvatures. Each delineated surface in the image was then reduced to a seed region by applying a morphological (erosion) operation. The seed regions were grown by iterative polynomial surface fitting with increasing model order. Yokoya and Levine [28] proposed a similar hybrid method in which surface identification is integrated with edge detection. Note that use of the curvatures requires local computations of partial derivatives and thus limits the tolerance of the method to noise. Intermediate and high-level processing of range images are described in [9] and [14].

A. Robust Facet Estimation

In noisy images, surfaces more complex than biquadratic are not very useful. If the degree of the surface is higher, more noise will be incorporated into the estimated parameters and the estimation procedure will be less sensitive to discontinuities. We use the biquadratic facet model

$$z_{u,v} = \beta_0 + \beta_1 u + \beta_2 v + \beta_3 u^2 + \beta_4 uv + \beta_5 v^2. \tag{22}$$

At location (u, v) in the lattice, the value of the pixel should obey relation (22). The parameters β_k are estimated from the data in a small $(2L+1) \times (2L+1)$ neighborhood centered on (u, v). The estimates are denoted $\hat{\beta}_k$ and the resulting pixel value $\hat{z}_{u,v}$. For the planar facet model $\beta_3 = \beta_4 = \beta_5 = 0$.

The parameters $\hat{\beta}_i$ can be computed by least squares estimation in the neighborhood. Least squares, however, is very sensitive to the presence of outliers, i.e., to sample values with large deviations from the uncorrupted surface. Note that if the window falls on a discontinuity, the data arises from more than one surface, several outliers result, and the estimates are erroneous. In recent years, new methods were proposed in statistics for reducing the effect of outliers on the estimates. They are known as robust estimators; in Meer *et al.* [19], we give a detailed analysis of these methods in the context of computer vision.

The facet parameters $\hat{\beta}_k$ are obtained by minimizing a penalty function of the residuals $r_{u,v}$, i.e., of the difference between the data and the value of the estimated fit. An important class is that of the continuous, symmetric, positive valued penalty functions $\rho(r_{u,v})$ with a unique minimum at $r_{u,v} = 0$. For this class, the estimation is equivalent to the minimization problem

$$\min_{\hat{\beta}} \sum_{u=-L}^{L} \sum_{v=-L}^{L} \rho\left(r_{u,v}\right). \tag{23}$$

The particular case $\rho(r_{u,v}) = r_{u,v}^2$ yields least squares estimation. The robust *M-estimators* use more complex $\rho(r_{u,v})$ functions to limit the influence of outliers on the estimates. The books of Huber [11] and Hampel *et al.* [7] offer a complete treatment of the subject. The M-estimators were applied to several computer vision problems, but for us, only local facet parameter estimation is of significance. Besl and Jain [2] proposed an M-estimator-based surface fitting algorithm, where the model order was also a variable. The optimal model structure is established by minimizing a fit quality measure over the sequential increase of the model order. For our purposes, a simple M-estimation procedure suffices.

The minimization problem (23) is solved for M-estimators as iteratively reweighted least squares with the definition of the weights depending on $\rho(r_{u,v})$. We have used the weight function (also known as Tukey's biweight)

$$w_{u,v} = \begin{cases} \left[1 - \left(\dfrac{r_{u,v}}{c\hat{\sigma}_{u,v}}\right)^2\right]^2 & |r_{u,v}| \le c\hat{\sigma}_{u,v} \\ 0 & \text{otherwise} \end{cases} \tag{24}$$

where $\hat{\sigma}_{u,v}$ is the locally estimated standard deviation of the fit, and c is a tuning constant taken equal to 4.685 to assure superior performance for Gaussian noise [10]. Note that $w_{u,v} \le 1$, and samples yielding large residuals receive small weights. The local standard deviation is robustly estimated by the expression

$$\hat{\sigma}_{u,v} = 1.4826 \operatorname{med}|r_{\mathrm{u,v}} - \operatorname{med} r_{\mathrm{u,v}}|. \tag{25}$$

where med denotes the median taken over the entire $(2L+1) \times (2L+1)$ neighborhood, and the factor 1.4826 compensates for the bias of the median estimator in Gaussian noise.

