Image Segmentation by Clustering

GUY B. COLEMAN, MEMBER, IEEE, AND HARRY C. ANDREWS, SENIOR MEMBER, IEEE

Abstract—This paper describes a procedure for segmenting imagery using digital methods and is based on a mathematical-pattern recognition model. The technique does not require training prototypes but operates in an "unsupervised" mode. The features most useful for the given image to be segmented are retained by the algorithm without human interaction, by rejecting those attributes which do not contribute to homogeneous clustering in N-dimensional vector space.

The basic procedure is a K-means clustering algorithm which converges to a local minimum in the average squared intercluster distance for a specified number of clusters. The algorithm iterates on the number of clusters, evaluating the clustering based on a parameter of clustering quality. The parameter proposed is a product of between and within cluster scatter measures, which achieves a maximum value that is postulated to represent an intrinsic number of clusters in the data. At this value, feature rejection is implemented via a Bhattacharyya measure to make the image segments more homogeneous (thereby removing "noisy" features); and reclustering is performed. The resulting parameter of clustering fidelity is maximized with segmented imagery resulting in psychovisually pleasing and culturally logical image segments.

INTRODUCTION

THIS PAPER describes a procedure for automatically segmenting images into regions using digital techniques. The background of this procedure lies in image-understanding systems, an expansion of image-processing systems that attempt to draw meaningful inferences from visual data. An important step to forming inferences about the visual data is to segment the image into regions of homogeneity to aid further analysis.

The goal of the research described herein is to develop a reasonably fast algorithm for segmenting images into regions that correspond in a large degree to areas that would be perceived as essentially homogeneous by a human interpreter. The procedure does not use context-related information such as shape and relative position. All forms of imagery may be segmented utilizing the same algorithm, with a somewhat expanded feature set for color and multispectral data.

The second section of the paper provides an overview of image-understanding systems in general and approaches to image segmentation in the past. The approach taken here is a procedure based entirely on clustering. However, while clustering has been used to refine and identify image segmentations in the past, it has previously been believed that a pure clustering approach was too cumbersome computationally to implement. The approach taken here avoids many of those pitfalls.

The third section consists of a theoretical development of the background of clustering. Additional tools of statistical data analysis are developed to determine the "intrinsic" number of clusters in the data, and a novel arrangement of these tools is proposed to provide a reliable and unambiguous stopping criterion for the algorithm.

The fourth section is a detailed description of the approach taken. Block diagrams and flowcharts of the algorithm are provided along with the rationale for the various procedures used. A complete description of the feature sets used to segment images are provided and the various rejection criteria for these features are justified, based on results obtained. To obtain an elementary preclassification of region character, a novel nonlinear filter based on the mode of the local area histogram is proposed and used to segment images.

The results obtained on several kinds of images are described in the fifth section. In some cases, images were segmented with more than one feature set in an attempt to improve performance. Monochrome, color, and movie frames were all segmented with varying degrees of success.

IMAGE-UNDERSTANDING SYSTEMS

An image-understanding system is a system that uses visual data to generate descriptions that are useful for desired applications. The descriptions generated can be at very different levels and degrees of detail. If an image is represented in digital form, then the image is represented by an array of numbers characterizing the brightness at each point on a (usually) rectangular grid. These brightness elements are called picture elements (pixels). In the limiting case, this array of numbers "describes" the image.

Image descriptions of this form are usually the starting point for image-understanding systems. The system generates a series of descriptions that are progressively more general until a descriptive level is reached that satisfies the system requirements. It has been observed that the successive levels of abstraction require that the higher levels of the system interact with the lower levels, based on the current descriptions [1]. This processing approach is called "heterarchical." The image-understanding system is therefore conceptualized as having a hierarchy of processing levels, as shown in Fig. 1.

The primitive description level extracts local features that are not related to context. The primary or "first-order" features of a pixel in a monochrome image are the brightness (with due consideration of the sensor-spectral response) and spatial location of the pixel. All other features are of higher order, that is they describe how the pixel is related to surrounding pixels in the image. These features describe such primitive local attributes of the picture as brightness, texture and color. A proper primitive description level of the image-understanding system would transform the features into a

Manuscript received October 10, 1977; revised October 11, 1978. This work was supported by the Advanced Research Projects Agency of the Department of Defense under Contract Number F-33615-76-C1203 ARPA Order No. 3119 and monitored by the Wright-Patterson Air Force Base, Dayton, OH.

G. B. Coleman was with the Image Processing Institute, Department of Electrical Engineering, University of Southern California, Los Angeles, CA. He is now with Hughes Aircraft Company, Canoga Park, CA.

H. C. Andrews was with the Image Processing Institute, Department of Electrical Engineering, University of Southern California, Los Angeles, CA. He is now with the Comtal Corporation, Pasadena, CA 91107.

0018-9219/79/0500-0773$00.75 © 1979 IEEE

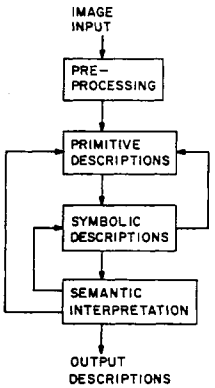


Fig. 1. Image understanding system.

coordinate system where numerical distance would be related to human perceptual difference [2].

The symbolic description level of the system takes the primitive descriptions and forms more global and symbolic descriptions of the image. Segmentation of the image takes place at this level. The initial segmentation is based purely on perceptual difference. After analysis by the semantic interpretation level of the system, the symbolic level may be directed to merge or to further divide regions in the image. Thus the decisions about dividing the scene into similar or homogeneous regions are made at this level of the image-understanding system. Feedback from the semantic interpretation level is necessary to ensure that the symbolic descriptions are consistent with the goals of the image-understanding system.

The semantic interpretation level of the system generates hypotheses for the contents of the image based on the symbolic descriptions. The semantic interpretation level then further directs the lower processing levels until the symbolic descriptions confirm one of the hypotheses.

A number of somewhat different models have been proposed other than the model of Fig. 1. It has been suggested, for example, that a goal directed or "top-down" approach be used to look for a specific object in, or test a specific hypothesis about, an image. Examples of this are discussed in [3], [4]. The problem with top-down approaches is that the specific circumstances under which the system operates must be well defined in advance. Any substantial departure from these circumstances will cause the system to fail to perform adequately.

Other models represent a middle ground between the completely top-down and the completely bottom-up approaches. These models differ mainly in that they use knowledge of the scene at the earliest possible stage of the image-understanding system to refine the scene description as it is generated [5], [6].

In all of these image-understanding system approaches, gross overall image segmentation is necessary to direct the attention of the higher system levels, form preliminary hypotheses about the image (such as whether it is an aerial photograph or indoor scene, etc.), and identify areas to be examined in greater detail or merged with other areas of lesser interest.

Segmentation of images into homogeneous regions has been a goal of image-understanding researchers for many years. Beginning with simple block-like objects [7], image segmentors have begun attempting to segment natural scenes. Early efforts [8]–[12] manipulated line drawings in different ways, but extracted these line drawings as a preprocessing step for higher level operations.

Extension of artificial intelligence based procedures to image segmentation often used top-down approaches based on *a priori* knowledge of the image content. Many of these approaches used training algorithms to train the classifier and highly heuristic features based on the *a priori* knowledge of the image and the purpose of the image understanding system [13]–[17]. An excellent description of each of these segmentation approaches and the context in which they were applied is contained in [18].

Common to all of these approaches is the extraction of line drawings by various methods. Thus the region boundaries represent the segmentation of the image. In some of these approaches, the edges are sought directly by edge detection [19]–[22], or functional approximation [23], [24]. In other approaches, the regions are detected first and the boundaries determined later. One method is a top-down procedure wherein the picture is segmented into progressively smaller regions until certain criteria are satisfied [25], [26]. Another method is a bottom-up approach, wherein the picture is divided into a large number of small regions (possibly as small as one pixel), which are successively merged to form larger regions [27], [28].

A few attempts at bottom-up approaches to image segmentation using clustering have been made in the past. The first of these was performed by Haralick and Kelly [29]. This procedure used a modified linking or "nearest neighbor" rule to form the clusters on multispectral image data. Further work has been performed using textural features and a classifier operating in the supervised mode [30]. The supervised mode requires that the cluster center be determined by "training." Finally, clustering has also been applied to images segmented by an edge-detection procedure [31]. An additional bottom-up approach to image segmentation is described by Ohlander [32]. This procedure uses histogram analysis to successively delete points contained in feature histogram peaks.

Pattern Recognition, Unsupervised Learning, and Clustering

A large methodology has been built up over the last several decades under the general subject heading of pattern recognition. It is convenient to divide this body of knowledge into two categories. The first category contains knowledge that is most closely related to computer-artificial intelligence. The second consists of mathematical theory and techniques from statistical data analysis and communication theory.

The artificial-intelligence approaches often use language theory to describe a scene in terms of primitive elements or subpatterns and their relationship to each other. The relationships are described in the syntactic-structure models of formal language. Visual patterns are considered to belong to a two-dimensional language. The structural descriptions of these patterns in terms of the grammar is the syntax. Recognition becomes syntax analysis (often called parsing). The limitations of these approaches are that relatively little work has been done in noisy syntax and that most existing linguistic schemes are in terms of shape which is but one of many features available to human observers. An exception to this is the work by Fu with recognizing and parsing noisy strings [47]. Nevertheless, context is easily visualized in such an approach as additional constraints on the relationships between the primitive elements.

The first results obtained in the general discipline now called pattern recognition were based on mathematical models [33].

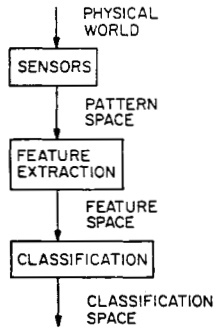


Fig. 2. Classical pattern recognition.

These models assume that a sensor or series of sensors measure physical quantities about an object in the real world (Fig. 2). In general, the measurements of the sensors form a vector that describes the object. In the case of visual data, the sensors are usually some form of camera, perhaps extracting multispectral measurements about the physical world. For purposes of manipulation of the data by digital computer, the image must be converted into digital form by appropriately sampling the image.

The pattern space consists of the image samples just described. The "first-order" features of an image are its brightness (possibly in several spectral regions), and the spatial coordinates of the appropriate point. Each point is usually called a picture element (pixel). Other features, such as texture, are properties of a region [2]. Thus the feature-extraction process may, in the case of images, enlarge the amount of data required to represent the image considerably.

The feature space, as described above, represents a high-dimensional (dimension >10 is not uncommon) space in which each point in the image is represented by a vector of features $x = (x_1, x_2, \ldots, x_n)^t$. Here n is the dimension of the feature space, and x_i is the value of the ith feature at a given pixel location. The classification problem is now to find separating surfaces in n dimensions which will partition the feature space into K mutually exclusive and collectively exhaustive regions. The classification which results from assigning the vectors in accordance with a particular partitioning of the feature space can then be evaluated as to the relevancy of such a partition to some image-understanding task.

This model of the feature space when applied to the image segmentation problem, implicitly assumes that numerical difference is directly proportional to perceptual difference, in the human perceptual system. This is an assumption which is almost certainly untrue, at the current state of knowledge about the human perceptual system and the current state of development of features used in digital image pattern-recognition techniques. Nevertheless, the existence of a (almost certainly) nonlinear transformation can be postulated which would map the feature vectors into a new space where the model described previously would be perceptually valid.

The determination of the separating surfaces in the traditional pattern-recognition system is made through the use of prototypes or training samples whose correct classification is known. These samples are fed to the system and establish the decision boundaries for use in classifying unknown samples. This approach is often called the "supervised" pattern-recognition approach.

Frequently, it is desirable to design a pattern classification system without the use of training samples [34]. The theoretical framework on which unsupervised pattern recognition is based is very tenuous. If nothing whatsoever is known about the data, the problem is not solvable in general. However, in the case of image-related data, it is known *a priori* (or at least assumed) that the data represents low-level perceptual differences. It is to be expected that regions of the image that appear the same would produce feature vectors that are near to each other, whereas regions that appear substantially different would produce feature vectors that are far apart. This assumption leads naturally to the expectation that similar appearing regions will produce groups of vectors that are close together in feature space. These groups of vectors will hereafter be called "clusters."

