RADC-TR-88-324, Vol IV (of nine)
Interim Report
March 1989

NORTHEAST ARTIFICIAL INTELLIGENCE CONSORTIUM ANNUAL REPORT 1987 Research in Automated Photointerpretation

Syracuse University

James Modestino

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, NY 13441-5700

This report has been reviewed by the RADC Public Affairs Division (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-88-324, Vol IV (of nine) has been reviewed and is approved for publication.

APPROVED: Michael D Richard

MICHAEL D. RICHARD, Capt, USAF
Project Engineer

APPROVED:

WALTER J. SENUS
Technical Director
Directorate of Intelligence & Reconnaissance

FOR THE COMMANDER: Igor G. Plonisch

IGOR G. PLONISCH
Directorate of Plans & Programs

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (IRRE) Griffiss AFB NY 13441-5700. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS N/A
2a SECURITY CLASSIFICATION AUTHORITY N/A 2b DECLASSIFICATION / DOWNGRADING SCHEDULE N/A	3 DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
4 PERFORMING ORGANIZATION REPORT NUMBER(S) N/A	5 MONITORING ORGANIZATION REPORT NUMBER(S) RADC-TR-88-324, Vol IV (of nine)
6a NAME OF PERFORMING ORGANIZATION Northeast Artificial Intelligence Consortium (NAIC) 6b OFFICE SYMBOL (If applicable)	7a NAME OF MONITORING ORGANIZATION Rome Air Development Center (IRRE)
6c ADDRESS (City, State, and ZIP Code) [illegible] Link Hall Syracuse University Syracuse NY 13244-1240	7b ADDRESS (City, State, and ZIP Code) Griffiss AFB NY 13441-5700
8a NAME OF FUNDING / SPONSORING ORGANIZATION Rome Air Development Center 8b OFFICE SYMBOL (If applicable) COES	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F30602-85-C-0008
8c ADDRESS (City, State and ZIP Code) Griffiss AFB NY 13441-5700	10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO
61102F (over)	2304	J5	01

11 TITLE (Include Security Classification)
NORTHEAST ARTIFICIAL INTELLIGENCE CONSORTIUM ANNUAL REPORT 1987 Research in Automated Photointerpretation

12 PERSONAL AUTHOR(S)
James Modestino

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Interim	FROM Dec 86 TO Dec 87	March 1989	96

16 SUPPLEMENTARY NOTATION
This effort was performed as a subcontract by Rensselaer Polytechnic Institute to Syracuse University, Office of Sponsored Programs.

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Artificial Intelligence, Expert Systems, Machine Vision Photointerpretation, Image Analysis
12	07		

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

The Northeast Artificial Intelligence Consortium (NAIC) was created by the Air Force Systems Command, Rome Air Development Center, and the Office of Scientific Research. Its purpose is to conduct pertinent research in artificial intelligence and to perform activities ancillary to this research. This report describes progress that has been made in the third year of the existence of the NAIC on the technical research tasks undertaken at the member universities. The topics covered in general are: versatile expert system for equipment maintenance, distributed AI for communications system control, automatic photointerpretation, time-oriented problem solving, speech understanding systems, knowledge base maintenance, hardware architectures for very large system, knowledge-based reasoning and planning, and a knowledge acquisition, assistance, and explanation system. The specific topics for this volume are the use of expert systems for automated photointerpretation and other AI techniques to image segmentation and region identification.

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL MICHAEL D. RICHARD, Capt, USAF	22b TELEPHONE (Include Area Code) (315) 330-7788 22c OFFICE SYMBOL RADC/IRRE

Block 10 (Cont'd)

Program Element Number	Project Number	Task Number	Work Unit Number
62702F	5581	27	13
62702F	4594	18	E2

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

4.1 Introduction:

The RPI task has been concerned with the development of expert systems techniques for automated photointerpretation. More specifically, our efforts have been directed toward the development, implementation and demonstration of techniques which will mimic the job of a trained photoanalyst in interpreting objects in monochrome, single-frame aerial images. This is a difficult task which requires a combination of numerical and symbolic image processing techniques.

During the course of this effort we have developed a novel hierarchical, region-based approach to automated photointerpretation (cf. [1]). Basically, this approach proceeds by first segmenting the input image into disjoint regions which differ in tonal or textural properties. The spatial relationships between different regions are then expressed in terms of the associated adjacency graph where nodes represent regions and the connectivity indicates regions which are spatially contiguous. Based upon knowledge of the underlying spatial adjacency graph, together with various self and mutual region attributes or features, the problem is then that of assigning interpretations, or object categories, to each of the nodes. *This is generally a computationally explosive task. The novelty of our approach* is that we have been able to develop a computationally feasible approach to this symbolic interpretation process.

The advantage of our approach is based upon two important properties: First, we model the interpretation process as a Markov random field (MRF) defined on the adjacency graph. Secondly, we make use of an efficient stochastic relaxation process to find the most likely interpretation. The first assumption allows us to localize the search for good interpretations while the second helps in avoiding the otherwise computationally explosive nature of the search for optimum interpretations.

Our major effort during FY'87 has been in refining this region hierarchical approach, improving the initial segmentation process and, finally, demonstrating the approach on real-world aerial photographs. The present report is an attempt to document this progress of the last year.

This final report is organized as follows: In the remainder of this Section we provide an overview of the current status of our hierarchical, region-based approach to automated photointerpretation. This is followed, in Section 4.2, by a detailed development of an unsupervised tonal segmentation scheme under a Gaussian modeling assumption. In Section 4.3 we describe a corresponding texture segmentation technique based upon MRF's. Subsequently, in Section 4.4, we describe a novel approach, based upon information theoretic concepts, for determination of the number of distinct image classes in an image. This latter issue is crucial to any fully automated image interpretation scheme. Finally, in Section 4.5, we provide a summary and an outline of research directions for FY'88.

4.1.1 Image Interpretation Approach:

In this section we will describe the current status of our automated photointerpretation system, review the pertinent details of the evolving testbed which will support it and illustrate some typical results obtained so far.

A block diagram of the overall testbed structure is illustrated in Fig. 4.1-1. The main function of the preprocessor is to provide a segmentation of the image into disjoint regions which are homogeneous within a region but differ in some sense from adjacent regions. In the next several Sections, we describe various segmentation schemes investigated for this purpose. For the time being then we assume that a segmentation has been obtained.

Once a segmentation is obtained, however preliminary, the regions are indexed and region maps are stored in the image database. That is, the actual pixel values associated with a region are stored separately for each region. In addition, various attributes associated with each region are stored. This includes such parameters as area, perimeter, boundary, elongation, etc. In addition, the spatial relationships between the various regions are maintained. This is most easily done by using an adjacency graph where the nodes correspond to regions and the connectivity indicates spatial relationships. In particular, two nodes are connected by an arc or edge if they are in some sense spatial neighbors. The values associated with arcs can include mutual information corresponding to the connected nodes. This information might include: mutual boundaries, spatial distances, strength of mutual edges, etc. Image interpretations are provided by the inferencing mechanism which has access to the region information stored in the image database, as well as the world knowledge stored in the knowledge database. Feedback to the image preprocessor is through the inferencing mechanism.

It should be noted from Fig. 4.1-1 that the testbed allows operator intervention through an interactive image processing and display terminal. More specifically, the operator can manually extract regions using a joystick or trackball and, if desired, actually provide interpretation of the various extracted regions. Once the disjoint regions are outlined by the operator, the various region attributes are automatically extracted and stored in the image database in exactly the same format as if they were automatically extracted by the image preprocessor. Furthermore, in cases where the operator provides region interpretations, the relevant spatial relationships are provided to the knowledge database allowing updating of our world knowledge.

Now suppose that an appropriate initial segmentation is obtained. Let the distinct regions be labeled $R_1, R_2, \ldots, R_N$ as, for example, in Fig. 4.1- 2 where $N = 7$. The corresponding first-order adjacency graph associated with this segmented image then appears as indicated in Fig. 4.1-3. By first-order adjacency we mean here that regions are adjacent, or are neighbors, if and only if they are spatially contiguous. The problem is now: given an *initial* segmentation, to provide a *global* interpretation for each of the nodes given measurement attributes associated with each node, context information associated with the mutual relationships specified in the adjacency graph and world knowledge as prescribed in the knowledge database. A detailed description of our approach to implementing this interpretation function was provided previously in [1]. As a result, the following discussion of the major characteristics of this approach will be abbreviated and will depend upon the more extensive development in [1] for details.

Suppose then that the segmented regions within the image are labeled $R_1, R_2, \ldots, R_N$ and let $I_1, I_2, \ldots, I_N$ be the corresponding global interpretations given to each of these regions where $I_i \epsilon \{\phi, 1, 2, \ldots, K\}$. Here, we have K specific object types whose labels are to be assigned to each of the regions plus the ambiguous or irrelevant object type represented by the label or symbol ϕ. Suppose we define the region information as $\mathcal{R} = (R_1, R_2, \ldots, R_N)$ and the interpretation vector $\mathbf{I} = (I_1, I_2, \ldots, I_N)$. Note there are at most $(K + 1)^N$ possible interpretation vectors although, in reality, there are many fewer than this since a valid global interpretation should not allow neighboring, or adjacent, regions to carry identical labels except for the uncertain symbol, ϕ. The exact number of interpretation vectors will then depend specifically upon the spatial arrangements of regions and is thus a random variable.

Our criterion will be to choose the estimated global interpretation $\hat{\mathbf{I}} = \mathbf{I}_0$ iff

$$\mathbf{I}_0 = \arg\max_{\mathbf{I}} p\{\mathbf{I}|\mathcal{R}, \mathcal{K}, \mathcal{X}\} \tag{1}$$

Here, $\mathcal{R}$ represents information describing the partitioning into regions, $\mathcal{K}$ represents information in the knowledge database and $\mathcal{X}$ represents the corresponding adjacency graph which includes all measurement information, both for each region separately as well as mutual measurement information between regions. The quantity $p\{\mathbf{I}|\mathcal{R}, \mathcal{K}, \mathcal{X}\}$ represents the conditional probability of $\mathbf{I}$ given $\mathcal{R}, \mathcal{K}$ and $\mathcal{X}$. This quantity may be difficult to specify theoretically, but the work in [1] provided a nice theoretical framework for specifiying the structure of this conditional probability. The optimization in (1) is then over all legitimate interpretation vectors; the resulting estimate is called the maximum a posteriori (MAP) estimate and is well-founded in statistical decision and estimation theory [2].

At this point we will make the assumption that, conditioned on $\mathcal{R}, \mathcal{K}$ and $\mathcal{X}$, the interpretation vector $\mathbf{I}$ is a Markov random field (MRF) defined on the corresponding adjacency graph. The concept of a MRF defined on a 2-D lattice has provided a useful model for images. However, as pointed out in [3], the concept of a MRF need not be restricted to lattices but can be defined on more general structures such as graphs. Thus, it appears quite natural to define the interpretation vector, $\mathbf{I}$, as a MRF defined on the associated adjacency graph.

Under the assumption that $\mathbf{I}$ is then a conditional MRF, it's well known through the equivalence of MRF's with Gibbs random fields (GRF's), that the the conditional probability must be of the form

$$p\{\mathbf{I}|\mathcal{R}, \mathcal{K}, \mathcal{X}\} = \frac{e^{-U(\mathbf{I}, \mathcal{R}, \mathcal{K}, \mathcal{X})}}{Z}, \tag{2}$$

where $U(\mathbf{I}; \mathcal{R}, \mathcal{K}, \mathcal{X})$ is the associated Gibbs energy function and Z is the corresponding partition function which serves the role of a normalization constant. More specifically, we have

$$Z = \sum_{\mathbf{I}} e^{-U(\mathbf{I},\mathcal{R},\mathcal{K},\mathcal{X})}, \tag{3}$$

where the summation is over all legitimate interpretation vectors. The energy function must then be designed to take into account the information represented by $\mathcal{R}, \mathcal{K}$ and $\mathcal{X}$.