The estimation procedure starts with an unweighted least squares (all $w_{u,v} = 1$) to obtain the initial parameter estimates $\hat{\beta}_k^{(0)}$. The residuals, the standard deviation (25), and the weights (24) can now be computed. In the next iteration, the new set of parameters $\hat{\beta}_k^{(1)}$ are obtained by weighted least squares. The standard deviation estimate $\hat{\sigma}_{u,v}$ should not be updated during the iterations [10]. At consecutive iterations, samples yielding large residuals (outliers) have their weights reduced, and the estimates are computed mostly from values distributed around the true surface. Often, for the latter samples, a Gaussian distribution is assumed.

The following quality measure was used by us as a stopping criterion for the iterations:

$$\varepsilon^{(l)} = \sqrt{\sum_{u=-L}^{L}\sum_{v=-L}^{L} w_{u,v} r_{u,v}^2 \Big/ \sum_{u=-L}^{L}\sum_{v=-L}^{L} w_{u,v}} \quad (26)$$

where the index l denotes the current iteration. Convergence is achieved if

$$|\varepsilon^{(l)} - \varepsilon^{(l-1)}| < 0.001 \quad l = 1, 2 \ldots . \quad (27)$$

The number of allowed iterations was bounded to ten. The estimates obtained in the final iteration are the facet parameters $\hat{\beta}_k$ recovered in the window. They are allocated to the window center (u, v). The parameters are not estimated at every pixel but at locations situated L pixels apart along either coordinate axes. Thus, the input range image is tessellated with $(2L+1) \times (2L+1)$ windows with almost a 50% overlap. Note that the M-estimators are window based, and this step of the segmentation algorithm can be performed in parallel.

The reliability of the initial guess used in the iterative process is extremely important, and optimum convergence is not assured when the biweights (24) are used. To reject the erroneous facet parameters, the global standard deviation estimate $\hat{\sigma}$ is computed as the mode (maximum value) of the $\hat{\sigma}_{u,v}$ distribution. The distribution is obtained by ordering the window estimates $\hat{\sigma}_{u,v}$. The mode is a robust, degree-0 least median of squares estimator and as long as the majority of the windows return an unbiased $\hat{\sigma}_{u,v}$ estimate, the correct value of $\hat{\sigma}$ is recovered. For a mode computation algorithm, see Meer *et al.* [19]. Note that we assume a homogeneously corrupted image, i.e., the signal-dependent noise component of the range acquisition process is considered negligible.

A p-tuple of facet parameters is mapped into the feature space only if the last iteration error satisfies

$$\varepsilon^{(l)} < 2.5\hat{\sigma}. \quad (28)$$

In spite of the sensitivity of M estimators to the initial estimates, there is no reason to use robust (least median of squares) estimators for the computation of initial facet parameters. The least median of squares technique tolerates discontinuities, but it requires more computations [19]. Since most of the M-estimate feature vectors (parameter p-tuples) that satisfy (28) are unbiased, the extracted cluster centers are satisfactory for any purpose.

B. Mapping Clusters into the Image

At the end of the clustering algorithm, not all the feature points were allocated to clusters, and the number of unlabeled feature points can exceed the minimum accepted cluster size. This is often the case when the image contains surfaces of degrees higher than the degree of the facet model being used. Clustering in the feature space does not take into account the spatial information available about the feature points, i.e., the coordinates (u, v) of the window center from which the point $\boldsymbol{x}_J$ was estimated. Consistency in the spatial domain thus can be used to post-process the segmentation.

The overlapping window tessellation used to estimate the feature vectors makes a straightforward feature space-image mapping unwieldy. Instead, first 'seed regions' are delineated containing the pixels that can unequivocally be mapped. These pixel values $f(u, v)$ must satisfy

$$|f(u, v) - f(u, v, \mathbf{T}_l)| < 2.5\hat{\sigma} \quad (29)$$

for only one label l, that is, the absolute valued residual must be less than 2.5 times the global standard deviation estimate of the noise. Connected component analysis then retains the larger seed regions. Note that by their definition, the seed regions are always good local fits and do not contain outliers. The final surface parameters are obtained by applying least squares.

The remaining pixels are incorporated through region growth. At every expansion step, a one-pixel-wide ring along the perimeter of each region is examined. If the difference between the estimated fit and the pixel value is less than $2.5\hat{\sigma}$, the pixel is incorporated into the expanding region. To avoid biasing the fit, the parameters of the initial surface are not updated during expansion. The expansion process stops at the collision of two regions or when no more pixels can be conquered. If two or more regions simultaneously reach a pixel, the one yielding the smallest residual wins. Note that at roof edges (discontinuities in the derivatives only), the threshold on the residual size introduces an ambiguous region.