In general, the term clustering refers to the grouping of a given set of objects into subsets according to the properties of each object. The subsets are required to contain objects that are in some sense more similar to each other than to the objects in other subsets. Clustering has been used for several decades, and was first applied by Tyron to numerical taxonomy problems [35].

There are any number of clustering procedures, each having its own peculiar characteristics [36]. When it is anticipated that the clusters are tight and widely spaced the chain method [37], [38] may be used. However, the procedure runs into trouble when the clusters are close together and the boundaries are indistinct.

There are a number of procedures which will iterate to a local minimum for the average distance, from each sample to the nearest cluster mean. Perhaps the best example of these procedures is the nearest means algorithm adapted by Ball and Hall [39].

This procedure begins with an assumed number of clusters. The means are arbitrarily assigned, although the initial mean assignment will affect the number of iterations required for convergence. The data is then assigned to the nearest mean. After all of the data points have been assigned, the cluster means are recomputed based on the assigned data points. This process continues until the data assignment does not change, at which point the process is said to have converged. This algorithm will iterate to a local minimum for the average within-cluster distance.

For clustering procedures of the nearest means type, the key obstacle to be overcome is the determination of the "correct" number of clusters. It has been suggested that a possible approach is to obtain a measure of the clustering quality represented by some parameter, beta, [33], [40].

A number of measures have been proposed for beta, one of which is the ratio of the between- to within-cluster scatter measure [41]. The within-cluster and between-cluster measures are derived from within- and between-cluster scatter matrices. These measures are intended to measure the separability of the data. The within-cluster scatter matrix is based on the scatter of the data about the cluster means, and is given by

$$S_\omega = \frac{1}{K} \sum_{k=1}^{K} \epsilon\{(x - \mu_k)(x - \mu_k)^t\}$$

where x is the feature vector,

$$\epsilon\{\cdot\} = \frac{1}{M_k} \sum_{x_i \in S_k} (x_i - \mu_k)(x_i - \mu_k)^t$$

and μ_k is the mean of the kth cluster. M_k is the number of elements in the kth cluster, x_i is an element in the Kth cluster

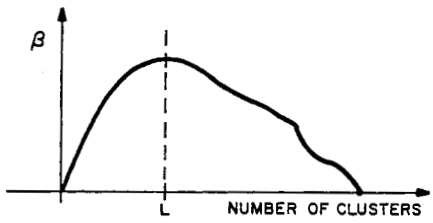


Fig. 3. Clustering fidelity criterion β_5.

(the set of such elements being given by S_k), and K is the total number of clusters. The between-cluster scatter matrix can be defined in numerous ways, but for $K \geqslant 2$ cluster, the most straightforward definition is given by

$$S_b = \frac{1}{K} \sum_{k=1}^{K} (\mu_k - \mu_0)(\mu_k - \mu_0)^t$$

where μ_0 is the overall mean vector of the entire mixture, and is given by

$$\mu_0 = \frac{1}{M} \sum_{i=1}^{M} x_i.$$

Here M represents the total number of points (pixels) to be clustered.

The goal of using the scatter matrices is a measure of cluster separability. It is therefore necessary to derive a number, parameter beta, from these matrices which is related to cluster separability. There are a number of ways of deriving such a number, among which are:

$$\beta_1 = \operatorname{tr}(S_w^{-1} S_b)$$

$$\beta_2 = \ln\{|S_w + S_b|/|S_w|\}$$

$$\beta_3 = \operatorname{tr} S_b - \mu(\operatorname{tr} S_w - c)$$

$$\beta_4 = \operatorname{tr} S_b / \operatorname{tr} S_w$$

$$\beta_5 = \operatorname{tr} S_b \cdot \operatorname{tr} S_w$$

where $\operatorname{tr}(\cdot)$ indicates "trace" or sum of the diagonal elements of a matrix, and $|\cdot|$ denotes the determinant of the matrix. When β_3 is used, the procedure is to maximize $\operatorname{tr} S_b$, subject to $\operatorname{tr} S_w = c$. Here μ is the Lagrange multiplier and c is a constant.

The terms β_1 and β_2 are invariant under any nonsingular linear transformation. The terms β_3 and β_4, while easier to compute, depend on the coordinate system.

The use of the parameter beta to measure the "goodness" of clustering requires that a knee in the beta versus number of clusters be detected. If the data is noisy and the curve is not smooth, this may be very difficult. A better procedure would be to observe a parameter β_5 which passes through a maximum at the intrinsic number of clusters (see Fig. 3).

When the number of clusters equals 1, $\operatorname{tr} S_w = \sigma^2$, the variance of the mixture, $\operatorname{tr} S_b = 0$, and $\beta_5 = 0$. When the number of clusters equals M, where M is the total number of vectors in the mixture,

$$\operatorname{tr} S_w = 0 \quad \text{and} \quad \operatorname{tr} S_b = \sigma^2.$$

Hence $\beta_5 = 0$.

This measure is zero at the limiting points of the clustering and greater than zero in the interval. Therefore, it must attain at least one (and perhaps several) maximum value(s) somewhere in the interval. The ideal behavior for β_5 would be for it to attain a unique maximum at a clustering of the data that would be regarded as "good" by a human observer.

The use of $\operatorname{tr} S_w$ and $\operatorname{tr} S_b$ to define clustering quality implicitly defines a weighting function on the cluster size. Each term in the within- and between- cluster scatter matrices is composed of a weighted sum of terms. The weighting is based on the relative frequency $(1/M_k)$ of the data points in each cluster.

An interesting relationship is true when this implicit weighting is used. In this case

$$\operatorname{tr} S_w + \operatorname{tr} S_b = \operatorname{tr} \phi = C$$

where ϕ is the covariance matrix of the data, and C is a constant (sum of the total variance of the data). Hence,

$$\beta_5 = \operatorname{tr} S_b \cdot \operatorname{tr} S_w = (C - \operatorname{tr} S_w) \operatorname{tr} S_w$$

differentiating with respect to $\operatorname{tr} S_w$, and setting the derivative equal to zero yields

$$\operatorname{tr} S_w = C/2$$

which implies that β_5 achieves a maximum at the clustering that causes $\operatorname{tr} S_w$ to equal one-half the $\operatorname{tr}[\phi]$.

Further,

$$\beta_4 = \operatorname{tr} \frac{S_b}{\operatorname{tr} S_w} = \frac{C - \operatorname{tr} S_w}{\operatorname{tr} S_w}.$$

When $\operatorname{tr} S_w = C/2$,

$$\beta_4 = \frac{C - C/2}{C/2} = 1.$$

Therefore, the ratio of between- to within- cluster scatter measures will be exactly 1 at the "product maximum" of β_5.

Knowledge of this relationship is an advantage for real-time applications, in that, determination of the product maximum (β_{max}) requires that clustering be performed on one greater number of clusters than the number at which the product maximum occurs in order to detect a decrease.

Different images can be expected to be segmented most efficiently by different sets of features, depending on the content of the scene. Once initial clustering has been performed, it may be desirable to discard those features not contributing to good clustering and recluster based on the most important features. In order to accomplish this, some means for evaluating the usefulness of the features is required. A related problem is that the features may be highly correlated in the original space. Thus several highly correlated features may be evaluated as good while conveying essentially the same information due to the high degree of correlation. Thus feature selection in an uncorrelated space is highly desirable.

After the product maximum of β_5 has determined the intrinsic number of clusters, the criterion of optimality for the selection of a feature set is the probability of misclustering (classification) of the samples. Several measures have been developed which upper bound the misclassification rate. Specifically, for a symmetric cost function classifier and Gaussian data, the Bayes error rate P_e has been shown to be upper-bounded inversely as the Bhattacharyya measure [41], [42].

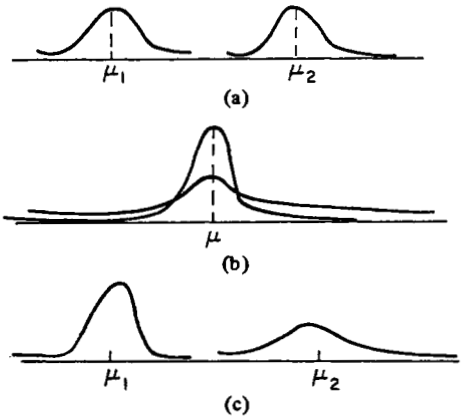


Fig. 4. Bhattacharyya (one-at-a-time) feature selection. (a) Equal variances, unequal means. (b) Equal means, unequal variances. (c) Unequal means and variances.

Hence,

$$P_e \leqslant P(S_1)\,P(S_2)\exp\,[-B(S_1, S_2)]$$

for a two-class (cluster) problem where

$$B(S_1, S_2) = \tfrac{1}{4}\ln\{\tfrac{1}{4}\,|[\phi_2]^{-1}[\phi_1] + [\phi_1]^{-1}[\phi_2] + 2[I]|\}$$
$$= \tfrac{1}{4}\,\mathrm{tr}\,\{([\phi_1] + [\phi_2])^{-1}(\mu_1 - \mu_2)(\mu_1 - \mu_2)^t\}$$

and $P(S_1)$, $P(S_2)$ are the prior probabilities of clusters S_1 and S_2. The Gaussian distributions are given by

$$P(x \mid S_k) = N(\mu_k, [\phi_k])$$

and $[\phi_k]$ is the covariance matrix of the kth class or cluster.

For a multiclass problem, P_e can be bounded by pairwise averaging, i.e.,

$$P_e \leqslant \sum_{i>j}^{K}\sum^{K} P(S_i)\,P(S_j)\exp\,[-B(S_i, S_j)].$$

The previous equation is called the many-at-a-time form of the Bhattacharyya distance measure. This equation requires that the covariance matrix of every class be invertible, a condition which may not be achievable in practice where the covariance matrices are sample determined. A computationally more simple form of the above results when the one-at-a-time form is utilized. This form is given by:

$$B_n(S_i, S_j) = \frac{1}{4}\ln\left\{\frac{1}{4}\left(\frac{\sigma_i^2(n)}{\sigma_j^2(n)} + \frac{\sigma_j^2(n)}{\sigma_i^2(n)} + 2\right)\right\} + \frac{1}{4}\left\{\frac{(\mu_i(n) - \mu_j(n))^2}{\sigma_i^2(n) + \sigma_j^2(n)}\right\}$$

where n refers to the nth dimension of the space and σ_i, σ_j are variances of the ith and jth cluster data on dimension n. This form involves only scalar means and variances.

Fig. 4 provides some insight into the behavior of the one-at-a-time Bhattacharyya measure. When the variances are equal but the means are not, as in Fig. 4(a), the first term of the Bhattacharyya measure will be zero but the second term will be nonzero. The second term will be large if the variance is small under this condition, implying that a large difference in means accompanied by small variances, is a desirable quality in a feature for distinguishing between two clusters. The situation depicted in Fig. 4(b) is the reverse, that is, the means are equal but the variance is not. If the variances are significantly different, the feature is still considered of potential usefulness in separating the clusters. Thus in this situation, the second term of the Bhattacharyya distance will be zero but the first term will be nonzero. Finally, in Fig. 4(c) both the mean and variance are unequal and both terms of the measure will be nonzero. The feature rejection criterion would be to only retain those features with large Bhattacharyya value.

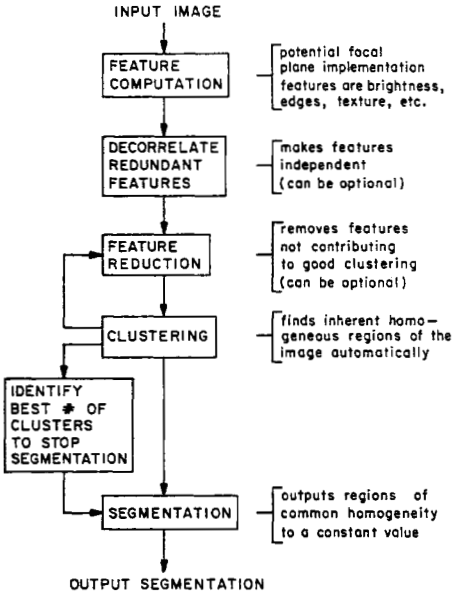


Fig. 5. General block diagram.