As can be seen from (1) and (2), the MAP estimate is obtained by minimizing the energy function. This is a difficult combinatorial problem since, as we have noted previously, there are as many as $(K + 1)^N$ possible interpretation vectors, **I**. In [1] we proposed and described the use of a stochastic relaxation procedure, called simulated annealing, to overcome these combinatorial problems. More specifically, simulated annealing was used to obtain the maximum of $p\{\mathbf{I}|\mathcal{R}, \mathcal{K}, \mathcal{X}\}$.

Now consider the choice of a Gibbs energy function. It's well-known (cf. [8]) that this must be of the form

$$U(\mathbf{I}; \mathcal{R}, \mathcal{K}, \mathcal{X}) = \sum_{c} V_c(\mathbf{I}_c; \mathcal{R}, \mathcal{K}, \mathcal{X}), \tag{4}$$

where $V_c(\mathbf{I}_c; \mathcal{R}, \mathcal{K}, \mathcal{X})$ is called a ***clique*** function and the summation in (4) is over all possible cliques with $\mathbf{I}_c$ the restriction of **I** to the clique c. Cliques and clique functions are described in more detail in [1]. In particular, we showed that the summation in [4] can be rewritten as

$$U(\mathbf{I}; \mathcal{R}, \mathcal{K}, \mathcal{X}) = \sum_{i=1}^{N} \sum_{c \epsilon C_i} V_c(\mathbf{I}_c; \mathcal{R}, \mathcal{K}, \mathcal{X}). \tag{5}$$

Here, the outer sum is over the individual nodes while the inner sum is over the set of distinct cliques, C_i, associated with $i = 1, 2, \ldots, N$.

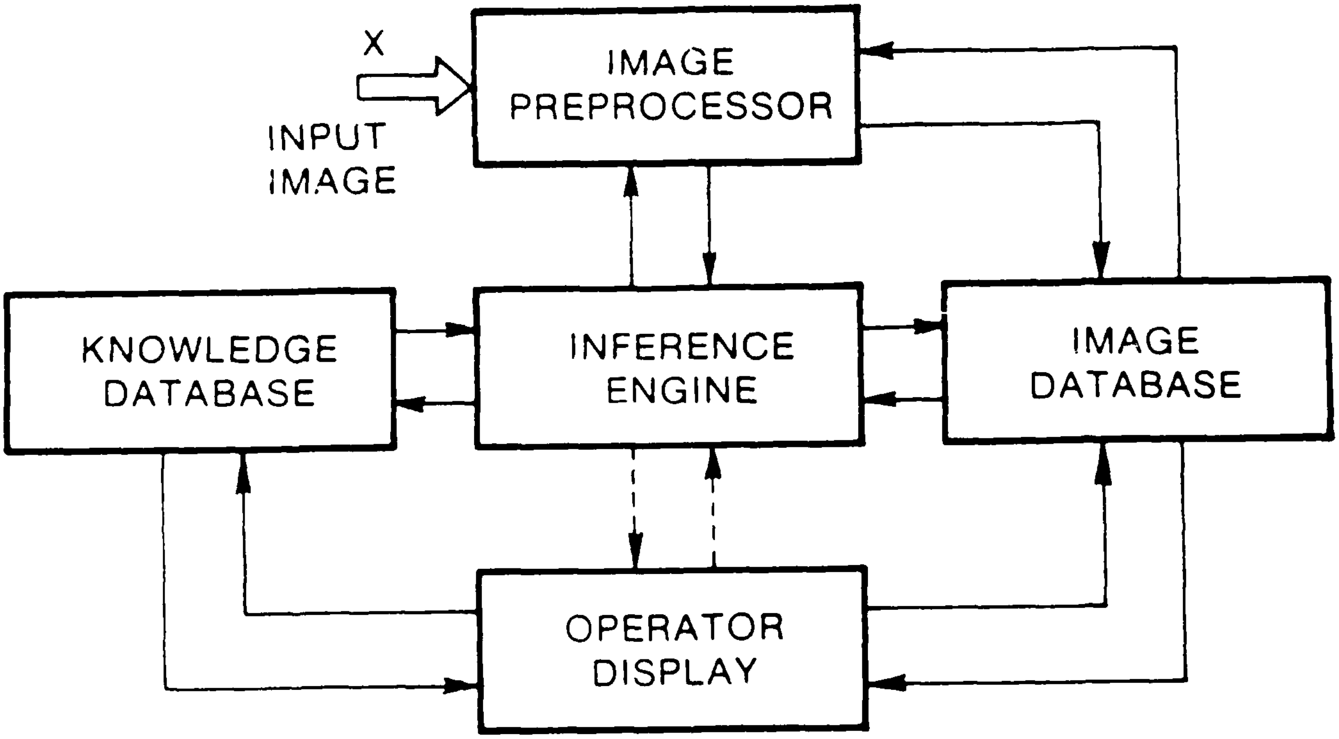
As pointed out in [1], the outstanding problem at this point then is in the determination and specification of an appropriate set of clique functions. At that time we suggested some ways in which these clique functions could be chosen in some simple illustrative problems. During the past year we have studied several refinements in the selection of clique functions and have applied this scheme to several sets of synthetic as well as real-world images. Our work here is incomplete and we expect to actively pursue these investigations throughout FY'88.

In the following Sections of this report we will describe in some detail the rather extensive work we have completed in FY'87 concerned with segmentation techniques. Again it must be emphasized that, regardless of the image interpretation technique employed, the results are highly dependent on having a good initial segmentation.

References for Section 4.1

1. J. W. Modestino, "A Hierarchial Region-Based Approach to Automated Photointerpretation", NAIC Final Report for FY'86.
2. H. L. Van Trees, *Detection, Estimation and Modulation Theory* I, Wiley and Sons, New York, 1968.
3. R. Kinderman and J. L. Snell, *Markov Random Fields and Their Applications*, American Mathematical Society, Providence, RI, 1980.

Fig. 4.1-1 Automated Photointerpretation Testbed



Automated Photointerpretation Testbed.

Fig. 4.1-2 An Initial Segmentation of an Image

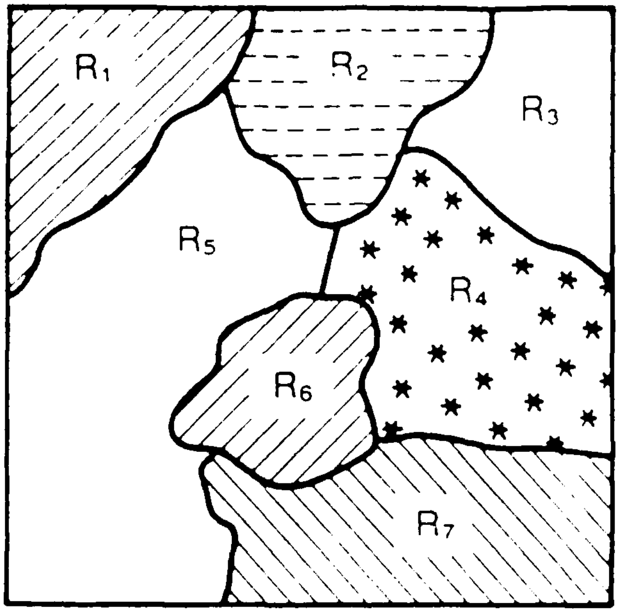
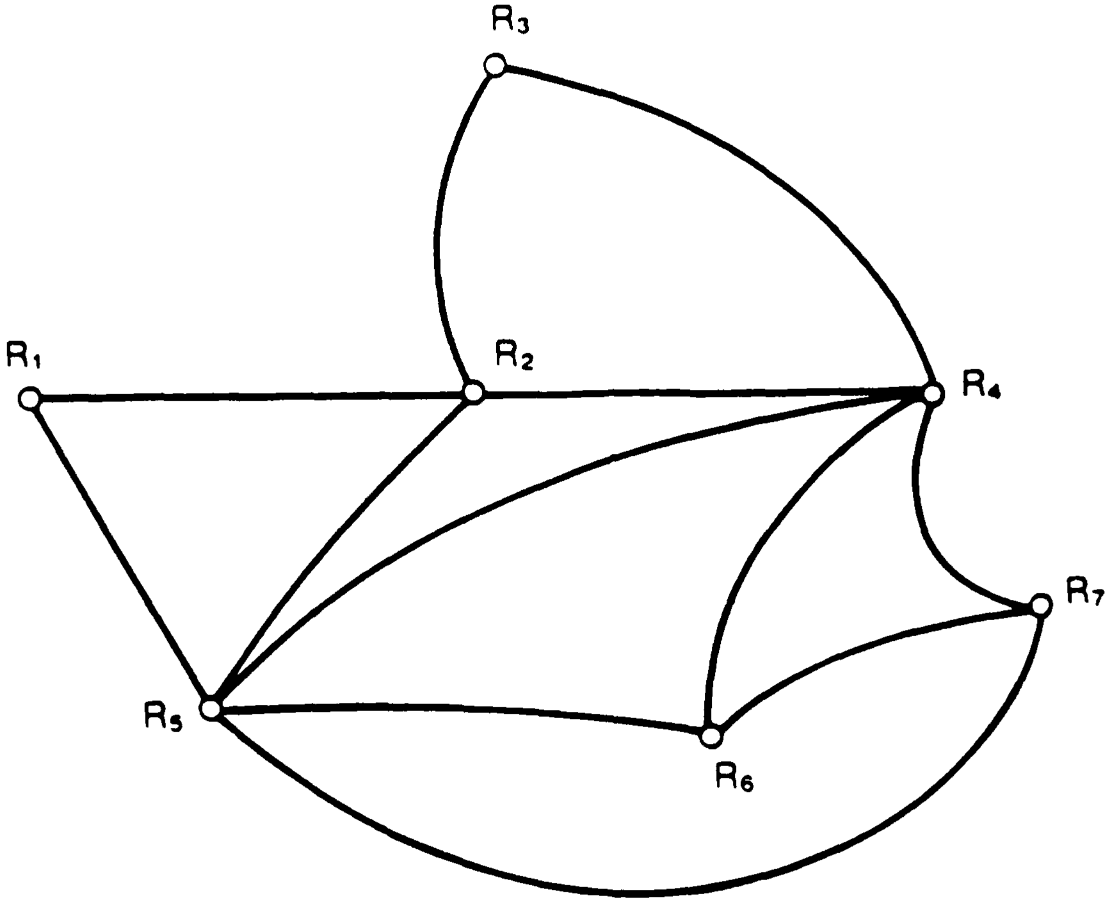


Fig. 4.1-3 Adjacency Graph for Segmented Image



4.2 Unsupervised Image Segmentation Using A Gaussian Model:

A Gaussian random field model-based maximum-likelihood (ML) approach to image segmentation is described in this section. In this approach, the segmentation problem is formulated as a statistical decision problem under a Gaussian modeling assumption for different image classes. The model parameters are estimated directly from the observed image, resulting in an unsupervised algorithm. The results of applying this algorithm to the segmentation of aerial images are also described.

