ADA044452

ENGINEERING

USCIPI Report 750

UNIVERSITY OF SOUTHERN CALIFORNIA

IMAGE SEGMENTATION BY CLUSTERING

by

Guy Barrett Coleman

July 1977

Image Processing Institute

University of Southern California

University Park

Los Angeles, California 90007

Sponsored by

Advanced Research Projects Agency

Contract No. F-33615-76-C-1203

ARPA Order No. 3119

DDC
SEP 22 1977
C

ORIGINAL CONTAINS COLOR PLATES: ALL DDC REPRODUCTIONS WILL BE IN BLACK AND WHITE.

AD No.
DDC FILE COPY

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

IPI

IMAGE PROCESSING INSTITUTE

IMAGE SEGMENTATION BY CLUSTERING

by

Guy Barrett Coleman

July 1977

Image Processing Institute
University of Southern California
University Park
Los Angeles, California 90007

This research was supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by the Wright Patterson Air Force Base under Contract No. F-33615-76-C-1203, ARPA Order No. 3119.

The views and conclusions in this document are those of the author and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Advanced Research Projects Agency or the U.S. Government.

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)
Image Processing Institute
University of Southern California, University Park
Los Angeles, California 90007

2a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

2b. GROUP

3. REPORT TITLE
IMAGE SEGMENTATION BY CLUSTERING

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
Technical Report, July 1977

5. AUTHOR(S) (First name, middle initial, last name)
Guy Barrett Coleman

6. REPORT DATE
July 1977

7a. TOTAL NO. OF PAGES
143

7b. NO. OF REFS
57

8a. CONTRACT OR GRANT NO.
F-33615-76-C-1203

b. PROJECT NO.
ARPA Order No. 3119

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)
USCIPI Report 750

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT
Approved for release: distribution unlimited.

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY
Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, Virginia 22209

13. ABSTRACT

The segmentation of imagery into homogeneous regions using digital techniques has been a goal of researchers for the past several years. Pattern recognition approaches using mathematical models have achieved results which are only partially satisfactory. The large dimension of the pattern space and the quantity of data involved in the digital representation of images are in part responsible for the limited applicability of these approaches. Other shortcomings are related to the demands for data with which to train the classifier.

Approaches based on linguistic models have also been tried, again with results which are partially satisfactory. The most serious shortcomings are related to the performance of these approaches in the presence of noise, a phenomenon with which man has learned to function effectively.

This dissertation describes a procedure for segmenting imagery using digital techniques and is based on the mathematical model. The classifer does not require training prototypes, that is, it operates in an "unsupervised" mode. The procedure is general in that the features most useful for the particular image to be segmented are selected by the algorithm. The algorithm operates without any human interaction.

The features used are based on brightness and texture in regions centered on every picture element in the image. To perform an elementary pre-classification of local regions, a filter based on the mode of the local area histogram is proposed and used in segmenting images.

The basic procedure is a K-means clustering algorithm which converges to a local minimum in the average squared inter-cluster distance for a specified

DD FORM 1 NOV 65 1473

391 111

UNCLASSIFIED
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT

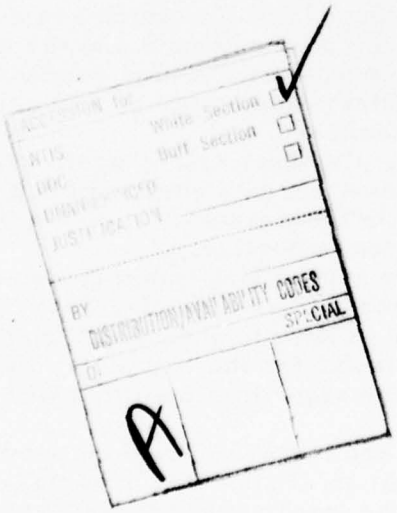
number of clusters. The algorithm iterates on the number of clusters, evaluating the clustering based on a parameter of clustering quality. The parameter proposed is a product of between and within cluster scatter measures, which achieves a maximum value that is postulated to represent an intrinsic number of clusters in the data.

It has been impossible in the past to compare different segmentations of the same image. A comparison measure based on the joint histogram of the two segmentations is proposed and examples of its use are presented.

It is within the state of the art to adapt the segmentation procedure described herein to operate in hardware at television rates. A functional diagram of such a system is presented, and estimates of the required capacities are given.

****

Key Words: Scene Analysis, Pattern Recognition, Image Segmentation, Clustering, Digital Image Processing.



ABSTRACT

The segmentation of imagery into homogeneous regions using digital techniques has been a goal of researchers for the past several years. Pattern recognition approaches using mathematical models have achieved results which are only partially satisfactory. The large dimension of the pattern space and the quantity of data involved in the digital representation of images are in part responsible for the limited applicability of these approaches. Other shortcomings are related to the demands for data with which to train the classifier.

Approaches based on linguistic models have also been tried, again with results which are partially satisfactory. The most serious shortcomings are related to the performance of these approaches in the presence of noise, a phenomenon with which man has learned to function effectively.

This dissertation describes a procedure for segmenting imagery using digital techniques and is based on the mathematical model. The classifier does not require training prototypes, that is, it operates in an "unsupervised" mode. The procedure is general in that the features most useful for the particular image to be segmented are selected by the algorithm. The algorithm operates without any human interaction.

The features used are based on brightness and texture in regions centered on every picture element in the image. To perform an elementary pre-classification of local regions, a filter based on the mode of the local area histogram is proposed and used in segmenting images.

The basic procedure is a K-means clustering algorithm which converges to a local minimum in the average squared inter-cluster 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.

It has been impossible in the past to compare different segmentations of the same image. A comparison measure based on the joint histogram of the two segmentations is proposed and examples of its use are presented.

It is within the state of the art to adapt the segmentation procedure described herein to operate in hardware at television rates. A functional diagram of such a system is presented, and estimates of the required capacities are given.

ACKNOWLEDGEMENT

I would like to express grateful appreciation to the members of my dissertation committee whose support and critique were invaluable. I feel special gratitude toward Professor Harry C. Andrews, who served as Chairman and whose guidance, support and enthusiasm made this work possible. I would also like to express appreciation to Mr. Ray Schmidt and the staff of the Image Processing Laboratory, without whose indulgence, this work might still be in progress.

TABLE OF CONTENTS

Chapter		Page
1	INTRODUCTION	9
	1.1 Research Objectives	9
	1.2 Organization of the Dissertation	10
2	IMAGE UNDERSTANDING SYSTEMS	13
	2.1 Image Understanding System Description	13
	2.2 Other Image Understanding System Models	17
	2.3 Image Segmentation Approaches	17
3	PATTERN RECOGNITION, UNSUPERVISED LEARNING, AND CLUSTERING	22
	3.1 Artificial Intelligence Approaches	22
	3.2 Mathematical Models	23
	3.3 Statistical Decision Theory Applications	27
	3.4 Supervised Pattern Recognition	28
	3.5 Unsupervised Pattern Recognition	28
	3.6 Clustering	31
	3.7 Clustering Quality Measures	35
	3.8 Feature Selection	46
	3.9 Segmentation Comparison Measure	50
4	IMAGE SEGMENTATION BY CLUSTERING - AN APPROACH	56
	4.1 Overall Approach	56

	4.2 Feature Rotation	59
	4.3 Clustering Algorithm	61
	4.4 Feature Computation	65
5	EXPERIMENTAL RESULTS	73
	5.1 APC Image Results	73
	5.2 Aerial Image Results	79
	5.3 House and Multi-Spectral Images Results	79
	5.4 Computer Time Required	81
	5.5 Comparison Measure Results	83
6	REAL TIME IMPLEMENTATION	117
	6.1 Feature Computation	117
	6.2 Segmentor	119
	6.3 Mean Computer	119
	6.4 Cluster Data Computer	120
	6.5 Preliminary Functional Requirements	123
	6.6 Motion Picture Segmentation Results	124
7	CONCLUSIONS	128

LIST OF FIGURES

Figure		Page
2.1	Image Understanding System	14
3.1	Classical Mathematical Pattern Recognition Model	24
3.2	Reformulated Pattern Recognition Model	29
3.3	Clustering Quality Measure	36
3.4	Alternate Clustering Quality Measure	40
3.5	Probability Weighting Function	42
3.6	Uniform Weighting Function	43
3.7	Bhattacharyya Measure Distribution	49
3.8	Comparison Measure Example	53
4.1	General Block Diagram	57
4.2	Flow Diagram of Image Segmentation Algorithm	60
4.3	Clustering Algorithm Flow Chart	63
5.1	APC Image Original Features	85
5.2	Segmentations - 12 Original APC Features	88
5.3	APC Image Rotated Features	90
5.4	Segmentations - 12 APC Rotated Features	93
5.5	Aerial Image Results	96
5.6	House Image Results	97
5.7	Multispectral Images Results	99

Figure		Page
5.8	Average Bhattacharyya Distance vs. Number of Clusters	100
5.9	Probability Weighted Product vs. Number of Clusters	102
6.1	Real Time Segmentation System	118
6.2	Mean Computer	121
6.3	Motion Picture Results	125

LIST OF TABLES

Table		Page
4.1	Feature Set Descriptions	70
5.1	Covariance, Eigenvalues and Eigenvectors	106
5.2	Eigenvalues vs. Bhattacharyya Distances	113
5.3	Computer Time	115
5.4	Comparison Measure Results	116
6.1	Automatic Real Time Segmentor Functional Requirements	123
6.2	Motion Picture Means	126

Chapter 1

INTRODUCTION

This dissertation 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.

For the purposes of this report, a "segmented image" is defined to be an image wherein each picture element (pixel) in the image is assigned a number corresponding to the index number of the segment to which it belongs.

1.1 Research Objectives

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 algorithm does not use context-related information such as shape and relative position. Either monochromatic or color imagery may be segmented utilizing the same algorithm, with a somewhat expanded feature set for the color imagery to take advantage of the multispectral information.

In the past, most evaluations of image understanding systems have been performed by subjective judgement. It is not completely clear how, given two different segmentations of the same image, one segmentation is judged better than the other. Ultimately, the value of an image segmentation system will lie in its potential usefulness and in its ability to segment imagery in a manner that to some degree emulates human perception. Nevertheless, it would be useful to be able to numerically compare two different segmentations of the same image. To that end, a comparison measure is proposed and examples of compared segmentations are given.

1.2 Organization of the Dissertation

The second chapter provides an overview of image understanding systems in general and approaches to image segmentation in the past. The approach taken in this dissertation is the only segmentation procedure based entirely on clustering that has been reported in the literature. 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 third chapter 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. Previous work in each of the areas brought to bear on the problem are outlined and their relationship to the work contained herein is discussed.

Chapter four is a detailed description of the approach taken. Block diagrams and flow charts 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 combinations of these features are justified, based on results obtained. To obtain an elementary pre-classification of region character, a novel 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 detail in Chapter 5. In some cases, images were segmented with more than one feature set in an attempt to improve performance. Segmentations of images produced under various conditions are compared using a comparison measure developed for that purpose.

The particular approach taken to image segmentation in this dissertation lends itself to "real time" implementation, that is, it is possible to construct electronic hardware to segment images at television rates. Chapter 6 is a functional description of a real time implementation of the algorithm which was programmed on a general purpose digital computer. The procedure described in Chapter 4 was

used to segment two frames of a motion picture. The segmentations are included to demonstrate the ability of the algorithm to produce essentially equivalent segmentations of spatially non-stationary images.

Finally, Chapter 7 draws conclusions from the results and provides directions in which further research is judged necessary.

Chapter 2

IMAGE UNDERSTANDING SYSTEMS

2.1 Image Understanding System Description

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 representing 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 [2-1] that the successive levels of abstraction require that the higher levels of the system interact with the lower levels, based on the current descriptions. This processing approach is called "heterarchical." The image understanding system is therefore conceptualized as having a hierarchy of processing levels, as shown in Figure 2.1.

The primitive description level extracts local features that are

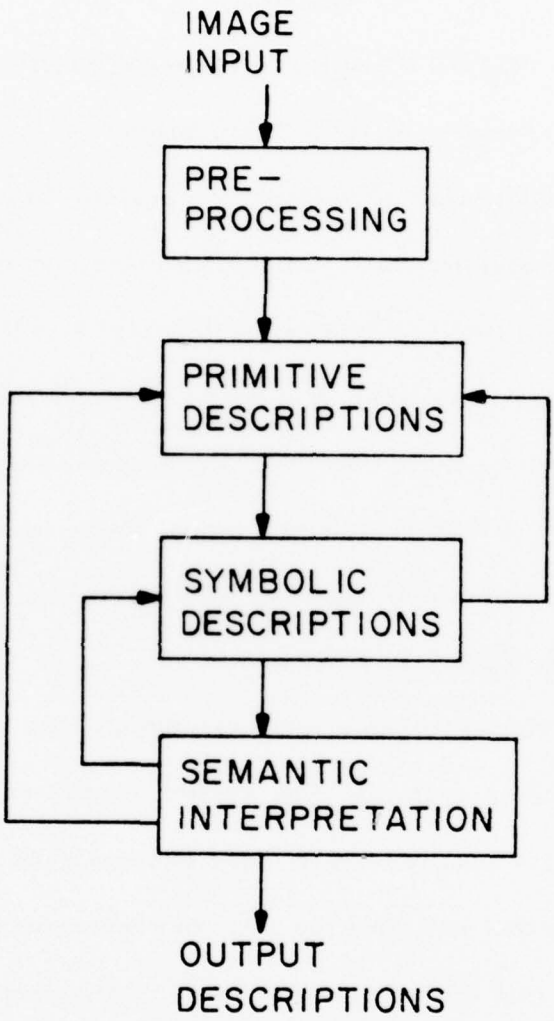


Figure 2.1. Image Understanding System

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

A large percentage of the difficulty with current schemes for image understanding can be related to the lack of understanding of the human perceptual system. Preliminary work towards relating texture features to human perception has been performed by Thompson [2-2]. He used a rank order experiment to define a combination of texture features. This combination forms a perceptual distance function which correlates to some degree with human perception. Further work along these same lines on other features combined with greater understanding of the human perceptual system will greatly improve the operation of image understanding systems.

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 ini-

tial 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.

The decisions about dividing the scene into similar or homogeneous regions are made at this level of the image understanding system. The notion of "similar" is a purely defined concept. Consider the problem of grouping automobiles, busses and airplanes. These three items are different in obvious respects, and in certain circumstances, it would be legitimate to group them separately. It is also true that they are all vehicles used for transportation, and they could be grouped together. The transportation specialist might group busses and airplanes together as representing forms of mass transit, whereas the average person might group busses and automobiles together because they are both land vehicles. For these reasons, it is obvious that any grouping of data must be performed with a specific intent. 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.

2.2 Other Image Understanding System Models

A number of somewhat different models have been proposed other than the model suggested here. 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 [2-3] and are typified by locating telephones in an indoor office scene. Other examples of this approach are the location of specific objects in x-ray radiographs [2-4].

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 [2-5, 2-6].

2.3 Image Segmentation Approaches

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. The image segmentation procedure must be sufficiently general that it will operate satisfactorily over a wide range of image types and within a wide range of possible implementations of image understanding systems. The segmentation procedure must be reasonably efficient in terms of computer time and storage required in order not to require unnecessary resources to implement. Many previous approaches have required several hours of computer time to implement. It has been observed that "It is usually possible to solve a difficult problem in a difficult manner by brute force and ignorance. However, real advances are made by recognizing difficulties and avoiding them." [2-7]

At the current state of the art, it is fashionable to invoke the "other level of the system" argument when the difficult interfaces between the image understanding system levels are encountered. This argument inevitably insists that some (usually the most challenging) aspect of the problem is that which the higher (lower) level of the system will solve. Not invoking this argument requires that the gross overall image segmentation be performed with some degree of autonomy, in other words, it must decide on a segmentation without close supervision from the higher levels of the system. If more or less detail about a particular region is desired, the higher level of the system can either merge regions or direct that regions be further segmented. The number of regions with which the higher levels of the

system must deal must be kept to a minimum, to permit reasonable implementation of the higher system levels.

Segmentation of images into homogeneous regions has been a goal of image understanding researchers for many years. Beginning with simple block-like objects and the work of Roberts [2-8], image segmentors evolved into those attempting to segment natural scenes. Roberts' work used intensity to detect object boundaries for further manipulation. Later efforts [2-9, 2-10, 2-11, 2-12, 2-13] manipulated the line drawings in different ways, but extracted these line drawings as a pre-processing 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 [2-14, 2-15, 2-16, 2-17, 2-18]. Some of these approaches were interactive, that is, a human operator provided guidance to the computer to direct the segmentation [2-19]. An excellent description of each of these segmentation approaches and the context in which they were applied is contained in [2-20].

Common to all of these approaches is the extraction of line drawings by varying means. Thus the region boundaries represent the segmentation of the image. In some of these approaches, the edges are

sought directly by edge detection [2-21, 2-22, 2-23, 2-24] or functional approximation [2-25, 2-26] . In other approaches, the regions are detected first and the boundaries determined later. There are two general approaches to region detection. The first is a top-down approach wherein the picture is segmented into progressively smaller regions until certain criteria are satisfied. Examples of these approaches are found in [2-27, 2-28] . The second approach is a bottom up approach wherein the picture is divided into a large number of small regions, possibly as small as one pixel. These regions are successively merged to form larger regions. Examples of this approach are given in [2-29, 2-30] .

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 [2-31] . This procedure used a modified linking or "nearest neighbor" rule to form the clusters on multi-image data. The procedure uses two arbitrary thresholds or parameters, the maximum number of clusters and a probability threshold parameter. The histogram is computed and peaks are isolated to accelerate the location of cluster centers. Naturally, the performance depends on the parameters selected.

Further work has been performed using textural features and a classifier operating in the supervised mode [2-32] . The supervised mode requires that the cluster center be determined by "training."

that is, samples whose classification is known are used to identify the cluster centers.

Clustering has also been applied to images segmented by a edge detection procedure [2-33]. The procedure used was: 1) compute a gradient image. 2) threshold the gradient image. 3) clean the thresholded gradient image. 4) label connected regions in the cleaned image. 5) cluster the labeled, connected regions. Thus, clustering is used to merge and identify segmentations after they are formed. A number of thresholds are required in forming and cleaning the gradient image and labeling the connected regions. This procedure is a combination of former types, utilizing both edge detection and region detection to form the segmentation.

An additional bottom-up approach to image segmentation is described by Ohlander [2-34]. This procedure uses histogram analysis to successively delete points contained in feature histogram peaks. The feature histograms are then recomputed and the process repeated. The initial system required considerable human interaction involving peak finding and selection, selection of connected regions and data base manipulation. Later work [2-20] refined and accelerated the procedure based on a priori knowledge about feature usefulness and sub-region analysis.

Chapter 3

PATTERN RECOGNITION, UNSUPERVISED LEARNING AND CLUSTERING

A large body of information and techniques 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 consists of theory and techniques from statistical data analysis and communication theory. The second category contains knowledge that is most closely related to computer artificial intelligence.

3.1 Artificial Intelligence Approaches

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 [3-1]. 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. Nevertheless, context is easily visualized in such an approach as additional constraints on the relationships between the primitive elements.

3.2 Mathematical Models

The first results obtained in the general discipline now called pattern recognition were based on mathematical models. These models [3-2] assume that a sensor or series of sensors measure physical quantities about an object in the real world as shown in Figure 3.1. 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 multi-spectral measurements about the physical world. At each point (x, y) the output of the i^{th} sensor at time t is

$$P_i(x, y, t) = \int_0^\infty F(x, y, t, \lambda) V_i(\lambda) d\lambda \qquad (3\text{-}1)$$

where $V_i(\lambda)$ is the spectral response of the i^{th} sensor and $F(x, y, t, \lambda)$ is the brightness of the physical world sensed at point x, y in the sensory plane of the i^{th} sensor at time t [3-3]. Similar to the definition for one dimensional time signals, the time average of the image at (x, y) is

$$\left\langle P_i(x, y, t) \right\rangle = \lim_{T \to \infty} \left\{ \frac{1}{2T} \int_{-T}^{T} P_i(x, y, t) L(t) dt \right\} \qquad (3\text{-}2)$$

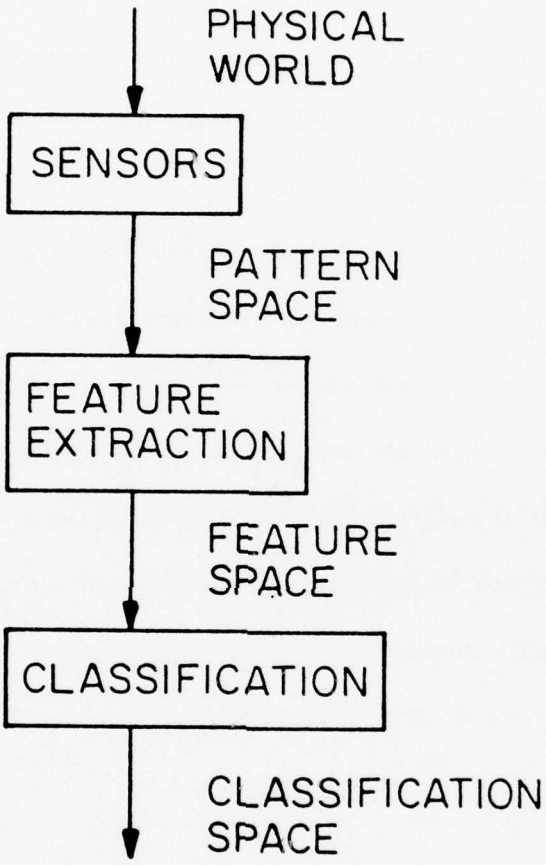


Figure 3.1. Classical Mathematical Pattern Recognition Model

where L(t) is a time weighting function. The image at this point is still in continuous form. For purposes of manipulation of the data by digital computer, the image must be converted into digital form by appropriately sampling the image. Thus the images are represented as real-valued functions of two spatial variables whose value at a point is related to the spectral and time integrals given in equations (3-1) and (3-2).

The pattern space consists of the spectral samples just described. The "first order" features of an image are its brightness (possibly in several spectral regions) and the x and y coordinates of the appropriate point. Each point is usually called a pixel (picture element). Other features, such as texture, are properties of a region [2-2]. Thus the feature extraction process may, in the case of images, enlarge the amount of data required to represent the image considerably. This increase causes the model to differ somewhat from the classical pattern recognition model where the feature extraction process usually performs a data compression by representing entire objects with a single vector of features.

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 $\overline{P}(x, y) = (P_1(x, y), P_2(x, y) \ldots P_n(x, y))$. Here n is the dimension of the feature

space and $P_i(x, y)$ is the value associated with dimension i at point (x, y).

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 based on the purpose of the classification.

The model just described often assigns the vectors by discriminant functions which are functionals of the feature vectors. Thus

$$g_k(\overline{P}(x, y)) > g_j(\overline{P}(x, y)) \qquad \text{for all } k = 1, 2, \ldots, n \ (k \neq j) \tag{3-3}$$

implies that the feature vector $\overline{P}$ is a member of class W_k. The discriminant functions usually assume the form of distance functions. The assumption is normally made that the feature space forms a metric space. The metric defined must satisfy the following conditions with respect to vectors $\overline{P}_1$, $\overline{P}_2$ and $\overline{P}_3$ in the space:

i) $$m(\overline{P}_1, \overline{P}_2) = m(\overline{P}_2, \overline{P}_1) \tag{3-4}$$

ii) $$m(\overline{P}_1, \overline{P}_2) \leq m(\overline{P}_1, \overline{P}_3) + m(\overline{P}_2, \overline{P}_3) \tag{3-5}$$

iii) $$m(\overline{P}_1, \overline{P}_2) \geq 0 \text{ and } m(\overline{P}_1, \overline{P}_2) = 0 \text{ iff } \overline{P}_1 = \overline{P}_2 . \tag{3-6}$$

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 theory and techniques currently being applied to pattern recognition approaches are equally applicable in the new space. It would be anticipated that the results obtained in this new space would more closely emulate the human perceptual system.

3.3 Statistical Decision Theory Applications

The extension of the model developed thus far to statistical decision theory is straightforward; each image is considered to be a sample function of a two dimensional random process and each feature vector is a (vector) random variable. The classes defined by the discriminant functions become decision regions. Depending on how much is known (or is assumed) about the underlying statistics of the feature vectors, the many different forms of statistical decision theory can be applied.

These methods implicitly define the concept of "similarity." The

scaling of the feature functionals and the selection of the features to be used implicitly defines how the pattern classifier is to interpret similarity. This suggests that, in reality, the classical pattern recognition system is better represented by Figure 3.2.

3.4 Supervised Pattern Recognition

The determination of the discriminant functions 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.

The selection of the training prototypes, the selection of features and the cost weighting of the feature space effectively define to the classifier what is intended by "similar." Since similarity is highly context dependent, the best results that can be hoped for using this methodology are to classify based on non-context related criteria. This conceptualization points out the reason for some of the disappointing results in past efforts which are based on classical pattern recognition. Similarity is a defined concept and depends on context. The mathematical approach does not readily lend itself to the application of contextual criteria.