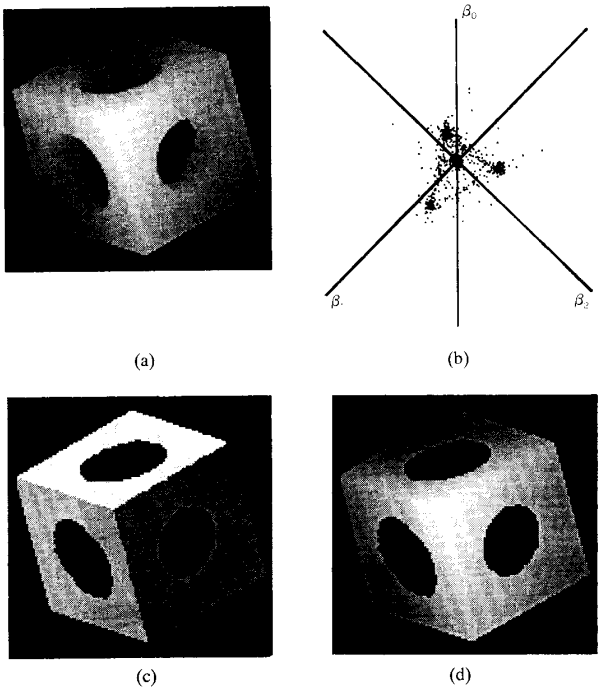
Most of the pixels that did not yield feature points belonging to labeled clusters remain unlabeled after region growth. Some pixels are not labeled because of noise and appear isolated in the image. They are removed by giving them the majority label in their centered 3×3 neighborhood.

C. Experimental Results

The algorithm described in the previous sections was applied to several 128×128 range images. The synthetic *cube* image was used to study the behavior of the method under increasing noise levels. The planar facet model was employed, i.e., the dimension of the feature space is three. The window size was 7×7, $L = 3$.

In Fig. 4(a), the *cube* is corrupted with zero mean Gaussian noise $\sigma = 2.34$ (measured from the image). The feature space contained 1342 points and had the configuration shown in Fig. 4(b). The four clusters extracted by the algorithm are well defined. The clusters correctly delineate the faces of the cube and the background. The labeled image, which is the output of the segmentation algorithm, is shown in Fig. 4(c). When the labeling gray level is lighter, the cluster was extracted later. Pixels not belonging to any cluster are black. Note that the small dark regions inside the holes are allocated to the same cluster as the background. Larger regions within the holes were not extracted since the planar facet model could not cope with nonplanar surfaces. The reconstructed input with the value of a labeled pixel computed from the surface to which it belongs is shown in Fig. 4(d).

In Fig. 5(a), the *cube* is corrupted with zero-mean Gaussian noise $\sigma = 13.09$. The 1572-point feature space is shown in Fig. 5(b). The clusters are less well defined, and the



(a) (b) (c) (d)

Fig. 4. Noisy *cube* image $\sigma = 2.34$: (a) Input; (b) feature space; (c) labeled image. Black pixels do not have a label. (d) Reconstructed image.

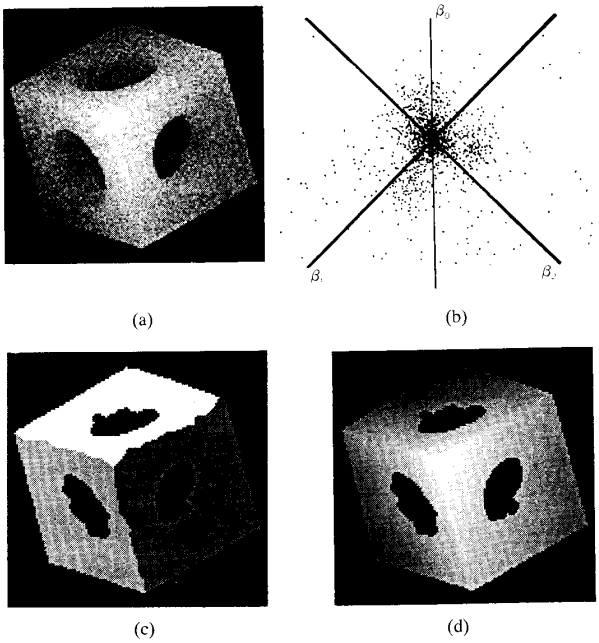
TABLE I
ESTIMATED CLUSTER CENTERS FOR THE *CUBE* IMAGES ($\beta_0, \beta_1, \beta_2$)