4.2.1 Background:

Image segmentation is a very important problem in many image processing applications. In an image segmentation problem, an observed image is separated into regions of different properties. Two of the most important properties used are *tone* and *texture* [1]. Tone is related to the average gray level of a region while texture corresponds to the spatial distribution of different gray levels in a region. As pointed out in [1], different regions in an image sometimes exhibit mainly one or the other of these two properties. When the spatial variation of gray levels in a region is small and uncorrelated, the region is dominated by tone. On the other hand, if the spatial variation of gray levels is large or correlated, the region is dominated by texture. This domination is not only determined by the particular image scene, but more often by the resolution of the image. In this paper, we are mainly concerned with images whose regions are dominated by tonal properties. Surveys of texture segmentation techniques can be found in [1], [12], [14]. Examples of the type of image for which tonal properties dominate can be found in many aerial photographs. In these images, the regions correspond to roads and fields, which have little texture originally, or vegetation regions which show little texture or gray level variation because of the low resolution of the image.

In many image analysis applications, image segmentation is the first stage of processing and the quality of segmentation is crucial to the overall performance of the system [2]. This is particularly the case in our application which is in automated photointerpretation [13]. Because of its importance in a wide variety of applications, a large number of image segmentation techniques have been proposed. These techniques can be classified into two different approaches; a *statistical* approach, where tonal or textural properties are characterized in statistical terms, such as mean, variance, correlation functions and probability distribution functions and a *structural* approach, where these imag properties are described by a properly defined formal language [1]. In this paper, we are interested in a purely statistical approach for image segmentation where the image regions exhibit mainly tonal properties.

Most of the previously proposed statistical techniques are heuristic or ad hoc in that they are either based on some ad hoc arguments or derived from certain heuristics about a specific set of images. The work of Haralick and Shapiro [3] provides a comprehensive survey of most of the existing heuristic statistical image segmentation techniques, ranging from the reasonably simple to the very complex. Although considerable success has been achieved by a number of them in some specific and well-defined situations, they have some unsatisfactory features. For example, it's often difficult to precisely define or choose the parameters involved in these algorithms, such as the valleys of histograms in various histogram-guided thresholding techniques or thresholds for closeness in most region growing algorithms. Many more sophisticated algorithms require an enormous amount of computation. In addition, there is little known, in general, on how effective these algorithms are and what type of images they can be applied to. More specifically, there is no specific modeling assumption made for the image properties and, consequently, the resulting solution cannot be optimal.

To overcome these difficulties, a number of stochastic model-based image segmentation techniques have been proposed [4]-[8], [14]. In a statistical model-based approach, stochastic modeling assumptions are made for regions of different statistical properties, we call *classes*, in an image. Then the segmentation problem is formulated as a statistical decision problem and an *optimal* solution is sought. As a result, the stochastic model-based approach usually provides image segmentation techniques that are more generally applicable and optimal according to some well-defined criterion.

Most of the stochastic model-based techniques, however, exploit textural properties rather than tonal properties; hence these are texture segmentation techniques. One of the few techniques which mainly makes use of tonal properties or, more precisely, attempts to model tonal properties, is an algorithm proposed by Derin and Elliot [5]. In this technique, different image regions are modeled by a constant gray level with additive white Gaussian noise which has the same mean and variance over the entire image while the distribution of different regions is modeled by a Markov random field (MRF), or Gibbs random field (GRF). The segmentation problem is then formulated as a maximum a posteriori (MAP) estimation problem. The maximization of the a posteriori probability functional is performed using an approximate dynamic programming procedure. This algorithm is partially *unsupervised* in that the model parameters for the regions are estimated directly from the observed image by the moment method of Gaussian mixture estimation although the model parameters for the MRF model which generates the regions must be pre-specified. The choice of underlying MRF parameters is made heuristically. While some successful examples are shown in [5], this algorithm is computationally quite involved. Both dynamic programming and the mixture estimation procedure require considerable computation. In their approach, the image classes are modeled as having constant gray levels corrupted by additive observation noise. This is a rather unrealistic assumption since many image regions that appear to have uniform gray-levels have gray level variation in them in addition to the additive observation noise. Finally, it's not very clear how the model parameters for the MRF of the region distribution should be selected. Recently, it has been shown that the parameters for the MRF can be estimated through an EM (Expectation- Maximization) type algorithm [12]. A disadvantage is that the amount of computation required is quite large.

In this paper, we describe a novel stochastic model-based image segmentation approach which provides a simpler alternative and overcomes some of the unsatisfactory features of Derin and Elliot's technique. First of all, we model different image classes, or region types, as independent Gaussian random fields with different spatially constant mean and variances. The constant mean of a class is used to model the flat gray level, or tone, of the region and the class-dependent variance is used to model the combined effects of variation of gray levels and additive observation noise which is assumed to be zero mean for that class. Assuming the variation of gray level in a region is relatively small, our model is a tonal model. Unlike Derin and Elliot's algorithm, we do not make any assumptions on the distribution of different regions in the image, since it is quite involved to estimate the MRF model parameters and perform the MAP operation. This results in a maximum-likelihood (ML) approach. By using the independence assumption, the likelihood functional can be maximized through a highly parallel operation; even using a raster scan, this can be quite simply done in one scan. Finally, the model parameters for different image classes can be estimated by using a computationally efficient clustering technique operating directly on the observed image. Hence this approach is entirely *unsupervised.* This algorithm has been applied to a set of aerial photographs and the results are shown to be quite promising.

4.2.2 The Gaussian Model and ML Segmentation:

In this paper, we consider an image as an array of gray levels defined on a two-dimensional (2-D) lattice of finite extent. In particular, we denote an image by x where

$$\mathbf{x} = \{x(m,n),(m,n)\epsilon L\}; \qquad L = \{(m,n),\ 1 \leq m,n \leq N\}. \tag{1}$$

A random field is a family of random variables defined over the lattice L. In this paper, we use capital letters for random fields and random variables, lower-case letters for realizations of random fields and sample values of random variables. A Gaussian random field representing an observed image can then be defined as

$$X(m,n) = f(m,n) + W(m,n); \qquad (m,n)\epsilon L, \tag{2}$$

where $f(m,n)$ is the mean and $W(m,n)$ is a zero-mean Gaussian random sequence, i.e., $W(m,n) \sim N(0,\sigma^2(m,n))$. In particular, we assume $f(m,n)$ and $\sigma^2(m,n)$ are constant, but unknown, for an image class and vary for different classes. In addition, we assume that the $X(m,n)$'s are independent. The probability density function of the observed random field is then simply

$$p(\mathbf{x}) = \prod_{(m,n)\epsilon L} \frac{1}{\sqrt{2\pi}\sigma(m,n)} exp\Big[- \frac{(x(m,n) - f(m,n))^2}{2\sigma^2(m,n)}\Big]. \tag{3}$$

Under a stochastic modeling assumption, the image segmentation problem can be formulated as a statistical decision problem. Here we take the basic formulation of the segmentation problem as in [4], [7], [8]. Assume that there are K possible image classes associated with the K hypotheses, $H_k, k = 1,2,\ldots,K$. Suppose that they are distributed in disjoint regions as shown in Fig. 4.2-1. Each of the image classes is modeled by an independent Gaussian model corresponding to a particular hypothesis. That is, we have the K hypothesis classes

$$H_k : X(m,n) = f(k) + W^{(k)}(m,n); \qquad k = 1,2,\ldots,K, \tag{4}$$

where

$$W^{(k)}(m,n) \sim N(0,\sigma^2(k)). \tag{5}$$

A typical realization of the K-class Gaussian random field image is shown in Fig 4.2-2., with $K = 3$. Here, the regions are first generated by a 2-D MRF and then "colored" by the appropriate Gaussian model. The model parameter vectors $\mathbf{a}_k$, k=1,2,3, are described in the next section.

In essence, image segmentation is the process of assigning each pixel in the image to a correct hypothesis class. According to statistical decision theory, an assignment rule which minimizes the classification error, assuming equally likely hypothesis, is a threshold test based on the ratios of the class-conditional likelihood functionals, or some monotone function of it [9]. More specifically, for each point (i,j) in the lattice L, we can construct a window of size $(2M+1)\times(2M+1)$, centered at (i,j) and denoted by $W_{i,j}$. The data contained in the window is denoted by $X_{i,j}$. That is, $X_{i,j} = \{x(m,n),(m,n)\epsilon W_{i,j}\}$ where

$$W_{i,j} = \{(m,n), i-M \leq m \leq i+M,\ j-M \leq n \leq j+M\}, \tag{6}$$

with $M << N$ and boundary effects are ignored. Define the class-conditional log likelihood functional, given H_k, at (i,j) by

$$L_k(X_{i,j}) = \log\{p(X_{i,j}|H_k)\}. \tag{7}$$

Then a maximum-likelihood approach is to assign pixel position (i,j) to image class k_0 if

$$k_0 = \underset{1\leq k\leq K}{\arg} \max\ L_k(X_{i,j}). \tag{8}$$

Notice here if we let $M = 0$, the window will contain only a single pixel which is a ML estimation approach under an independence assumption on the pixels [7]. As will be shown later, the segmentation result with $M = 0$ is somewhat "spotty", and a proper choice of $M > 0$ can smooth out most of the noise spots. Notice also that in this segmentation algorithm, the decision on any pixel position is independent of those of the others, hence it can be implemented in parallel. However, in our implementation we utilize a raster scan processing approach which can be summarized as follows:

1.) Process all the pixels in a raster scan order.

2.) At each pixel position, a decision window centered at the pixel is constructed (ignoring the boundary effects).

3.) The class-conditional likelihood functional defined in expression (7) can then be evaluated for each hypothesis.

4.) Assign the pixel to image class k_0, $1 \leq k_0 \leq K$, if it maximizes the class-conditional likelihood functional as in expression (8).

To carry out the computations in 3.) and 4.) above, the model parameters for each of the image classes are needed. In the next section, we will describe a method for estimating the model parameters directly from the image.

4.2.3 Model Parameter Estimation and Segmentation Results:

The parameter estimation technique used in the ML segmentation approach is similar to those in our previous work [7],[8], which were quite successful in unsupervised texture segmentation. More specifically, define the model vector for each class or hypothesis,

$$\mathbf{a}_k = (f(k), \sigma(k)), \qquad k = 1, 2, \ldots, K. \tag{9}$$

Then the $\mathbf{a}_k$'s are the model parameters to be estimated from the observed image. As in [7] and [8], consider a sliding window of size $M_1 \times N_1$, where $M_1 << N$, $N_1 << N$, with each step of the sliding window being displaced M_2 pixels vertically and N_2 pixels horizontally, as shown in Fig. 4.2-3. At each position of the sliding window, a Gaussian model vector is estimated by computing the sample mean and sample variance. This vector is then stored as a sample vector. Finally, all the sample vectors obtained this way are then used as input to a particular clustering algorithm known as the K-means algorithm [10]. The centroids of the clusters found in the clustering process are then used as model parameter vectors for the underlying image classes and used in the model-based windowed ML segmentation algorithm described in the previous section.

A remaining question with this estimation approach is how K, the number of different image classes, is to be determined. In related work [11], we have proposed use of an information-theoretic criterion, known as the Akaike Information Criterion (AIC), to determine the number of classes from the observed image. This scheme has been shown to provide correct results for synthetic mixture data and reasonable results for real-world images that are in close agreement with subjective observations. This scheme is directly applicable to the present situation. In this paper, however, our interest is to see how effective the segmentation is under reasonable assumptions on the number of classes. By reasonable, we mean the number of classes is approximately equal to the number of perceptively different tone classes in the image. In the segmentation experiments to follow then, we assign the number of classes by observing the images.