3.5 Unsupervised Pattern Recognition

Frequently, it is desirable to design a pattern classification

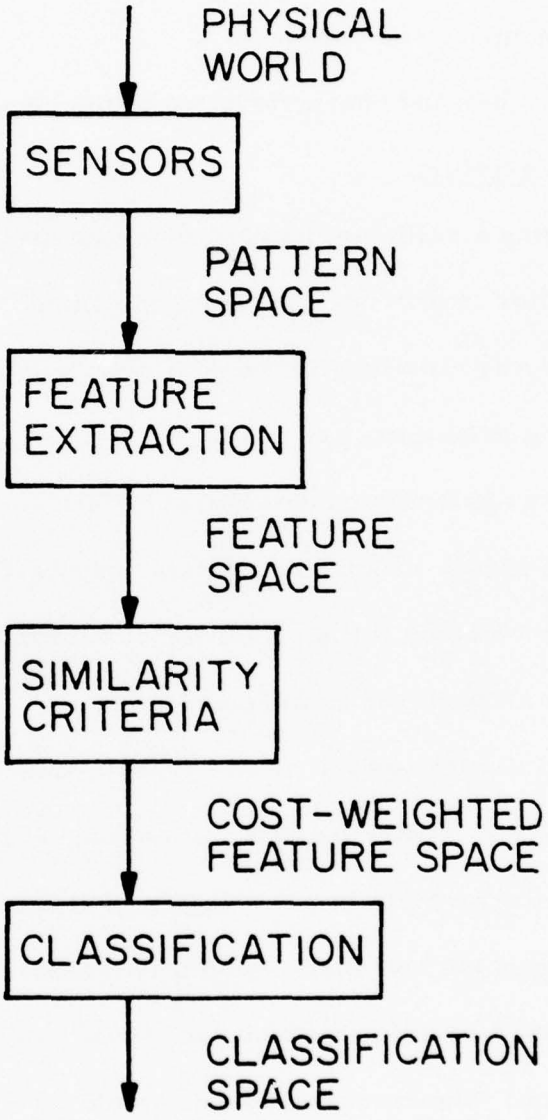


Figure 3.2 Reformulated Pattern Recognition Model

system without the use of training samples [3-4]. This is often called the "unsupervised" approach. There are a number of reasons for using the unsupervised approach.

i) The number and characteristics of the classes may not be known a-priori.

ii) Obtaining a sufficient number of prototypes to train the classifier is difficult and time consuming.

iii) Underlying structure in the data may be overlooked if training prototypes are used.

iv) In many applications, the characteristics of the patterns change slowly with time. Satisfactory performance can still be obtained if the classifier can track the changes using unsupervised techniques.

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. That this is so is obvious from the fact that a nonlinear transformation on the feature space could be defined which would reorganize the data in any desired form. The reorganized data would be equally as valid as the original data if nothing whatsoever is known about the data at the outset.

In the case of image related data, it is known a priori (or at least assumed) that the data represents low level perceptual differ-

ences. 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."

3.6 Clustering

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 [3-5].

It has been previously pointed out that the theoretical basis for unsupervised learning using clustering techniques is weak at best. It has been observed by Watanabe [3-6] that under certain conditions, there is no theoretical basis at all for clustering and unsupervised learning. His observations eminate from philosophical grounds and proceed as follows. Suppose every object to be clustered is described by n binary descriptions. No loss of generality is incurred since any object described by a finite number of finite precision

numbers can be described in this manner.

Thus, in theory, there are $N = 2^{2n}$ different possible descriptions if the set of descriptions includes all complementary descriptions. Hence, every object is described by an equal number of binary ones and zeros. It is also true that if there is at least one binary description that is different between two objects, then it follows that there are exactly N/4 binary positions in which the objects are described identically (a proof of this is contained in [3-7]). From this follows the somewhat startling conclusion that any two objects are as similar to one another as any other two objects when the degree of similarity is measured by the number of identical binary descriptions. It follows that there is no such thing as a class of similar objects in the world.

The above conclusion does not coincide with intuition or empirical observation. The apparent conflict is eliminated if it is assumed that some of the binary descriptions are more "important" than others. For example, binary descriptions that correspond to most significant bits are much more "important" than descriptions that correspond to least significant bits. As has been concluded previously, it is obviously of great importance in the classification process to define the concept of similarity in the selection and weighting (importance) of the descriptions or features.

There are any number of clustering procedures, each having its own peculiar characteristics. An extremely detailed discussion of numerous different clustering techniques is contained in [3-8]. There are, however, certain similarities between the various techniques that permit them to be categorized. Ball [3-8] defines 7 different cluster-seeking techniques for finding similar subsets in data. One of these he calls "clustering techniques" which are distinguished by the iterative sorting of the data using multiple cluster points until the cluster means "adequately" describe the data.

When it is anticipated that the clusters are tight and widely spaced the chain method [3-9], [3-10] may be used. The first data point is taken to be the starting point of the first cluster. If the distance to the second data point exceeds a threshold, the second data point becomes the starting point of a new cluster. The distance from each succeeding data point to every member of every cluster is computed, and the point is included in an existing cluster if its minimum distance is below a threshold. 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 in 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[3-11] and called ISODATA.

This procedure begins with an assumed number of clusters. The means are arbitrarily assigned, although the initial mean assignment will affect the clustering through 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 in the average within cluster distance.

Various methods have been proposed to use procedures of the type just described to find the "correct" or "best" number of clusters in the data. The algorithms developed by Ball and Hall [3-11] use merging and splitting to arrive at a final number of clusters. Thus clusters having variances that are larger than a threshold will be split and clusters whose means are separated by less than a threshold will be merged. One major shortcoming of this approach is that the merging and splitting thresholds must be established a priori. A procedure for determining these thresholds from the data has been developed by Fromm and Northouse [3-12].

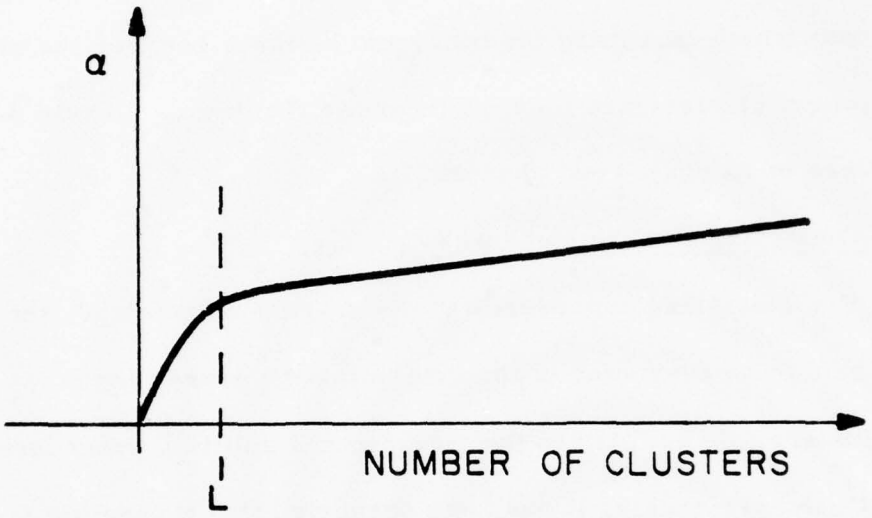
It has been observed by Nagy [3-13] that the procedures based on

minimization of a distance function, such as the procedure just described, are most appropriate for fairly isotropic clusters. The methods which maximize the minimum distance between the members of distinct clusters are most appropriate for dense, clearly separated, clusters of (perhaps) odd shapes.

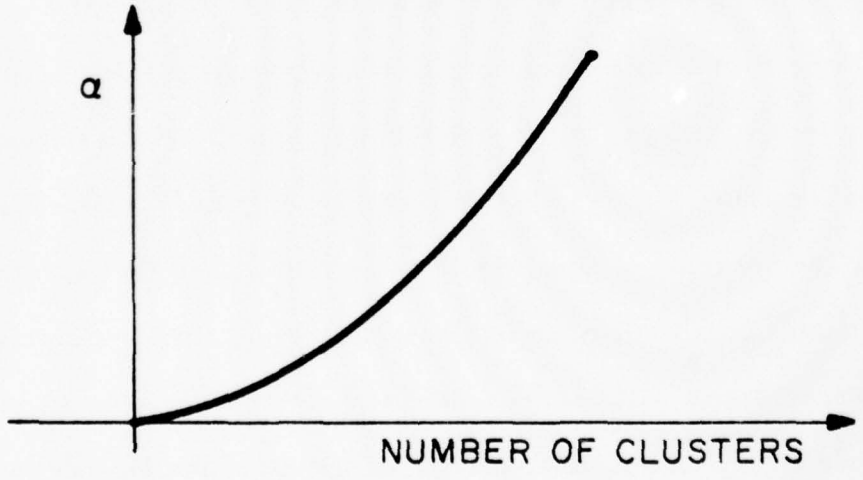
3.7 Clustering Quality Measures

For clustering procedures of the nearest means type, the key obstacle to be overcome is the determination of the "correct" number of clusters. In addition to the merging and splitting procedures mentioned previously, it has been suggested that a possible approach is to obtain a measure of the clustering quality represented by some parameter alpha [3-2]. This parameter might be expected to vary with the number of clusters as shown in Figure 3.3.

A number of measures have been proposed for alpha, one of which is the ratio of the between to within cluster scatter measure [3-14]. Thus, if it is true that there are intrinsic clusters in the data, the behavior of alpha would be as follows. If the initial number of clusters is less than the intrinsic number, L, the within cluster scatter measure will be large, and alpha will be small. As the number of clusters is increased, the within cluster scatter measure decreases rapidly, increasing alpha rapidly. When the intrinsic number of clusters (L) is reached, the rate at which the within



(a) GOOD CLUSTERING



(b) POOR CLUSTERING

Figure 3.3 Clustering Quality Measure

cluster scatter measure increases becomes small. The between cluster scatter measure changes very little after L clusters are reached, because the new cluster centers are close to the old cluster centers. Thus, alpha might be expected to behave as shown in Figure 3.3a.

The within-cluster and between-cluster measures are derived from within-cluster and between-cluster scatter matrices. These measures are intended to measure the separability of the data [3-14]. The within cluster scatter matrix is based on the scatter of the data about the cluster means and is given by (3-7)

$$S_w = \sum_i P(W_i) E\{(\bar{x}-M_i)(\bar{x}-M_i)^T | W_i\} \qquad (3-7)$$

where W_i is the i^{th} cluster, $P(W_i)$ is the relative frequency (or probability) of the data in that cluster, and M_i is the i^{th} cluster mean. $E\{\cdot\}$ denotes the expected value or average, and $(\cdot)^T$ denotes the transpose of the vector quantity in parentheses.

The between cluster scatter matrix can be defined in numerous ways, but for multi-cluster problems, (that is, problems having more than two clusters) the most straightforward definition is given by:

$$S_b = \sum_i P(W_i)(M_i-M_0)(M_i-M_0)^T \qquad (3-8)$$

M_0 is the overall expected vector of the entire mixture and is given by:

$$M_0 = E(\bar{x}) = \sum_i P(W_i)M_i \qquad (3\text{-}9)$$

The goal of using the scatter matrices is a measure of cluster separability. It is therefore necessary to derive a number from these matrices which is related to cluster separability. If this number is to behave like the parameter alpha discussed earlier, it should increase when the within cluster scatter decreases or the between cluster scatter increases. There are a number of ways of deriving such a number, among which are:

$$\begin{aligned} \alpha_1 &= \mathrm{tr}(S_w^{-1} S_b) \\ \alpha_2 &= \ell n\{|S_w+S_b|/|S_w|\} \\ \alpha_3 &= \mathrm{tr}\ S_b - \mu(\mathrm{tr}\ S_w - c) \\ \alpha_4 &= \mathrm{tr}\ S_b / \mathrm{tr}\ S_w \end{aligned} \qquad (3\text{-}10)$$

where $\mathrm{tr}(\cdot)$ indicates "trace" or sum of the diagonal elements of a matrix, and $|\cdot|$ denotes the determinant of the matrix. When α_3 is used, the procedure is to maximize $\mathrm{Tr}\ S_b$ subject to $\mathrm{Tr}\ S_w = c$. Here μ is the Lagrange multiplier and c is constant.

The terms α_1 and α_2 are invariant under any non-singular linear transformation. The terms α_3 and α_4, while easier to compute, depend on the coordinate system.

The use of the parameter alpha to measure the "goodness" of clustering requires that a knee in the alpha vs. number of clusters be

detected (see Figure 3.3). 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 beta which passes through a maximum at the "intrinsic" number of clusters (see Figure 3.4).

A candidate for this measure is

$$\beta = \text{Tr } S_w \cdot \text{Tr } S_b \tag{3-11}$$

When the number of clusters equals 1, $\text{Tr } S_w = \sigma^2$, the variance of the mixture, $\text{Tr } S_b = 0$ and $\beta = 0$. When the number of clusters equals N, where N is the total number of vectors in the mixture,

$$\text{Tr } S_w = 0 \quad \text{and} \quad \text{Tr } S_b = \sigma^2$$

Hence $\beta = 0$.

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 values somewhere in the interval. The ideal behavior for β 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 $\text{Tr } S_w$ and $\text{Tr } S_b$ to define clustering quality implicitly defines a weighting function $W(n_i)$ 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 (probability) of the data points in each cluster.

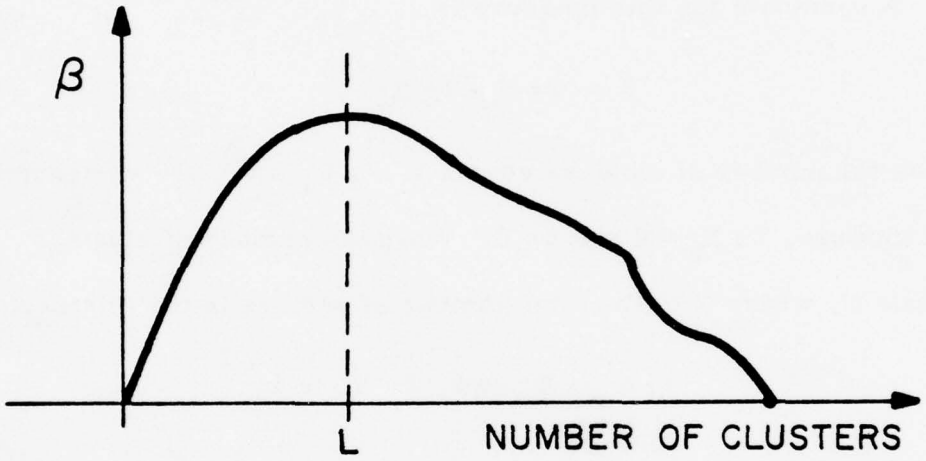


Figure 3.4 Alternate Clustering Quality Measure

The weighting function is depicted in Figure 3.5. Here, n_i is the number of points in the i^{th} cluster and N is the total number of points.

This weighting causes large clusters to have a greater effect on the clustering quality measure than small clusters. The probability weighting is correct if large clusters are indeed more important or if the clusters are of approximately the same size.

If small clusters are of equal importance as large clusters, the quality measure should be based on a weighting which gives equal weight to every cluster, regardless of size. This suggests a uniform weighting as shown in Figure 3.6. Reformulating (3-7) and (3-8) as

$$S_w = \frac{1}{I} \sum_i W(n_i) E\{(\bar{x}-M_i)(\bar{x}-M_i)^T\} \qquad (3\text{-}12)$$

$$S_b = \frac{1}{I} \sum_i W(n_i)(M_i-M_0)(M_i-M_0)^T \qquad (3\text{-}13)$$

maintains the property that the quality measure is not directly affected by the number of clusters.

Other situations can be postulated. If it is assumed that the clusters are normally distributed in size about an average value, n_m, a Gaussian weighting is suggested. If the criterion is to minimize the maximum cluster variance, a weighting which is 1 at the maximum diagonal element of S_w and zero elsewhere is correct. All

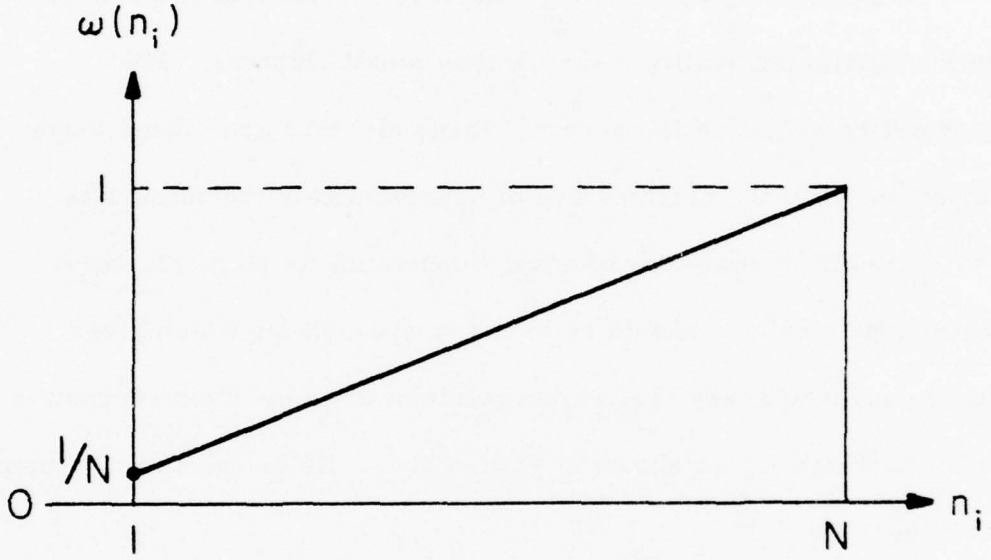


Figure 3.5 Probability Weighting Function

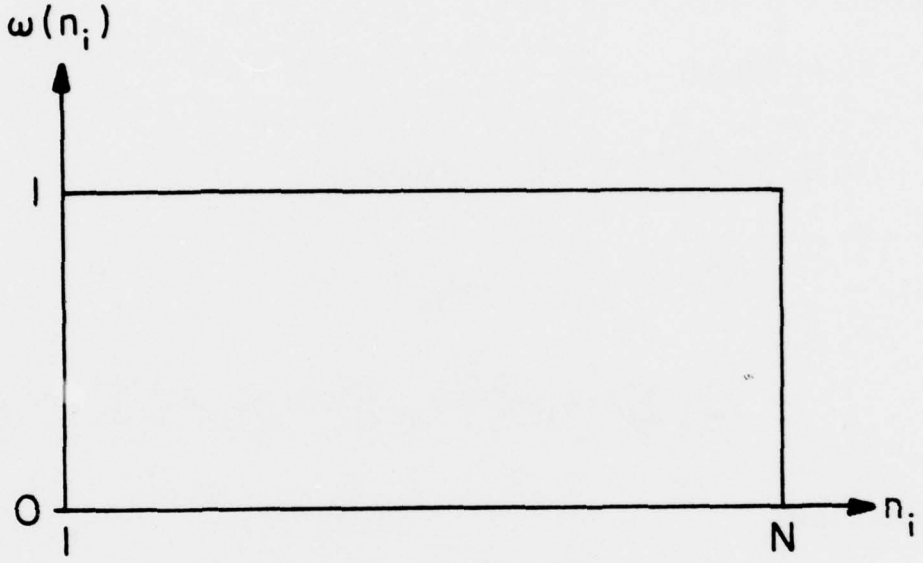


Figure 3.6. Uniform Weighting Function

of these weightings attempt to quantify in different ways what is meant by "good clustering."

An interesting relationship is true when the probability weighting function of Figure 3.5 is used.

By definition:

$$\text{Tr } S_b \triangleq \text{Tr} \sum_i P(W_i)(M_i - M_0)(M_i - M_0)^T \tag{3-14}$$

$$= \sum_i P(W_i) \sum_j (m_{ij} - m_{0j})^2 \tag{3-15}$$

$$= \sum_j \left[\sum_i P(W_i) m_{ij}^2 - 2m_{0j} \sum_i P(W_i) m_{ij} + m_{0j}^2 \sum_i P(W_i) \right] \tag{3-16}$$

Noting that:

$$\sum_i P(W_i) = 1 \tag{3-17}$$

and

$$\sum_i P(W_i) m_{ij} = m_{0j} \tag{3-18}$$

yields

$$\text{Tr } S_b = \sum_j \sum_i P(W_i) m_{ij}^2 - \sum_j m_{0j}^2 \tag{3-19}$$

additionally, by definition:

$$\text{Tr } S_w \triangleq \text{Tr} \sum P(W_i) E\{(\bar{x}_i - m_i)(\bar{x}_i - m_i)^T\} \tag{3-20}$$

$$= \sum_i P(W_i) \sum_j E\{(x_{ij} - m_{ij})^2\} \tag{3-21}$$

$$= \sum_j \sum_i P(W_i) E\{x_{ij}^2\} - \sum_j \sum_i P(W_i) m_{ij}^2 \tag{3-22}$$

Therefore,

$$\text{Tr } S_b + \text{Tr } S_w = \sum_j \sum_i P(W_i) E\{x_{ij}^2\} - \sum_j m_{0j}^2 \tag{3-23}$$

$$= \sum_i P(W_i) \sum_j E\{x_{ij}^2\} - \sum_i P(W_i) \sum_j m_{j0}^2 \tag{3-24}$$

$$= \sum_i P(W_i) \left[\sum_j E\{x_{ij}^2\} - m_{j0}^2 \right] \tag{3-25}$$

$$= \sum_i P(W_i) \text{ Tr } E\{(\bar{x}_i - M_0)(\bar{x}_i - M_0)^T\} \tag{3-26}$$

$$= \text{Tr}[\emptyset] = K \tag{3-27}$$

where $[\emptyset]$ is the covariance matrix of the data. Thus $\text{Tr } S_b + \text{Tr } S_w$ = constant and

$$\text{Tr } S_b = K - \text{Tr } S_w \tag{3-28}$$

Hence

$$\beta = \text{Tr } S_b \cdot \text{Tr } S_w = (K - \text{Tr } S_w) \text{ Tr } S_w \tag{3-29}$$

differentiating with respect to $\text{Tr } S_w$ and setting the derivative equal to zero yields

$$\text{Tr } S_w = K/2 \tag{3-30}$$

which implies that β achieves a maximum at the clustering that causes $\text{Tr } S_w$ to equal one-half the $\text{Tr } [\emptyset]$.

Further,

$$\alpha = \text{Tr } S_b / \text{Tr } S_w \tag{3-31}$$

$$= \frac{K - \text{Tr } S_w}{\text{Tr } S_w} \tag{3-32}$$

When $\text{Tr } S_w = K/2$,

$$\alpha = \frac{K-K/2}{K/2} = 1 \tag{3-33}$$

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

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.

This relationship does not hold in general but is a phenomenon which is peculiar to the weighting of terms by the cluster probabilities.

3.8 Feature Selection

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. It has been concluded by Andrews [3-15] that feature selection in an uncorrelated space is highly desirable.

The criterion of optimality for the selection of a feature set is the probability of misclassification of the samples. Several measures have been developed which upper bound the misclassification rate. Specifically, for a Bayes symmetric cost function classifier and Gaussian data the error rate has been shown to be upper bounded inversely as the Bhattacharyya measure [3-16], [3-17], [3-18].

Hence

$$P_e \leq P(S_1)P(S_2)e^{-B(S_1, S_2)} \tag{3-34}$$

for a two class problem where

$$B(S_1, S_2) = \tfrac{1}{4} \ell n\{\tfrac{1}{4}|[\phi_2]^{-1}[\phi_1] + [\phi_1]^{-1}[\phi_2] + 2[I]|\}$$
$$+ \tfrac{1}{4} tr\{([\phi_1] + [\phi_2])^{-1}(\mu_1 - \mu_2)(\mu_1 - \mu_2)^t\} \tag{3-35}$$

The Gaussian distributions are given by

$$P(x|S_K) = N(\mu_k, [\phi_k]) \tag{3-36}$$

and $[\phi_k]$ is the covariance matrix of the k^{th} class.

For a multi-class problem, P_e can be bounded as in Eq. (3-34) by pairwise averaging, i.e.,

$$P_e \leq \sum_{i>}^{K} \sum_{j}^{J} P(S_i)P(S_j)e^{-B(S_i, S_j)} \tag{3-37}$$

Equation (3-35) 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:

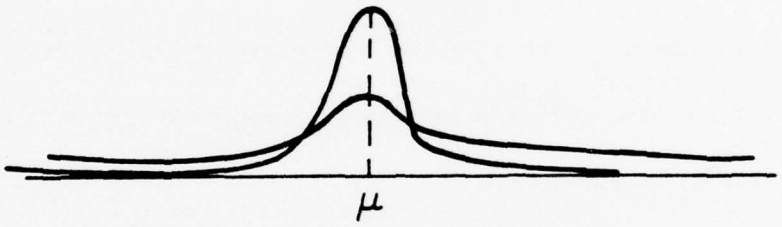
$$B_n(S_1, S_2) = \tfrac{1}{4} \ell n \left\{ \tfrac{1}{4} \left(\frac{\sigma_1^2(n)}{\sigma_2^2(n)} + \frac{\sigma_2^2(n)}{\sigma_1^2(n)} + 2 \right) \right\}$$

$$+ \tfrac{1}{4} \left\{ \frac{(\mu_1(n) - \mu_2(n))^2}{\sigma_1^2(n) + \sigma_2^2(n)} \right\} \tag{3-38}$$

where n refers to the n^{th} dimension of the space. This form involves only scalar means and variances.