	$\sigma = 0$	$\sigma = 2.34$
background	(0,0,0)	(8.19, 0, 0)
top face	(0,4,1)	(-19.26, 4.01, 0.997)
left face	(200, -1.6, 3)	(196.86, -1.718, 2.968)
right face	(400, -0.6, -1.7)	(369.8, -0.583, -1.658)

	$\sigma = 13.09$	$\sigma = 6$, 20% impulse
background	(8.52, 0.004, 0.008)	(7.28, 0, 0)
top face	(-20.13, 3.968, 0.981)	(-11.17, 3.75, 0.91)
left face	(196.37, -1.701, 2.896)	(192.25, -1.68, 2.9)
right face	(368.98, -0.603, -1.652)	(368.52, -0.584, -1.655)

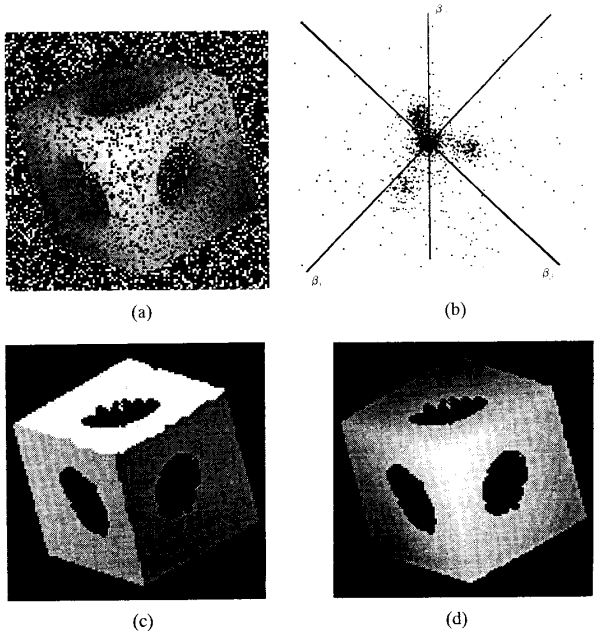
importance of the mapping from the feature space to the image increases. The labeled image had four extracted clusters (Fig. 5(c)). Note the more ragged edges between the faces of the cube and the integration of hole boundaries into the faces. The reconstructed input is shown in Fig. 5(d). The background is extracted as a slightly slanted plane since the presence of significant half-wave Gaussian noise introduces bias.

In Fig. 6(a), the *cube* is corrupted with zero-mean Gaussian noise $\sigma = 6$. After that, about 20% of pixels chosen at random have their values subtracted from 255 to simulate a second, impulse-type noise process. Note that the impulse noise is not zero mean. The feature space had 1483 points (Fig. 6(b)). The robust M-estimator succeeds in filtering out some of the



(a) (b) (c) (d)

Fig. 5 Noisy *cube* image $\sigma = 13.09$: (a) Input; (b) feature space; (c) labeled image. Black pixels do not have a label. (d) Reconstructed image.



(a) (b) (c) (d)

Fig. 6 Noisy *cube* image $\sigma = 6$ and 20% of the pixel shave their values inverted modulo 255: (a) Input; (b) feature space; (c) labeled image. Black pixels do not have a label. (d) Reconstructed image.

impulse noise (although least median squares would have been more effective), and the feature space appears less distorted than the one in Fig. 6(c) corresponding to a high Gaussian noise level. The labeled image with four clusters is shown in Fig. 6(c) and the reconstructed input in Fig. 6(d).