The performing of feature evaluation–rejection in uncorrelated space implies that an eigenvector (or discrete Karhunen–Loeve) transformation is required [43]. While the dimensions having the largest eigenvalues will be the best under certain conditions, they will not be optimal in general. However, the one-at-a-time Bhattacharyya measure will pick the correct eigenvector regardless of their mixture eigenvalues. The resulting reduced set of features provides computational savings as well as a tighter more dense clustering.

Image Segmentation by Clustering

The overall approach taken to segment images by clustering is depicted in the general block diagram of Fig. 5. The feature computation block computes several features at each pixel. These features are related to brightness and texture for several window sizes centered on every pixel.

The feature decorrelation is performed by a multidimensional axis rotation (Karhunen–Loeve transformation). The rotation is performed so that the new feature set is uncorrelated.

Feature reduction, which is accomplished subsequently, will retain only those features necessary for good clustering as defined by the Bhattacharyya measure. If feature reduction is not performed on decorrelated features, several highly correlated features may be retained, but convey essentially the same information.

Clustering is again performed but now on the reduced feature set. When the optimum number of clusters is determined, by the product maximum of β_5, the cluster means are forwarded to the segmentation phase of the algorithm. The segmentation phase assigns every pixel (vector) in the image to the closest cluster mean received from the clustering algorithm.

The feature decorrelation is necessary in order that the feature reduction will retain the minimum number of features contributing to good clustering. The feature reduction improves the quality of the segmentation by discarding noisy and less useful features. The first clustering is performed explicitly for the purpose of evaluating the features. The algorithm iterates to a correct number of clusters, and the features are evaluated at that point. The second clustering is performed for the purpose of finding the means with which to segment the image in the segmentation phase of the algorithm.

A detailed flow-diagram of the algorithm is illustrated in Fig. 6. The feature computation, which will be described in detail, subsequently produces as described previously a vector at each pixel location. These vectors are forwarded to the covariance computation routine and to the Karhunen–Loeve rotation.

The covariance matrix is computed over the feature set and the diagonal elements of this matrix are the feature variances over the image. The matrix which diagonalizes the covariance matrix is computed yielding

$$A^t \phi A = \Lambda$$

where Λ is diagonal having the eigenvalues of the covariance matrix as diagonal elements, i.e.,

$$\Lambda = \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_n \end{bmatrix}.$$

This transformation corresponds to a multidimensional axis rotation and is the discrete form of the Karhunen–Loeve transformation. The covariance matrix in the rotated space will be diagonal and will be

$$\phi_R = A^t \phi A = \Lambda.$$

This rotated space of features is forwarded to the clustering algorithm for clustering.

The clustering algorithm uses the K-means algorithm for $K = 2, 3, 4, \cdots, 16$ clusters. At each step, the quality of clustering is computed as $\beta_5 = \operatorname{tr} S_b \cdot \operatorname{tr} S_w$. The average pairwise Bhattacharyya distance is computed for every feature. At the product maximum of β_5, the Bhattacharyya distance for all features is computed. Features having a Bhattacharyya distance which far exceeds the overall average are identified for use in the final clustering. Since these features are uncorrelated, only the minimum necessary are retained for good clustering. The flowchart of the algorithm is shown in Fig. 7.

The algorithm begins at 2 clusters. The initial means are established by computing the mean and variance of each feature over the image. The 2 initial cluster centers are chosen to be evenly spaced on the diagonal of positive correlation at ±1 standard deviation in the hyperspace of the feature set. As the number of clusters is incremented, one of the vectors will have a largest distance to the cluster center it is closest to. This vector then becomes a new cluster center. Final segmentation is performed on every pixel, utilizing the means or cluster centers computed during the clustering algorithm.

An aspect of clustering which has a major effect on the results is the feature set used to describe the image. For monochrome imagery, the most obvious features that are intuitively important to human observers are brightness and

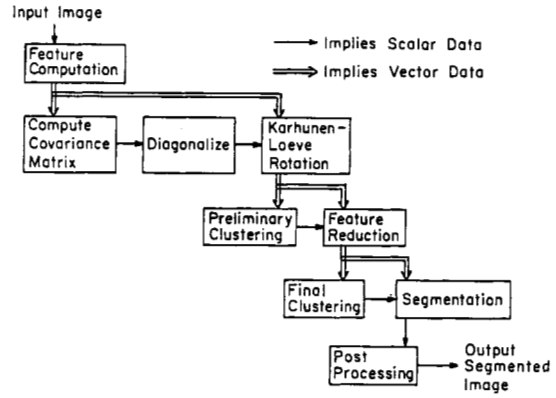


Fig. 6. Flow diagram of image segmentation algorithm.

texture. Brightness is a relatively straightforward concept, but texture is not. Much research has been performed regarding human perception of texture, and the subject is far from closed.

To date, the most promising results obtained with texture operators utilize the grey level probability dependancy matrices proposed by Haralick [44]. The normal approach followed with these measures is to compute the grey level dependency matrices and then to derive texture measures from the matrices themselves. A large number of measures can be computed from these matrices, but Thompson [2] found that perhaps five or less correlate significantly with human perception.

Other texure measures which have been proposed are the "edges per unit area" as a measure of the local edge density. This measure was computed and used in segmenting several types of scenes. The basic edge detector is the Sobel operator which is defined as follows:

$$[s_1] = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

and

$$[s_2] = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}.$$

At each pixel, the image is multiplied by the $[s_1]$ and $[s_2]$ masks yielding $s1$ and $s2$. A function of the Sobel magnitude (SM) is then defined as

$$\mathrm{SM} = \log (s1^2 + s2^2)^{1/2}$$

and the Sobel phase (SP) is given as

$$\mathrm{SP} = \arctan\left(\frac{s2}{s1}\right).$$

A modified Sobel phase texture operator was computed as

$$\mathrm{SP}^* = (\mathrm{SP} + \pi)\,\mathrm{SM}$$

in an attempt to suppress the phase operator when no texture is present. These primitive operators permitted segmentation of numerous monochrome images, with varying degrees of success.

The goal of the algorithm developed here is to perform gross overall scene segmentation. For this reason, very small

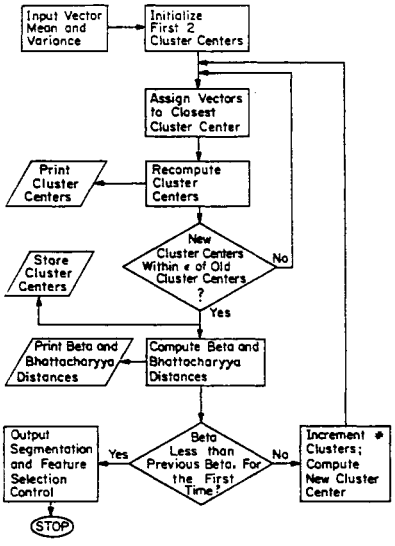


Fig. 7. Clustering algorithm flowchart.

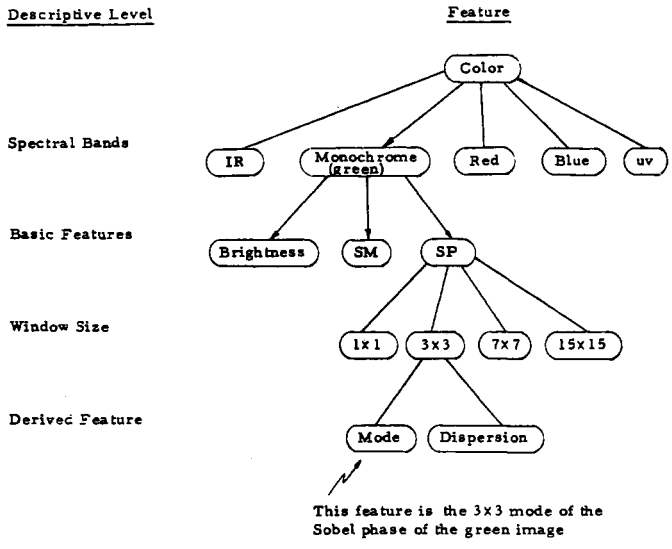


Fig. 8. Possible feature combinations.

"fine grain" segments were considered undesirable. It was decided to perform a prefiltering to make some basic decisions about region character on a local level as a first step prior to segmentation.

Linear operators tend to blur the region boundaries and reduce the region boundary resolution. A nonlinear filter that does not blur the boundaries was conceived and called a "mode filter." This filter computes a local area histogram centered on each pixel, for different region sizes, and outputs the mode or most frequently occurring value. The height of the histogram at each pixel may also be used as a measure of the local dispersion (standard deviation) of the region. Region sizes of 3 X 3, 7 X 7, and 15 X 15 were computed.

The effect of the mode filter is to replace every pixel with the most frequently occurring value in a small region centered on the pixel. This removes small variations in brightness and tends to create relatively large regions of completely uniform character. The mode filter causes almost no loss in boundary resolution because the output of the filter does not change until a majority of the values change. Then the filter output changes value at the point where the center of the window crosses the region boundary.[1]

Color and multispectral features were computed by performing mode filtering on the three or more spectral images (i.e., red, green, and blue for visible color). This expanded feature set produced more satisfying results, as would be expected from the increased information available from the multispectral data. However, as seen from Fig. 8, even this limited set of features tends to increase quite rapidly for the large number of combinations available. There were generally five feature sets which were used for image clustering. These feature sets are clearly combinations of a few different types of basic features, and are based on the Sobel operators and the mode dispersion for texture information and on the mode value for brightness information. The individual sets will be developed in the next section.

As has been previously discussed, the primary goal of this effort was to develop the segmentation algorithm. Feature experimentation was done as required to investigate the performance of the segmentation algorithm, but was not pursued in great depth as a topic having its own merit. A great deal of investigation into features, especially texture, obviously remains to be done. However, the features used here were selected mainly for their potential use of implementation possibly with charge-coupled-device (CCD) circuitry near the focal plane of a sensor.

[1] This filter is not to be confused with the median filter, both of which are nonlinear operators the latter being used mainly for outlier rejection and noise filtering.

Experimental Results

An enormous amount of data was collected during the performance of numerous experiments in segmenting various kinds of images. A representative sampling of that data is included in the photographs and figures in this section. The photographic data consists of photographs of the features, both correlated and decorrelated, and the resulting segmentations. The graphs depict behavior of the Bhattacharyya distance measure, used for feature selection, and of the clustering quality measure β, used to stop the algorithm at the correct number of clusters.

The first image to be segmented is a monochrome scene of an armored personnel carrier (APC) shown in Fig. 9. The feature computation stage of Fig. 5 computed the following features as input to the vector-space. For this image the features were:

x_1 = monochrome brightness
x_2 = Sobel log magnitude
x_3 = Sobel phase
x_4 = (3 X 3) mode of brightness
x_5 = (3 X 3) mode of Sobel log magnitude
x_6 = (3 X 3) mode of Sobel phase
x_7 = (7 X 7) mode of brightness
x_8 = (7 X 7) mode of Sobel log magnitude
x_9 = (7 X 7) mode of Sobel log magnitude
x_{10} = (15 X 15) mode of brightness
x_{11} = (15 X 15) mode of Sobel log magnitude
x_{12} = (15 X 15) mode of Sobel phase.