Figure 3.7 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 Figure 3.7a, the first term of the Bhattacharyya measure will be zero but the second term will be non-zero. The second term will be large if the variance is small under

(a) EQUAL VARIANCES, UNEQUAL MEANS



(b) EQUAL MEAN, UNEQUAL VARIANCES

(c) UNEQUAL MEANS AND VARIANCES

Figure 3.7. Bhattacharyya Measure Distribution

this condition, implying that a large difference in means accompanied by small variances is a desirable quality in a feature for distinguishing between two classes. The situation depicted in Figure 3.7b 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 classes. Thus, in this situation, the second term of the Bhattacharyya distance will be zero but the first term will be non-zero. Finally, in Figure 3.7c both the mean and variance are unequal and both terms of the measure will be non-zero.

The performing of feature evaluation in uncorrelated space implies that an eigenvector (or discrete Karhunen-Loeve) transformation is required [3-19]. While the dimensions having the largest eigenvalues will be the best under certain conditions, they will not be optimal in general. The one-at-a-time Bhattacharyya measure will pick the correct eigenvector regardless [3-15].

3.9 Segmentation Comparison Measure

There exist a host of techniques developed over the last few years for forming clusters and segmenting images. A common shortcoming is that it is nearly impossible to compare these methods since no quantitative conditions of optimality exist. Typically in the literature a statistical or heuristic argument is made that a proposed

method should work well for a particular type of scene. A mathematical argument sometimes proceeds and classified images are displayed to support the predicted performance. Virtually no means exist for comparing performance among methods and it is suspected that no one method works well for all types of scenes.

An approach to this shortcoming would be for there to exist a standard data set of segmented pictures that human observers agreed were "correctly" segmented. If pictures segmented by a candidate procedure could be compared to the standard segmented data base, a primitive means would then exist for comparing different segmentations of the same scene. A comparison measure that is proposed is:

$$C = \left[\sum_{i,j} h_{max}(i,j)\frac{1}{N} - \frac{1}{I \cdot J}\min(I,J)\right]\Gamma\% \qquad (3\text{-}39)$$

$$\Gamma = \frac{100}{1 - \frac{1}{I \cdot J}\min(I,J)}$$

I and J are the total number of segments in images 1 and 2 respectively. $h_{max}(i,j)$ are the elements of the joint histogram of the segmented images which are maximum in both the rows and the columns of the joint histogram. Γ is a normalizing factor which forces C to be bounded as $0 \leq C \leq 100$ and N is the total number of pixels in each image.

Pictures which are identical will have C = 100% while pictures which are completely unrelated and have equal sized segments will

have a uniform joint histogram and have C = 0%. This measure penalizes the segmented images for having non-equal numbers of segments. Pixels in the image having the greater number of segments that are located in the smaller sized segments are regarded as being misclassified.

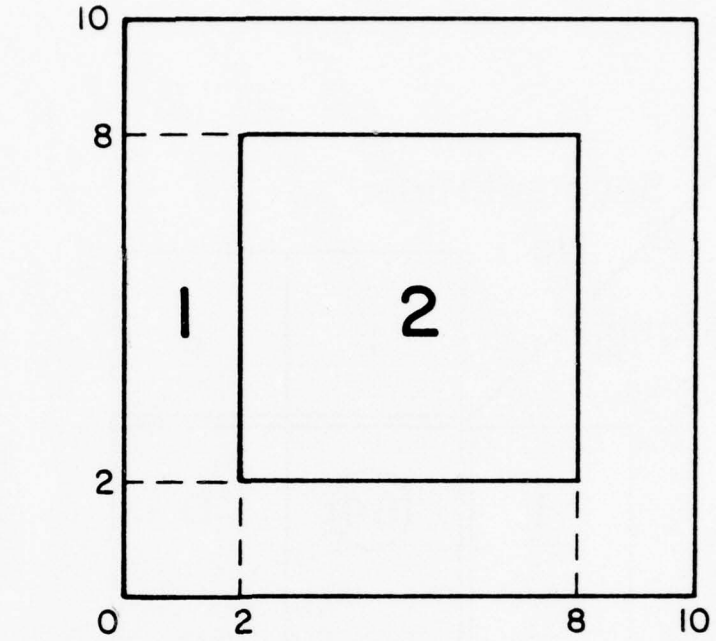
As an example of the behavior of this measure, consider the segmentations of Figure 3.8. The joint histogram of these segmentations is also given in Figure 3.8. The comparison measure between these two segmentations will be:

$$C = \left[\frac{64}{100} + \frac{24}{100} - \frac{1}{6} \times 2\right] \Gamma = .547 \times \Gamma$$

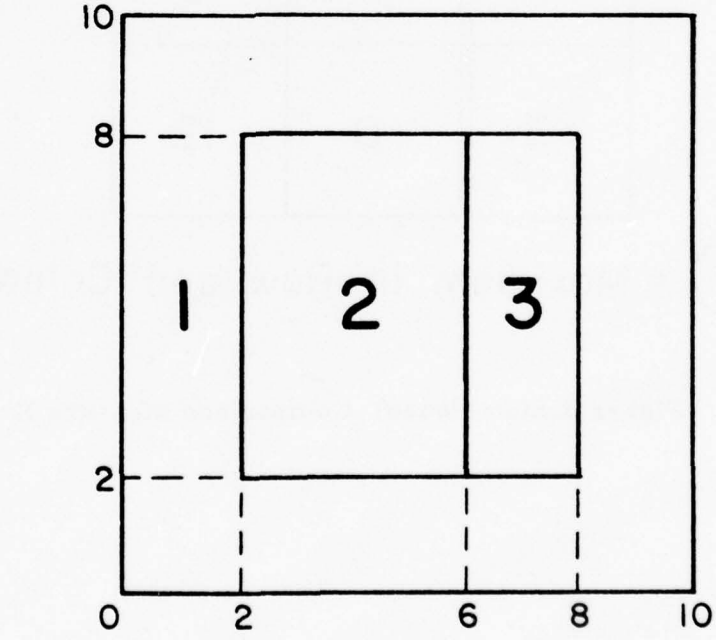
$$\Gamma = \frac{100}{1 - \frac{1}{6} \cdot 2} = 1.5$$

$$C = 82\%$$

As another example of the behavior of this measure, suppose that the two segmentations to be compared were similar to segmentation number 1 except that segment number 2 is of different size. When segment number 2 consists of only one pixel, the comparison measure will equal 47.5%. At the other extreme, where segment number 2 consists of all but 1 pixel of the segmentation, the comparison measure will equal 46.0%. The measure will, in this case, reach a maximum of 100% when segment number 2 is of



SEGMENTATION NO. 1



SEGMENTATION NO. 2

Figure 3.8 Comparison Measure Example

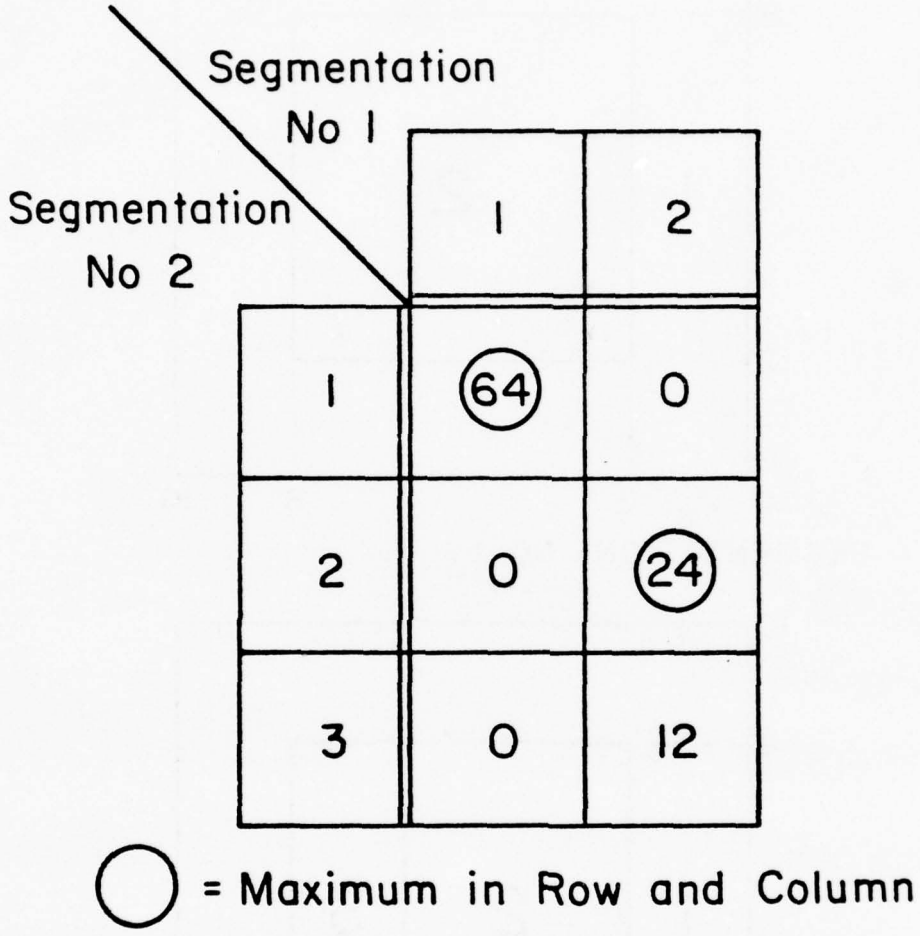


Figure 3.8(continued) Comparison Measure Example

equal size in both segmentations. If both segmentations are completely unrelated and have equal sized segments, the joint histogram will be uniform and each entry will equal $\frac{1}{I \cdot J}$ min(I, J) where I and J are the number of segments in pictures 1 and 2 respectively. The comparison measure will equal 0% in this case.

Chapter 4

IMAGE SEGMENTATION BY CLUSTERING - AN APPROACH

4.1 Overall Approach

The overall approach taken to segment images by clustering is depicted in the general block diagram of Figure 4.1. The feature computation block computes several features at each pixel. These features are related to brightness and texture at several window sizes centered on every pixel.

The feature decorrelation is performed by a multi-dimensional 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. If feature reduction is not performed on decorrelated features, several highly correlated features may be retained, but convey essentially the same information.

The feature reduction is accomplished by performing clustering on the full feature set on a sample basis. In other words, only samples of the image are used for clustering to reduce the time required. At the number of clusters which is determined to be "optimum," those features having above average one-at-a-time Bhattacharyya measure are retained, and the remainder are

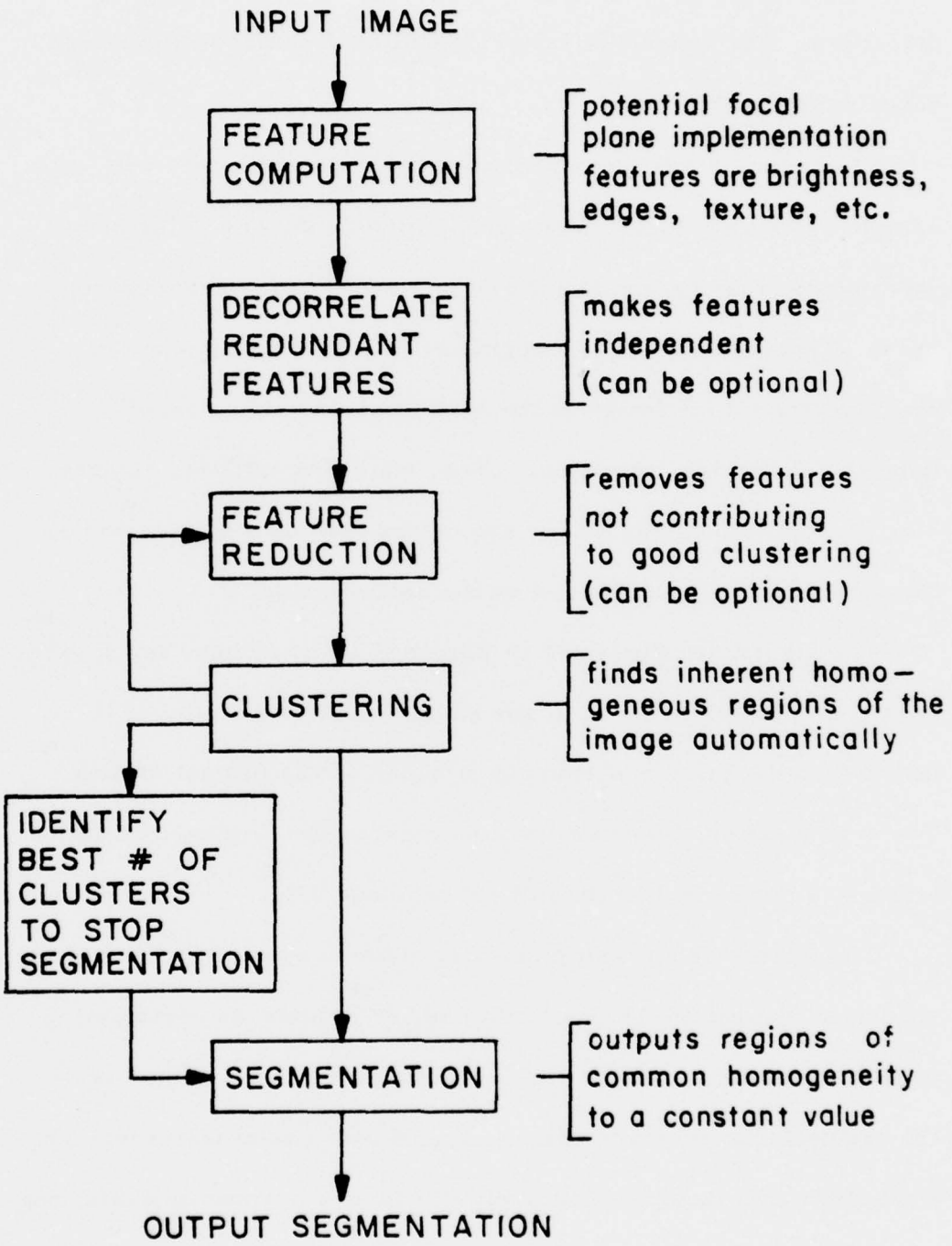


Figure 4.1 General Block Diagram

discarded. The optimality criterion will be discussed in greater detail subsequently.

Clustering is again performed on the reduced feature set, on a sample basis as before. When the optimum number of clusters is determined, 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. Thus, while the optimum number of clusters and the cluster means are determined on a sample basis, the segmentation is performed on the entire image.

The algorithm illustrated in Figure 4.1 was adopted for several reasons. Clustering on a sample of the image is a factor of 16 faster than the same procedure performed on the full set of data. The segmentation, however, retains most of the original resolution since it is performed on the full set of image data.

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 Figure 4.2. 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.

4.2 Feature Rotation

The covariance matrix is computed over the feature set as

$$[\phi(i,j)] = \underset{x,y}{<} (p_i(x,y)-\bar{p}_i)(p_j(x,y)-\bar{p}_j)^T > \qquad (4-1)$$

Here $<\cdot>$ denotes averaging and $\bar{p}_i$ is the average of the i^{th} feature over the image. The average is performed on every fourth pixel and every fourth line to reduce computation time. The diagonal elements of this matrix are the feature variances over the image. The matrix which diagonalizes the covariance matrix is computed yielding

$$A^T \phi A = \Lambda \qquad (4-2)$$

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

$$\Lambda = \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_N \end{bmatrix} \qquad (4-3)$$

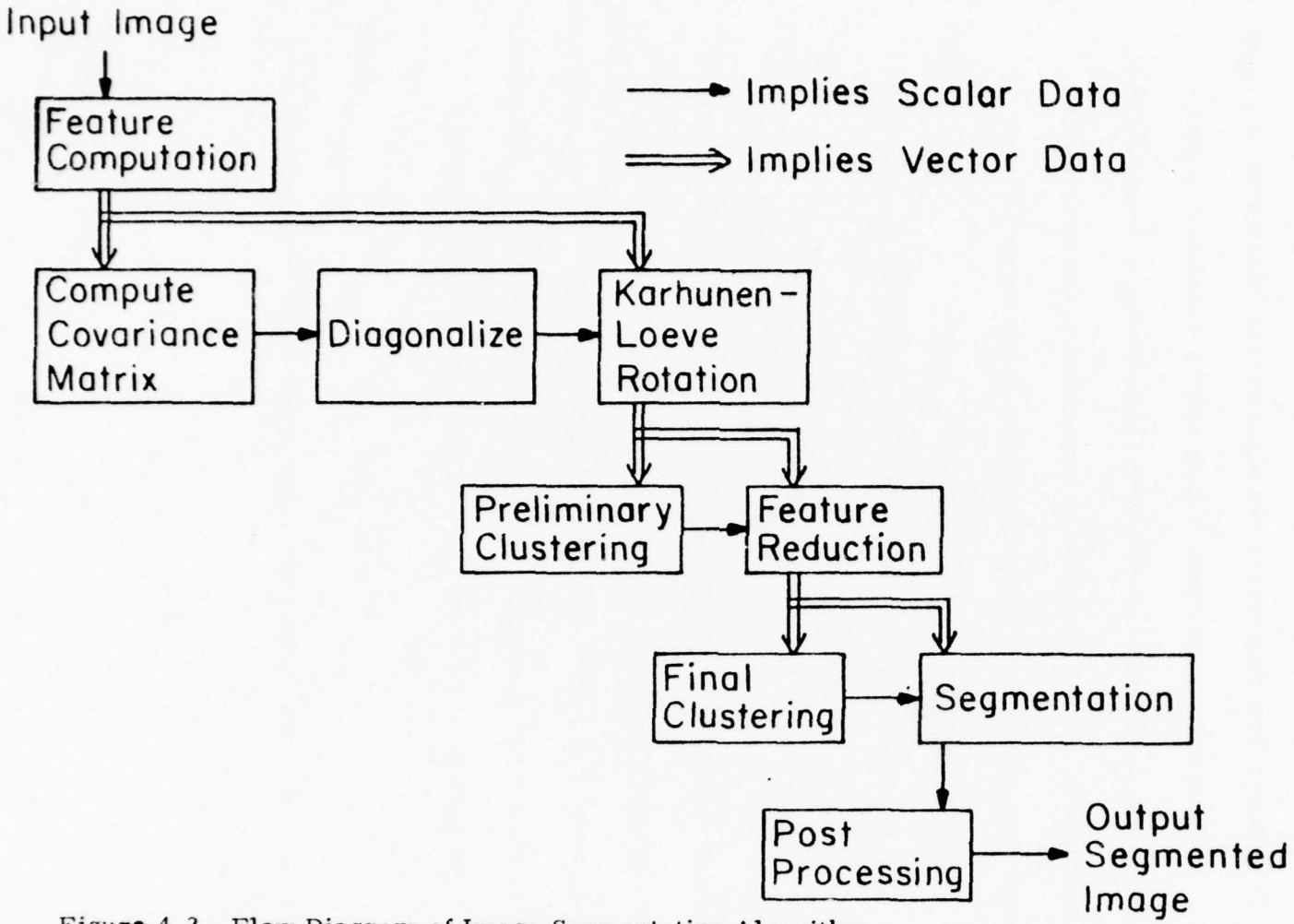


Figure 4.2 Flow Diagram of Image Segmentation Algorithm

The matrix A which accomplishes this diagonalization is the well-known matrix of eigenvectors, i.e.,

$$A = [\bar{a}_1 \; \bar{a}_2 \; \ldots \; \bar{a}_N] \tag{4-4}$$

where $\bar{a}_i$ is an eigenvector, i.e.,

$$\emptyset \bar{a}_i = \lambda_i \bar{a}_i \tag{4-5}$$

A new feature set is computed by multiplying every vector in the original space by A^T, i.e.,

$$\bar{q}(x, y) = A^T \bar{p}(x, y) \tag{4-6}$$

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

$$[\emptyset_R(i, j)] = A^T [\emptyset(i, j)] A = \Lambda \tag{4-7}$$

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

4.3 Clustering Algorithm

The clustering algorithm uses the k-means algorithm for $2, 3, 4, \ldots, 16$ clusters. At each step, the quality of clustering is computed as $\beta = \text{Tr } S_b \cdot \text{Tr } S_w$ (see Ch. 3 and equations (3-3), (3-4) and (3-7)). The average pairwise Bhattacharyya distance is

computed for every feature. At the product maximum, the Bhattacharyya distance for all features is computed. Features having a Bhattacharyya distance which 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 Figure 4.3.

The vectors are assigned to the nearest cluster mean in accordance with the L_1 distance measure, i.e.,

$$d_1(\bar{p}^{(1)}, \bar{p}^{(2)}) = \sum_{i=1}^{N} |p_i^{(1)} - p_i^{(2)}| \qquad (4\text{-}8)$$

The clustering of spatial sources has been shown to be very insensitive to the distance measure used [4-1]. Therefore, the absolute value measure was chosen over the L_2 distance measure

$$d_2(\bar{p}^{(1)}, \bar{p}^{(2)}) = \left[\sum_{i=1}^{N} (p_i^{(1)} - p_i^{(2)})^2 \right]^{\frac{1}{2}} \qquad (4\text{-}9)$$

to reduce the computation time required.

The clustering algorithm computes the cluster means on every fourth line and every fourth pixel to further reduce computation time. For a given number of clusters, the algorithm iterates until it converges. Convergence is assumed to have been reached when the means on the K-1st iteration and the means on the Kth iteration

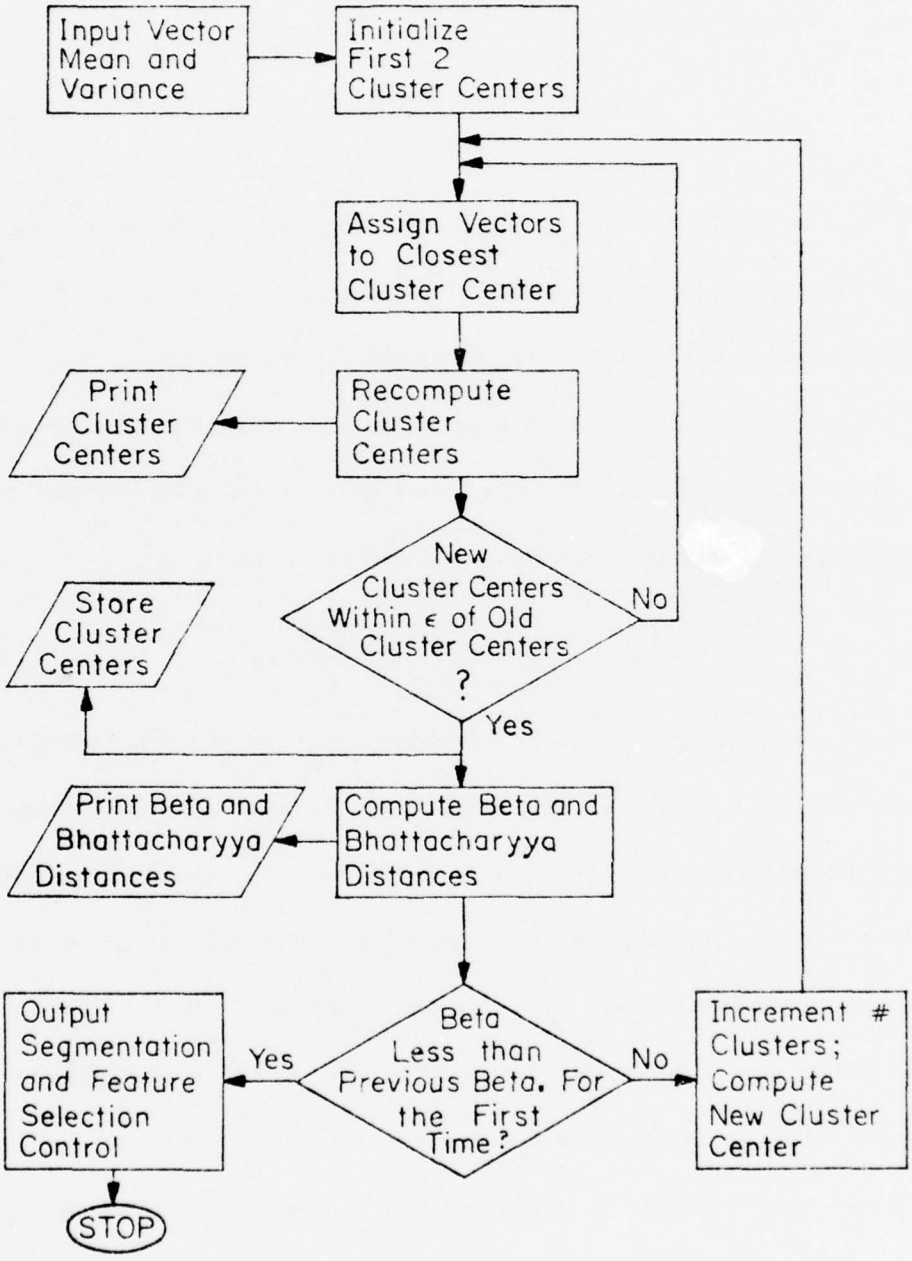


Figure 4.3 Clustering Algorithm Flow Chart

differ by less than one brightness level in any dimension for any cluster. This is equivalent to

$$\| C_j^{(K)} - C_j^{(K-1)} \|_\infty < 1 \quad \text{for all } j \tag{4-10}$$

Since this is the limiting resolution of the data, further iterating is performed on the quantization noise of the data and does not yield results which will significantly affect the segmentation.