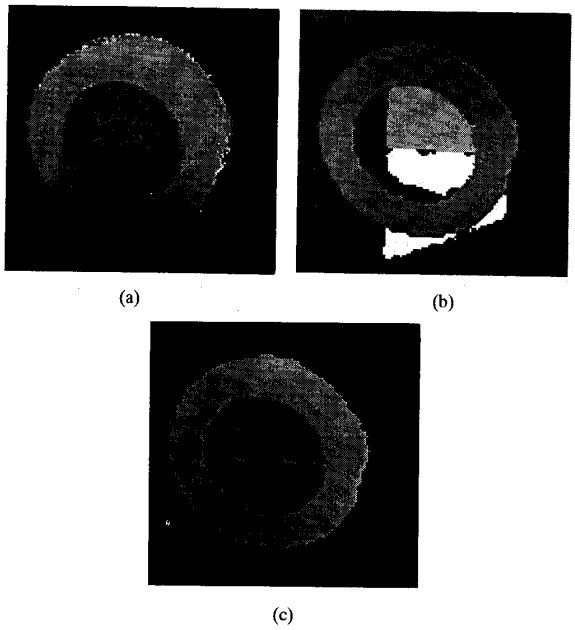


Fig. 7 The *ring* image: (a) Input; (b) labeled image. Black pixels do not have a label. (c) Reconstructed image.

The experimental results are summarized in Table I. The slopes of the facets are correctly estimated in all cases, as a comparison with the ideal values ($\sigma = 0$) shows. The variations of the estimated intercept $\hat{\beta}_0$ are of lesser importance for large regions like the ones delineated in the *cube* images.

The *ring* image is a real range image taken with an ERIM sensor (Fig. 7(a)). From the 889-point feature space obtained with the planar facet model, five clusters were extracted (Fig. 7(b)). Nonplanar surfaces, and the region around the roof edge at the bottom, were not labeled. The reconstructed image is shown in Fig. 7(c).

The *coffee cup* is also a real range image (Fig. 1(a)). We have used the biquadratic facet model; thus, the dimension of the feature space is six. For reliable facet parameter estimation of the low-curvature biquadratic surface (body of the cup), the window size had to be increased to 15×15, $L = 7$. The larger window reduces the size of the feature space; we had only 289 points. The three extracted clusters are shown in the labeled image (Fig. 8(a)). The handle, which is an elongated feature with a higher degree surface, cannot be labeled since the facet parameter estimation is large-window based. Note the discrimination of the hole in the cup. The several small unlabeled regions in the background near the cup were detected as outliers in the image. The reconstructed image is shown in Fig. 8(b).

The segmentation algorithm does not require a large amount of computation. On a HP9000 computer, the planar facets were extracted in less than 30 s from a 128×128 image with overlapping 7×7 windows. A feature space of about 1300 points was decomposed into clusters in less than 60 s and mapped back into the image in less than 30 s.

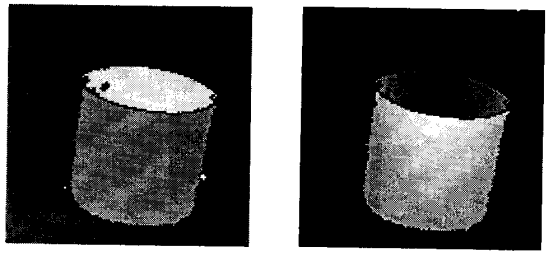


Fig. 8. *Coffee cup* image: (a) Labeled image. Black pixels do not have a label. (b) Reconstructed image.

VI. Discussion

We have presented a novel clustering method in which no *a priori* knowledge is required about the number of clusters to be extracted. The robustness of our clustering algorithm is due to its tolerance at every stage of the procedure of half the data being severely corrupted, i.e., not belonging to the extracted cluster. The use of the equivalent multivariate normal cluster to dichotomize the data (9) is a weak assumption. Rousseeuw and van Zomeren [25] suggested that a unimodal elliptical distribution assumption suffices for a cluster to be correctly detected. Note also that all our examples made use of real data in which normality was not assured. The algorithm also tolerated the presence of impulse noise (Fig. 6).

The algorithm iteratively identifies the regions of maximum density in the feature space. These regions correspond to the modes of the continuous probability distribution. Mode-seeking clustering methods are not new in the literature (see [13]). In the proposed methods, the feature space is quantized and the structure of the underlying probability density is estimated by either locating the k nearest neighbors or by a Parzen window approach. We do not use these techniques. By extracting the clusters (identifying the modes) in the order of decreasing size, we are also less prone to errors.

Small changes in the shapes of the large clusters extracted at the beginning of the algorithm result in significantly different numbers of small clusters extracted at later iterations. The effect of small groups of points on the final decomposition of the feature space can be reduced by assuming overlap between adjacent clusters. Recall that in our algorithm, all the points within a delineated cluster are removed from the feature space. Far from the cluster center, however, we can also assume that some of the points belong to the tail of an adjacent cluster. Thus, removal of only a subset of points far from the center (based on the assumed probability distribution) can improve the definition of large clusters and therefore eliminate discrimination of small groups of points.