There are two other problems encountered when implementing the estimation algorithm. The first is how the sliding window size should be selected for model parameter estimation. Although it is not clear quantitatively how the window size effects the estimation accuracy, we can make some qualitative observations. In general, if the window is too large, it might contain a significant amount of data from different classes, resulting in unreliable estimates. On the other hand, if the window is too small, the data might not be enough to arrive at reasonably accurate estimates. At this point, we choose the sliding window size heuristically. For example, we noticed in our experiments that most of the regions have a size greater than 16×16. As a result, we choose the size of the sliding window to be 16×16. Notice, however, as long as the window is not too large or too small, the size is not very critical and the same size can be used for a number of images.

Secondly, even by proper selection of the window size, we still might come to a situation in which a window contains data from different classes in about equal amounts, i.e., the window is"sitting" on a boundary. The sample vectors arising from such situations will affect the accuracy of the estimated class model vectors. As a result, the performance of the segmentation may be degraded. For example, regions that should be well separated if the class model vectors are reasonably accurate may be mixed together, or not separated at all. However, we observed that when a sliding window contains data from different classes, the estimated variance is usually quite large, especially when the difference in gray level is large. Hence, to improve the estimation accuracy, we can reject those sample vectors which have large variance components. For the simple scheme considered here, a threshold, denoted T_σ, is selected and if the square root of the variance component of a sample vector exceeds the threshold, T_σ, it will be discarded from the clustering procedure. Currently, this threshold is selected heuristically through observing the quality of corresponding segmentation results. Later in this section, we will show, through experimental results, this simple scheme does improve the segmentation. For a completely automatic process, it has to be selected according to a fixed rule or algorithm. Other more sophisticated techniques can also be used to obtain reliable model parameter estimates. For example, a χ^2 type of test can be performed on the data contained in a number of subwindows of a sliding window to see if they have the same distribution; that is, if the gray level in the sliding window is "uniform". If the data is uniform, an estimated sample model vector is stored. Otherwise, it is rejected. In this approach, we still need to decide the size of the sliding window, the number of subwindows, and the significance level of the test. Another approach is to use robust estimation techniques treating unreliable sample model vectors as "outliers". These approaches are currently under investigation [12].

We have applied the algorithm described in the previous sections to the segmentation of aerial photographs. The images are of size 256×256 and digitized to 256 gray-levels. The segmentation is performed for each image under different assumptions on the number of classes. In the following we will present and discuss the experimental results.

First, we show that by rejecting sample model vectors with large estimated variance component, using the simple scheme described previously, the segmentation results can indeed be improved In Fig. 4.2-4, an image containing fields and oil tanks is segmented under the assumption of 3 classes. The size of the decision window is 3×3; that is, $M = 1$. The segmentation results, along with the estimated model parameter vectors, are shown in Fig.4.2-4b and 4.2-4c, respectively, for the case of not rejecting any sample vector and rejecting sample vectors with large variance component. In the latter case we have taken T_σ=15. It can be seen that the results are improved considerably. In the rest of the experimental results, we reject the sample vectors which have large variance component using $T_\sigma = 15$.

Next, in Fig. 4.2-5, we show the effect of the window size M in the ML segmentation approach. It can be seen that when the decision window size is selected properly, the windowed approach smoothes out some noisy spots in the segmentation and significantly improves the segmentation results. In the rest of the segmentation experiments, we used decision windows with size 3×3, or $M = 1$.

Finally, in Fig.'s 4.2-6, 4.2-7, 4.2-8 we show some segmentation results for three different aerial photographs under the assumptions of both 3 and 6 classes. In each case, different regions of the imagc are separated reasonably well by a 3-class assumption. Buildings, roof, roads, and vegetation areas, are well separated. Finer segmentation is obtained under a 6-class assumption. It should be pointed out, however, that the segmentations here are still coarse in that different real world objects are assigned to the same class as long as they are close in tonal properties. Differentiation of regions of the same class which are really different objects could be achieved using other properties, for example, texture or shape information.

4.2.4 Summary:

In this paper, we have described an unsupervised Gaussian model-based ML approach to image segmentation. In this approach, different regions are modeled by independent and spatially varying Gaussian random fields. The segmentation problem is formulated as a statistical decision problem and an ML solution is proposed. The model parameters are estimated directly using a clustering-estimation method. Experiments on the segmentation of aerial photograph images are shown to be promising.

This work brings up a number of problems for future investigation. First, we need to study methods to determine the threshold, T_σ, for rejecting erroneous sample model vectors directly from the data. Possible solutions are outlined in the previous section and experiments are needed to thoroughly investigate their efficacy. Another interesting problem is the characterization of the image classes. The independent Gaussian model employed in this work basically aimed at the tonal properties of the image classes, while spatial variation or texture is only reflected in the variance of the model. In addition, the independence assumption further limits the characterization of texture properties in image classes. On the other hand, a number of texture-based segmentation schemes do not perform well when the image classes exhibit strong tonal differences [12]. What is needed is a more robust approach that combines the merits of both tonal model-based and texture model-based approaches.

References for Section 4.2

1. R. M. Haralick, "Statistical and structural approaches to texture", Proc. IEEE, Vol. 67, pp. 786-804, May 1979.

2. T. O. Binford, "Survey of model-based image analysis systems", *The International Journal of Robotics Research, Vol. 1*, No. 1, pp. 587-633, Spring 1982.

3. R. M. Haralick and L. G. Shapiro, "Image segmentation techniques", *Computer Vision, Graphics and Image Processing,* Vol. 29, pp. 100-132, 1985.

4. J. W. Modestino, R. W. Fries and A. L. Vickers, "Texture discrimination based upon an assumed stochastic texture model", IEEE Trans. Pattern Anal. Machine Intel., Vol. PAMI-3, pp. 557-580, September 1981.

5. H. Derin and H. Elliot, "Modeling and segmentation of noisy and textured images using Gibbs random fields", IEEE Trans. Pattern Anal. Machine Intel., Vol. PAMI-9, pp. 39-55, January 1987.

6. C. W. Therrien, T. F. Quatieri and D. E. Dudgeon, "Statistical model-based algorithms for image analysis", Proc. IEEE, Vol. 74, pp. 532-551, April 1986.

7. J. Zhang and J. W. Modestino, "Unsupervised AR random field model- based image segmentation", unpublished RPI report, March 1987.

8. J. Zhang and J. W. Modestino, "Texture classification and discrimination using the Markov random field model", submitted to IEEE Trans. Pattern Anal. Machine Intel., Dec. 1987.

9. H. L. Van Trees, *Detection, Estimation, and Modulation Theory*, Part I., John Wiley and Sons, Inc., New York, 1986.

10. J. T. Tou and R. C. Gonzalez, *Pattern Recognition Principles*, Addison-Wesley Pub. Co., Reading, MA, 1974.

11. J. Zhang and J. W. Modestino, "A model-fitting approach to cluster validation with applications to stochastic model based image segmentaiton", *Proc. ICASSP '88*, New York, April 1988.

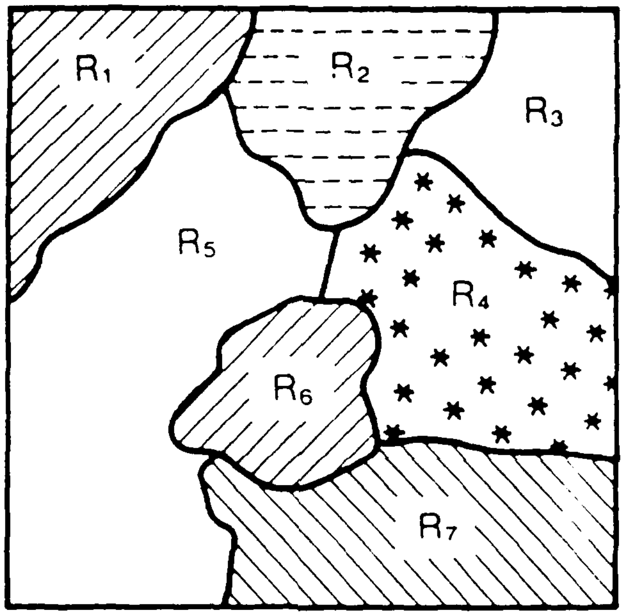
12. J. Zhang, "Two-dimensional stochastic model-based image analysis", Ph.D. Thesis, Rensselaer Polytechnic Institute, Troy, New York, in preparation.

13. J. W. Modestino, "A hierarchical region-based approach to automated photointerpre-

tation", RPI report, March 1987.

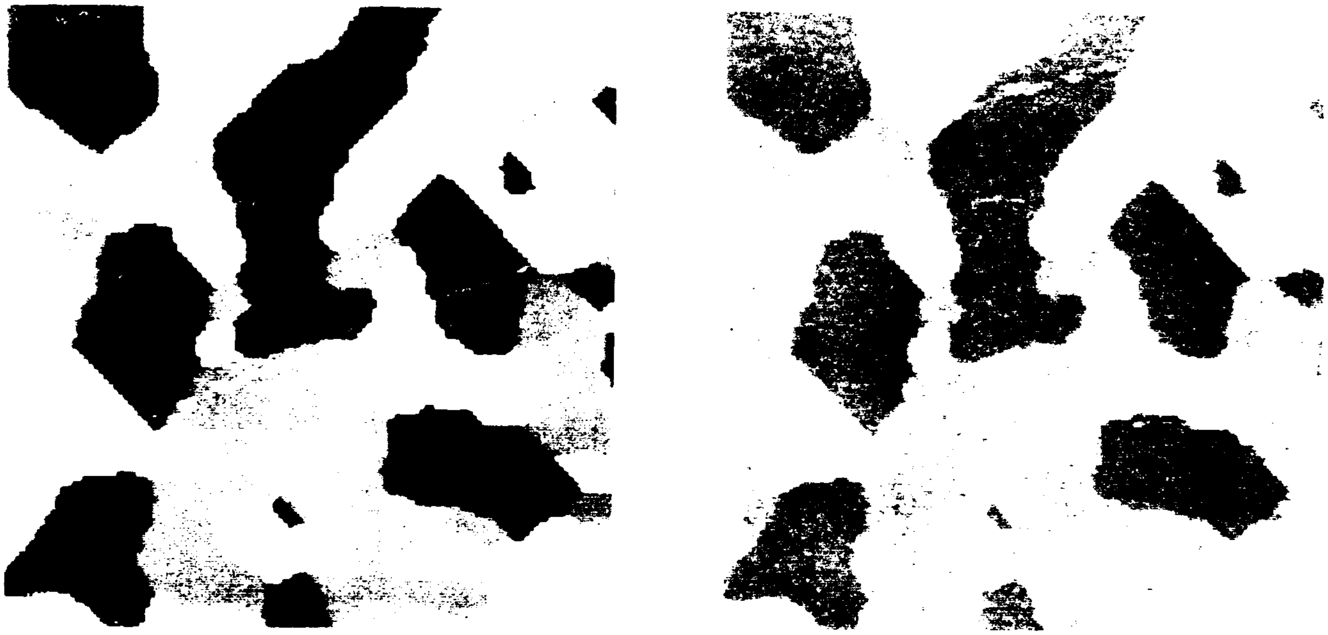
14. F. S. Cohen and D. B. Cooper, "Simple, parallel, hierachical and relaxation algorithms for segmenting non-casual Markovian random field models", Proc. IEEE Pattern Anal. Machine Intel., Vol. PAMI-9, pp. 195-219, March, 1987.

Fig. 4.2-1 An Image Containing Multiple Regions



An Initial Segmentation of an Image.

Fig. 4.2-2 A Realization of a 3-Class Gaussian Random Field



a.) MRF Generated Region Map.

b.) 3-Class Image;

$\hat{\underline{a}}_1$=(70,8.9),

$\hat{\underline{a}}_2$=(100, 14.1),

$\hat{\underline{a}}_3$=(150, 10.9).