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 as

$$C_K^j = C_K + (V_K)^{\frac{1}{2}}(2(j-1) - 1) \tag{4-11}$$

where C_K is the mean of the K^{th} feature, V_K is the variance of the K^{th} feature and $j = 1, 2$ is the appropriate cluster number. Equation (4-11) places the initial cluster centers evenly spaced on the diagonal of positive correlation at plus and minus 1 standard deviation in the hyper-space of the feature set. As the number of clusters is incremented, the new cluster center is initialized at the vector whose distance from its respective cluster center is the greatest.

Final segmentation is performed on every pixel, utilizing the means or cluster centers computed during the clustering algorithm. This procedure permits segmentation of the image to nearly the original resolution, while performing the tedious computations on

one-sixteenth of the data.

4.4 Feature Computation

An aspect of clustering which has a major effect on the results is the feature set used to describe the image. While this research was not intended to probe deeply into an optimal feature set, some exploration of the subject was necessary to permit development of the clustering procedure. For monochrome imagery, the most obvious features that are intuitively important to human observers are brightness and 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 dependancy matrices proposed by Haralick [4-2]. 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 [2-2] found that perhaps 5 or less correlate significantly with human perception. If the original 256 possible brightness levels in the original picture are quantized to 16 levels, and if 4 angles and 4 distances are used, then 16, 16 x 16 matrices must be computed at every pixel. This amount of computation was considered excessive

in view of the goal that this procedure is intended to be reasonably fast.

Other texture 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:

$$[S_1] = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad \text{and} \quad [S_2] = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{4-12}$$

At each pixel, the image is multiplied by the $[S_1]$ and $[S_2]$ masks yielding S1 and S2. The Sobel magnitude is then defined as

$$SM = (S1^2 + S2^2)^{\frac{1}{2}} \tag{4-13}$$

and the Sobel phase is given as

$$SP = \arctan\left(\frac{S2}{S1}\right) \tag{4-14}$$

These measures are designed to detect well defined edges. As a result, they tend to have large value at such edges and very low values elsewhere. When quantized to 8 bits, much of the region is quantized to zero and only clearly defined edges remain visible. For that reason, the logarithm of the magnitude was taken to expand the lower range of values, i.e.,

$$SM^* = \log(1 + SM) \tag{4-15}$$

The Sobel phase has the opposite shortcoming. It will have large (although nearly random) value in regions where no discernible texture exists. The Sobel phase texture operator was therefore computed as

$$SP^* = (SP+\pi)SM^* \qquad (4\text{-}16)$$

in an attempt to suppress the phase operator when no texture is present. These primative 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 "fine grain" segments were considered undesirable. It was decided to perform a pre-filtering 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. The Tukey median filter [4-3], while much better in this respect than a local averaging, still causes some blurring of the boundaries. A 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 of the region. Region

sizes of 3 x 3, 7 x 7 and 15 x 15 were computed. The local area histogram was computed to N^2 levels, where N is the linear dimension of the window.

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 disadvantage of the mode filtering is that it creates artificial boundaries when regions of slowly varying brightness cross a threshold of the histogram. The resulting image looks much like a "paint-by numbers" painting.

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. The square window will clip corners of square regions, however. The dispersion or histogram height does not have the nice properties of the mode value. The height of the mode will decrease as a region boundary is approached, and increase as the region boundary is left. Some blurring of region boundaries will therefore occur.

If the region size to be detected is near the size of the mode filter window, it will pass undetected (or be severely reduced) by the filtering process. The window size must therefore be selected to be

smaller than the smallest region size to be detected.

Color features were computed by performing mode filtering on the three spectral images (red, green and blue). This expanded feature set produced more subjectively satisfying results, as would be expected from the increased information available from the multi-spectral data.

There are generally five feature sets which are used for image clustering. These feature sets are summarized in Table 4.1. 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 filter for brightness information.

The original set was feature set number 1. Feature set number 2 was motivated by the attempt to obtain a more satisfying result from the aerial image. In the case of polychromatic imagery, feature set number 3 is basically similar to feature set number 2 for monochrome imagery, with the larger (15 x 15) mode filters eliminated to conserve computer time. Feature set number 4 is an obvious choice, and feature set number 5 was chosen in an attempt to introduce texture information into the segmentation process for the ten band multi-spectral image.

As has been previously discussed, the primary goal of this effort was to develop the segmentation algorithm. Feature experimentation

MONOCHROME IMAGERY

Feature Set Number 1

Feature No.	Description
1	Original
2	Log Sobel Magnitude
3	Sobel Phase x Log Sobel Magnitude
4	Feature 1 Mode Filtered, 3 x 3
5	Feature 2 Mode Filtered, 3 x 3
6	Feature 3 Mode Filtered, 3 x 3
7	Feature 1 Mode Filtered, 7 x 7
8	Feature 2 Mode Filtered, 7 x 7
9	Feature 3 Mode Filtered, 7 x 7
10	Feature 1 Mode Filtered, 15 x 15
11	Feature 2 Mode Filtered, 15 x 15
12	Feature 3 Mode Filtered, 15 x 15

Feature Set Number 2

Feature No.	Description
1	Original Mode Filtered, 3 x 3
2	Dispersion of Feature 1, 3 x 3
3	Original Mode Filtered, 7 x 7
4	Dispersion of Feature 3, 7 x 7

Table 4.1. Feature Set Descriptions

POLYCHROMATIC IMAGERY

Feature Set Number 3

Feature No.	Description
1	Red Original
4	Red Original Mode Filtered, 3 x 3
5	Dispersion of Feature 2, 3 x 3
10	Red Original Mode Filtered, 7 x 7
11	Dispersion of Feature 4, 7 x 7
2, 6-7, 12-13	Similar to above for Green
3, 8-9, 14-15	Similar to above for Blue

Feature Set Number 4

Feature No.	Description
1 - 10	Ten unmodified bands of multispectral imagery.

Feature Set Number 5

Feature No.	Description
1 - 2	Best two rotated features of 10 band multispectral imagery.
3	Feature 1 Mode Filtered, 3 x 3
4	Dispersion of Feature 3, 3 x 3
5	Feature 2 Mode Filtered, 7 x 7
6	Dispersion of Feature 5, 7 x 7

Table 4. 1 (Continued) Feature Set Descriptions

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.

Chapter 5

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, charts and tables at the end of this chapter. 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, used to stop the algorithm at the "correct" number of clusters. The tables consist of the statistical data used to decorrelate the features for the various images, a comparison of the Bhattacharyya distance with the decorrelated feature eigenvalues, results of running the segmentation comparison measure and a table of computer time required to run the algorithm in its various configurations. The table of Bhattacharyya distance measure versus feature eigenvalues illustrates the superiority of this measure to the eigenvalues in identifying the best features.

5.1 APC Image Results

Examples of features and segmented images are shown in Figures 5.1 through 5.7. Figure 5.1 consists of the 12 original features

computed from the APC image. These features were subjected to clustering without rotation, producing the segmentations of Figure 5.2. The probability weighted product maximum occurred at 9 clusters. A graph of the average Bhattacharyya distance versus the number of clusters for this image is shown in Figure 5.8. 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 $d_B = 1.0$. 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. The probability weighted product of the between cluster scatter measure and the within cluster scatter measure was computed for each number of clusters. The between and within scatter measures are normalized by Γ^2 so that they range between 0 and 1. These products are plotted versus the number of clusters for the APC image under various conditions as well as for several other images in Figure 5.9. At the probability weighted product maximum for the original APC features (9 clusters in this case), the above average

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 respectively. Thus all of the texture information has been discarded. These features were used to again cluster the image and the results are shown in Figure 5.2. The probability weighted product maximum occurred at 2 clusters for the reduced feature set.

The covariance matrix of the 12 original feature set was computed and diagonalized. The covariance matrix and the diagonalization (eigenvector) matrix as well as the eigenvalues of the covariance matrix are tabulated in Table 5.1. The eigenvectors are shown as column vectors in the table. In the case of the APC image, rotated feature number 1 consists of -.37 x original feature number 1 plus .13 x original feature number 2, etc. Each vector of the rotated feature set is computed in this manner from the spatially corresponding vector in the original feature set. The actual rotated feature set is shown in Figure 5.3. The version shown in Figure 5.3 has been rescaled for ease of viewing. The set of 12 features used for clustering was rescaled as a set to cover the range 0, 255. The feature set displayed in Figure 5.3 has had each feature individually rescaled to cover 0, 255 for viewing convenience. The covariance matrix of the rotated features is diagonal and each diagonal entry is equal to the variance for the respective feature. The features are arranged approximately in order of descending eigenvalue (variance) by the computer

routine that diagonalizes the covariance matrix. The lower variance (energy) of the higher number rotated features is evident from their appearance. The columns of the rotation matrix are the eigenvectors of the covariance matrix corresponding to the eigenvalues listed above it. The average Bhattacharyya distance for each feature is listed in Table 5.2, along with the eigenvalues and the rank of the feature with respect to average Bhattacharyya distance. It can be seen that the relative eigenvalue does not exactly correspond to the average Bhattacharyya distance.

An interesting phenonemon can be observed in the behavior of the clustering quality measure (product) in Figure 5.9. The behavior of the quality measure for the rotated and non-rotated 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.

The segmentation at the probability weighted product maximum

for the 12 non-rotated features (Figure 5.2) and the segmentation at the probability weighted product maximum for the 12 rotated features (Figure 5.4) appear very similar. That this is so is expected, since the multidimensional rotation of the axes by the rotation matrix is a linear invertible map and should not change the shape of the clusters. The differences which do exist are due in small part to numerical (round off) errors in the computation and it is conjectured that they are due in larger part to the fact that the clustering algorithm is somewhat sensitive to cluster initialization. The nearest means algorithm will converge to a local minimum in the average inter-cluster 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 the rotated features and are plotted in Figure 5.8. The above average features at the probability weighted product maximum (4 in this case) were used to again cluster the image. The results of this are shown in Figure 5.4. The comparison between these segmentations and the segmentations performed with the 4 best non-

rotated features is interesting. The feature reduction in the non-rotated space retained all of the brightness related features and none of the texture related features. Therefore, the 4 retained features were highly correlated and all of the texture information was lost. The feature reduction in the rotated space, on the other hand, discarded only non-correlated information. The features that remained can be expected to contain most of the information necessary for clustering. The segmentation that resulted from the 4 best rotated features differs from the segmentation of the 4 non-rotated features mainly in that the background has been split into two segments. Otherwise, the segmentations are very similar and suggest that the intrinsic structure of the data has been retained.

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 Figure 5.4. The classification of the bushes in the images as being the same as the vehicle constitutes an error or misclassification in the process. Substantially more errors were made when the segmentation was done with only one feature, which is expected due to the large reduction in dimension.

5.2 Aerial Image Results

The segmentation procedure was applied to an aerial image. The original as well as the segmentation results are shown in Figure 5.5. The photograph labeled "5 best of 12 rotated features" was segmented using the same 12 features used for the APC image. This image was also segmented using a different feature set consisting of 4 features. To attempt to measure the local dispersion of grey level values, the height of the local histogram mode as well as the grey scale value of the mode was used as a feature. The height was rescaled 0, 255 before clustering. An example of the dispersion feature for a 3 x 3 window is shown in Figure 5.5. The set of four features consisted of the mode and dispersion for a 3 x 3 window and the mode and dispersion for a 7 x 7 window. These features were rotated; clustering was performed; the above average features (1 in this case) were selected and clustering was again performed. The results of this segmentation are labeled "1 Best of 4 Rotated Features" in Figure 5.5. Covariance and rotation matrices, eigenvalues and Bhattacharyya distances, and probability weighted product behavior for this image are tabulated in Table 5.1, 5.2 and Figure 5.9 respectively.

5.3 House Multispectral Images Results

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 first polychromatic image to which the procedure was applied was a color image of a house. The red, green and blue original images for this picture are shown in Figure 5.6. The feature set consisted of the mode and dispersion for each of the three color planes, in both 3 x 3 and 7 x 7 windows. Thus there were 4 features plus the original for each of the three colors, yielding 15 features. Examples of the dispersion feature for the red image are shown in Figure 5.6. The results of the segmentation for the full feature set and the best (1 in this case) feature are shown in Figure 5.6 and are remarkably similar. The windows were classified as "sky" because the sky is reflected in them. The covariance and rotation matrices, eigenvalues and Bhattacharyya distances and probability weighted product behavior for this image are tabulated in Tables 5.1, 5.2 and Figure 5.9 respectively.

A series of 10 multi-spectral images were also used for segmentation. Two bands of the original set and the resulting segmentations are shown in Figure 5.7. The covariance matrix of the rotated feature set was approximately singular, indicating that the multispectral data is highly redundant. The variance of the higher numbered features was low and approximately the same value as the off-diagonal elements in the covariance matrix. As a result, the ratio of these very small numbers was large in some cases and the Bhattacharyya distance was found by the algorithm to be large even though the features themselves

had very little energy. As a result, the best two features selected by the algorithm included a feature which had extremely low variance and the resulting segmentation looked almost identical to the segmentation produced by the full 10 feature set. In an attempt to improve performance, the best two features having significant energy (variance) were used to produce the segmentation labeled "2 Rotated Features - Augmented." The augmentation consisted of the mode and dispersion computed on each of the "hand selected" features. The covariance and rotation matrices, eigenvalues and Bhattacharyya distances, and probability weighted product behavior are tabulated in Tables 5.1, 5.2 and Figure 5.9 respectively.

5.4 Computer Time Required

The computer time, in CPU minutes, for various steps in this procedure is tabulated in Table 5.3. The "full runs" are permitted to run out to 16 clusters, regardless of the occurrence of a product maximum. The "abbreviated run" consists of stopping the clustering algorithm when the product maximum is realized. This occurs when clustering is performed with one more cluster than the number corresponding to the product maximum. The segmentation produced, however, corresponds to the previous clustering, that is, the product maximum.

The computation was performed on a PDP-10 computer utilizing a BBN Tenex operating system. The programs were written in FOR-

TRAN. The clustering is performed on every fourth pixel and every fourth line, while the segmentation is performed on every line and every pixel. Only one line is stored in core memory at a time, to reduce core requirements. The effect of one-line-at-a-time clustering also speeds up the procedure considerably, since the machine is time-shared, and the computer core image must be stored on disk each time the allocated time for the particular user expires. Excessive core storage requires excessive switch in/switch out time, and substantially increases the CPU time required for program execution.

Several steps might be taken to reduce the computer time required for performance of the algorithm. The mode filters might be implemented more efficiently by modifying the local histogram as the window is moved instead of recomputing it at every pixel. If the region size desired in the segmentation is known a-priori, the mode filtering might be performed on only one window size, instead of several window sizes, as is currently done.

It might be that a suitable fixed transformation of the features would suffice for a class of pictures. If this were so, the transformation could be computed once, and used for every picture to be segmented. Alternately, a preliminary feature rejection based on eigenvalue could be performed, eliminating some of the features prior to the first clustering operation.

Since feature computation comprises roughly one-third (see

Table 5. 3) of the computer time required, a smaller number of easy-to-compute features would clearly speed up the process.

Improved computational efficiency could also be obtained by writing some of the repetitive calculations as machine-language subroutines, instead of in FORTRAN, as the algorithm is currently implemented.

5. 5 Comparison Measure Results

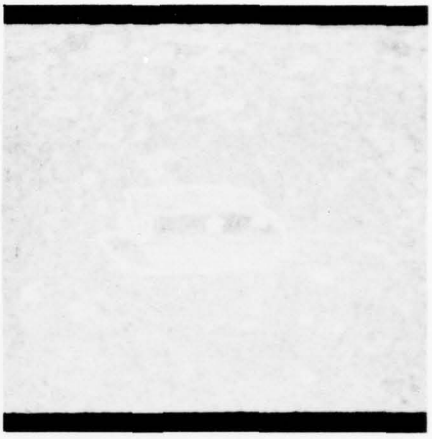
The results of comparing numerous segmentations of the APC image using the comparison measure are tabulated in Table 5. 4. The highest number in the table is 96% and occurs in the comparison of the 2 cluster segmentation of the 4 best non-rotated APC feature set with the 2 cluster segmentation of the 1 best rotated APC feature set. Both of these segmentations were the product maximum for the respective feature sets. It can be seen from figures 5. 2 and 5. 4 that these segmentations appear almost identical, which is intuitively satisfying since the comparison measure should correlate with human perception. It should also be noted that the two segmentations are negatives of each other, that is, the numerical values assigned to "APC" and "background" are opposite in the respective segmentations. The comparison measure effectively ignores the absolute value of the points by selecting the elements in the joint histogram which are larger.

The ideal use for the quality measure would be to compare seg-

mentations made by different procedures to a "standard" segmentation, perhaps created by a human observer. If a "standard" set of segmentations existed, different procedures for segmenting images could be compared to the standard segmentations, and a numerical indication of effectiveness could be derived.

Ultimately, the effectiveness of a segmentation procedure will depend on the purpose for which the segmentation is performed. If the subdivision of one segment recognized by a human observer into several segments is of no consequence, then those elements of the joint histogram corresponding to the subsegments could be combined and the comparison measure would be higher. In any event, informed use of the comparison measure should permit numerical measure of segmentation performance, an ability long lacking in image understanding system research.

Original

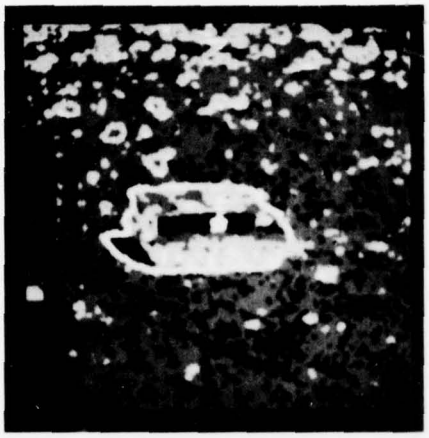


Sobel Log Magnitude

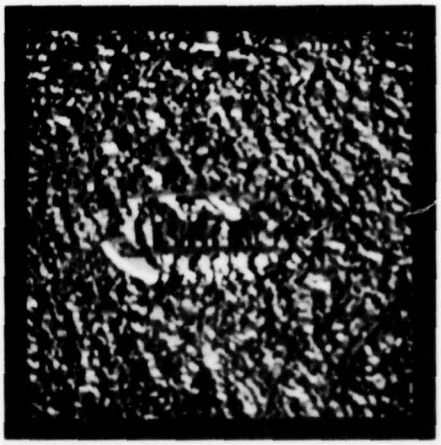
Sobel Phase Product

Original, Mode Filtered 3 x 3

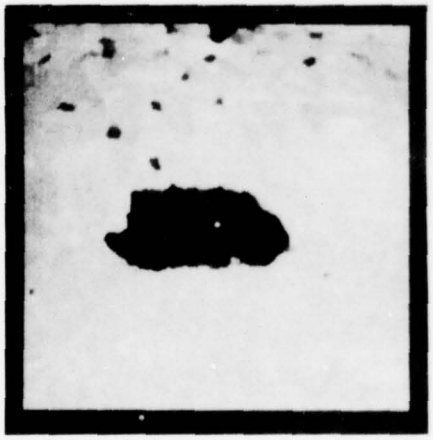
Figure 5.1 APC Image Original Features



Log Magnitude, Mode
Filtered 3 x 3



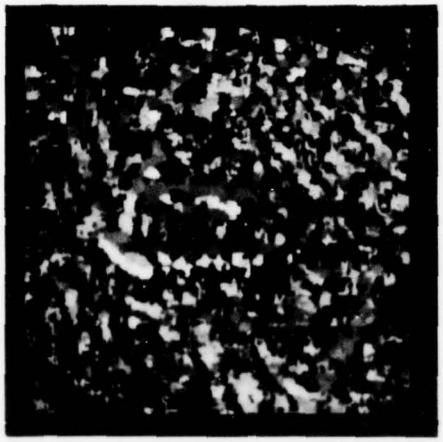
Phase Product, Mode
Filtered 3 x 3



Original, Mode
Filtered 7 x 7

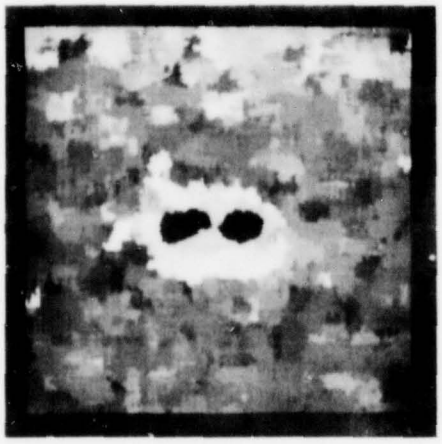
Log Magnitude, Mode
Filtered 7 x 7

Figure 5.1 (continued) APC Image Original Features

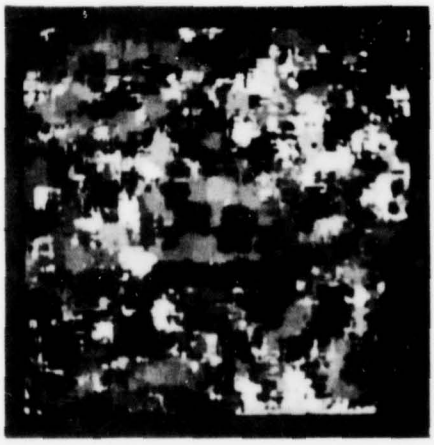


Phase Product, Mode
Filtered 7 x 7

Original, Mode
Filtered 15 x 15



Log Magnitude, Mode
Filtered 15 x 15



Phase Product, Mode
Filtered 15 x 15

Figure 5.1 (continued) APC Image Original Features

7 Clusters

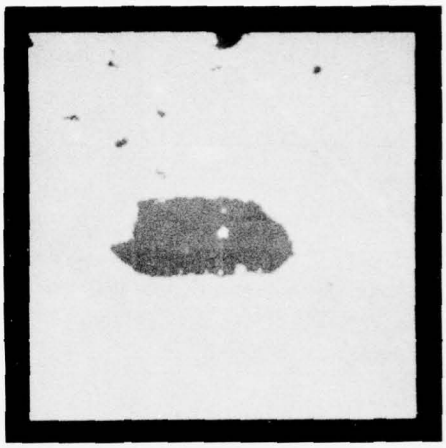
8 Clusters

9 Clusters -
Product Maximum

12 Clusters -
Most Similar To
12 Rotated Segmentations

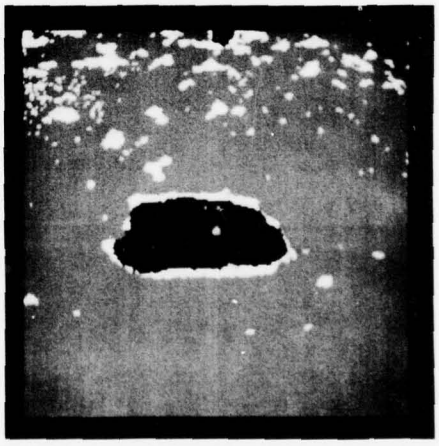
Figure 5.2 Segmentations - 12 Original APC Features

2 Clusters -
Product Maximum



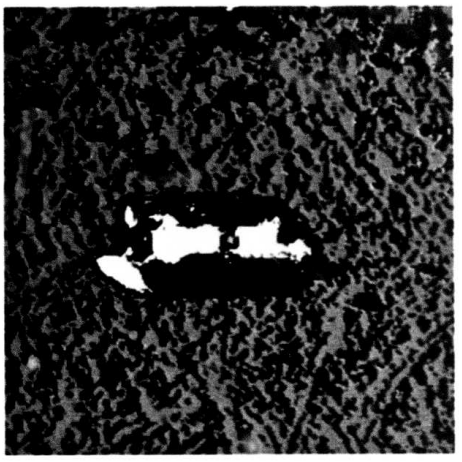
3 Clusters

4 Clusters



5 Clusters

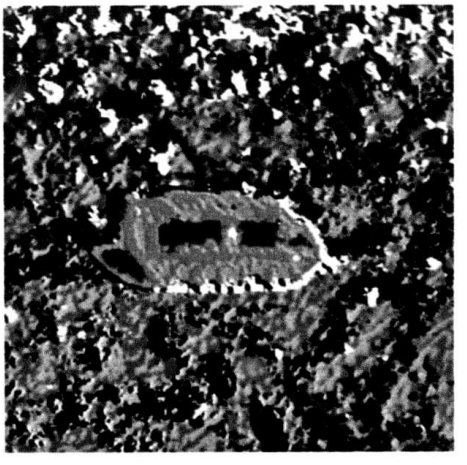
Figure 5.2 (continued) Segmentations - 4 Best APC
Original Features