The employment of random sampling in the analysis of the feature space should introduce, at least theoretically, an uncertainty in the decomposition of the space. For the relative large data size of the feature space, the sensitivity of clustering to random sampling should be reduced [25]. If desired, repeated application of the clustering algorithm on the same data can provide a stable description of the most well-defined

clusters against a background of small unstable ones. This redundancy-based procedure belongs to the class consensus vision techniques, described in Meer *et al.* [18].

The range image segmentation examples show a good tolerance of the noise corrupting the images. The insensitivity of global methods to noise is coupled, however, with an insensitivity to relative small regions in the input image. The proposed segmentation algorithm fails when applied to outdoor scenes where the homogeneous regions have reduced sizes. Cluster detection becomes unreliable. In industrial environments, where only a few well-defined objects are present in the visual field, global methods should be superior.

Mapping of the extracted clusters into the image does not necessarily preserve spatial coherence of uniform (under the assumed feature measure) regions. Often, post processing is required. Such a technique was discussed in Section V-B. There are several ways of directly incorporating spatial information into a feature space-based analysis. The spatial coordinates can be added to the feature vector, increasing the dimension of the space p by two or three. Then, the extracted clusters must correspond to contiguous regions in the image. In a different approach, clustering is implemented hierarchically, i.e., on an image pyramid. The data is analyzed in parallel within increasing sized windows (higher pyramid levels). Clusters extracted in adjacent windows are fused. We are currently investigating this method.

Combining spatial (image-based) and feature-space information can help recover features that are not identified as clusters. For example, the handle of the *coffee cup* is not recovered by a biquadratic facet model. A connected component analysis of the unlabeled pixels, however, reveals the feature. Similarly, when separate regions in the image belong to the same surface (i.e., are detected as one cluster), they are separately delineated because of the spatial constraints. We believe that a combined spatial/feature space method using robust estimation techniques can overcome the limitations of global methods while preserving most of their excellent noise-resistant properties.

ACKNOWLEDGMENT

We are grateful to P. Besl for providing us with the range images.

REFERENCES

[1] J. R. Bergen and H. Shvaytser, "A probabilistic algorithm for computing Hough transforms," Tech. Rep. SRI David Sarnoff Res. Cen., Princeton, NJ; submitted to *J. Algorithms*, 1989.

[2] P. J. Besl and R. C. Jain, "Segmentation through variable-order surface fitting," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 10, pp. 167–192, 1988.

[3] P. J. Besl, J. B. Birch, and L. T. Watson, "Robust window operators," in *Proc. Second Int. Conf. Comput. Vision* (Tampa, FL), Dec. 5–8, 1988, pp. 591–600; see also *Machine Vision Applications*, vol. 2, pp. 179–214, 1989.

[4] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.

[5] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Comm. ACM*, vol. 24, pp. 381–395, 1981.

[6] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. New York: Academic, 1972.

[7] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel, *Robust Statistics: An Approach Based on Influence Functions*. New York: Wiley, 1986.

[8] R. M. Haralick and L. Watson, "A facet model for image data," *Comput. Graphics Image Proc.*, vol. 15, pp. 113–129, 1981.

[9] R. Hoffman and A. K. Jain, "Segmentation and classification of range images," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-9, pp. 608–619, 1987.

[10] P. W. Holland and R. E. Welsch, "Robust regression using iteratively reweighted least squares," *Commun. Stat*, vol. A6, pp. 813–828, 1977.

[11] P. J. Huber, *Robust Statistics*. New York: Wiley, 1981.

[12] J. Illingworth and J. Kittler, "A survey of the Hough transform," *Comput. Vision Graphics Image Proc.*, vol. 44, pp. 87–116, 1988.

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

[14] A. K. Jain and R. Hoffman, "Evidence-based recognition of 3-D objects," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 10, pp. 783–803, 1988.

[15] J. M. Jolion, P. Meer, and A. Rosenfeld, "Generalized minimum volume ellipsoid clustering with applications in computer vision," Comput. Vision Lab., Univ. Maryland, College Park, CAR-TR-485, 1990.