Thus each pixel generates a point or sample in 12-dimensional vector space x. Fig. 9 shows each coordinate or feature of this space in pictorial form, for viewer identification. These features are referred to as the 12 original features as they have not yet been subjected to feature decorrelation or feature

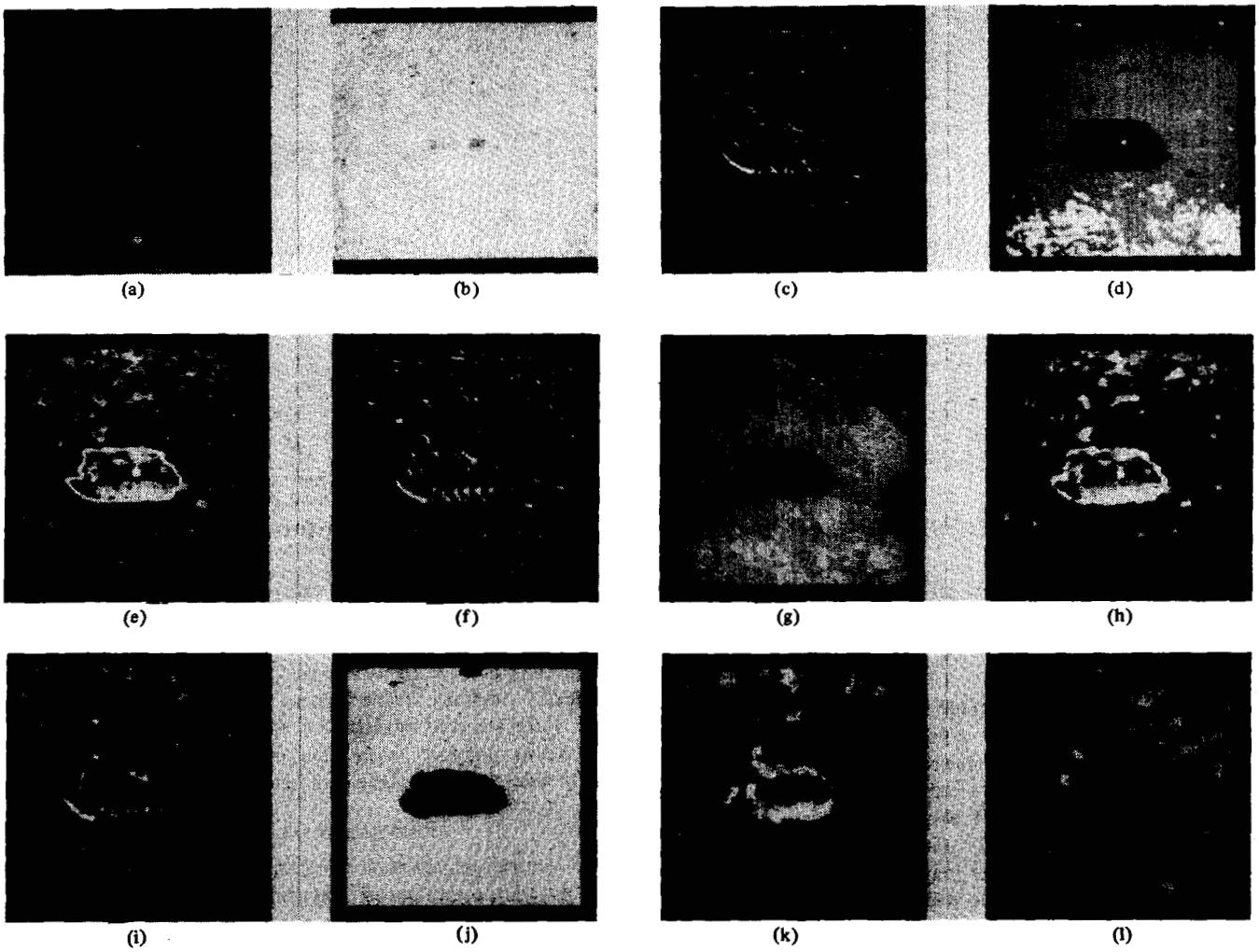


Fig. 9. APC image original features. (a) Original. (b) Sobel log magnitude. (c) Sobel phase product. (d) Original, mode filtered 3 X 3. (e) Log magnitude, mode filtered 3 X 3. (f) Phase product, mode filtered 3 X 3. (g) Original, mode filtered 7 X 7. (h) Log magnitude, mode filtered 7 X 7. (i) Phase product, mode filtered 7 X 7. (j) Original, mode filtered 15 X 15. (k) Log magnitude, mode filtered 15 X 15. (l) Phase product, mode filtered 15 X 15.

reduction (see Fig. 5). These features were subjected to clustering without decorrelation or feature reduction producing the segmentations of Fig. 10. The product maximum of β_5 occurred at 9 clusters. A graph of the average Bhattacharyya distance versus the number of clusters for this image is shown in Fig. 11. This graph is constructed such that the average Bhattacharyya distance for each feature is normalized by the average for all features at each number of clusters. The normalized overall average therefore consists of the horizontal line at 1.0. While there is some changing of relative position between the features as the number of clusters is varied, those features which are above average tend to remain above average, and those which are below average tend likewise to remain below average. The graph shows reasonably consistent behavior of the Bhattacharyya distance measure as the number of clusters varies. Thus feature selection based on this measure is a consistent procedure. It is interesting to observe that the four dominant features from Fig. 11 according to the Bhattacharyya measure are x_1, x_4, x_7 and x_{10}. These four features are all derived functions of the brightness of the original image. Thus no texture features are needed (as one might intuitively believe), as simple thresholding appears sufficient for segmenting this image.

Returning to the results of Fig. 10, it is apparent that 9 clusters appear to be psychovisually unpleasant because as human viewers, we might believe the imagery is broken up into too many segments or regions. Thus feature reduction is in order. At the product maximum of β_5 the four best features were 7, 1, 4, and 10, in that order. These features are original mode filtered 7 X 7, original unmodified, original mode filtered 3 X 3, and original mode filtered 15 X 15 as mentioned earlier. Thus all of the texture information has been discarded. These features were used to again cluster the image and the results are shown in Fig. 12. The product maximum of β_5 occurred at 2 clusters for this reduced feature set. Pictorially, this segmentation is far more pleasing. Even when the algorithm is forced beyond 2 clusters, the additional segments are culturally acceptable (that is, 3 regions pick up the star of the APC, etc.).

The covariance matrix of the 12 original feature set was computed and diagonalized. Each vector of the rotated feature set is computed from the spatially corresponding vector in the original feature set. The first four features of the actual rotated feature set is shown in Fig. 13. The version shown in Fig. 13 has been rescaled for ease of viewing. The covariance matrix of the rotated features is diagonal and each

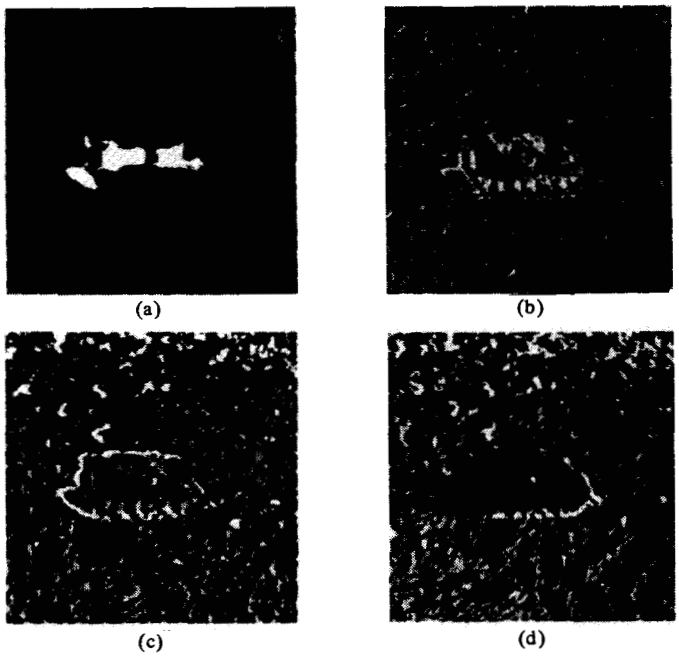


Fig. 10. Twelve nonreduced correlated features. (a) Seven regions. (b) Eight regions. (c) Nine regions (best number of regions). (d) Ten regions.

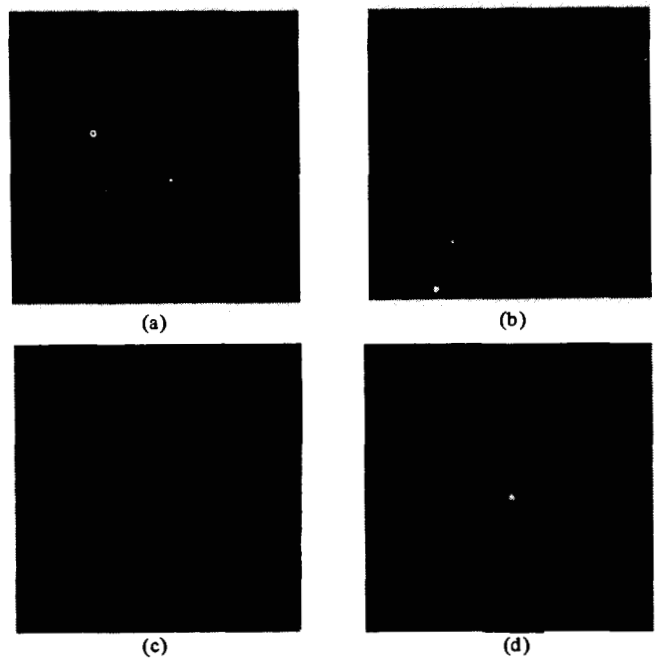


Fig. 12. Four induced correlated features. (a) Two regions (best number of regions). (b) Three regions. (c) Four regions. (d) Five regions.

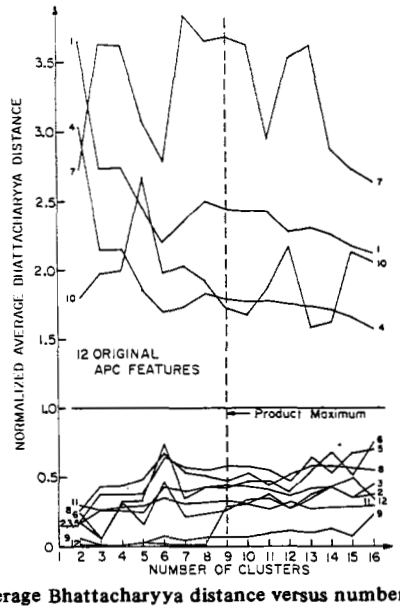


Fig. 11. Average Bhattacharyya distance versus number of clusters.

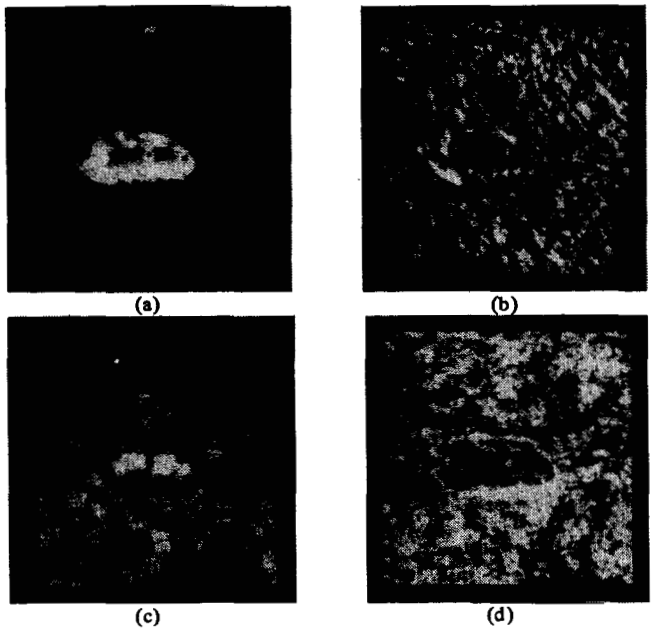


Fig. 13. APC image rotated features. (a) Rotated feature 1. (b) Rotated feature 2. (c) Rotated feature 3. (d) Rotated feature 4.

diagonal entry is equal to the variance for the respective feature. The segmentation at the product maximum of β_5 for the 12 rotated features appear very similar. That this is so is expected since the multidimensional rotation of the axes by the rotation matrix is a linear unitary invertible-map and should not change the shape of the clusters. The differences which do exist are due in small part to numerical (roundoff) errors in the computation. The nearest means algorithm will converge to a local minimum in the average intercluster distance. In addition, since convergence of the algorithm for a fixed number of clusters is considered to occur when the means change less than one brightness value in any dimension, the final clustering is also slightly sensitive to the direction from which the convergence is approached. Nevertheless, the agreement is surprisingly good, and supports the hypothesis that intrinsic clusters do in fact exist in the data.