7 REGIONS

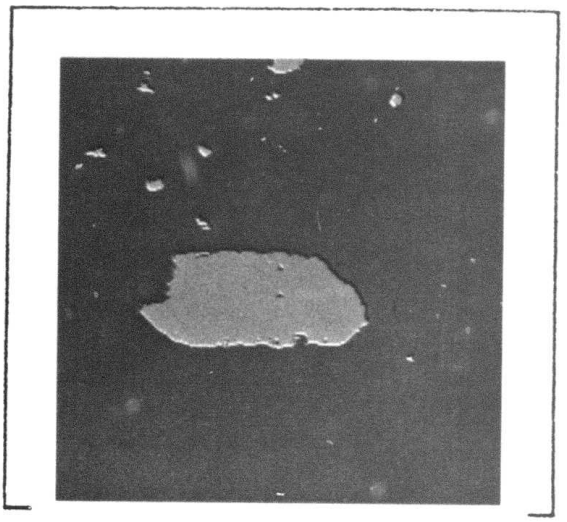
8 REGIONS

9 REGIONS
(BEST NUMBER OF
REGIONS)

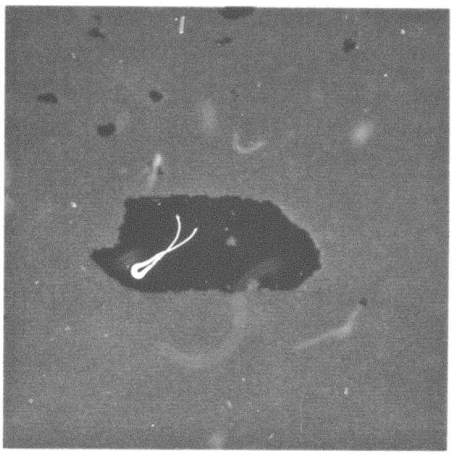


10 REGIONS

FIGURE 2. 12 NON REDUCED CORRELATED FEATURES

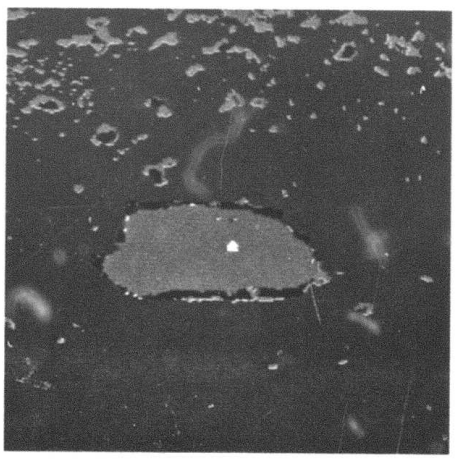


2 REGIONS
(BEST NUMBER
OF REGIONS)



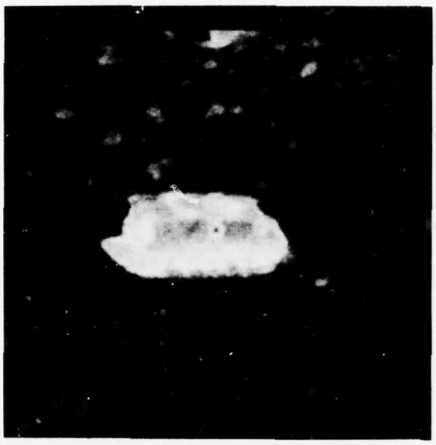
3 REGIONS

4 REGIONS

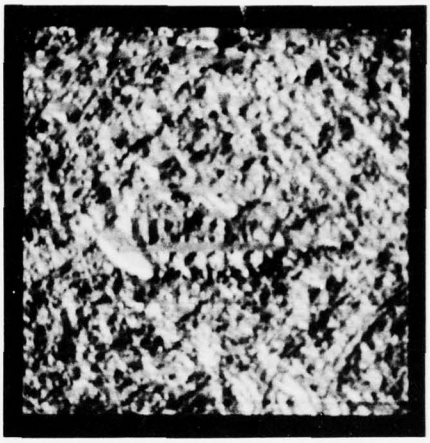


5 REGIONS

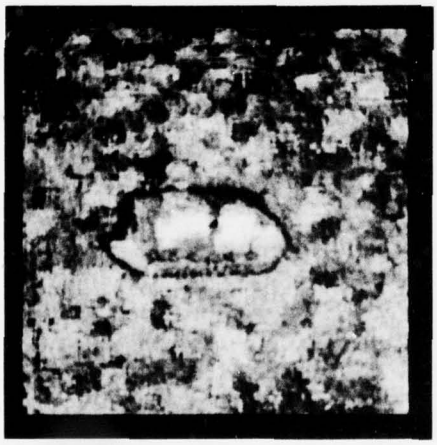
FIGURE 3. 4 REDUCED CORRELATED FEATURES



Rotated Feature 1



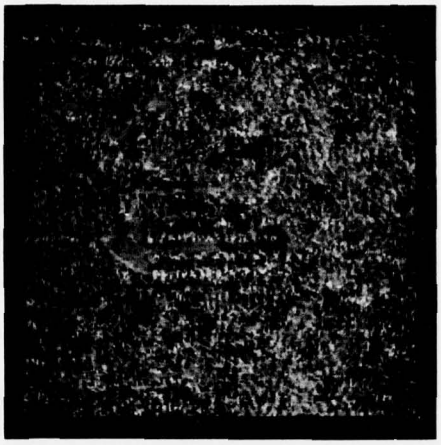
Rotated Feature 2



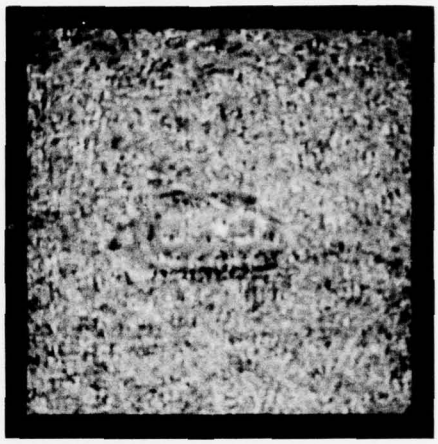
Rotated Feature 3

Rotated Feature 4

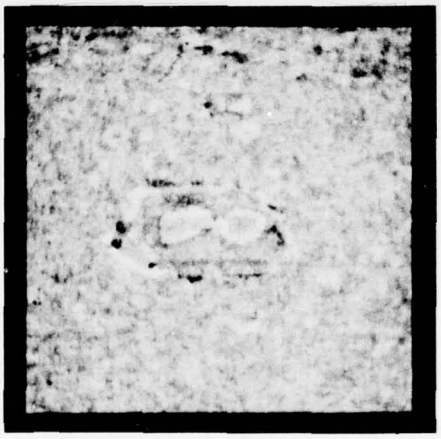
Figure 5.3 APC Image Rotated Features



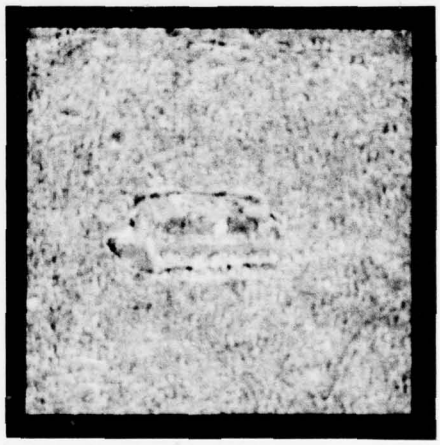
Rotated Feature 5



Rotated Feature 6

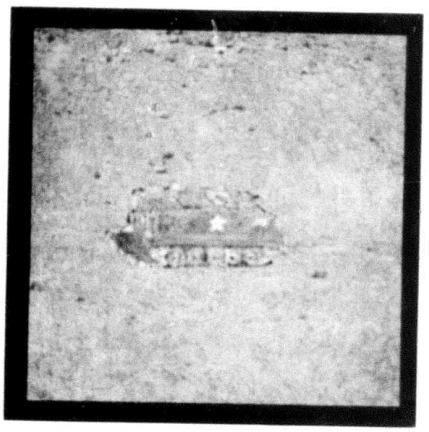


Rotated Feature 7

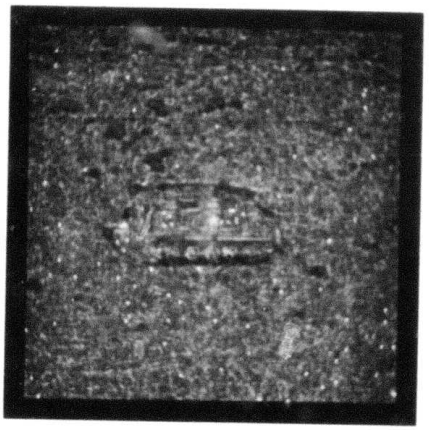


Rotated Feature 8

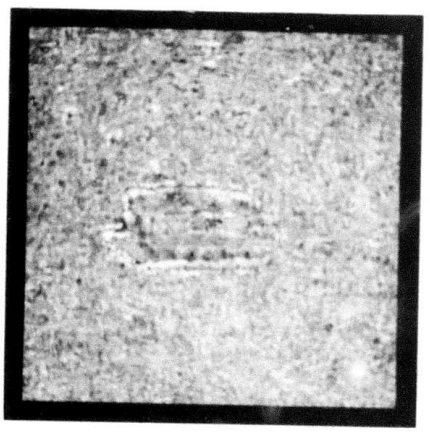
Figure 5.3 (continued) APC Image Rotated Features



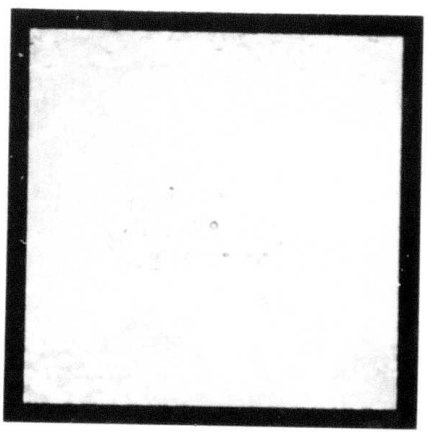
Rotated Feature 9



Rotated Feature 10

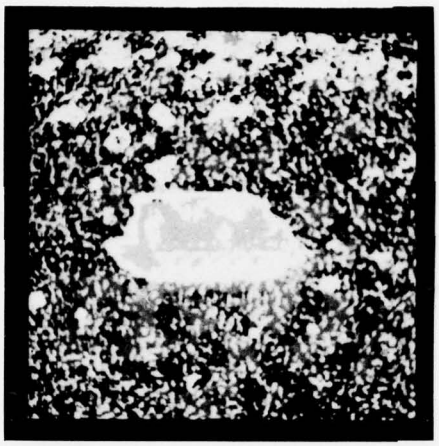


Rotated Feature 11

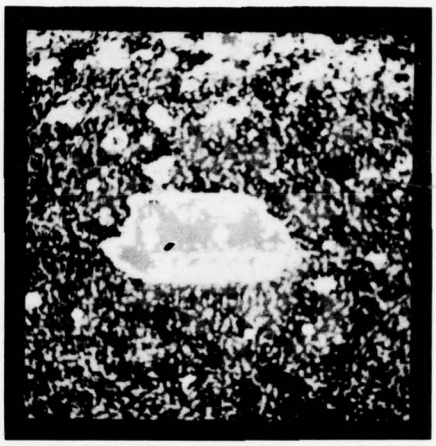


Rotated Feature 12

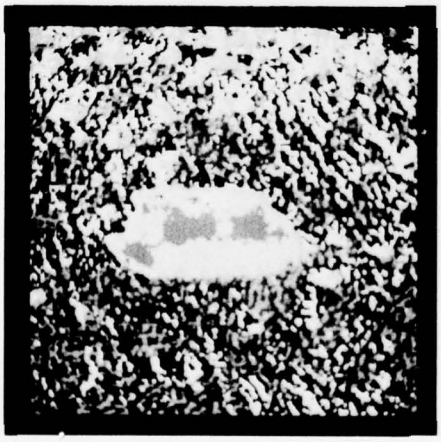
Figure 5.3 (continued) APC Image Rotated Features



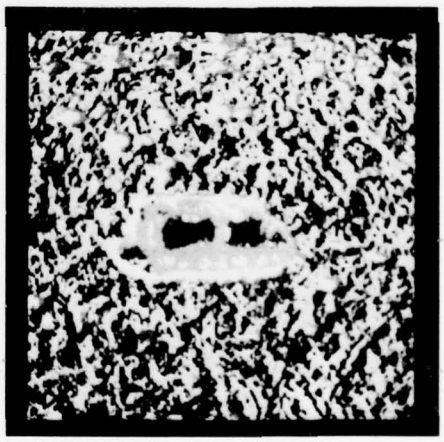
6 Clusters -
Most Similar to
4 Best Rotated
Segmentations



7 Clusters

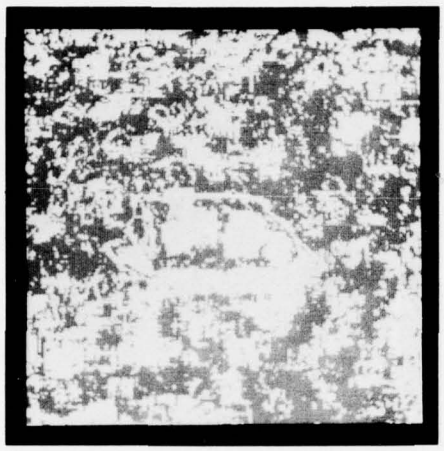


8 Clusters -
Product Maximum

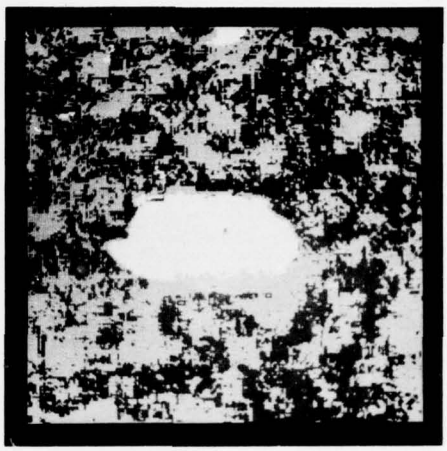


12 Clusters -
Most Similar to
12 Non-Rotated
Segmentations

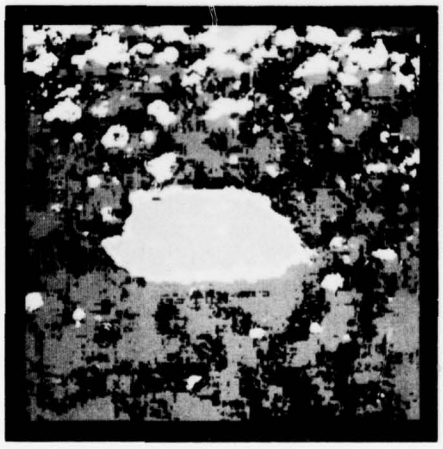
Figure 5.4 Segmentations - 12 APC Image Rotated Features



2 Clusters



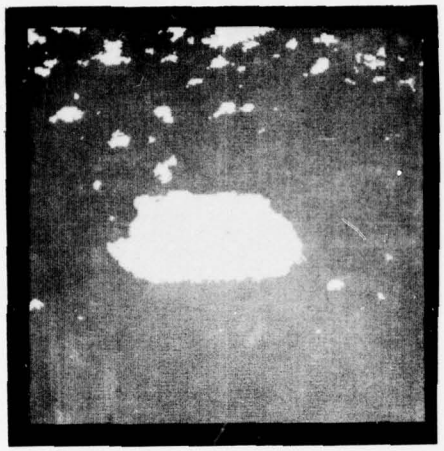
3 Clusters -
Product Maximum



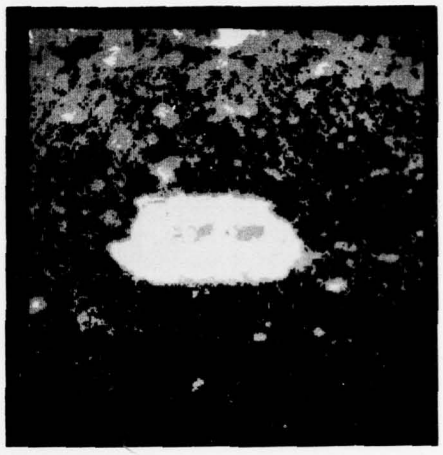
4 Clusters

6 Clusters - Most
Similar to 12 Rotated
Segmentations

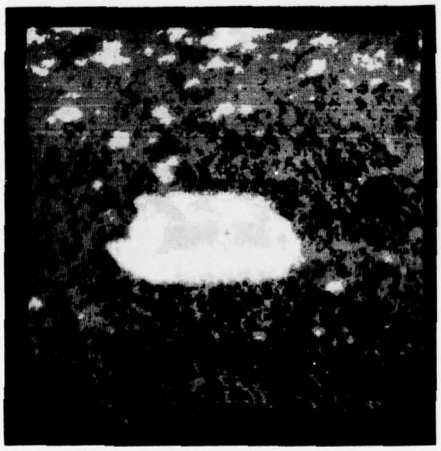
Figure 5.4 (continued) Segmentations 4 Best APC
Rotated Features



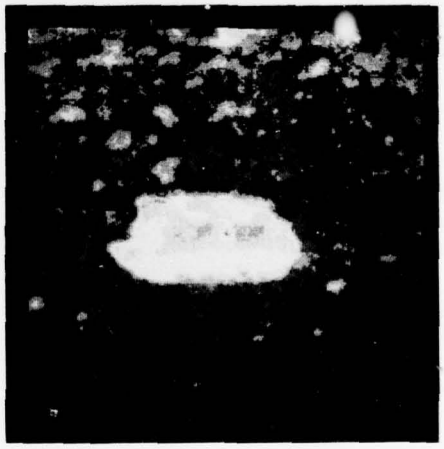
2 Clusters -
Product Maximum



3 Clusters

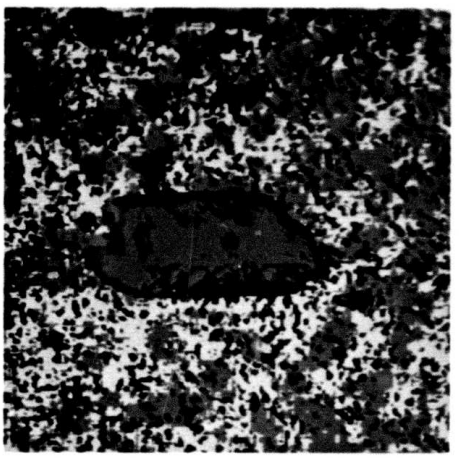


4 Clusters

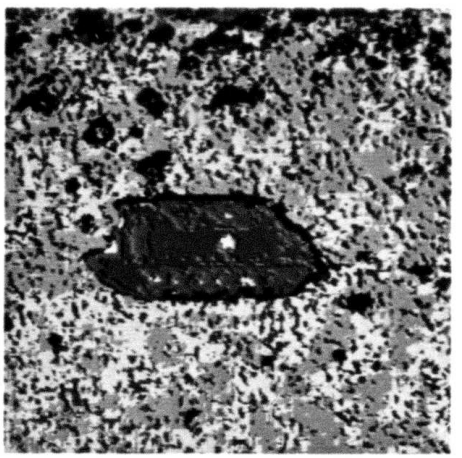


5 Clusters

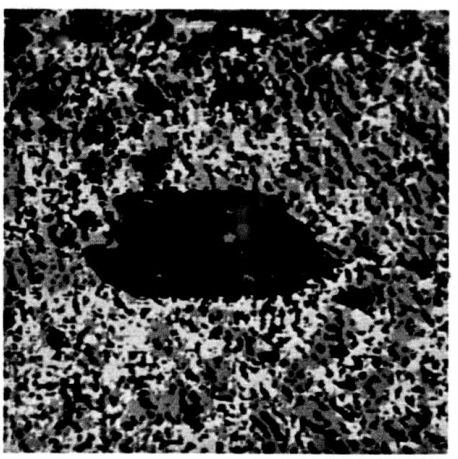
Figure 5.4 (continued) Segmentations - 1 Best APC Rotated Feature



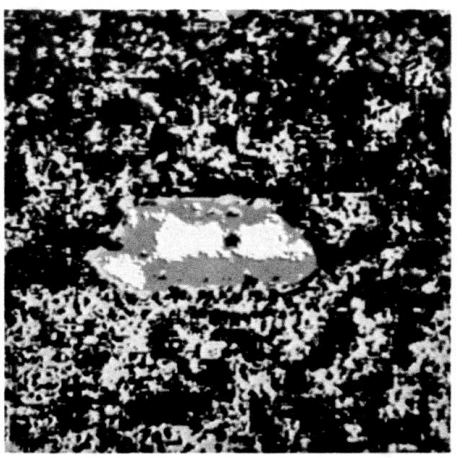
6 REGIONS



7 REGIONS



8 REGIONS
(BEST NUMBER
OF REGIONS)



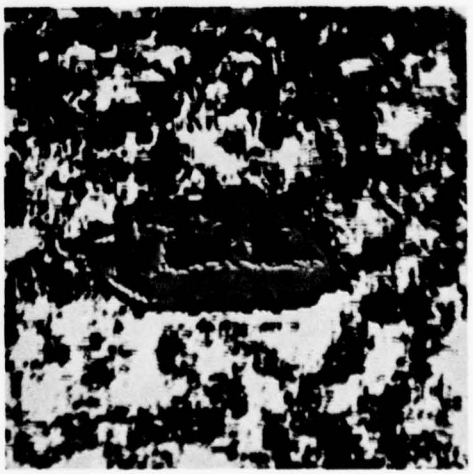
9 REGIONS

FIGURE 4. 12 NON REDUCED DECORRELATED FEATURES

2 REGIONS

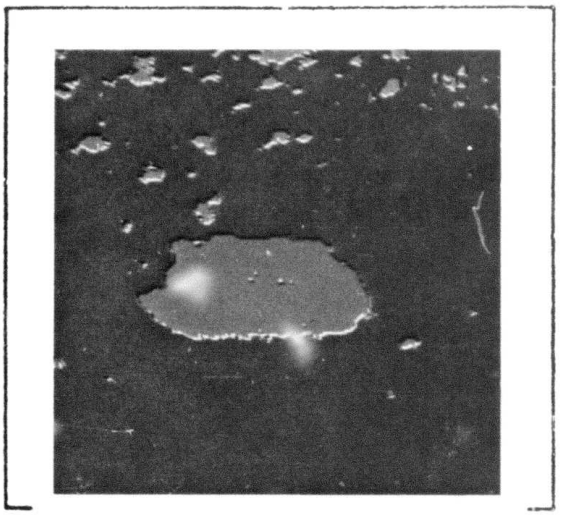
3 REGIONS
(BEST NUMBER
OF REGIONS)

4 REGIONS

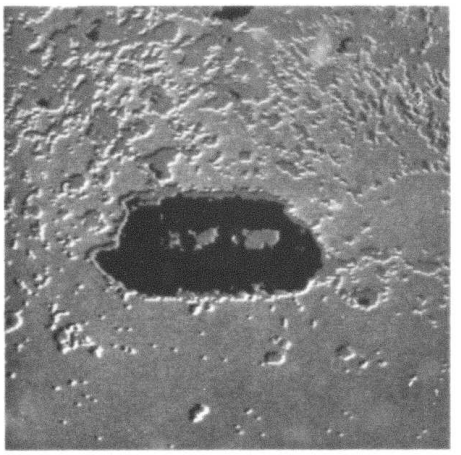


5 REGIONS

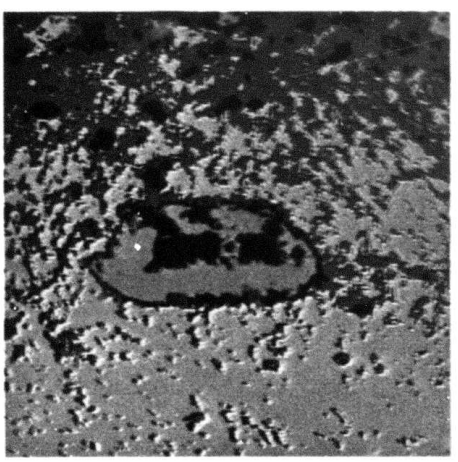
FIGURE 5. 4 REDUCED DECORRELATED FEATURES



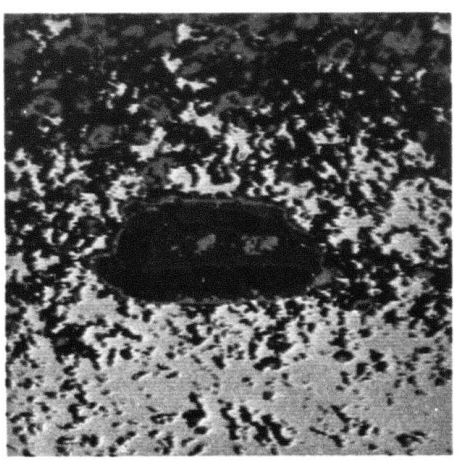
2 REGIONS
(BEST NUMBER
OF REGIONS)