Fig. 4.2-3 A Sliding Window on the Image Plane

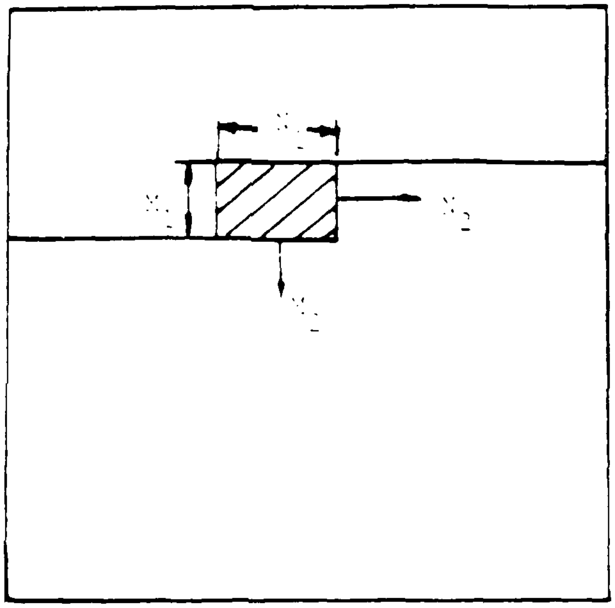
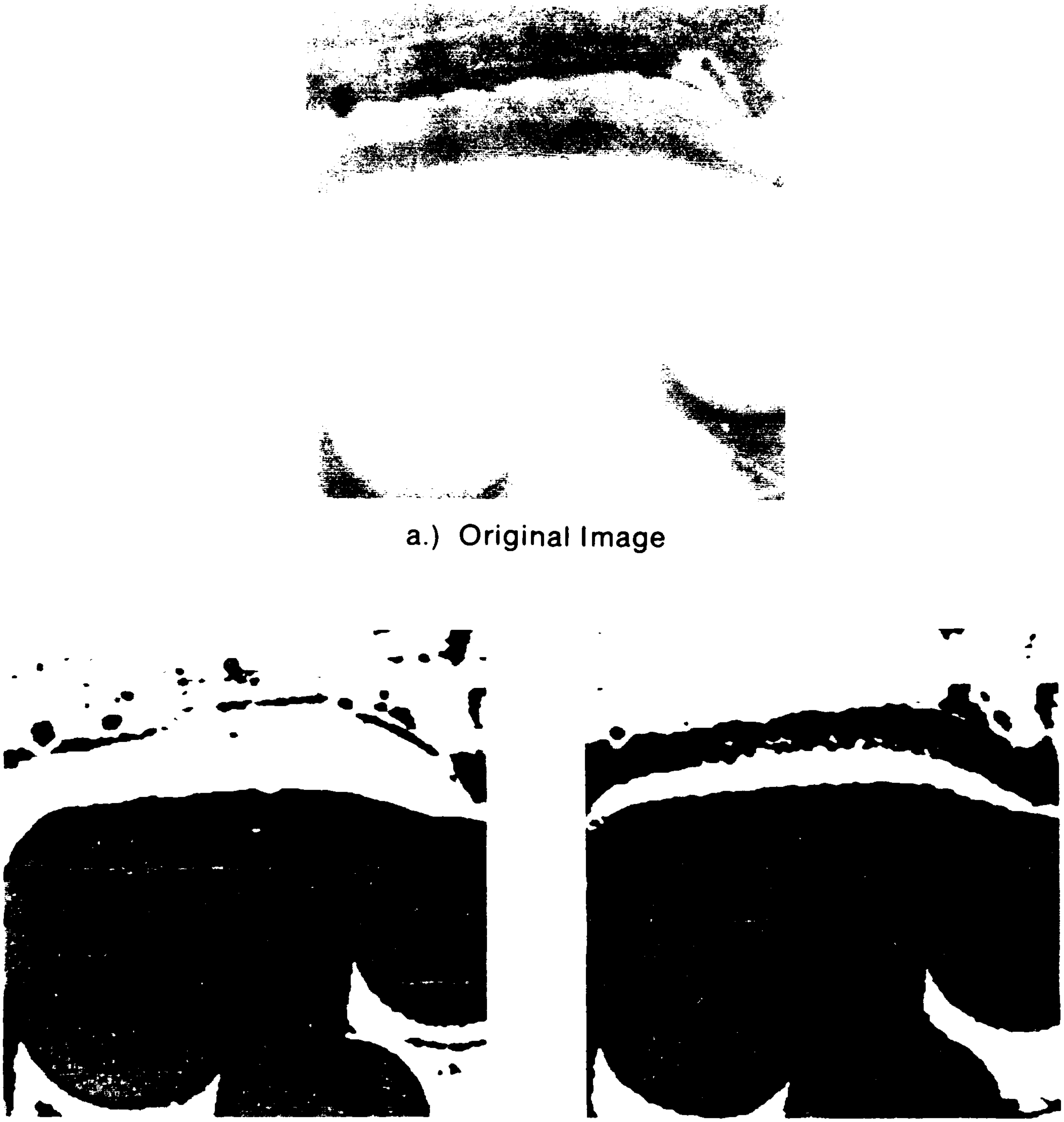


Fig. 4.2-4 Performance Improvement by rejecting large variance components

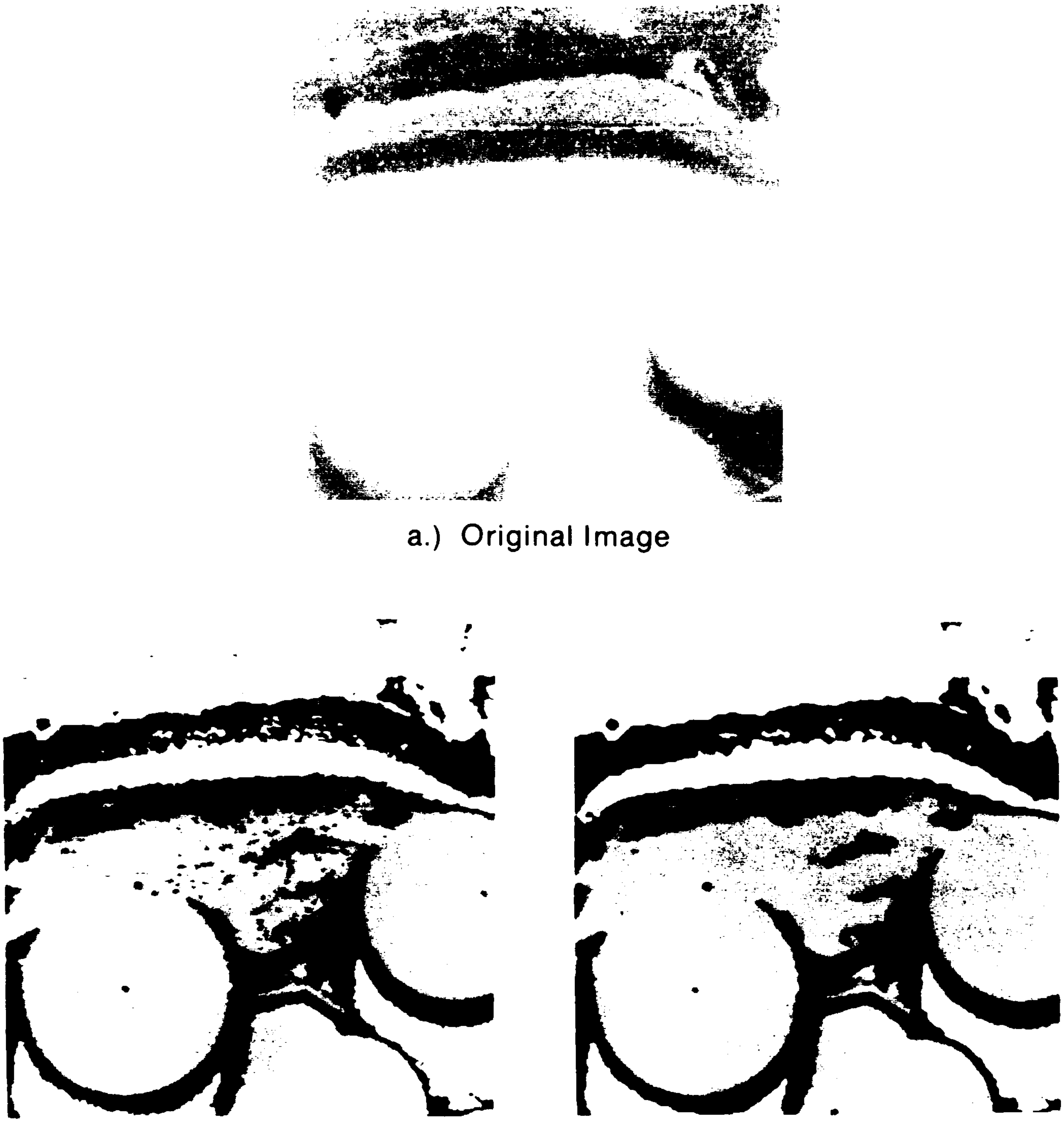


a.) Original Image

b.) Segmentation 1, 3-classes, without rejecting model vectors with large variance term; M=1, $\hat{\underline{a}}_1$=(150, 57), $\hat{\underline{a}}_2$=(203.4, 7.5), $\hat{\underline{a}}_3$=(135.2, 9.0)

c.) Segmentation 2, 3-classes, rejecting model vectors with large variance term; T_σ=15,M=1, $\hat{\underline{a}}_1$=(158.1, 11.5), $\hat{\underline{a}}_2$=(208.7,3.4), $\hat{\underline{a}}_3$=(131.0, 5.0)

Fig. 4.2-5 Improvement by decision window of size greater than one, $T_\sigma = 15$