The average Bhattacharyya distances for the rotated features were computed for all cluster numbers and are plotted in Fig. 14. The best of the rotated features was substantially higher in Bhattacharyya distance than any of the other features. This is to some extent expected, since the rotation process will compact the maximum amount of information into the features having the largest eigenvalues. Accordingly, it was decided to perform clustering on this one exceptionally good feature. The results of this are also shown in Fig. 15. The classification of the bushes in the images as being the

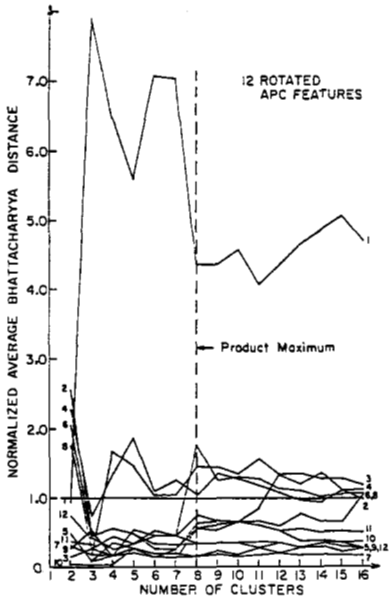


Fig. 14. Average Bhattacharyya distance versus number of clusters.

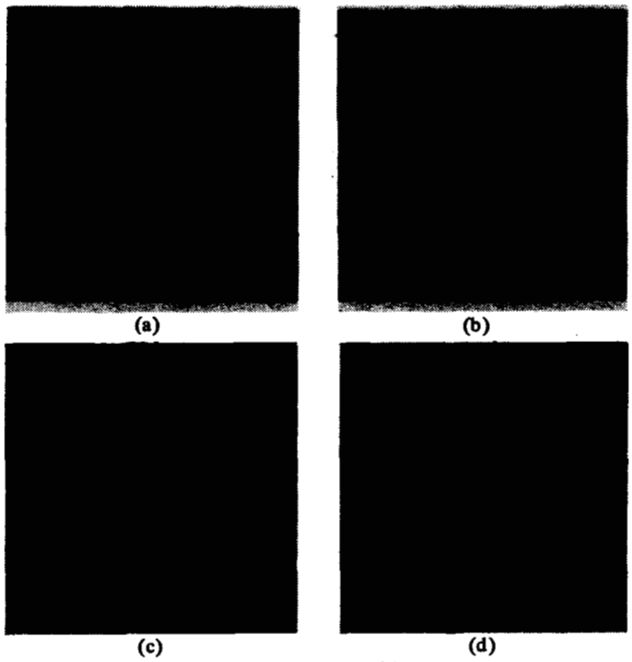


Fig. 15. Single best decorrelated feature. (a) Two regions (best number of regions). (b) Three regions. (c) Four regions. (d) Five regions.

same as the vehicle constitutes an error or misclassification in the process. However, the similarity of these results is quite comparable with that of the four best nonrotated features of Fig. 12.

The product of the between- and the within-cluster scatter measure was computed for each number of clusters. The between-scatter and within-scatter measures are normalized so that they range between 0 and 1. These products are plotted versus the number of clusters for the APC image under various conditions in Fig. 16.

An interesting phenomenon can be observed in the behavior of the clustering quality measure (product) in Fig. 16. The behavior of the quality measure for the rotated and nonrotated feature sets is almost identical, which is to be expected if the intrinsic structure of the data is unchanged by the feature rotation process. The clustering quality measure maximum is rather broad in both cases for the full feature sets. The reduced feature sets on the other hand, show a sharper, more clearly defined peak in the quality measure, suggesting that the intrinsic clusters in the data are more clearly defined in the reduced sets of features. For all images tested, the quality measure tended to demonstrate a more clearly defined maximum when computed on feature sets that were expected to yield "better" clustering.

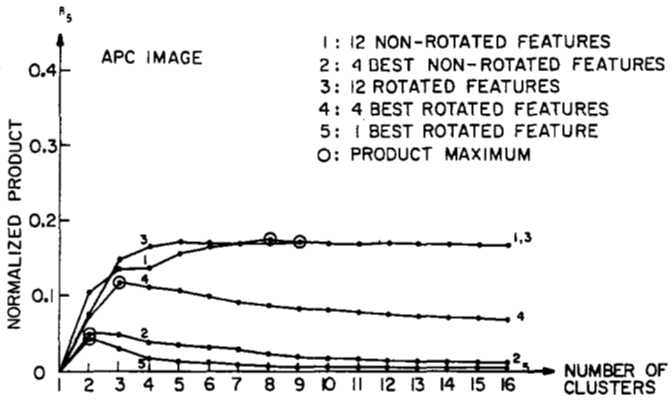


Fig. 16. Product versus number of clusters.

The segmentation procedure was also applied to polychromatic imagery. It would be expected that somewhat improved results would be obtained from the expanded feature set, and the results seem to confirm this expectation. The features utilized were:

x_1 = red brightness
x_2 = green brightness
x_3 = blue brightness
x_4 = (3 X 3) mode of red brightness
x_5 = (3 X 3) dispersion of red brightness
x_6 = (3 X 3) mode of green brightness
x_7 = (3 X 3) dispersion of green brightness
x_8 = (3 X 3) mode of blue brightness
x_9 = (3 X 3) dispersion of blue brightness
x_{10} = (7 X 7) mode of red brightness
x_{11} = (7 X 7) dispersion of red brightness
x_{12} = (7 X 7) mode of green brightness
x_{13} = (7 X 7) dispersion of green brightness
x_{14} = (7 X 7) mode of blue brightness
x_{15} = (7 X 7) dispersion of blue brightness.

The results of segmenting in a 15-dimensional space are presented in Fig. 17. The product maximum of β_5 was determined to be two clusters. The additional segmentations are the result of permitting the algorithm to continue beyond the best number of clusters. Again, it is gratifying that the additional segments appear to be culturally relevant, in the sense that new regions are structural entities (i.e. windows, drain pipes, shadowed areas, etc.) and are not just randomly scattered.

Fig. 18 presents the clustering fidelity criterion versus number of clusters and appears to peak at two clusters. Also the single best feature in rotated space produces an image segmentation identical to that of Fig. 17(b) to within a very few number of pixels. Thus a 15:1 compression appears available. Note also that the full feature set still provided a sharply peaked maximum in contrast to the monochrome

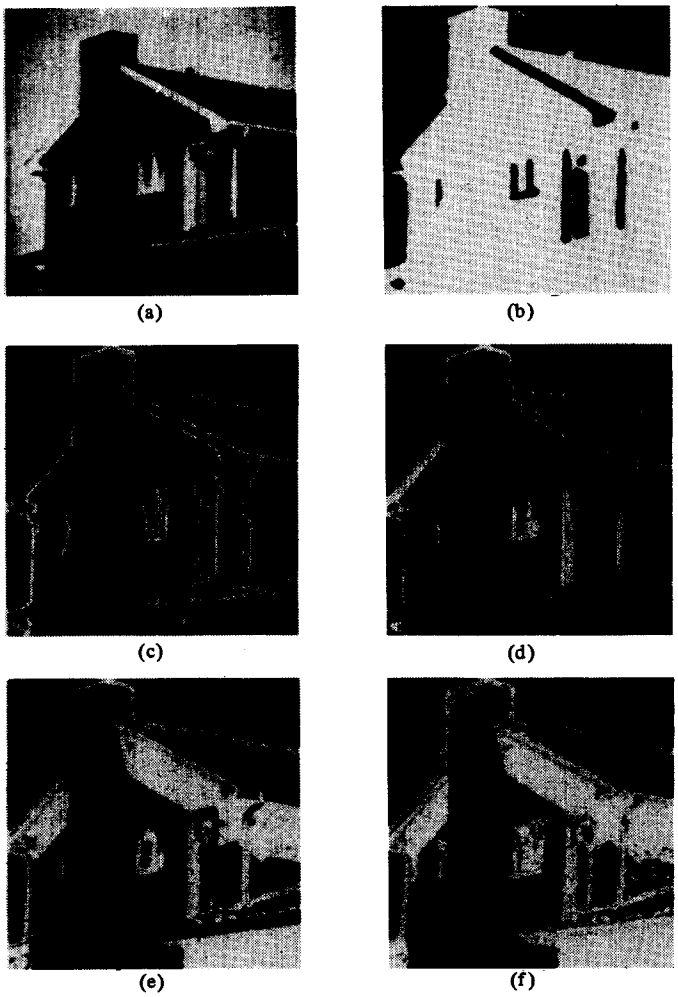


Fig. 17. Segmentation of house picture. (a) House original. (b) Two regions (best number of regions). (c) Three regions. (d) Four regions. (e) Five regions. (f) Six regions.

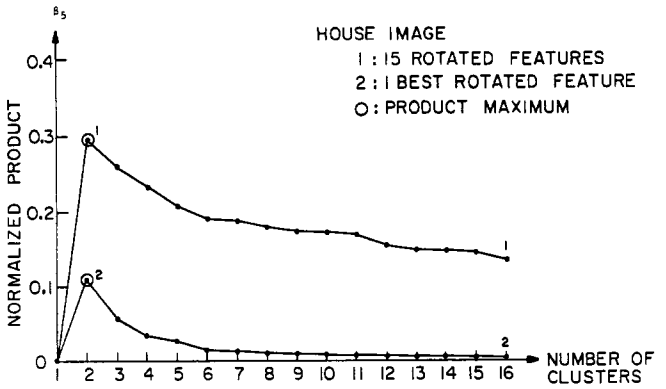


Fig. 18. Product versus number of clusters.

imagery. Indeed this well-behaved cluster maximum can be attributed to the additional information that color provides.

With certain minor modifications, the segmentation algorithm described in this paper can be adapted to near real-time operation. In the sense used here, near real-time implies operation at standard television rates.

Fig. 19 is a block diagram of a hypothetical system. The feature computer computes the features in real-time from the input television image. The technology for this block of

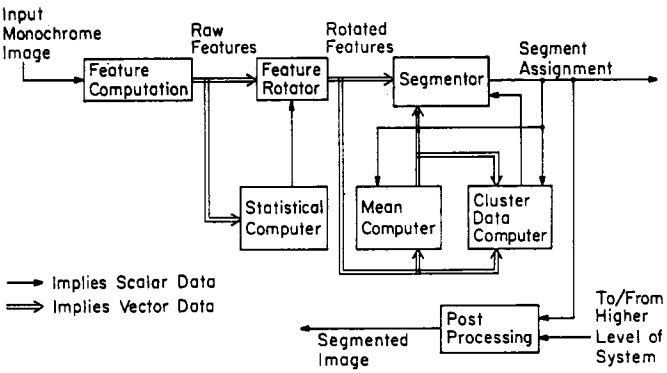


Fig. 19. TV rate image segmentor.

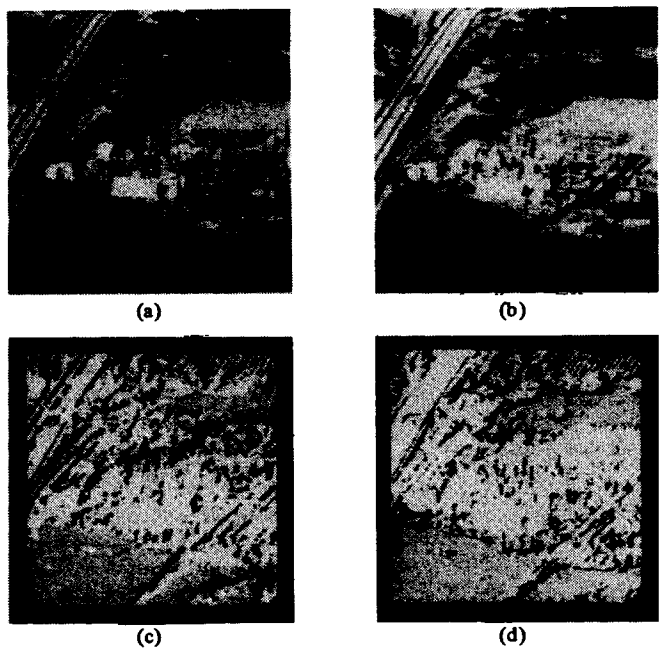


Fig. 20. Motion picture results. (a) Original—Frame 1. (b) Original—Frame 5. (c) Segmentation—Frame 1 (four clusters). (d) Segmentation—Frame 5 (four clusters).

the system is in development [45] on CCD hardware and may even be implemented on the focal plane of a multielement sensor. This conceptualization is sometimes called the "smart sensor" design.