3 REGIONS



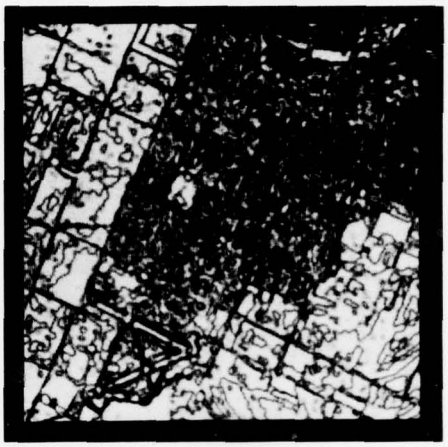
4 REGIONS



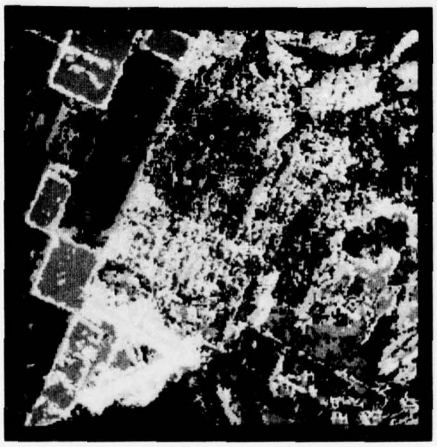
5 REGIONS

FIGURE 6. SINGLE BEST DECORRELATED FEATURE

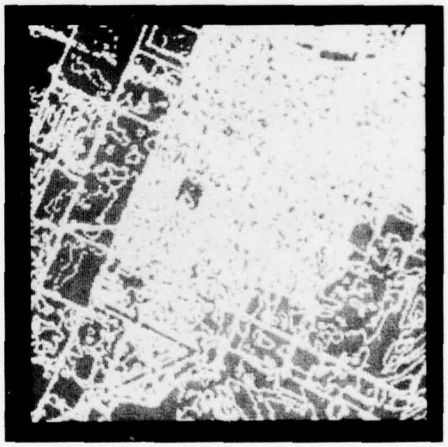
Original



Dispersion 3 x 3

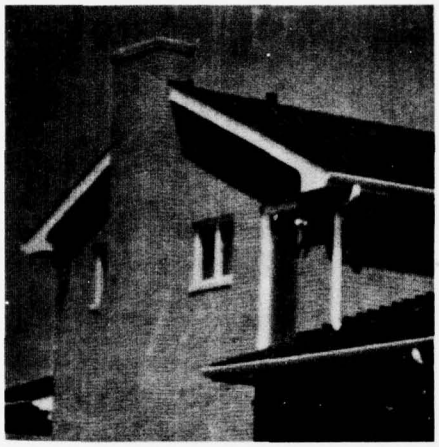


7 Clusters - Product Maximum
5 Best of 12 Rotated Features

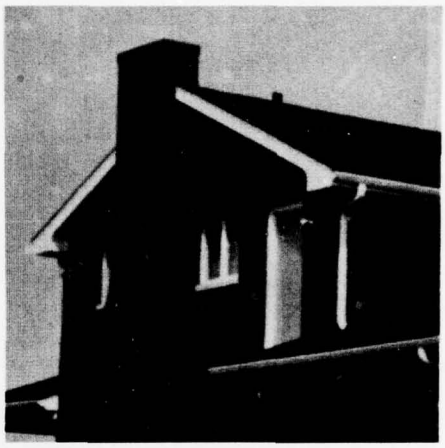


2 Clusters - Product Maximum
1 Best of 4 Rotated Features

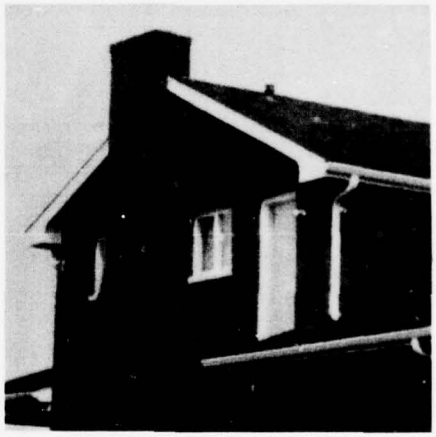
Figure 5.5 Aerial Image Results



Red Original



Green Original

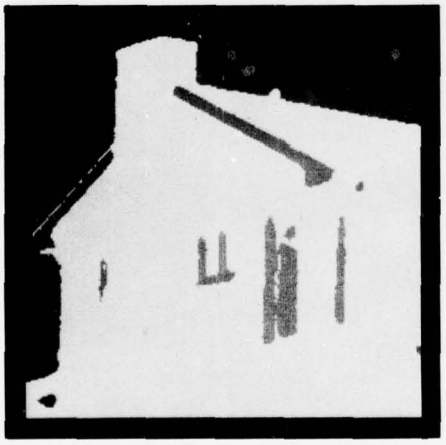


Blue Original

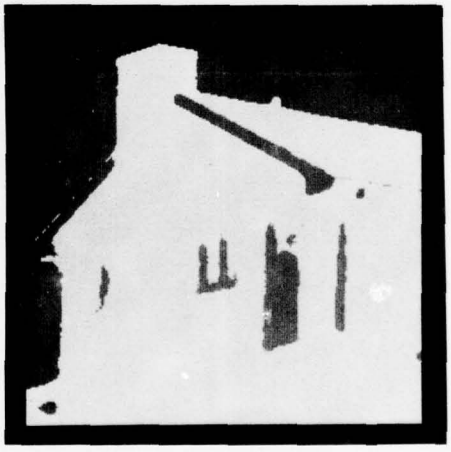
Figure 5.6. House Image Results

Red Dispersion, 3 x 3
Window

Red Dispersion, 7 x 7
Window

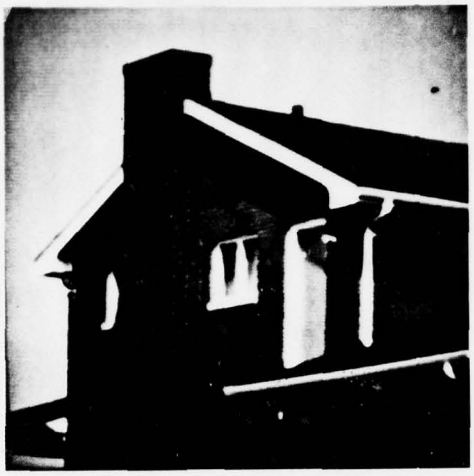


2 Clusters - Product Maximum
15 Rotated Features



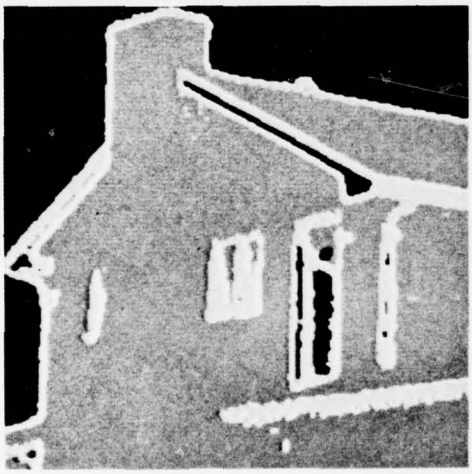
2 Clusters - Product Maximum
1 Best Rotated Feature

Figure 5.6 (continued) House Image Results

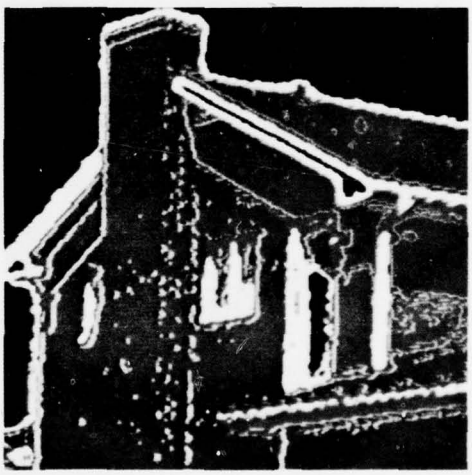


(a) House Original

(b) 2 Regions (Best Number of Regions)



(c) 3 Regions



(d) 4 Regions

(e) 5 Regions

(f) 6 Regions

Figure 7. Segmentation of House Picture.

(g) 7 Regions

(h) 8 Regions

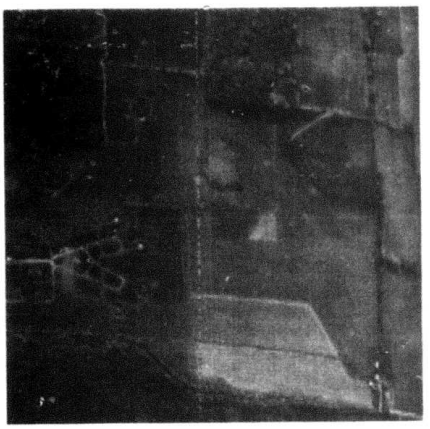
(i) 9 Regions

(j) 10 Regions

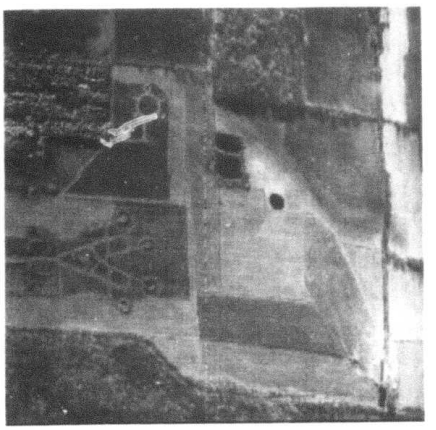
(k) 11 Regions

(l) 12 Regions

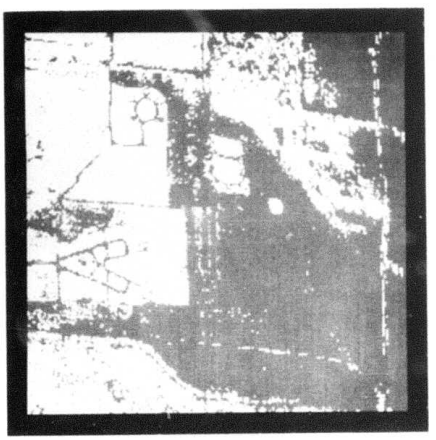
Figure 8. Segmentation of House Picture.