[16] N. Kiryati and A. Bruckstein, "Antialiasing the Hough transform," *Proc. 6th Scandinavian Conf. Image Anal.* (Oulu, Finland), June 19–22, 1989, pp. 621–628.

[17] N. Kiryati, Y. Eldar, and A. Bruckstein, "A probabilistic Hough transform," *Patt. Recog.*, vol. 24, pp. 303–316, 1991.

[18] P. Meer, D. Mintz, A. Montanvert, and A. Rosenfeld, "Consensus vision," in *Proc. AAAI-90 Workshop Qualitative Vision* (Boston, MA), July 1990, pp. 111–115.

[19] P. Meer, D. Mintz, and A. Rosenfeld, "Least median of squares based robust analysis of image structure," in *Proc. DARPA Image Understanding Workshop* (Pittsburgh, PA), Sept. 1990, pp. 231–254.

[20] N. Otsu, "A threshold selection method from gray-level histogram," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-9, pp. 62–66, 1979.

[21] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes*. Cambridge, England: Cambridge University Press, 1988.

[22] T. Risse, "Hough transform for line recognition: Complexity of evidence accumulation and cluster detection," *Comput. Vision Graphics Image Proc.*, vol. 46, pp. 327–345, 1989.

[23] J. O'Rourke and K. R. Sloan Jr., "Dynamic quantization: Two adaptive data structures for multidimensional spaces," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-6, pp. 266–280, 1984.

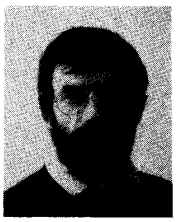
[24] P. J. Rousseeuw and A. M. Leroy, *Robust Regression & Outlier Detection*. New York: Wiley, 1987.

[25] P. J. Rousseeuw and B. C. van Zomeren, "Unmasking multivariate outliers and leverage points," with comments by R. D. Cook and D. H. Hawkins; D. Ruppert and D. G. Simpson; P. J. Kempthorne and M. B. Mendel; and rejoinder. *J. Amer. Stat. Assoc.*, vol. 85, pp. 633–651, 1990.

[26] P. K. Sahoo, S. Soltani, and A. K. C. Wong, "A survey of thresholding techniques," *Comput. Vision Graphics Image Proc.*, vol. 41, pp. 233–260, 1988.

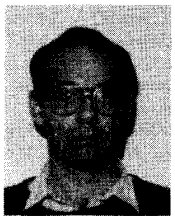
[27] L. Xu, E. Oja, and P. Kultanen, "A new curve detection method: Randomized Hough transform (RHT)," *Patt. Recognition Lett.*, vol. 11, pp. 331–338, 1990.

[28] N. Yokoya and M. D. Levine, "Range image segmentation based on differential geometry: A hybrid approach," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 11, pp. 643–649, 1989.



Jean-Michel Jolion was born in Moulins, France, on July 20, 1961. He received the Ph.D. degree in computer science from the Institut National des Sciences Appliquées, Lyon, France, in 1987.

From 1987 to 1988, he was a Research Associate in the Computer Vision Laboratory, Center for Automation Research, University of Maryland. He is currently with the Computer Science Department, Université Lyon I, as an Associate Professor. His research interests are in computer perception, pattern recognition, robust statistics, and hierarchical image analysis.



Peter Meer (S'84–M'87) was born in Oradea, Romania, on February 14, 1949. He received the Dipl. Engn. degree from the Bucharest Polytechnic Institute, Bucharest, Romania, in 1971 and the D.Sc. degree from the Technion, Isreal Institute of Technology, Haifa, Israel, in 1986, both in electrical engineering.

From 1971 to 1979, he was with the Computer Research Institute, Cluj, Romania, working on research and development of digital hardware. Between 1986 and 1990, he was Assistant Research Scientist at the Center for Automation Research, University of Maryland, College Park. In 1991, he joined the Department of Electrical and Computer Engineering, Rutgers University, Piscataway, NJ, as an Assistant Professor. His research interests include application of estimation techniques and probablistic algorithms to machine vision problems.

Samira Bataouche was born in Algiers, Algeria, on September 21, 1964. She is currently working towards the Ph.D. degree in computer science at Université Lyon I, Lyon, France. Her thesis work is on hierarchical information retrieval processes.

From 1983 to 1988, she was a software engineering student at the Institut National de Formation en Informatique in Algiers. Her current research interests are in image analysis, pattern recognition, and algorithms for parallel computer vision.