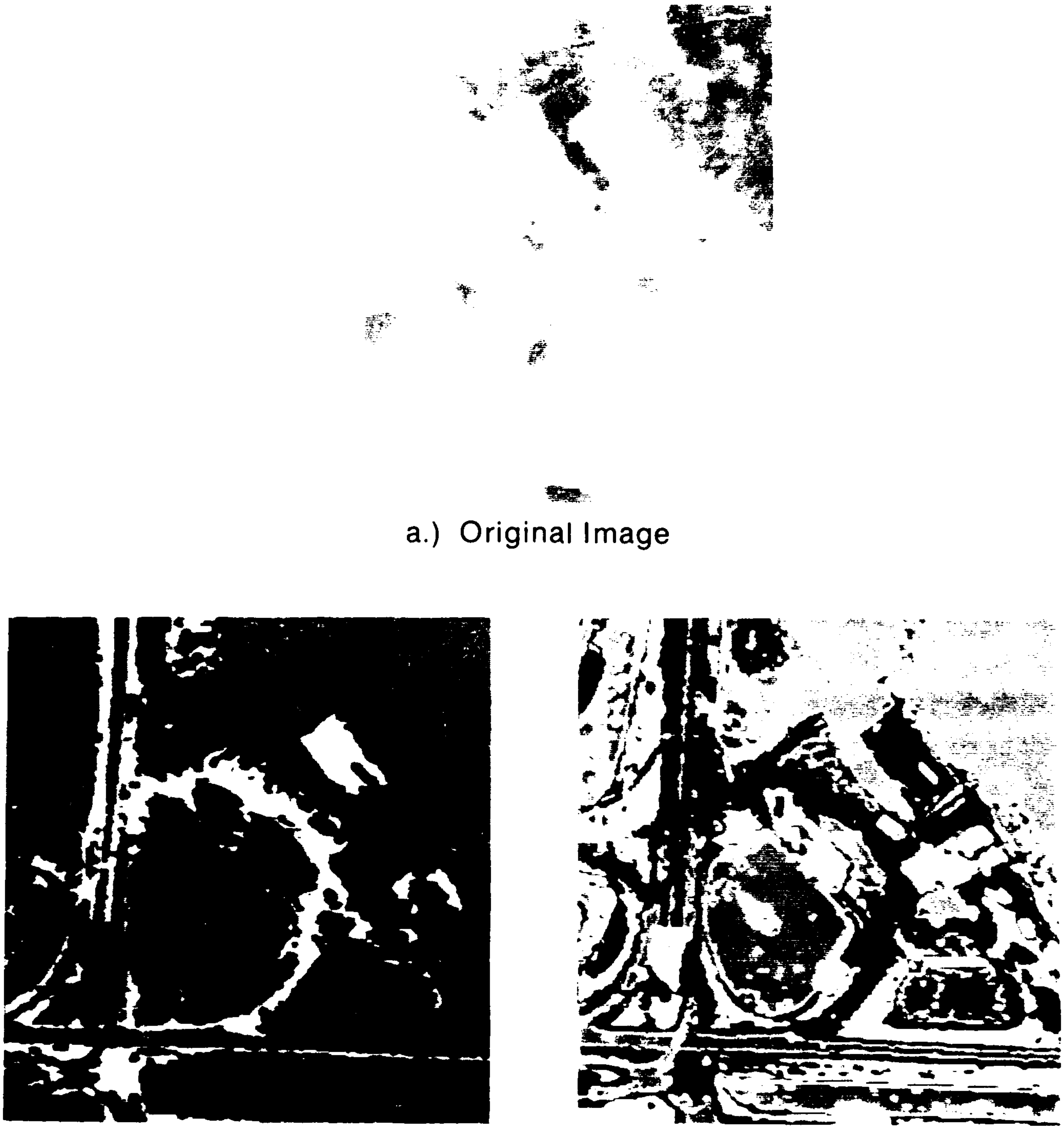


a.) Original Image

b.) Decision Window 1x1
(M=0)

c.) Decision Window 3x3
(M=1)

Fig. 4.2-6 Segmentation of Aerial Photo 1, $T_\sigma = 15, M = 1$

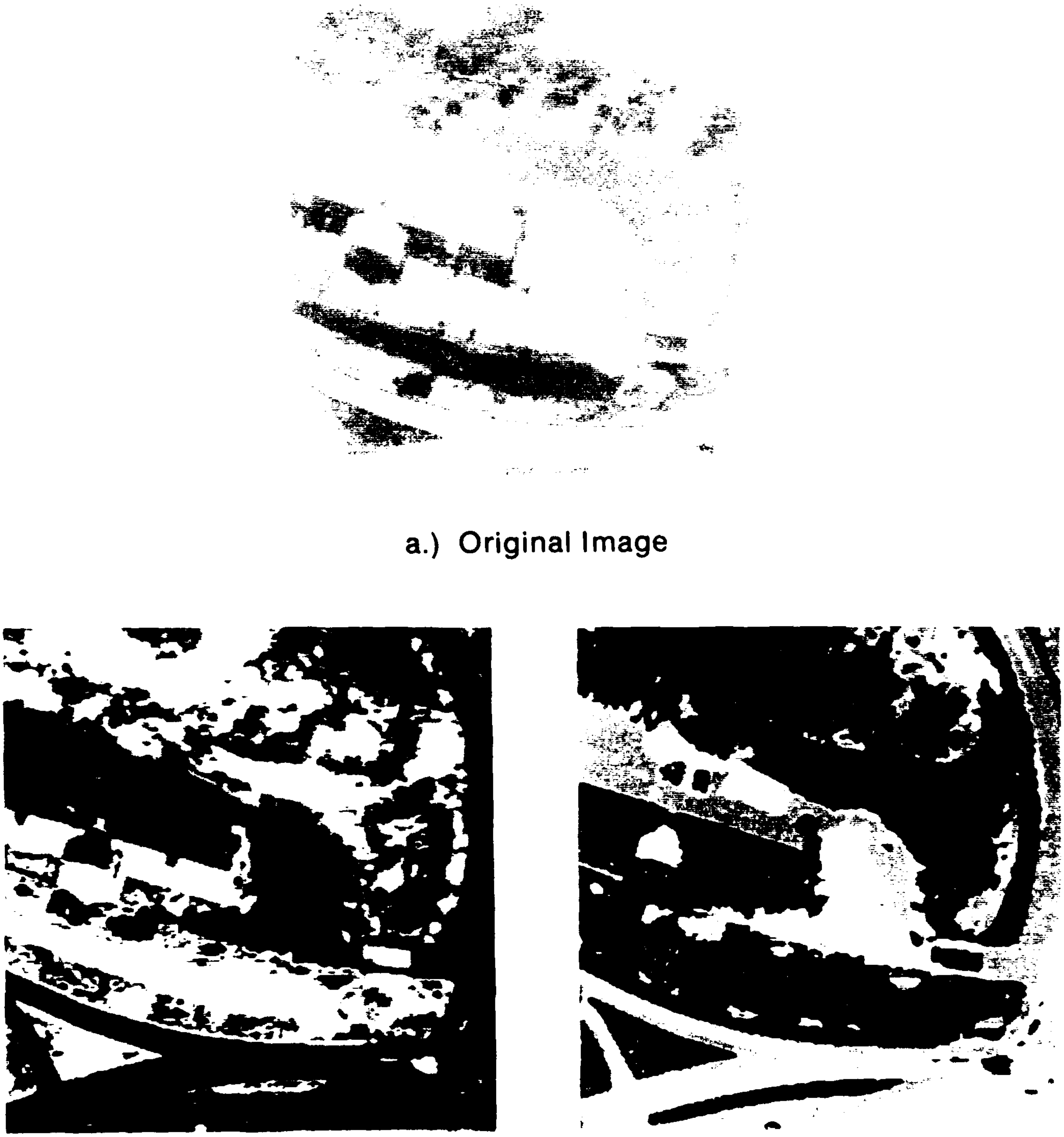


a.) Original Image

b.) 3-Class Segmentation

c.) 6-Class Segmentation

Fig. 4.2-7 Segmentation of Aerial Photo 2, $T_\sigma = 15, M = 1$

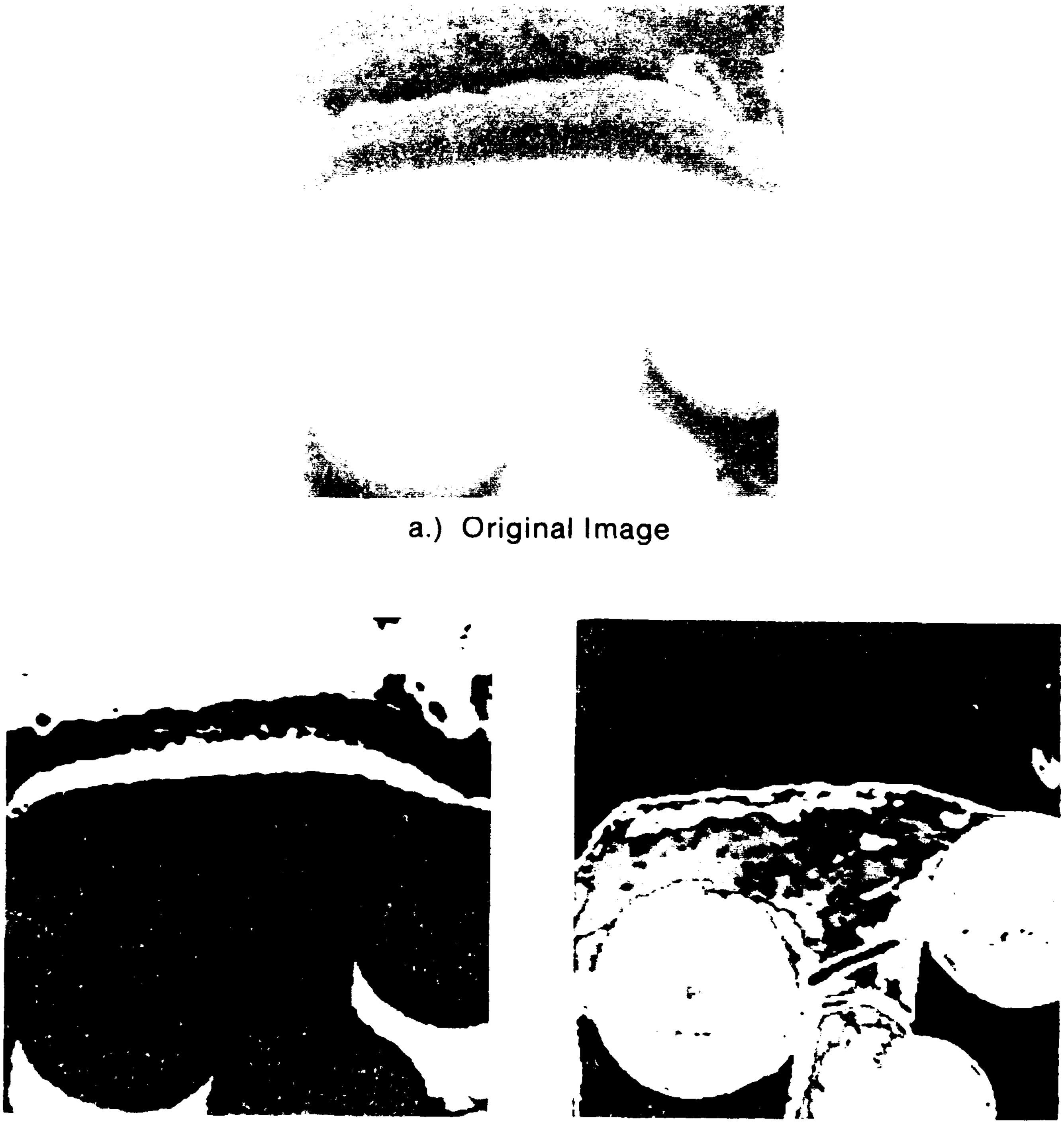


a.) Original Image

b.) 3-Class Segmentation

c.) 6-Class Segmentation

Fig. 4.2-8 Segmentation of Aerial Photo 3, $T_\sigma = 15, M = 1$



a.) Original Image

b.) 3-Class Segmentation

c.) 6-Class Segmentation

4.3 Texture Classification and Discrimination Using the Markov Random Field Model:

Over the last ten years, texture analysis has become a very important area in image processing applications and many techniques have been proposed and investigated. These techniques can be classified as either statistical or structural. In a statistical approach, texture is characterized in terms of its statistical properties such as mean, variance, or probability distribution. In a structural approach [1-3], texture is described as a formal language which contains specified primitives as elements and uses a placement rule as its grammar. In this paper, we will be only interested in a purely statistical approach.

Most of the existing statistical techniques, as summarized by Haralick in [1], are ad hoc in that no stochastic modeling assumptions are made for the texture classes. Textures are described in terms of some lower-level features such as mean, variance and correlation functions. Although these features provide some useful information about the texture classes, they are quite limited. As a result, while considerable success has been achieved for some special applications, there is little known in general as to how good these techniques are and what type of texture classes they can be applied to. In response to this shortcoming, Modestino, et al. [4-5] introduced a particular random field model, called the random tessellation process, for texture. Under this modeling assumption, texture analysis applications, such as classification and discrimination, can be formulated as classical statistical decision problems. More generally applicable and optimal solutions can then be obtained. However, as pointed out in [4-5], there are some unsatisfactory features of their model.