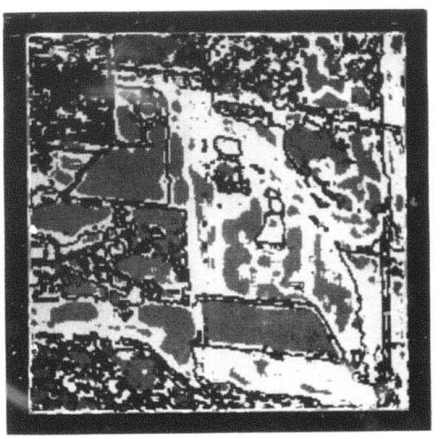
Original - Band 2



Original - Band 8



2 Clusters
Product Maximum -
10 Rotated Features



3 Clusters
Product Maximum -
2 Rotated Features
Augmented (6 Features Total)

Figure 5.7 Multispectral Images Results

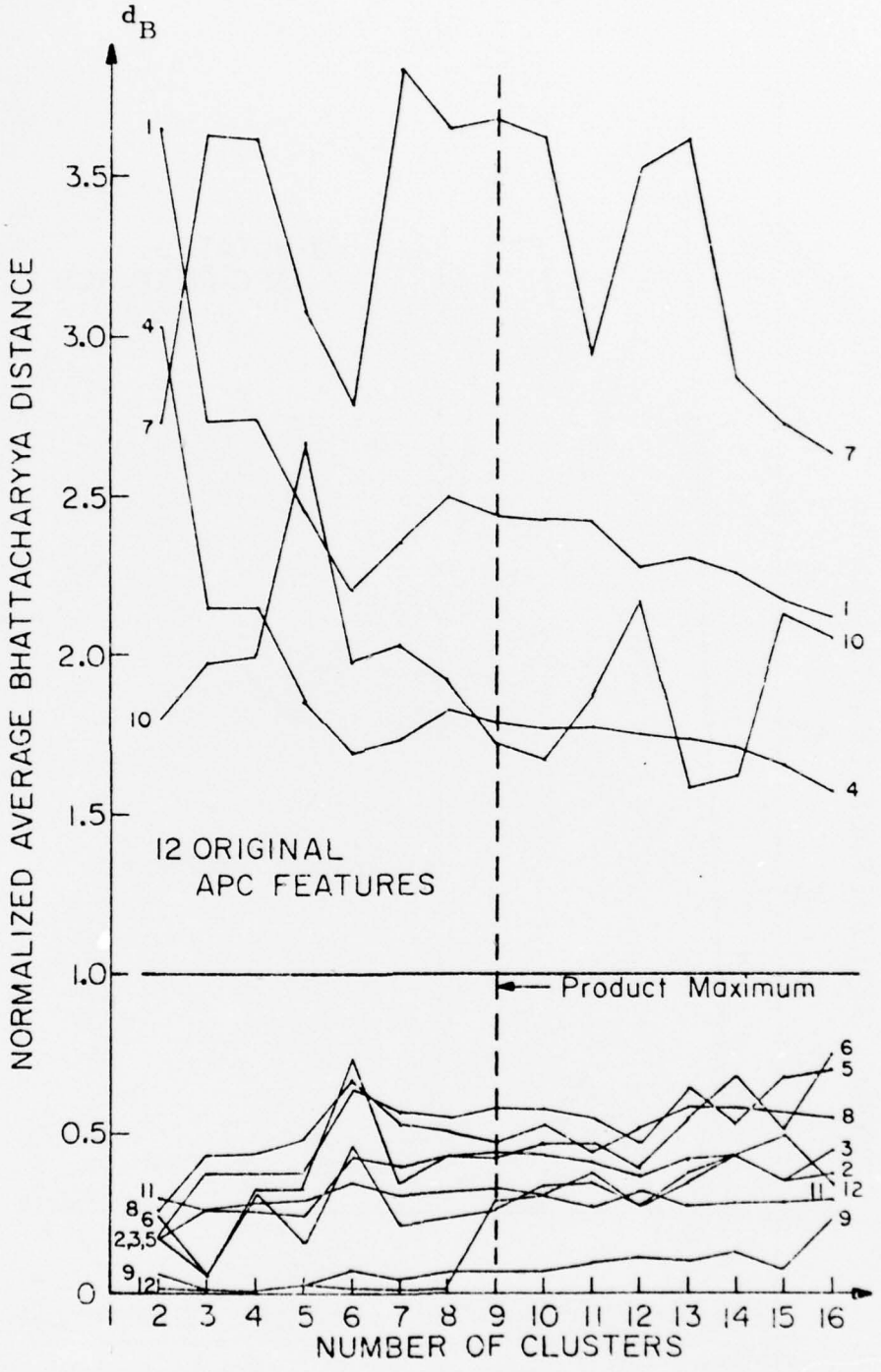


Figure 5.8 Average Bhattacharyya Distance vs. Number of Clusters

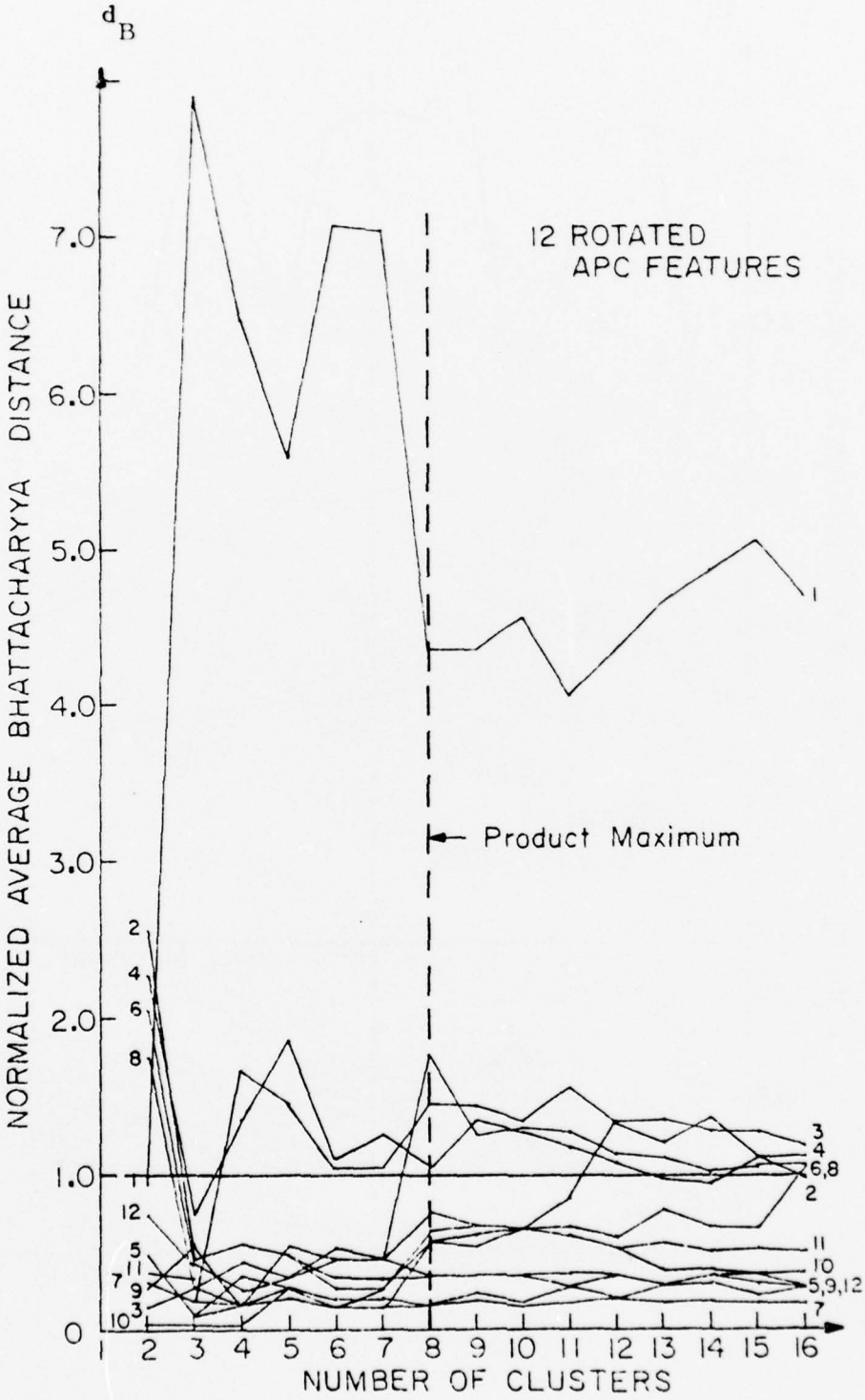


Figure 5.8 (Continued) Average Bhattacharyya Distance vs. Number of Clusters

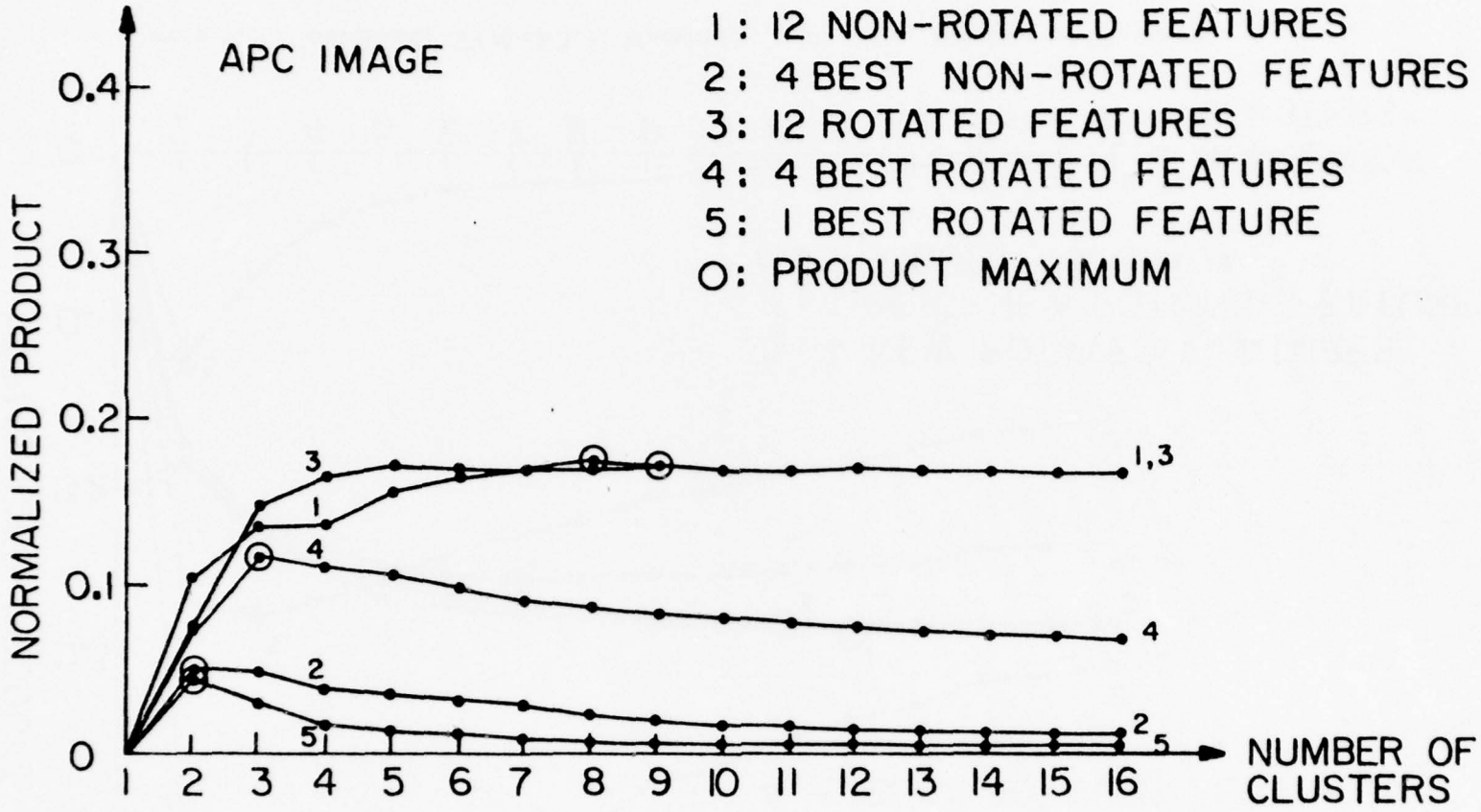


Figure 5.9 Probability Weighted Product vs. Number of Clusters

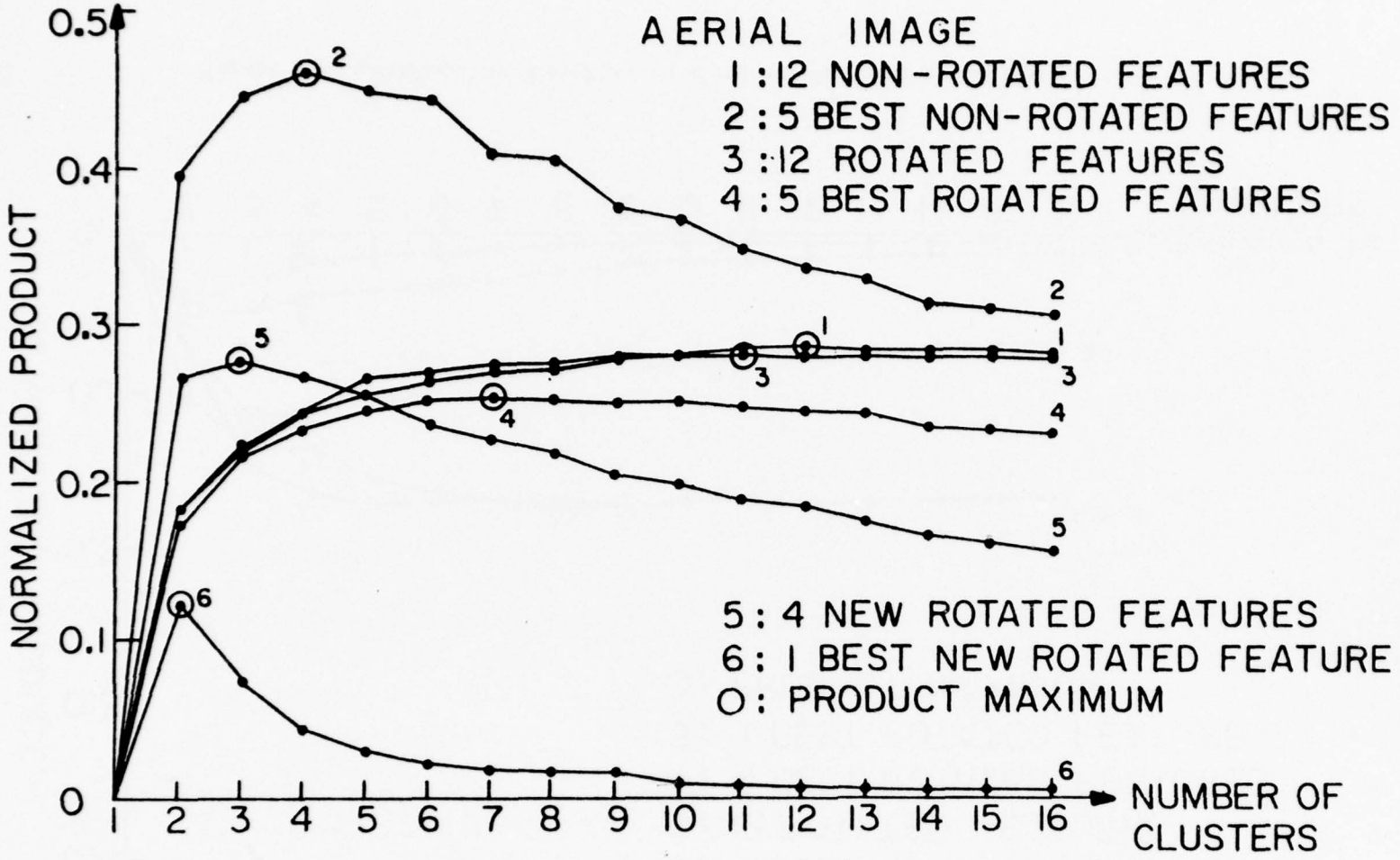


Figure 5.9 (Continued) Probability Weighted Product vs. Number of Clusters

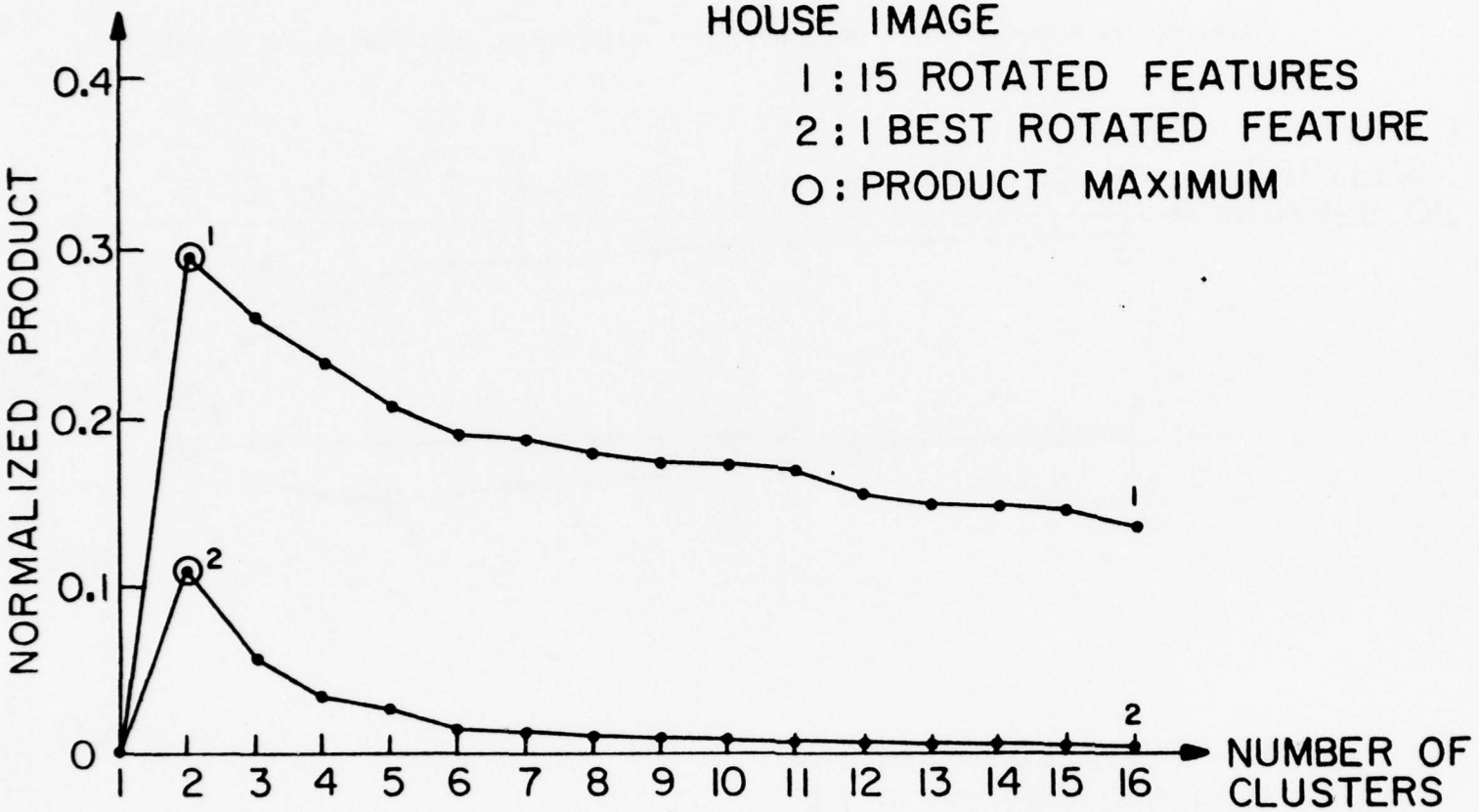


Figure 5.9 (Continued) Probability Weighted Product vs. Number of Clusters

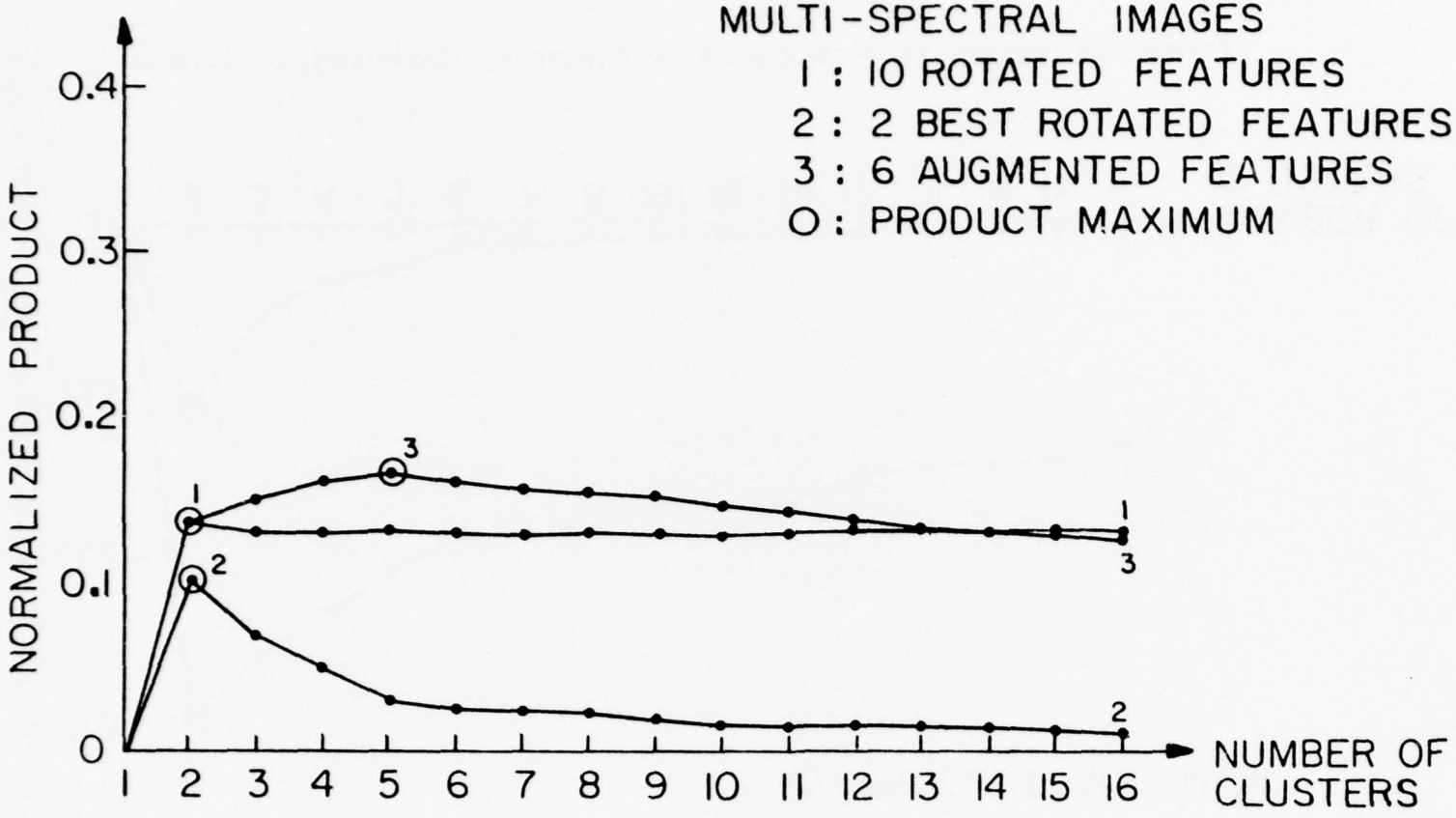


Figure 5.9 (Continued) Probability Weighted Product vs. Number of Clusters

TABLE 5.1 COVARIANCE, EIGENVALUES AND EIGENVECTORS

APC IMAGE - ORIGINAL FEATURES

COVARIANCE MATRIX

2030	-564	-382	2305	-1393	-252	2341	-1218	-185	2426	-708	540
-564	530	96	-573	952	-13	-612	793	256	-647	465	86
-382	96	2645	-466	88	1664	-533	1	1239	-631	-277	-73
2305	-573	-466	2798	-1477	-378	2726	-1337	-241	2895	-788	665
-1393	952	88	-1477	2913	-29	-1585	2109	481	-1282	1418	60
-252	-13	1664	-378	-29	3187	-292	-262	1846	-389	6	407
2341	-612	-533	2726	-1585	-292	2906	-1434	-409	3030	-817	635
-1218	793	1	-1337	2109	-262	-1434	2045	216	-1374	1172	-72
-185	256	1239	-241	481	1846	-409	216	2705	-780	238	805
2426	-647	-631	2985	-1282	-389	3030	-1374	-780	3995	-536	727
-708	465	-277	-788	1418	6	-817	1172	238	-536	1896	548
540	86	-73	665	60	407	635	-72	805	727	548	3484

EIGENVALUES

13734	6144	4552	2391	1331	1253	753	420	49	177	226	104

EIGENVECTORS **COLUMN VECTORS**

-0.37	0.06	-0.07	0.11	-0.06	0.17	-0.09	0.24	0.81	0.29	-0.07	0.10
0.13	-0.02	-0.14	0.10	0.05	0.13	0.11	0.14	0.29	-0.87	-0.21	-0.12
0.10	0.47	0.20	0.25	0.76	0.06	-0.30	0.01	-0.01	0.02	-0.05	0.04
-0.42	0.05	-0.12	0.17	-0.03	0.23	-0.06	0.24	-0.32	0.09	-0.20	-0.71
0.32	-0.09	-0.42	0.44	0.07	0.10	0.36	-0.26	0.01	0.30	-0.47	0.05
0.08	0.63	0.09	0.15	-0.29	-0.54	0.34	0.27	0.03	0.02	-0.04	-0.09
-0.44	0.04	-0.11	0.16	-0.04	0.08	0.02	0.29	-0.39	-0.14	-0.22	0.67
0.28	-0.13	-0.30	0.28	0.09	0.15	0.12	0.52	-0.07	0.07	0.64	0.01
0.11	0.53	-0.13	0.05	-0.43	0.56	-0.19	-0.32	-0.05	-0.06	0.19	0.08
-0.48	-0.01	-0.23	0.36	0.12	-0.30	0.10	-0.51	0.06	-0.17	0.42	-0.04
0.17	-0.06	-0.41	0.13	-0.19	-0.41	-0.75	0.06	-0.01	-0.02	-0.03	-0.00
-0.09	0.27	-0.63	-0.64	0.29	-0.04	0.14	0.02	-0.00	0.04	0.01	0.00

TABLE 5.1 (Continued) COVARIANCE, EIGENVALUES AND EIGENVECTORS

AERIAL IMAGE - ORIGINAL FEATURES

COVARIANCE MATRIX

784	-136	-374	656	-136	-629	648	-31	-72	616	264	-16
-136	586	228	-168	982	536	-159	889	561	-185	831	126
-374	228	3213	-378	292	2843	-336	-96	678	-548	313	455
656	-168	-378	856	-269	-453	726	-172	-142	644	-344	38
-136	982	292	-269	2318	861	-148	1896	1217	-261	1635	234
629	536	2843	-458	861	5176	-511	562	1373	-836	985	1039
648	-159	-336	726	-148	-511	807	-147	-240	704	-342	180
-31	889	-96	-172	1896	562	-147	2313	905	-317	1578	17
-72	561	678	-142	1217	1373	-240	905	4153	-379	703	-286
616	-185	-548	644	-261	-836	704	-317	-379	905	-403	-23
-264	831	313	-344	1635	985	-342	1578	703	-403	2454	434
-16	126	455	38	234	1039	180	17	-286	-23	434	3347

EIGENVALUES

9814	5526	3750	2893	2028	1113	817	438	128	159	181	64

EIGENVECTORS COLUMN VECTORS

-0.10	0.06	-0.03	-0.32	0.34	-0.04	0.05	-0.24	-0.46	-0.70	-0.07	0.02
0.16	0.17	-0.07	-0.05	-0.04	0.06	0.09	-0.16	-0.76	0.44	0.08	0.34
0.37	-0.43	0.12	-0.08	-0.27	0.75	0.02	0.13	0.04	0.06	0.04	-0.04
-0.11	0.01	-0.03	0.35	-0.36	-0.09	-0.10	0.17	-0.06	0.48	-0.49	-0.47
0.32	0.40	-0.15	-0.02	-0.15	0.16	0.33	-0.60	-0.13	0.22	-0.33	-0.17
0.60	-0.44	0.01	0.01	-0.22	-0.62	0.02	-0.08	-0.02	-0.07	0.03	0.01
-0.11	0.02	-0.08	0.35	-0.36	0.01	-0.01	-0.05	-0.38	0.12	-0.12	0.75
0.26	0.45	0.17	-0.09	-0.20	-0.07	0.40	0.60	0.16	0.09	0.30	-0.05
0.37	0.27	0.59	0.56	0.35	0.05	-0.09	0.04	0.03	0.02	0.05	0.01
-0.15	0.03	-0.05	0.30	-0.33	0.01	-0.11	-0.38	0.06	0.09	0.73	-0.27
0.31	0.33	-0.28	-0.18	-0.05	0.05	-0.82	0.05	0.07	-0.03	-0.01	0.03
0.14	-0.21	-0.70	0.46	0.45	0.11	0.08	0.06	-0.01	0.00	0.03	-0.04

TABLE 5.1 (Continued) COVARIANCE,EIGENVALUES & EIGENVECTORS

AERIAL IMAGE - ORIGINAL NEW FEATURES

COVARIANCE MATRIX

856	174	726	335
174	5557	234	1944
726	234	807	306
335	1944	306	1733

EIGENVALUES

6413	1647	104	790

EIGENVECTORS

-0.06	0.67	0.70	0.25
-0.92	-0.23	0.02	0.33
-0.07	0.63	-0.71	0.30
-0.39	0.32	-0.03	-0.86

HOUSE IMAGE -ORIGINAL FEATURES

TABLE 5.1 (Continued) COVARIANCE, EIGENVALUES & EIGENVECTORS

COVARIANCE MATRIX

1208	1081	832	1349	-677	1149	32	916	-233	1559	344	1137	200	1094	235
1081	2903	3130	1314	662	3045	-253	3609	-5	1417	1330	3062	1843	4382	2404
832	3130	3587	1034	1240	3243	-241	4080	226	1049	1592	3256	2355	4958	3111
1349	1314	1034	1653	-770	1455	-26	1227	-367	1821	446	1453	310	1489	372
-677	662	1240	-770	3138	555	766	1225	1732	-943	1078	553	1719	1515	2406
1149	3045	3243	1455	555	3414	-701	3932	-493	1595	1260	3355	1756	4765	2206
32	-253	-241	-26	766	-701	3662	-673	2523	15	734	-404	1204	-683	1309
916	3609	4080	1227	1225	3932	-673	5039	-503	1340	1637	3984	2477	6075	3315
-233	-5	226	-367	1732	-493	2523	-503	4320	-423	1119	-308	1654	-332	2184
1559	1417	1049	1821	-943	1595	15	1340	-423	2215	485	1637	305	1647	322
344	1330	1592	446	1078	1260	734	1637	1119	485	1613	1340	1819	2081	2465
1137	3062	3256	1453	553	3355	-404	3984	-308	1637	1340	3412	1890	4851	2463
200	1843	2355	310	1719	1756	1204	2477	1654	305	1819	1890	3299	3198	3623
1094	4382	4958	1489	1515	4765	-683	6075	-332	1647	2081	4851	3198	7429	4202
235	2404	3111	372	2406	2206	1309	3315	2184	322	2465	2463	3623	4202	5568

EIGENVALUES

31349	10904	4897	1514	1391	926	602	362	273	96	69	31	13	7	25

EIGENVECTORS COLUMN VECTORS

-0.09	0.14	-0.36	0.04	-0.15	0.20	-0.01	0.04	-0.31	0.22	-0.22	0.36	-0.67	0.04	-0.02
-0.29	0.09	-0.07	0.15	0.01	-0.07	0.02	-0.17	-0.39	0.06	-0.15	-0.47	0.03	-0.65	0.14
-0.33	0.02	0.07	0.12	0.03	-0.12	0.02	-0.20	-0.56	0.25	0.08	0.01	0.32	0.54	-0.20
-0.11	0.17	-0.40	0.00	-0.16	0.23	-0.02	0.10	-0.11	-0.31	0.76	-0.09	0.03	0.00	0.05
-0.12	-0.35	0.35	0.43	0.01	0.73	-0.07	0.12	-0.01	0.01	0.00	-0.06	-0.03	0.01	-0.00
-0.31	0.17	-0.03	0.18	0.01	-0.01	-0.09	-0.06	0.01	-0.67	-0.30	0.41	0.20	-0.12	-0.28
0.00	-0.41	-0.46	-0.05	0.76	0.07	0.15	-0.06	-0.03	-0.05	-0.02	0.02	0.02	-0.00	-0.05
-0.39	0.11	0.15	0.06	0.19	-0.06	0.11	0.05	0.14	0.24	0.17	0.49	0.14	-0.14	0.61
-0.03	-0.54	-0.27	0.53	-0.36	-0.44	0.03	0.06	0.13	0.03	0.06	0.07	-0.00	-0.01	0.04

HOUSE IMAGE -ORIGINAL FEATURES

TABLE 5.1 (Continued) COVARIANCE,EIGENVALUES & EIGENVECTORS

EIGENVECTORS (Continued)

-0.12	0.20	-0.48	0.00	-0.16	0.28	-0.05	0.20	0.33	0.37	-0.29	-0.09	0.47	0.02	-0.09
-0.16	-0.17	-0.08	-0.19	-0.19	0.17	-0.12	-0.85	0.30	0.05	0.06	0.03	-0.06	0.00	0.01
-0.32	0.13	-0.07	0.10	0.07	-0.06	0.04	0.04	0.22	-0.29	-0.24	-0.44	-0.26	0.49	0.41
-0.24	-0.29	-0.00	-0.34	0.01	-0.09	-0.81	0.24	-0.09	0.00	-0.01	-0.01	-0.01	-0.01	0.08
-0.47	0.11	0.15	0.05	0.18	-0.16	0.05	0.13	0.36	0.22	0.24	-0.11	-0.31	-0.10	-0.55
-0.32	-0.39	0.05	-0.54	-0.31	0.08	0.52	0.22	-0.09	-0.12	-0.10	0.00	0.04	-0.03	-0.02

TABLE 5.1(continued) COVARIANCE, EIGENVALUES & EIGENVECTORS

MULTI-SPECTRAL IMAGES
ORIGINAL FEATURES

COVARIANCE MATRIX

544	614	477	429	409	452	426	463	400	92
614	728	573	515	493	545	523	579	518	115
477	573	504	446	442	455	457	520	466	85
429	515	446	406	407	424	390	419	380	66
409	493	442	407	434	434	351	340	317	43
452	545	455	424	434	460	380	373	350	62
426	523	457	390	351	380	508	664	602	139
463	579	520	419	340	373	664	963	857	206
400	518	466	380	317	350	602	857	825	238
92	115	85	66	43	62	139	206	238	187

EIGENVALUES

4454	810	143	108	24	10	7	2	1	2

EIGENVECTORS COLUMN VECTORS

-0.32	0.25	-0.28	0.54	-0.20	0.55	-0.35	0.03	-0.05	-0.04
-0.38	0.25	-0.23	0.37	0.38	-0.34	0.52	0.15	0.16	0.14
-0.33	0.16	0.15	-0.14	-0.39	0.11	0.49	-0.49	-0.43	-0.04
-0.28	0.22	0.10	-0.19	-0.09	-0.10	-0.10	-0.19	0.60	-0.63
-0.27	0.32	0.16	-0.48	-0.21	0.21	0.03	0.54	0.18	0.39
-0.29	0.32	-0.02	-0.24	0.37	-0.30	-0.45	0.05	-0.55	-0.16
-0.33	-0.16	0.12	0.02	0.00	-0.22	-0.36	-0.50	0.26	0.60
-0.40	-0.52	0.28	0.25	-0.35	-0.30	-0.06	0.40	-0.14	-0.18
-0.37	-0.50	-0.04	-0.25	0.51	0.51	0.12	-0.03	0.01	-0.09
-0.09	-0.22	-0.84	-0.32	-0.31	-0.19	-0.02	0.01	-0.00	0.00

TABLE 5.1 (Continued) COVARIANCE, EIGENVALUES & EIGENVECTORS

MULTI-SPECTRAL IMAGES

ORIGINAL AUGMENTED FEATURES

COVARIANCE MATRIX

1545	-139	1010	-322	-149	-147	-103	-179
-139	4253	-75	2245	-250	736	-0	1741
1010	-75	1470	-487	-95	166	640	-166
-322	2245	-487	2324	-206	799	-282	1962
-149	-250	-95	-206	125	-91	79	64
-147	736	166	799	-91	2470	277	1348
-103	-9	640	-282	79	277	881	235
-179	1741	-116	1962	64	1348	235	4548

EIGENVALUES

8305	3002	2643	1832	1201	485	102	46

EIGENVECTORS COLUMN VECTORS

-0.06	0.20	-0.57	0.29	-0.53	0.13	-0.49	0.09
0.57	-0.54	-0.43	-0.10	0.22	0.36	0.01	-0.00
-0.05	0.33	-0.62	0.04	0.19	-0.29	0.56	-0.28
0.46	-0.22	0.07	0.02	-0.29	-0.78	-0.13	-0.15
-0.03	0.05	0.10	0.04	0.14	0.16	-0.34	-0.91
0.28	0.42	-0.00	-0.81	-0.26	0.13	-0.03	-0.03
0.01	0.24	-0.16	-0.13	0.67	-0.29	-0.55	0.25
0.61	0.53	0.25	0.48	0.09	0.19	0.09	0.07

APC Image

Rotated Feature No.	Eigenvalue	Normalized Average Bhattacharyya Distance	Rank
1	849.3	4.39	1
2	377.3	0.61	6
3	279.3	1.46	3
4	149.3	1.05	4
5	81.8	0.17	11
6	78.6	0.72	5
7	46.8	0.17	11
8	26.4	1.72	2
9	3.2	0.34	9
10	11.1	0.57	7
11	13.6	0.53	8
12	6.6	0.27	10

Overall Average = .36
(Normalizing factor)

Aerial Image

Rotated Feature No.	Eigenvalue	Normalized Average Bhattacharyya Distance	Rank
1	550.6	3.02	1
2	308.6	2.51	2
3	209.7	1.11	5
4	161.7	1.28	4
5	115.1	1.80	3
6	62.9	0.30	8
7	46.7	0.19	11
8	25.2	0.29	9
9	7.2	0.17	12
10	8.9	0.23	10
11	10.8	0.77	6
12	3.9	0.32	7

Overall Average = 0.19
(Normalizing factor)

Table 5.2 Eigenvalues vs. Bhattacharyya Distances

House Image

Rotated Feature No.	Eigenvalue	Normalized Average Bhattacharyya Distance	Rank
1	1559.2	12.35	1
2	541.8	0.09	9
3	242.8	0.24	5
4	76.3	0.03	11
5	69.4	0.09	9
6	47.2	0.15	8
7	30.3	0.15	8
8	17.6	0.04	10
9	13.3	0.17	6
10	4.7	0.16	7
11	3.8	0.24	5
12	1.9	0.41	3
13	0.8	0.15	8
14	0.4	0.25	4
15	1.3	0.47	2

Overall Average = .43
(Normalizing factor)

Multi-Spectral Images (Non-Augmented)

Rotated Feature No.	Eigenvalue	Normalized Average Bhattacharyya Distance	Rank
1	508.3	4.88	1
2	93.6	0.45	6
3	16.2	0.02	9
4	12.3	0.02	9
5	3.1	0.58	5
6	1.3	0.38	7
7	0.9	0.23	8
8	0.4	0.66	4
9	0.2	0.96	3
10	0.3	1.82	2

Overall Average = .18
(Normalizing factor)

Table 5.2 (Continued) Eigenvalues vs. Bhattacharyya Distances

12 Feature Set - Full Runs

Operation	CPU Time Required (Hours : Minutes: Seconds)
Compute Basic Features	1: 54
3 x 3 Mode Filters	4: 02
7 x 7 Mode Filters	9: 48
15 x 15 Mode Filters	36: 02
Total Feature Computation	51: 46
Feature Rotation	8: 09
Initial Clustering	1:31: 32
Final Clustering	21: 10
Segmentation	1: 13
TOTAL	2:53: 50

4 Feature Set - Abbreviated Run (Typical)

Operation	CPU Time Required (Minutes: Seconds)
3 x 3 Mode Filter	1: 25
7 x 7 Mode Filter	3: 56
Total Feature Computation	5: 21
Feature Rotation	1: 33
Initial Clustering	6: 10
Final Clustering	1: 59
Segmentation	1: 35
TOTAL	16: 38

Table 5.3 Computer Time

APC Image	Comparison in Percent			
No. of Clusters	12 Non-Rotated vs. 12 Rotated	12 Rotated vs. 4 Best Rotated	4 Best Non-Rotated vs. 4 Best Rotated	4 Best Non-Rotated vs. 1 Best Rotated
2	44	59	31	96
3	51	59	50	73
4	35	61	45	56
5	50	67	43	44
6	41	68	41	44
7	44	57	37	35
8	36	49	29	48
9	59	47	28	46
10	53	46	28	29
11	72	52	16	28
12	74	46	14	28
13	52	37	24	25
14	50	34	24	38
15	53	36	23	34

$$C = \left[\sum_{i,j} h_{MAX}(i,j) \quad \frac{1}{N} - \frac{1}{I \cdot J} \quad MIN(I, J) \right] \Gamma$$

$$\Gamma = \frac{100}{1 - \frac{1}{I \cdot J} \quad MIN(I, J)}$$

Table 5.4 Comparison Measure Results

Chapter 6

REAL TIME IMPLEMENTATION

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

6.1 Feature Computation

Figure 6.1 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 the system is in development [6-1] on charge-coupled-device (CCD) hardware and may even be implemented on the focal plane of a multi-element 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.