The raw features are then forwarded to the feature rotator. The feature rotator performs a real-time multidimensional rotation of the input vector, that is, each component of the output vector is a weighted sum of the input vector components. The weights are a function of the picture statistics, specifically the picture covariance matrix which is computed and diagonalized by the statistical computer. The statistical computer may consist of a combination of a microprocessor and other hardware. It is a reasonable assumption that the picture statistics will not change substantially over a small number of frames. In order to verify this, the algorithm described above was used to segment two frames of a motion picture of a chemical plant. The results of these segmentations are shown in Fig. 20, along with the original photographs. The motion picture was taken from a moving aircraft, and the originals are not spatially registered, as can be seen. They are five frames apart in the motion picture.

The two segmentations however, appear quite similar, and support the hypothesis that the statistical structure of the data can be identified for the purposes of segmentation even when the pictures are not spatially registered.

If a real-time system is implemented, and frame to frame amplitude differences are expected, either appropriate scaling will be required or the rotation matrix will have to be found to change slowly. The effect of this procedure would be to rotate image feature sets with a nonoptimal rotation matrix. Since the rotation is performed to permit feature rejection in decorrelated space, the penalty for this procedure will most likely be small.

Conclusions

This paper has presented a procedure for gross segmentation of digital imagery. The procedure uses an unsupervised method, and requires no human interaction or adjustable thresholds. There are disadvantages to using an unsupervised approach. So little is known about the human-perceptual system that the resulting segmentations will usually not be as satisfying as segmentations made by a human being or those performed by a carefully trained segmentor operating in a supervised mode. Additionally, the segmentor has no knowledge of the intent of the segmentation except that provided implicitly through the features selected to be used.

There are however, advantages to the unsupervised approach. The construction of a data set to use during the training phase of the supervised approach is time consuming and tedious. Additionally, the supervised method is incapable of satisfactory performance in situations where the statistics of the scene vary substantially. Situations that are likely to encounter such statistics are those in which the sensor characteristics vary and those in which near real-time segmentation of real images is desired. The difference is appearance with weather, time of day, and terrain makes an unsupervised procedure mandatory.

The procedure outlined herein lends itself conceptually to near real-time implementation. While the design of such a system to operate at television rates will require considerable ingenuity on the part of the circuit designers, such a system should find wide application in target recognition/tracking systems and possibly may be used to solve the problem of cross correlation of the same scene observed by sensors of radically different characteristics. With some generalization of the concept of cross correlation, segmentations of the same scene viewed by different sensors can be compared.

The unsupervised approach may also reveal characteristics in the data (image) that were unobserved by the human observer. There may exist inherent clusters in the data that passed unnoticed by human beings. Use of a supervised procedure will tend to further mask these unobserved characteristics, as the training of the classifier effectively instructs the classifier to ignore these characteristics. The unsupervised approach may eventually find usefulness in image enhancement because of the ability to detect unnoticed structure in the data.

Further work is certainly necessary in understanding the human-perceptual system at its intermediate level and using this knowledge to develop features to improve the performance of the segmentor. It may well develop that some textural recognition processes occur at a fairly high level in the human-perceptual system and do not lend themselves to implementation in the lower levels of an image-understanding system. If so, the improved understanding of the human-perceptual system will prove valuable as much for what it indicates cannot be done as it is for its indications of what can be done.

The clarification of what is meant precisely by a "segmented image" is also an avenue for further investigation. If a "well-segmented image" can be represented by a mathematical criteria, then analysis based on picture statistics will almost certainly provide suggestions on how to improve segmentor performance. In addition, it will provide means for predicting hypothetical system performance without having to build and test the system.

Much of the usefulness of an image-segmentation system must be determined by application. The current state of the art in image-understanding systems is such that applications are just now being postulated, much less implemented and tested. The advantages of the procedure described herein seem to be two-fold.

First, the procedure provides the cluster means directly as a by-product of the segmentation process. This is opposed to the previous procedures, which segment the scene with boundary-detection methods, compute features inside the boundaries, and only then perform clustering to determine the means.

A second advantage of this procedure is its potential for real-time implementation. Many previous procedures have required exact spatial stationarity of the image data to permit the iterations necessary to perform segmentation. This procedure requires only that the picture statistics change slowly with time, and does not require storing the entire image at one time. Such a procedure will have clear advantages when the sensor is mounted on a moving platform as in target detection/recognition systems.

However a final comment must be made at this point. The procedure as implemented, currently has enormous computational requirements. These are many orders of magnitude greater than those of the techniques described in [19]-[30]. This limitation will require a possible order of magnitude improvement in computational efficiency and/or processing elements before real-time or near real-time applications are contemplated.

Acknowledgment

The authors wish to acknowledge the support provided by the personnel of the Image Processing Institute in developing and maintaining the facilities necessary to carry out the research report herein. Particular thanks go to Professor W. K. Pratt and Mr. Ray Schmidt in this regard. In addition, the authors are indebted to Major David Carlstrom of the IPTO Office-ARPA for suggesting this important image understanding research task.

References

[1] P. H. Winston, "Heterarchy in the MIT robot," M.I.T. Artificial Intelligence Lab., Cambridge, MA, Vision Flash 8.
[2] A. L. Zobrist and W. B. Thompson, "Building a distance function for gestalt grouping," *IEEE Trans. Comput.*, vol. C-24, pp. 711-719, July 1975.
[3] J. M. Tenenbaum, "On locating objects by their distinguishing features in multisensory images," Artificial Intelligence Cent., Stanford Research Inst., Menlo Park, CA, Tech. Note 84.
[4] C. A. Harlow and S. A. Esenbeis, "The analysis of radiographic images," *IEEE Trans. Comput.*, vol. C-22, pp. 678-689, July 1973.
[5] E. C. Freuder, "Suggestion and advice," M.I.T. Artificial Intelligence Lab., Cambridge, MA, Vision Flash 43, Mar. 1973.
[6] ——, "Active knowledge," M.I.T. Artificial Intelligence Lab.,

Cambridge, MA, Vision Flash 53, Oct. 1973.
[7] L. G. Roberts, "Machine processing of three-dimensional solids," in *Optical and Electro Optical Information Processing*, J. T. Tippett *et al.*, Eds. Cambridge, MA: M.I.T. Press, 1965, pp. 159-197.
[8] A. Guzman-Arean, "Computer recognition of three-dimensional objects in a visual scene," Thesis (EE), M.I.T. Project MAC, Cambridge, MA, MAC-TR-59, Dec. 1968.
[9] P. Winston, "Learning structural descriptions from examples," Thesis (EE), M.I.T. Project MAC, Cambridge, MA, MAC-TR-76, Sept. 1970.
[10] D. L. Waltz, "Generating semantic descriptions from drawings of scenes with shadows," Thesis (EE), M.I.T. Artificial Intelligence Lab., Cambridge, MA, AI-TR-271, Nov. 1972.
[11] Y. Shirai, "A heterarchial program for recognition of polyhedra," M.I.T. Artificial Intelligence Lab., Cambridge, MA, AI Memo 263, June 1972.
[12] G. R. Grape, "Model based (intermediate level) computer vision," Thesis (CS), Stanford Univ., Stanford, CA, AIM-201, May 1973.
[13] M. D. Kelly, "Visual identification of people by computer," Thesis (CS), Stanford Univ., Stanford, CA, AIM-130, July 1970.
[14] T. M. Sakai and T. Kanode, "Computer analysis and classification of photographs of human faces," Kyoto Univ., Kyoto, Japan, Rep., 1972.
[15] Y. Yakimovsky, "Scene analysis using a semantic base for region growing," Thesis (CS), Stanford Univ., Stanford, CA, AIM-209, July 1973.
[16] G. J. Agin, "Representation and description of curved objects," Thesis (CS), Stanford Univ., Stanford, CA, AIM-173, Oct. 1972.
[17] H. G. Barrow and R. J. Popplestone, "Relational descriptions in picture processing," in *Machine Intelligence*, vol. 6, B. Meltzer and D. Mitchie Eds., Edinburgh, Scotland: University Press, 1970, pp. 377-396.
[18] K. Price, "Change detection and analysis in multi-spectral images," Thesis (CS), Carnegie-Mellon Univ., Pittsburgh, PA, 1976.
[19] A. Rosenfeld and M. Thurston, "Edge and curve detection for visual scene analysis," *IEEE Trans. Comput.*, vol. C-20, May 1971.
[20] A. Rosenfeld *et al.*, "Edge and curve detection: Further experiments," *IEEE Trans. Comput.*, vol. C-21, pp. 677-715, July 1972.
[21] A. Martelli, "Edge detection using heuristic methods," *Comput. Graphics and Image Processing*, vol. 1, pp. 169-182, 1972.
[22] M. J. Hueckel, "A local visual operator which recognizes edges and lines," *J. ACM*, vol. 20, no. 4, pp. 634-647, Oct. 1973.
[23] T. Pavlidis, "Segmentation of pictures and maps through functional approximation," *Comput. Graphics and Image Processing*, vol. 1, pp. 360-372, 1972.
[24] S. L. Horowitz and T. Pavlidis, "Picture segmentation by a tree traversal algorithm," *J. ACM*, vol. 23, no. 4, pp. 368-388, Apr. 1976.
[25] T. V. Robertson *et al.*, "Multispectral image partitioning," Purdue Univ., Lafayette, IN, TR-EE 73-26, Aug. 1973.
[26] A. Klinger, "Data Structures and Pattern Recognition," in *Proc. 1st Int. Joint Conf. on Pattern Recognition* (Washington, DC), pp. 497-498, Oct. 1973.
[27] C. R. Brice and C. L. Fennema, "Scene analysis using regions," *Artif. Intel. J.*, vol. 1, pp. 205-226, Fall 1970.
[28] T. Pavlidis, "Linguistic analysis of waveforms," in *Software Engineering*, J. Tou, Ed. New York: Academic Press, 1971, pp. 203-205.
[29] R. M. Haralick and G. L. Kelly, "Pattern recognition with measurement space and spatial clustering for multiple images," *Proc. IEEE*, vol. 57, pp. 654-665, Apr. 1969.
[30] R. M. Haralick *et al.*, "Textural features for image classification," *IEEE Trans. Syst., Man, and Cybern.*, vol. SMC-3, pp. 610-621, Nov. 1973.
[31] R. M. Haralick and I. Dinstein, "A spatial clustering procedure for multi-image data," *IEEE Trans. Circuits Syst.*, vol. CAS-22, May 1975.
[32] R. Ohlander, "Analysis of natural scenes," Thesis (CS), Carnegie-Mellon Univ., Pittsburgh, PA, June 1975.
[33] H. C. Andrews, *Introduction to Mathematical Techniques in Pattern Recognition*. New York: Wiley, 1972.
[34] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.
[35] R. C. Tyron, *Cluster Analysis*. Ann Arbor, MI: Edwards, 1939.
[36] G. H. Ball, "A comparison of some cluster-seeking techniques," Stanford Res. Inst., Stanford, CA, Tech. Rep. RADC-TR-66, 514, Nov. 1966.
[37] R. E. Bonner, "A logical pattern recognition program," *IBM J. Res. Develop.*, July 1962.
[38] R. E. Bonner, "On Some Clustering Techniques," *IBM J. Res. Develop.*, Jan. 1964.
[39] G. H. Ball and D. J. Hall, "ISODATA, a novel method of data analysis and pattern classification," Stanford Res. Inst., Menlo Park, CA, April 1965.
[40] G. Nagy, "State of the art in pattern recognition," *Proc. IEEE*, vol. 56, May 1968.
[41] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. New York: Academic Press, 1972.
[42] T. Kailath, "The Divergence and Bhattacharyya Distance Measure in Signal Detection," *IEEE Trans. Commun. Technol.*, vol. 15, pp. 52-60, 1967.
[43] T. L. Henderson and D. G. Lainiotis, "Comments on Linear Feature Extraction," *IEEE Trans. Inform Theory*, vol. IT-15, pp. 728-730, Nov. 1969.
[44] K. S. Fu, *Sequential Methods in Pattern Recognition and Machine Learning*. New York: Academic Press, 1968.
[45] R. M. Haralick, K. Shannugam, and I. Dinstein, "Textural Features for Imaging Classification," *IEEE Trans. Syst. Man, Cybern.*, vol. SMC-3, Nov. 1973, pp. 610-621.
[46] G. R. Nudd, "Progress on the Sobel CCD Chip and Circuit II," Semi-annual Technical Report, Report No. 740, Image Processing Institute, University of Southern California, Mar. 1977.
[47] K. S. Fu and P. H. Swain, "On Syntactic Pattern Recognition," *Software Engineering*, vol. 2, J. T. Tou, Ed. New York: Academic Press, 1971.