The recent developments in Markov random field (MRF) theory provide a powerful alternative texture model and have resulted in intensive research activity in MRF model-based texture analysis techniques [6-9]. Comparing to the previously proposed techniques, the MRF model-based approach has several distinguishing features.

First of all, the MRF, also known as the Gibbs Random Field (GRF), is characterized by the joint probability distribution function of the random variables on the entire lattice over which the MRF is defined. This provides complete information about the statistical properties of the random field. Secondly, the joint probability of the random field can be specified in terms of a few parameters, which makes the model mathematically tractable.

Finally, synthetic textures that closely resemble real-world textures can be generated by properly selecting a specific MRF model. In this paper, we describe a novel MRF model-based maximum-likelihood (ML) approach to texture classification and discrimination.

In a texture *classification* problem, an observed image is to be assigned to one of a finite number of classes according to its texture. Abend, et al. [10] proposed a Markov mesh model-based approach for texture classification. As an extension of the Markov chain to two dimensions, it has a causal structure and recently has been shown [11] to be a subclass of the MRF which is non-causal in general. Chellappa, et al. [12] have shown some success on texture classification using a non-causal Gaussian Markov Random Field model which is again a specific MRF model [11]. A similar approach is proposed in [13]. In this paper, we will consider the general MRF model which is noncausal and non-Gaussian. As can be seen later, this class of MRF models is more convenient for the classification of textures with few gray levels, as with binary textures, for example.

In a texture *discrimination* problem, an observed image is to be separated into disjoint regions of different textures, hence is also known as texture image segmentation. Derin, et al. [8] proposed a *maximum a posteriori (MAP) estimation* approach for texture discrimination using the MRF model. More specifically, they have considered a hierarchical image model. First, the distribution of the texture regions on the image lattice is modeled as a MRF. Then, different texture types are modeled by different MRF models. The maximization in this MAP approach is performed through dynamic programming with some approximations made on the a posteriori probability functional. The model parameters for different textures are assumed to be estimated from training data while the model parameters for the distribution of regions are chosen heuristically. In other words, this is a *supervised* approach. A similar supervised MAP approach is developed in [13] under a Gaussian Markov modeling assumption.

In this paper, we use a novel *unsupervised* ML approach. First of all, we do not assume a model for the distribution of regions since, even in those cases for which training data for different textures are available, the training data for the region distribution is rarely available. It is for this reason, in particular, that we have avoided a MAP approach and made use of a simpler ML approach which requires less a priori knowledge. Secondly, we have considered the case when the training data for each texture type is not available. A clustering technique is used to estimate the MRF model parameters directly from the observed image, resulting in an *unsupervised* scheme. Finally, texture discrimination is accomplished by assigning each pixel of the image into different texture classes through an ML test performed on the basis of neighboring pixels. This results in a highly parallel algorithm. As will be shown, compared to a dynamic programming approach, this algorithm requires far less computation.

After a brief review of the MRF theory in the next section, the ML approach for texture classification and discrimination with corresponding experimental results will be presented in Section 4.3.2 and Section 4.3.3, respectively. A summary is provided in Section 4.3.4.

4.3.1 The Markov Random Field Model:

The MRF model used in this paper originated from studies in statistical physics and recently has been adapted to an image processing context. In this section, we review some basic theory and some specific MRF models that will be used later.

For simplicity, we consider only digital images. That is, images with finite size and a finite number of grey levels. In particular, we define an image to be a two-dimensional (2-D) array over a finite square lattice, denoted by $\mathbf{f} = \{f(i,j), (i,j)\epsilon\mathbf{L}\}$ where $\mathbf{L} = \{(i,j), 1 \leq i \leq N, l \leq j \leq N\}$, and $f(i,j)$ can assume only a finite number of values.

A random field is defined to be a family of random variables defined over the 2-D lattice $\mathbf{L}$. Denote the random field by $\mathbf{X}$, the random variable at (i,j) by $\mathbf{X}(i,j)$, then $\mathbf{X} = \{X(i,j), (i,j)\epsilon\mathbf{L}\}$. In statistical image modeling, images are considered as realizations of random fields. In this paper, capital letters are used for random fields or random variables while lowercase letters are used for realizations or sample values.

A MRF on a 2-D lattice is a random field with the special property that the statistics of a point in the lattice given those of the rest of the lattice depends only on a few points known as its neighbors. More rigorous definitions are presented in what follows, starting with the concept of a neighborhood system.

Definition 1: A collection of subsets of $\mathbf{L}$, $\mathbf{n} = \{n(i,j), (i,j)\epsilon\mathbf{L}, n(i,j) \subset \underline{\mathbf{L}}\}$ is a neighborhood system on $\mathbf{L}$, if and only if

(i) $(i,j) \notin n(i,j)$.

(ii) if $(k,\ell)\epsilon n(i,j)$, then $(i,j)\epsilon n(k,\ell)$.

Some typical neighborhood system configurations are shown in Fig. 4.3-1. As indicated there, a neighborhood system can be classified as first-order, second-order, etc., according to the number of neighbors each lattice point has. To avoid boundary problems, a periodic lattice structure is assumed. Under this condition, all the points in **L** will have the same number of neighbors. A MRF is then defined with respect to a specified neighborhood system.

<u>Definition 2</u>: Let n be a neighborhood system over the 2-D lattice **L**. A random field $\underline{\mathbf{X}} = \{X(i,j), (i,j)\epsilon\underline{\mathbf{L}}\}$ is a MRF with respect to n, if and only if

$$(i) \quad P[\underline{X} = \underline{x}] > 0, \text{for all } \underline{x} \tag{1a}$$

$$(ii) \quad P[X(i,j) = x(i,j)|X(k,\ell) = x(k,\ell), (k,\ell)\epsilon\underline{L}, (k,\ell) \neq (i,j)]$$

$$= P[X(i,j) = x(i,j)|X(k,\ell) = x(k,\ell), (k,\ell)\epsilon n(i,j)] \tag{1b}$$

where P[·] and P[·|·] indicate the joint and conditional probability distributions of the random field, respectively. The order of the neighborhood system **n** is called the *order* of the MRF and the conditional probabilities in (1b) are also called the *local* characteristics.

The concept of the MRF would not be very useful for practical applications if it were not for the Hammersley and Clifford theorem which establishes the relation between the MRF and the Gibbs Random Field (GRF) and hence provides the functional form of the joint probability distribution function for a MRF. Before the GRF can be defined, the important concept of a clique must be introduced.

<u>Definition 3</u>: Given a lattice and neighborhood system pair, (**L**, **n**), a clique on the lattice, denoted by c, is a subset of **L**, such that

(i) c contains at least a single point of **L**

(ii) if $(k,\ell)\epsilon c, (i,j)\epsilon c$ and $(i,j) \neq (k,\ell)$, then $(i,j)\epsilon n(k\ \ell)$.

In particular, the collection of all the cliques of the pair (**L**, **n**) is denoted by C(**L**, **n**). Examples of clique types under different neighborhood systems are shown in Fig. 4.3-2. Now, the GRF can be defined as follows:

Definition 4: A random field $\mathbf{X} = \{X(i,j), (i,j)\epsilon\mathbf{L}\}$ is a GRF with respect to a given neighborhood system $\mathbf{n}$, if and only if its joint probability distribution function is of the following form:

$$P[\mathbf{X} = \mathbf{x}] = Z^{-1} exp[-U(\mathbf{x})], \tag{2a}$$

where

$$U(\mathbf{x}) = \sum_{c\epsilon C(\mathbf{L},\mathbf{n})} V_c(\mathbf{x}), \tag{2b}$$

and

$$Z = \sum_{all \mathbf{x}} exp[-U(\mathbf{x})]. \tag{2c}$$

Here, $V_c(\mathbf{x})$ is called the ***clique function*** and it depends only on the points in clique c while Z, called the partition function, is a normalizing factor to make (2a) a valid probability distribution. Notice that the GRF is defined in terms of its joint probability distribution, which provides complete information about the random field, while in the case of a MRF, there is little known about the joint probability distribution. Similarly, the conditional probabilities or local characteristics of a GRF can be found from the joint probability distribution, while in the case of the MRF the conditional probabilities are not readily apparent. Hammersley and Clifford have established the equivalence between the MRF and GRF, hence making the MRF a feasible model for practical applications such as texture modeling. This theorem will be simply stated in what follows. The proof is rather involved and can be found in Besag's work [14].

Theorem: A random field $\mathbf{X} = \{X(i,j), (i,j)\epsilon\mathbf{L}\}$ defined over $\mathbf{L}$ is a MRF with respect to the given neighborhood system $\mathbf{n}$ if and only if it is a GRF with respect to $\mathbf{n}$.

In this paper, two MRF models will be used as texture models, presented in the following examples in terms of their conditional probability distributions. These models have been widely used for real-world two-dimensional (2-D) phenomena, including textures, and have been shown to be simple and effective [6-10]. They are the main MRF texture models used in this paper. However, other models can also be defined for applications of interest by properly selecting the clique functions [14].

A.) Example of a First-Order MRF:

Consider a first-order MRF with the neighborhood system and its clique types shown in Fig.'s 4.3-1a and 4.3-2a. The joint probability distribution function of this MRF is:

$$\begin{aligned} P[\mathbf{X}=\mathbf{x}] = Z^{-1} exp[a \sum_{(i,j)\epsilon\mathbf{L}} x(i,j) \\ + b_1 \sum_{(i,j)\epsilon\mathbf{L}} x(i,j)x(i,j-1) \\ + b_2 \sum_{(i,j)\epsilon\mathbf{L}} x(i,j)x(i-1,j)]. \end{aligned} \tag{3}$$

Notice that in the above summations the periodic lattice structure is assumed. The local characteristics of the MRF can be found easily by Bayes' conditional probability formula as:

$$\begin{aligned} P[X(i,j) &= x(i,j)|X(k,\ell) = x(k,\ell),(k,\ell)\epsilon n(i,j)] \\ &= \frac{\exp[x(i,j)s(i,j)]}{\sum_{x(i,j)} \exp[x(i,j)s(i,j)]}, \end{aligned} \tag{4}$$

where

$$\begin{aligned} s(i,j) &= a + b_1[x(i,j-1) + x(i,j+1)] \\ &+ b_2[x(i-1,j) + x(i+1,j)] \end{aligned} \tag{5}$$

and the sum is over all possible values of $x(i,j)$. A special case is when $b_1 = b_2 = b$. This is called an *isotropic* MRF.

B.) Example of a Second-Order MRF:

This MRF model has the second-order neighborhood system and clique types shown in Fig.'s 4.3-1b and 4.3-2b with the following joint probability distribution function

$$\begin{aligned}P[\mathbf{X}=\mathbf{x}] = Z^{-1}exp[a & \sum_{(i,j)\epsilon\mathbf{L}} x(i,j) \\ & + b_1 \sum_{(i,j)\epsilon\mathbf{L}} x(i,j)x(i,j-1) \\ & + b_2 \sum_{(i,j)\epsilon\mathbf{L}} x(i,j)x(i-1,j) \\ & + c_1 \sum_{(i,j)\epsilon\mathbf{L}} x(i,j)x(i-1,j-1) \\ & + c_2 \sum_{(i,j)\epsilon\mathbf{L}} x(i,j)x(i-1,j+1)].\end{aligned} \tag{6}$$

Again, the periodic assumption is made for the above summations. Similar to the previous example, the local characteristics can be found as:

$$\begin{aligned}P[X(i,j) &= x(i,j)|X(k,\ell) = x(k,\ell),(k,\ell)\epsilon n(i,j)] \\ &= \frac{exp[x(i,j)t(i,j)]}{\sum_{x(i,j)} exp[x(i,j)t(i,j)]},\end{aligned} \tag{7}$$

where now

$$\begin{aligned}t(i,j) = a &+ b_1[x(i,j-1) + x(i,j+1)] \\ &+ b_2[x(i-1,j) + x(i+1,j)] \\ &+ c_1[x(i-1,j-1) + x(i+1,j+1)] \\ &+ c_2[x(i-1,j+1) + x(i+1,j-1)].\end{aligned} \tag{8}$$

Suppose a given image $\mathbf{f} = \{f(i,j),(i,j)\epsilon\mathbf{L}\}$ is modeled by a specific MRF. It is desired to estimate the model parameters of the MRF from the image data. Since the ML approach to be developed later bears close relationship with the parameter estimation algorithm, we will describe it in detail.