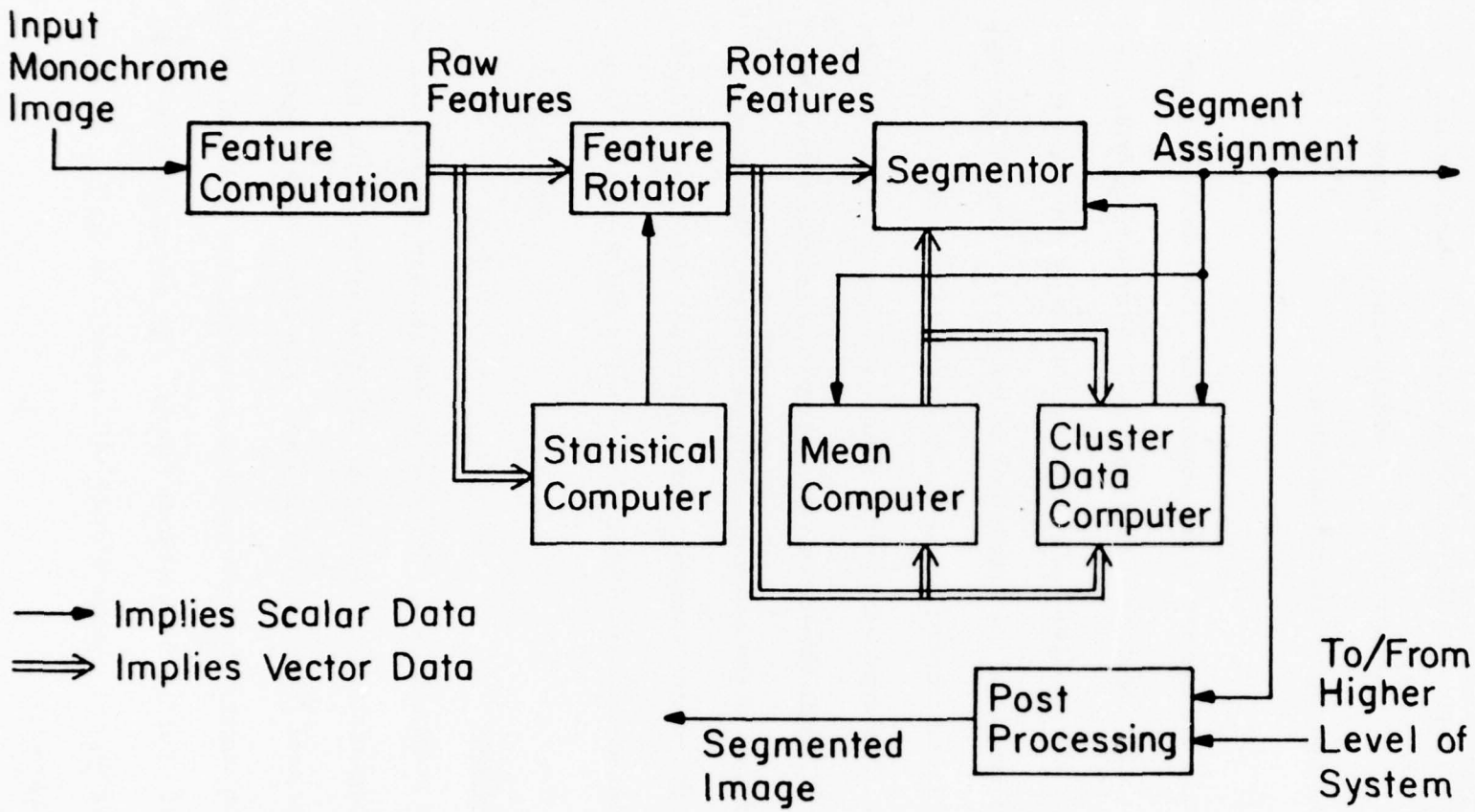


Figure 6.1 Real Time Segmentation System

The statistical computer will therefore not have to compute and diagonalize the covariance matrix in a single scan, but may take several scans to perform this computation.

6.2 Segmentor

The heart of the system is the segmentor. This device accepts the incoming rotated feature vector and the cluster means from the mean computer and assigns each incoming vector to the nearest cluster mean. The output of the segmentor is therefore a scalar corresponding to the index number of the cluster to which the vector has been assigned. The segmentor must accept an input from the cluster data computer through the mean computer which defines those features to be ignored in the assignment of the vectors. The ignoring of features in the vector assignment is equivalent to feature reduction in the algorithm which has been implemented in software on a digital computer.

6.3 Mean Computer

The mean computer accepts the incoming feature vectors and the output of the segmentor and recomputes the cluster means for use in segmenting the next frame of picture data. The effect of this procedure is that the current frame is always being segmented based on the means computed during the previous frame. The assumption is made that the cluster means change slowly with respect to the frame time (usually 1/60 second).

A functional diagram of the mean computer is shown in Figure 6.2. The vector switch accepts the input feature vector and switches it to the summing register specified by the output of the segmentor. The classification summing register also records the number of vectors in each cluster, for division of the vector sums at the completion of the frame. The output at the end of the frame consists of the vector sums in the vector summing registers divided by the number of vectors that contributed to each respective sum. These new cluster means are then forwarded to the segmentor for segmentation of the next incoming frame.

6.4 Cluster Data Computer

The purpose of the cluster data computer (Figure 6.1) is threefold. First, it decides when the mean computer/segmentor loop has converged for a fixed number of clusters. Convergence will be assumed to have occurred when the previous means and the current means differ by less than some amount in an appropriate norm. The second function of the cluster data computer will be to evaluate the rotated features with respect to usefulness and specify to the segmentor those features to be ignored. The remaining function of the cluster data computer is to decide on the number of clusters in the data and to specify to the mean computer how many clusters are present. At this point, the real time algorithm deviates from the computer algorithm as implemented currently. Implementation of the current

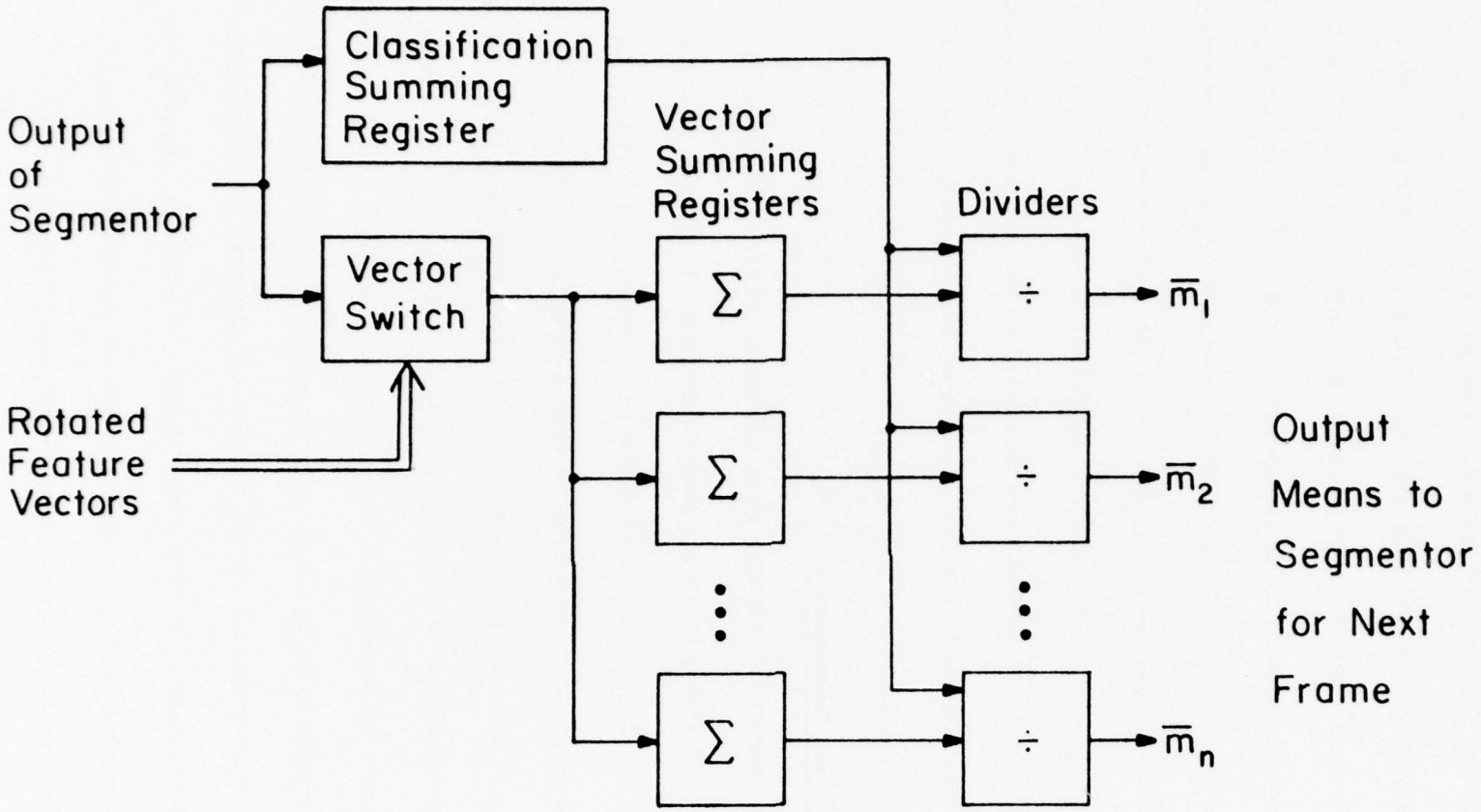


Figure 6.2 Mean Computer

algorithm would require three sets of segmentation hardware, that is, 3 segmentors and 3 mean computers. One of these sets would perform clustering for the assumed number of clusters, N. The other two sets would perform clustering for N-1 and N+1 clusters, respectively. The number of clusters would be incremented or decremented as necessary to maintain the quality parameter (see Chapter 4) at maximum for N clusters.

The requirement for 3 sets of segmentation hardware to determine the intrinsic number of clusters in the data is cumbersome and inefficient. It is suggested that alternate procedures be used to accomplish the same result with one set of segmentation hardware. A suggested starting point would be for the algorithm to attempt to maintain the ratio of within to between scatter measures at some fixed (possibly operator adjustable) value. Suitable hysteresis would be necessary to prevent a low level limit cycle about the fixed value.

The cluster data computer will most likely require more than one frame to compute the measures necessary to set the number of clusters. In addition, it must wait until the inner loop comprised of the segmentor and mean computer has converged in order to begin computation of these measures. Further, as the number of clusters is incremented or decremented, the cluster data computer must decide which cluster to eliminate or where to initialize a new cluster center. The suggested procedure is to combine the cluster center

pair having minimum separation and to initialize new clusters at the vector furthest from its respective cluster center, as is done currently by the computer algorithm. An alternate procedure would be to split the cluster having greatest variance into two clusters, each 1 standard deviation removed from the previous cluster center. It should be noted that an additional respect in which the real time procedure suggested here differs from the computer procedure described previously is that the real time procedure uses every pixel for every calculation. This differs from the computer algorithm in which a sample procedure is used to reduce computer time required.

6.5 Preliminary Functional Requirements

A preliminary estimate of the accuracy and overall register size necessary to achieve useful results for this system is given in Table 6.1. These values are derived from the subjective judgment of the author only, based on results obtained with the computer algorithm. Minimal, most probable, and maximal requirements are given since the cost of implementation in terms of hardware may be very non-linear and an intelligent compromise can often mean large hardware cost savings.

Parameter	Minimal	Most Probable	Maximal
Feature Precision (bits)	3	6	8
Number of Features	4	8	16
Max. Number of Clusters	6	16	32

Table 6.1 Automatic Real Time Segmentor Functional Requirements

6.6 Motion Picture Segmentation Results

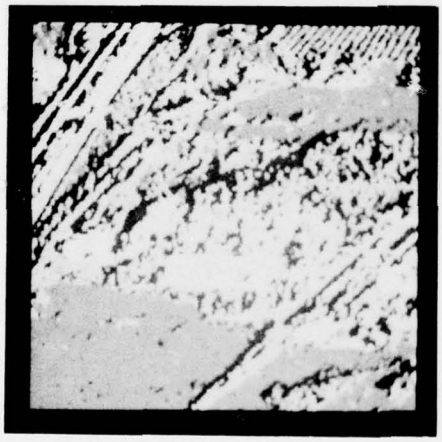
The algorithm described in Chapter 4 was used to segment two frames of a motion picture of a chemical plant. The results of these segmentations are shown in Figure 6.3, along with the original photographs. The motion picture was taken from a moving aircraft, and the originals are not spatially registered, 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. The mean vectors for the 12 feature sets are shown in Table 6.2. The first mean, which is the mean of the original image, differs between the two images by about 12%. This is presumed to be caused by frame to frame exposure/development differences between the two frames. The difference in means causes a corresponding difference in variances, and the rotation matrices for the two feature sets are sufficiently different to prevent ideal tracking of the cluster means, since they are effectively represented with respect to different bases.

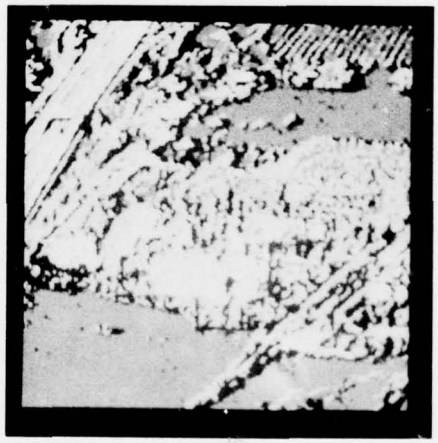
If a real time system is implemented, and frame to frame amplitude differences are expected, either appropriate scaling will be required or the rotation matrix will have to be forced to change slowly. The effect of this procedure would be to rotate image feature sets

Original - Frame 1

Original - Frame 5



Segmentation - Frame 1
(4 Clusters)



Segmentation - Frame 5
(4 Clusters)

Figure 6.3. Motion Picture Results.

Feature Mean	Frame 1	Frame 5
1	88	100
2	199	198
3	94	96
4	83	97
5	166	166
6	94	97
7	84	90
8	137	132
9	93	93
10	79	87
11	134	128
12	94	88

Table 6.2 Motion Picture Means

with a non-optimal rotation matrix. Since the rotation is performed to permit feature rejection in decorrelated space, the penalty for this procedure will most likely be small.

Chapter 7

CONCLUSIONS

This dissertation 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 set data 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 in appearance with weather, time of day and terrain makes an unsupervised procedure mandatory.

The procedure outlined herein lends itself 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, it is felt that such a system is well within the state of the art at this writing. 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.

A comparison measure was introduced and utilized to compare different segmentations of the same scene. This comparison measure would be particularly useful in comparing segmentations of an image performed by a candidate procedure with a standard segmentation of

the same image. In addition, the comparison measure may be the basis from which to proceed on defining a generalized cross-correlation function for use in cross-correlating different sensor outputs.

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 criterion, 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 twofold.

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.

REFERENCES

[2-1] P. H. Winston, "Heterarchy in the MIT Robot," MIT Artificial Intelligence Lab, Vision Flash 8, March 1971.

[2-2] A. L. Zobrist and W. B. Thompson, "Building a Distance Function for Gestalt Grouping," IEEE Trans. Comput., Vol. C-24, July 1975, pp. 711-719.

[2-3] J. M. Tenenbaum, "On Locating Objects by Their Distinguishing Features in Multisensory Images," Technical Note 84, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, Calif.

[2-4] C. A. Harlow and S. A. Esenbeis, "The Analysis of Radiographic Images," IEEE Trans. Comp., Vol. C-22, July 1973, pp. 678-689.

[2-5] E. C. Freuder, "Suggestion and Advice," MIT Artificial Intelligence Lab, Vision Flash 43, March 1973.

[2-6] E. C. Freuder, "Active Knowledge," MIT Artificial Intelligence Lab, Vision Flash 53, October 1973.

[2-7] L. B. Arguimbau, Vacuum-Tube Circuits, John Wiley, New York, 1948, p. 122.

[2-8] L. G. Roberts, "Machine Processing of Three-Dimensional Solids," in Optical and Electro Optical Information Processing, MIT Press, Cambridge, J. T. Tippett et al., eds., 1965, pp. 159-197.

[2-9] A. Guzman-Aerna, "Computer Recognition of Three-Dimensional Objects in a Visual Scene," MAC-TR-59, MIT Thesis (EE), Project MAC, December 1968.

[2-10] P. Winston, "Learning Structural Descriptions from Examples," MAC-TR-76, MIT Thesis (EE), Project MAC, September 1970.

[2-11] D. L. Waltz, "Generating Semantic Descriptions from Drawings of Scenes with Shadows," AI-TR-271, MIT Thesis (EE), AI Laboratory, November 1972.

[2-12] Y. Shirai, "A Heterarchial Program for Recognition of Polyhedra," AI Memo 263, MIT AI Laboratory, June 1972.

[2-13] G. R. Grape, "Model Based (Intermediate Level) Computer Vision," AIM-201, Stanford Univ. Thesis (CS), May 1973.

[2-14] M. D. Kelly, "Visual Identification of People by Computer," AIM-130, Stanford Univ. Thesis (CS), July 1970.

[2-15] T. M. Sakai and T. Kanode, "Computer Analysis and Classification of Photographs of Human Faces," Kyoto University Report, Kyoto, Japan, 1972.

[2-16] Y. Yakimovsky, "Scene Analysis Using a Semantic Base for Region Growing," AIM-209, Stanford Univ. Thesis (CS), July 1973.

[2-17] G. J. Agin, "Representation and Description of Curved Objects," AIM-173, Stanford Univ. Thesis (CS), October 1972.

[2-18] H. G. Barrow and R. J. Popplestone, "Relational Descriptions in Picture Processing," Machine Intelligence 6, B. Meltzer and D. Mitchie (Eds.), University Press, Edinburgh, 1970, pp. 377-396.

[2-19] J. M. Tenenbaum, et al., "An Interactive Facility for Scene Analysis Research," Artificial Intelligence Center Technical Note 87, Stanford Research Institute, Menlo Park, Calif., January 1974.

[2-20] K. Price, "Change Detection and Analysis in Multi-Spectral Images," Carnegie-Mellon Univ. Thesis (CS), 1976.

[2-21] A. Rosenfeld and M. Thurston, "Edge and Curve Detection for Visual Scene Analysis," IEEE Trans. Computers, C-20 May 1971, pp. 562-569.

[2-22] A. Rosenfeld, et al., "Edge and Curve Detection: Further Experiments," IEEE Trans. Computers, C-21 July 1972, pp. 677-715.

[2-23] A. Martelli, "Edge Detection Using Heuristic Methods," Comput. Graphics and Image Processing 1, 1972, pp. 169-182.

[2-24] M. J. Hueckel, "A Local Visual Operator Which Recognizes Edges and Lines," J. ACM 20, 4, October 1973, pp. 634-647.

[2-25] T. Pavlidis, "Segmentation of Pictures and Maps Through Functional Approximation," Comput. Graphics and Image Processing 1 1972 , pp. 360-372.

[2-26] S. L. Horowitz and Theodosios Pavlidis, "Picture Segmentation by a Tree Traversal Algorithm," J. ACM, 23, 4, April 1976, pp. 368-388.

[2-27] T. V. Robertson, et al., "Multispectral Image Partitioning," TR-EE 73-26, Purdue Univ., Lafayette, Indiana, August 1973.

[2-28] A. Klinger, "Data Structures and Pattern Recognition," Proc. First Int. Joint Conf. on Pattern Recognition, Washington, D. C. October 1973, pp. 497-498.

[2-29] C. R. Brice and C. L. Fennema, "Scene Analysis Using Regions," Artif. Intel. J. 1, Fall 1970, pp. 205-226.

[2-30] T. Pavlidis, "Linguistic Analysis of Waveforms," in Software Engineering, J. Tou, Ed., Academic Press, New York, 1971, p. 203-205.

[2-31] R. M. Haralick and G. L. Kelly, "Pattern Recognition with Measurement Space and Spatial Clustering for Multiple Images," Proc. IEEE, 57, 4, April 1969, pp. 654-665.

[2-32] R. M. Haralick, et al., "Textural Features for Image Classification," IEEE Trans. Sys. Man. and Cyber., SMC-3, 6, November 1973, pp. 610-621.

[2-33] R. M. Haralick and I. Dinstein, "A Spatial Clustering Procedure for Multi-Image Data," IEEE Trans. on Cir. and Sys., CAS-22, 5, May 1975. pp. 440-450.

[2-34] R. Ohlander, "Analysis of Natural Scenes," Carnegie-Mellon Univ. Thesis (CS), June 1975.

[3-1] B. L. Bullock, "Pattern Recognition Methodology," Hughes Research Laboratories, Research Report 450, December 1971.

[3-2] H. C. Andrews, Introduction to Mathematical Techniques in Pattern Recognition, Wiley, 1972. New York.

[3-3] W. K. Pratt, Digital Image Processing, to be published.

[3-4] R. O. Duda and P. E. Hart, Pattern Classification and Scene Analysis, Wiley, 1973. New York.

[3-5] R. C. Tyron, Cluster Analysis, Edwards Brothers, Ann Arbor, Mich. 1939.

[3-6] S. Watanabe, Methodologies of Pattern Recognition, Academic Press, New York, 1969.

[3-7] S. Watanabe, Knowing and Guessing, John Wiley and Sons, New York Press.

[3-8] G. H. Ball, "A Comparison of Some Cluster-Seeking Techniques," Stanford Research Institute, Technical Report No. RADC-TR-66-514, November 1966.

[3-9] R. E. Bonner, "A Logical Pattern Recognition Program," IBM J. Research and Development, July 1962. pp 353-359.

[3-10] R. E. Bonner, "On Some Clustering Techniques," IBM J. Research and Development, January 1964. pp. 22 - 31.

[3-11] G. H. Ball and D. J. Hall, "ISODATA, A Novel Method of Data Analysis and Pattern Classification," Stanford Research Institute, Menlo Park, Calif., April 1965.

[3-12] F. R. Fromm and R. A. Northouse, "CLASS: A Nonparametric Clustering Algorithm," Pattern Recognition, July 1976. pp. 107 - 114.

[3-13] G. Nagy, "State of the Art in Pattern Recognition," Proc. of IEEE, May 1968. pp. 836 - 862.

[3-14] K. Fukunaga, Introduction to Statistical Pattern Recognition, Academic Press, 1972. New York.

[3-15] H. C. Andrews, "A Poor Man's Feature Selector or Avoidance of Singular Covariance Matrices," unpublished tutorial, Image Processing Institute, University of Southern California, Los Angeles, California.

[3-16] T. Kailath, "The Divergence and Bhattacharyya Distance Measure in Signal Detection," IEEE Trans. on Communication Technology, 15, 1, 1967, pp. 52-60.

[3-17] T. L. Henderson and D. G. Lainiotis, "Comments on Linear Feature Extraction," IEEE Trans. PGIT, IT-15, 6, November 1969, pp. 728-730.

[3-18] A. Caprihon and R. J. P. de Fiquairedo, "On the Extraction of Pattern Features from Continuous Measurements," IEEE Symposium on Adaptive Processes, 1969, pp. 3-e-1 to 3-e-5.

[3-19] K. S. Fu, Sequential Methods in Pattern Recognition and Machine Learning, Academic Press, New York, 1968.

[4-1] E. E. Hilbert, "A Joint Clustering/Data Compression Concept," Ph.D. Dissertation, University of Southern California, May 1975.

[4-2] R. M. Haralick, K. Shonnugam and I. Dinstein, "Textural Features for Image Classification," IEEE Trans. Syst. Man Cybern., SMC-3, November 1973, pp. 610-621.

[4-3] J. W. Tukey, Exploratory Data Analysis, Addison-Wesley, 1971, Reading, Massachusetts.

[6-1] G. R. Nudd, "Progress on the Sobel CCD Chip and Circuit II," Semiannual Technical Report, Report No. 740, Image Processing Institute, University of Southern California, March 1977.