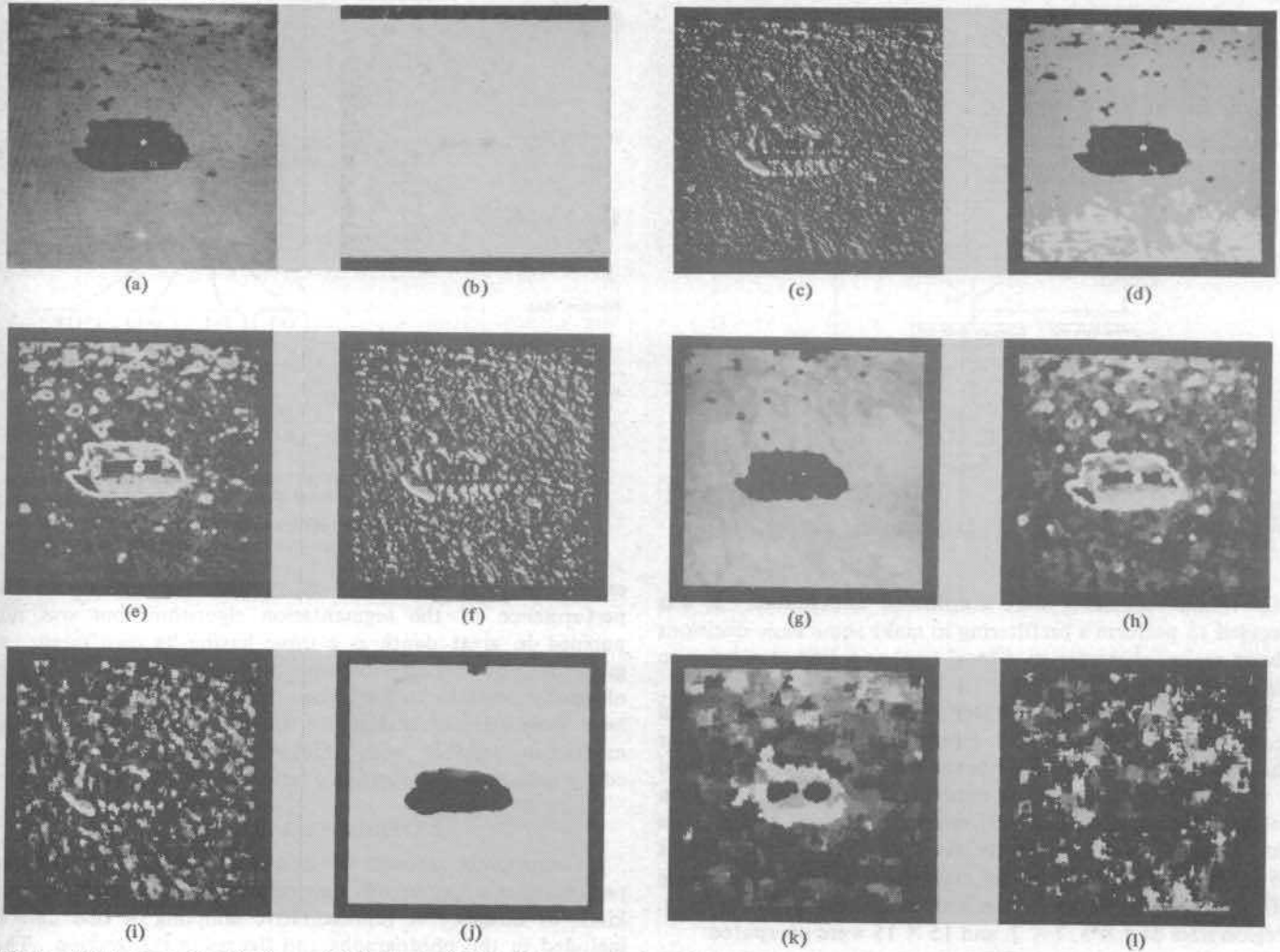


Fig. 9. APC image original features. (a) Original. (b) Sobel log magnitude. (c) Sobel phase product. (d) Original, mode filtered 3 × 3. (e) Log magnitude, mode filtered 3 × 3. (f) Phase product, mode filtered 3 × 3. (g) Original, mode filtered 7 × 7. (h) Log magnitude, mode filtered 7 × 7. (i) Phase product, mode filtered 7 × 7. (j) Original, mode filtered 15 × 15. (k) Log magnitude, mode filtered 15 × 15. (l) Phase product, mode filtered 15 × 15.

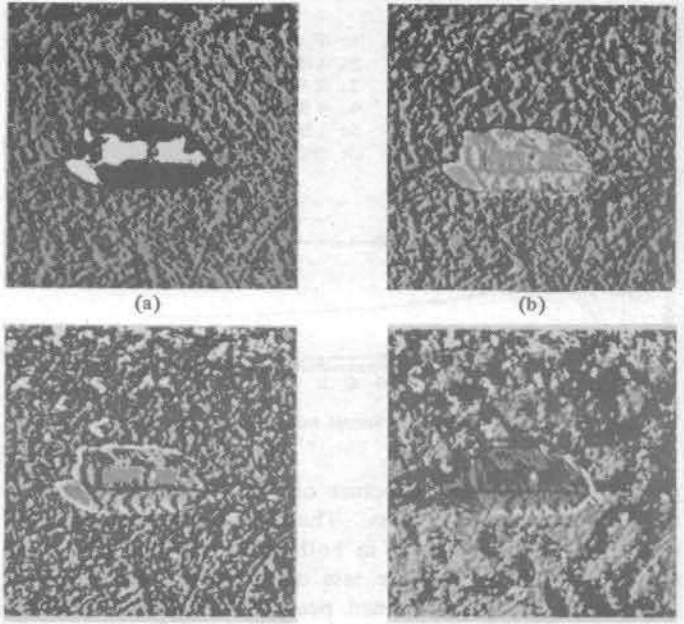


Fig. 10. Twelve nonreduced correlated features. (a) Seven regions. (b) Eight regions. (c) Nine regions (best number of regions). (d) Ten regions.

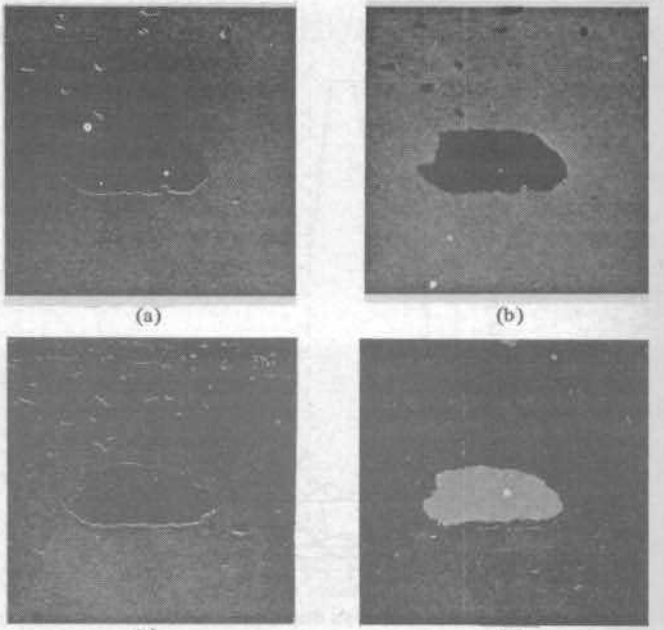
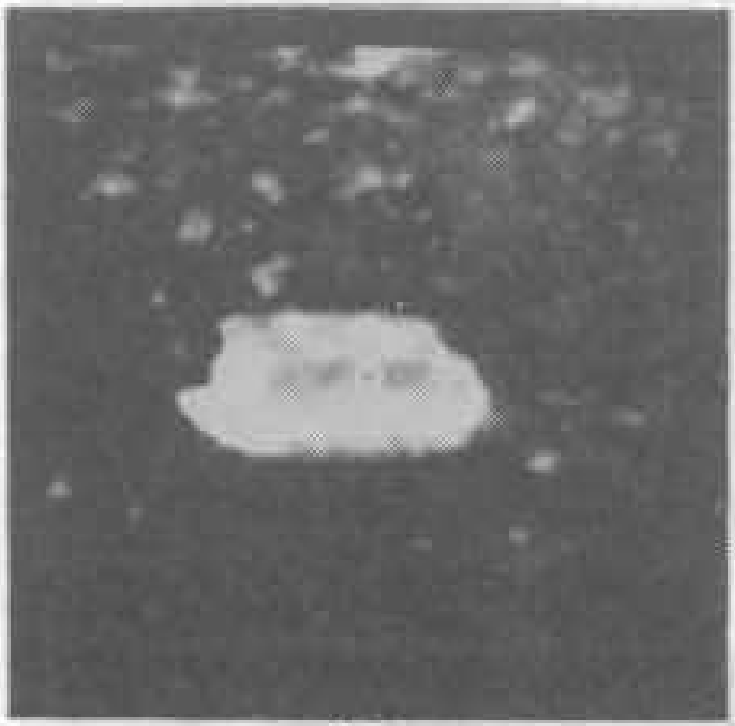
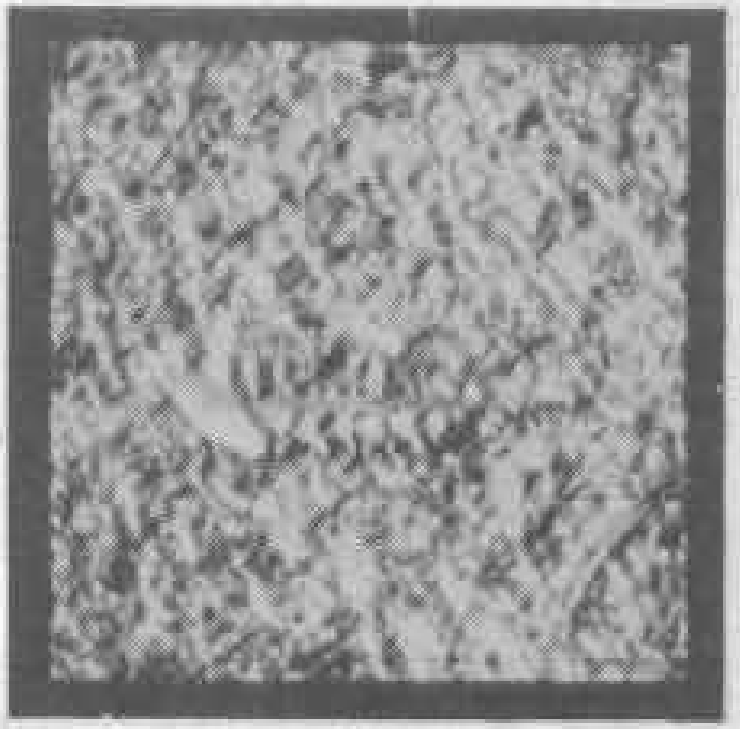


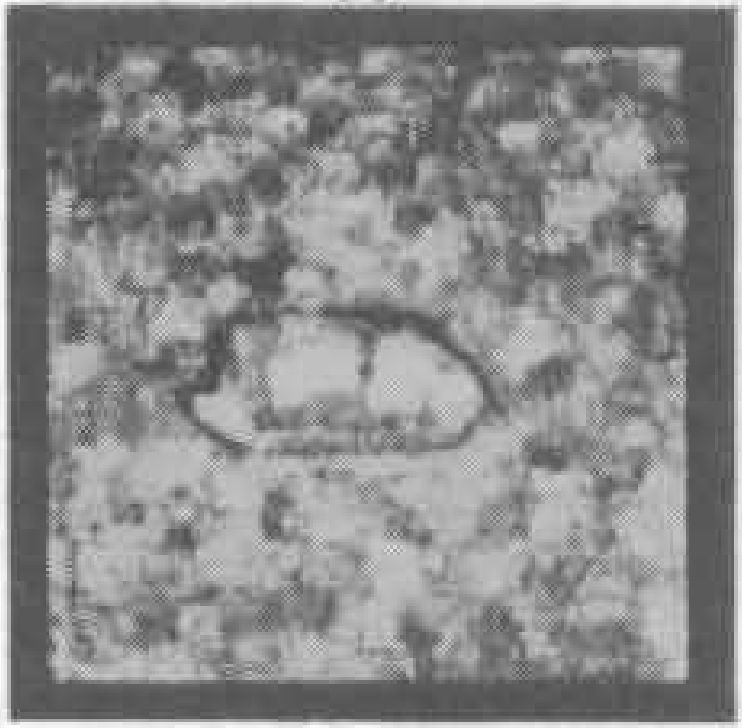
Fig. 12. Four induced correlated features. (a) Two regions (best number of regions). (b) Three regions. (c) Four regions. (d) Five regions.



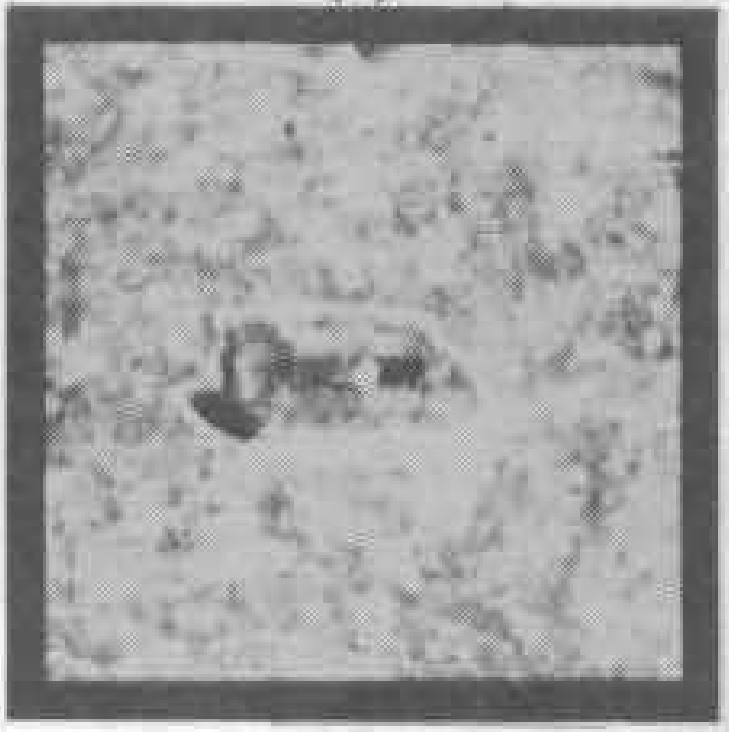
(a)



(b)



(c)



(d)

Fig. 13. APC image rotated features. (a) Rotated feature 1. (b) Rotated feature 2. (c) Rotated feature 3. (d) Rotated feature 4.

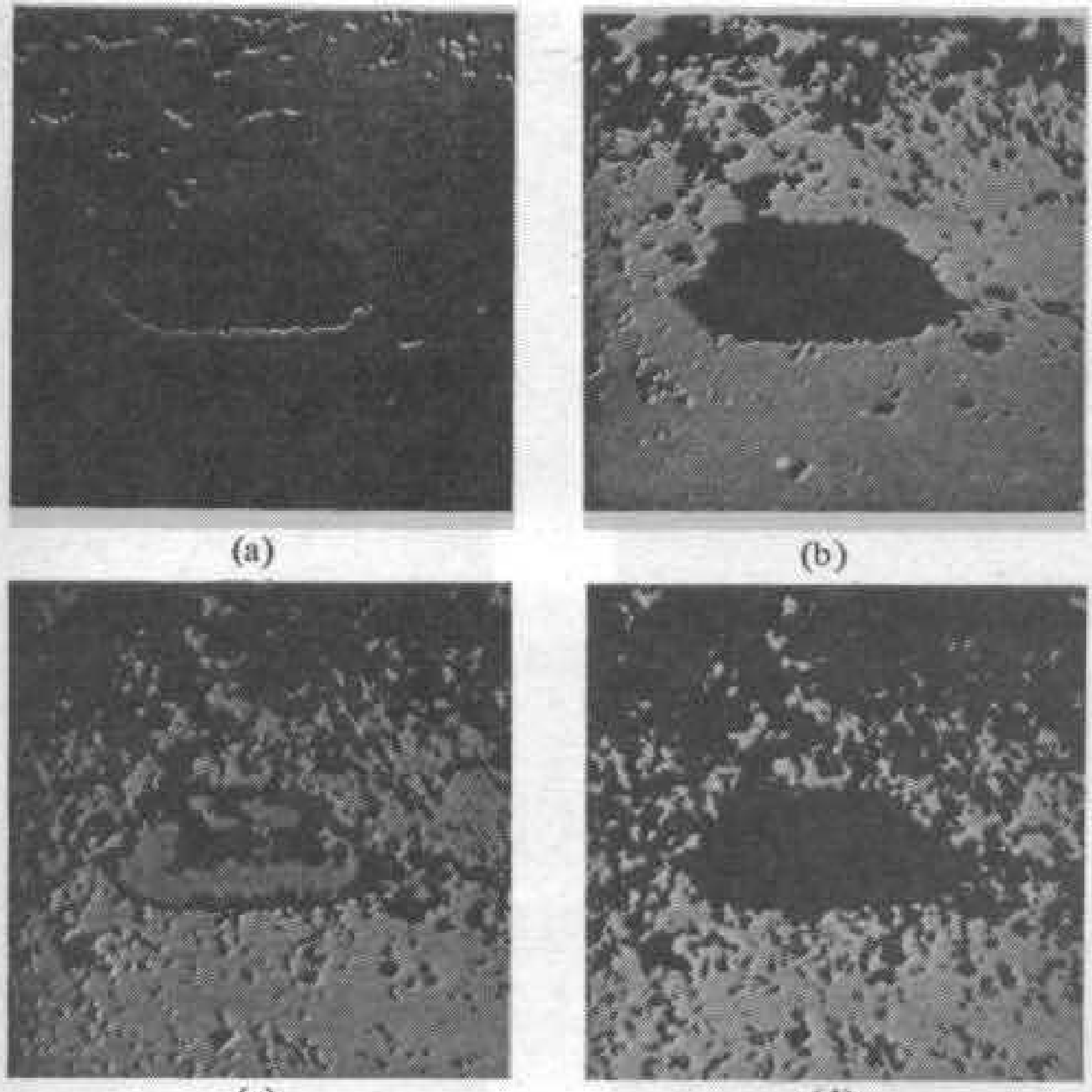
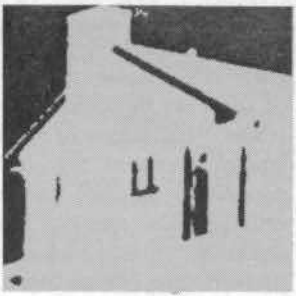
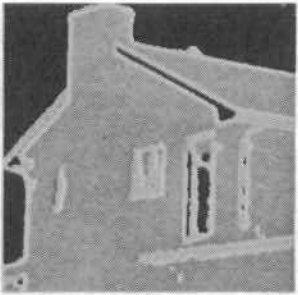


Fig. 15. Single best decorrelated feature. (a) Two regions (best number of regions). (b) Three regions. (c) Four regions. (d) Five regions.

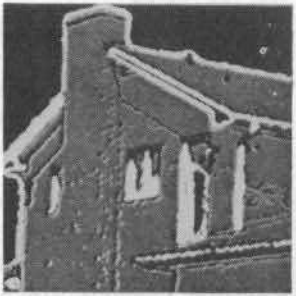
(a)



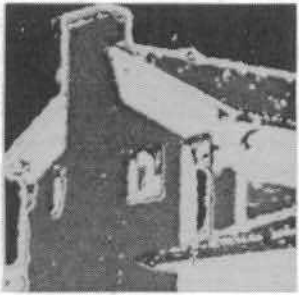
(b)



(c)



(d)



(e)

(f)

Fig. 17. Segmentation of house picture. (a) House original. (b) Two regions (best number of regions). (c) Three regions. (d) Four regions. (e) Five regions. (f) Six regions.

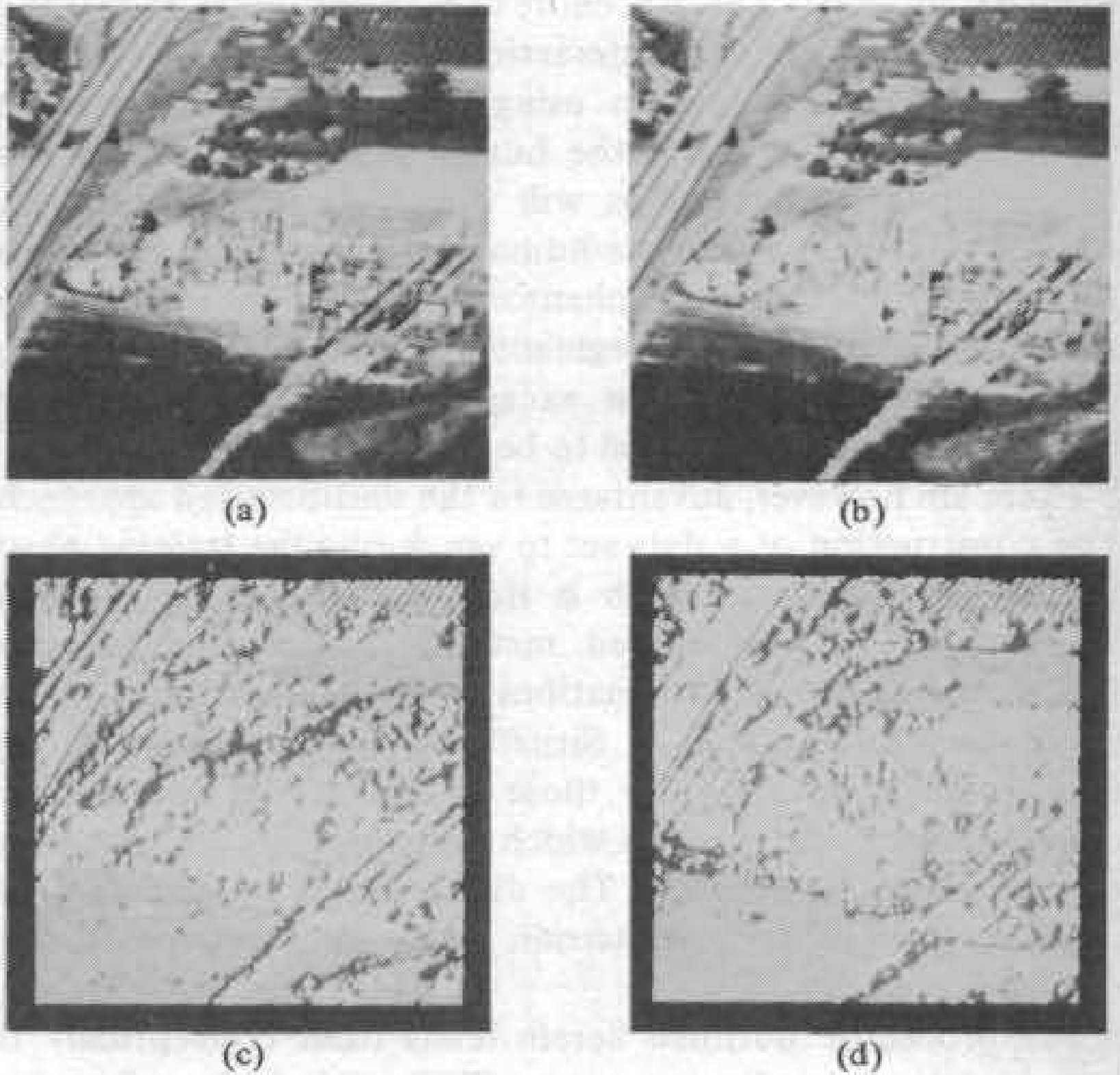


Fig. 20. Motion picture results. (a) Original—Frame 1. (b) Original—Frame 5. (c) Segmentation—Frame 1 (four clusters). (d) Segmentation—Frame 5 (four clusters).