Let the parameters be denoted by the vector $\mathbf{a}$. For example, for the first-order isotropic MRF, $\mathbf{a} = (a,b)$. The *maximum-likelihood* (ML) estimate of $\mathbf{a}$, denoted $\hat{\mathbf{a}}_{ML}$, is obtained by maximizing the *likelihood functional* $L(\mathbf{f};\mathbf{a}) \overset{\triangle}{=} P(f|a)$ where $P(\mathbf{f}|\mathbf{a})$ is the joint probability of $\mathbf{f}$ as a realization of the MRF given $\mathbf{a}$ as the parameter vector. Once the model is chosen, $P(\mathbf{f}|\mathbf{a})$ is a functional of the vector $\mathbf{a}$. Although this estimate is optimum, it is difficult to compute. This is because the computation of the conditional joint probability functional, $P(\mathbf{f}|\mathbf{a})$, involves the computation of the normalization factor Z in (2), which in turn contains all the possible realizations of the MRF, and in this case is also a functional of $\mathbf{a}$. The computation is almost impossible even for a binary MRF (BMRF) on a reasonably small lattice. Obviously, a suboptimum technique needs to be used which preserves some optimality of the (ML) approach and yet is computationally feasible. Besag's coding method [14] is such a technique.

In this coding method, the 2-D lattice is separated into disjoint sets of points, called codings, according to the neighborhood system assumption of the MRF. The codings are defined in such a way that the points in each coding are conditionally independent given the random variables on the other codings. From this property, no two points in the same coding are neighbors. Examples of codings are shown in Fig. 4.3-3 for the first and second-order MRF's discussed in this section.

Suppose for a given MRF there are M codings, denoted by $C_1, C_2, ..., C_M$. Define for the *m'th* coding the following *coding-likelihood*:

$$L_m(\mathbf{f};\mathbf{a}) = P[F(i,j) = f(i,j), (i,j)\epsilon C_m | a, F(k,\ell) = f(k,\ell), (k,\ell)\epsilon C_q$$
$$\text{for all } q,\ 1 \leq q \leq M, q \neq m]; \quad m = 1,2,...,M. \tag{9}$$

Since the points in a fixed coding are conditionally independent, $L_m(\mathbf{f};\mathbf{a})$ can also be written as

$$L_m(\mathbf{f};\mathbf{a}) = \prod_{(i,j)\epsilon C_m} P[F(i,j) = f(i,j)|\mathbf{a}, F(k,\ell) = f(k,\ell), (k,\ell)\epsilon n(i,j)], \tag{10}$$

where $P[\cdot|\cdot]$ is the local characteristic which can be easily computed. Therefore, $L_m(\mathbf{f};\mathbf{a})$ is also easy to compute. The *m'th* coding estimate of the parameter vector can be obtained by maximizing $L_m(\mathbf{f};\mathbf{a})$ with respect to $\mathbf{a}$ for $m = 1,2,...,M$. The resulting estimate will be denoted by $\hat{\mathbf{a}}^{(m)}_{MCL}, m = 1, 2, ..., M$, where MCL indicates *maximum coding likelihood estimation* and the superscript indicates the *m'th* coding. It has been noticed by Besag [14], Cross and Jain [7], and the present authors that this coding method provides very accurate estimates. Also, the estimates obtained by using different codings are very close to each other. In the remainder of this paper, for definiteness, we compute the maximum coding-likelihood estimate as the average over all $\hat{\mathbf{a}}^{(m)}_{MCL}$. That is, we take

$$\hat{\mathbf{a}}_{MCL} \triangleq \frac{1}{M} \sum_{m=1}^{M} \hat{\mathbf{a}}^{(m)}_{MCL}. \tag{11}$$

4.3.2 Texture Classification:

In this section, we will first develop the MRF model-based ML approach for texture classification for the case where training data is available and then show that it can be combined with a clustering algorithm when training data is not available. A block diagram outlining an approach to the texture classification problem is shown in Fig. 4.3-4. The inputs to the classifier are digital images containing texture data from one of a finite number of texture classes. These images are separated into the unknown or test set of images, whose texture class is unknown, and the training set of images, whose texture class is known a priori. The training set is necessary to provide information that will be used by the classifier in the decision process. In the parameter estimation stage, information essential for differentiating the texture classes is estimated from the training set of images and used to adapt the classifier for all the possible texture classes. Then the unknown images will be processed by the classifier to decide which texture class is presented.

Let the image data on which the classifier is to operate be denoted by $\mathbf{f} = \{f(i,j), (i,j)\epsilon\mathbf{L}\}$. Assume that there are K texture classes or hypotheses labeled by $H_k, k = 0, 1, 2, ..., K-1$. The class-conditional likelihood functional [15], assuming the *k'th* hypothesis is acting, is then defined as:

$$L_k(\mathbf{f}) = P(\mathbf{f}|H_k); \quad k = 0, 1, ..., K-1, \tag{12}$$

where $P(\mathbf{f}|H_k)$ is the joint probability distribution of the random field assuming that hypothesis H_k is acting.

If the MRF described in Section 4.3-2 is used as the texture model, the likelihood functional in (12) is the joint probability of the MRF model for the *k'th* texture class. In this paper, we assume different texture classes are modeled by MRF's with the same functional forms of probability distribution but different in the parameters in these functionals. For example, if the first-order isotropic model is used, the likelihood functional is

$$L_k(\mathbf{f}) = P[\mathbf{f}|\mathbf{a}_k] = P[\mathbf{f}|(a_k, b_k)]; \quad k = 0, 1, ..., K - 1, \tag{13}$$

where $\mathbf{a}_k = (a_k, b_k)$ is the parameter vector for class k. According to the ML decision rule, which minimizes the classification error probability, the data is assigned to texture class k_0 corresponding to the index that maximizes the class-conditional likelihood functional in (12).

Although this approach is optimum, it is usually difficult to implement, since the computation of the normalization factor Z in $P[\mathbf{f}|H_k]$, just as the case of parameter estimation, involves all the possible realizations of the MRF. A reasonable suboptimum approach is to use a likelihood functional which is closely related to the joint probability function and is yet easy to compute. The coding likelihood used in Besag's coding method for model parameter estimation can be used to develop such a suboptimum approach. Instead of computing the joint probability, the coding likelihood, defined as follows, is used as the likelihood functional. More specifically, suppose there are M codings denoted by $C_1, C_2, ..., C_M$, the class-conditional likelihood evaluated on the *m'th* coding is

$$\begin{aligned} L_{m,k}(\mathbf{f}) &= P[f(i,j), (i,j)\epsilon C_m | \mathbf{a}_k, f(k,\ell), (k,\ell) \notin C_m] \\ &= \prod_{(i,j)\epsilon C_m} P[f(i,j)|\mathbf{a}_k, f(k,\ell), (k,\ell)\epsilon n(i,j)]; \\ & m = 1, 2, ..., M; k = 0, 1, ..., K - 1. \end{aligned} \tag{14}$$

As mentioned previously, the coding likelihoods computed for different codings are very close. In principle, we could choose any value of $m = 1, 2, ..., M$ and perform ML classification on the basis of $L_{m,k}(\mathbf{f}), k = 0, 1, ..., K - 1$. However, for definiteness, we have chosen again to average the various coding likelihoods. More specifically, define

$$L_k(\mathbf{f}) \triangleq \sum_{k=1}^{M} L_{m,k}(\mathbf{f}). \tag{15}$$

The decision rule then becomes: assign the data to class k_0 if

$$L_{k_0}(\mathbf{f}) = \max_{0 \le k \le K-1} L_k(\mathbf{f}). \tag{16}$$

When training data is available, this suboptimum ML classifier can be implemented with the estimated parameter vectors for each texture class from the training data set using Besag's coding method. When the training data is not available, as is often the case in many practical applications, the parameter vector for each texture class can be obtained as follows. First, model parameter vectors are estimated from every observed image. Then these vectors, also called samples, are grouped into several disjoint sets called clusters. Finally the centroids of the clusters are used as the estimated class model vectors in the ML classifier. This is usually referred to as a clustering procedure in pattern recognition and the algorithm which performs the grouping is called a clustering algorithm. There are many clustering algorithms available [16], [17]. In this paper we make use of the K-means algorithm [17]. It has been shown that this algorithm is optimal under a specific cluster criterion function and convergent under a well defined condition for the distribution of samples. It is also simple to implement. The major disadvantage of this algorithm is that the number of clusters has to be known before applying the algorithm, which is sometimes an impractical assumption. A number of techniques, mostly heuristic, have been proposed to determine the number of clusters, or classes, and there is no well accepted theory [16], [18]. In this paper we assume that the number of classes is known or predetermined and pursue this problem in other separate work.

Texture classification experiments have been performed on both synthetic texture classes and natural texture classes to test the efficacy of the ML approach described above. The synthetic textures used are realizations of binary Markov random fields (BMRF's) while the natural textures are equal probability quantized binary images from Brodatz's photo album [19]. For each case, the classification is performed with the aid of training data or using the clustering method. The experimental results are presented as follows:

A.) Supervised Classification of Synthetic Textures:

The texture classes used in this experiment are generated as follows: First, a realization of a binary MRF of size 240 × 24 specified by a parameter vector $\mathbf{a}_k, k = 0, 1, ..., K - 1$, is generated using Geman and Geman's algorithm [9]. Next, each 240 × 240 image is cut into nine 80 × 80 subimages. Finally, the subimage at the upper-left corner is used as the training data for that texture class, while the rest of the subimages are taken as test data.

The first-order isotropic BMRF is used in both texture generation and parameter estimation. Four texture classes are generated. The estimated model parameters from the training data for each texture class are shown in Table 4.3-1a along with the actual model parameters used to generate the texture classes. The 240 × 240 image for each texture class is shown in Fig. 4.3-5. It can be seen from these images that they have different clusterings. The classification results are shown in the contingency table in Table 4.3-2a. All the data are correctly classified. Similar results have been obtained for second-order MRF's [19].

B.) Unsupervised Classification of Synthetic Textures

In this experiment all the subimages in A.) are used as test data. The clustering algorithm described previously is applied on the set of model vectors estimated from the thirty-six subimages, using the same BMRF model as A.) and assuming the number of classes is known to be four. The cluster centroids as shown in Table 4.3-1b along with the model parameters that generates the synthetic textures and the classification result is shown in Table 4.3-2b. The estimated model vectors obtained by clustering are very close to the actual values and all data are correctly assigned.

C.) Supervised Classification for Natural Textures

The natural textures used in this experiment are four texture images from Brodatz' photo album [20]. They were originally 256 grey level images of size 128 × 128. An equal probability quantization is performed to transform these textures into binary images. Figure 4.3-6 shows the binary quantized images of these texture classes. The training data and test data sets are obtained as follows: First, each image is cut into four 64 × 64 subimages. Then the subimage at the upper-left corner is used as training data while the rest are used as test data for that texture class. Usually, natural textures are modeled by MRF models of order higher than one [7]. However, we found that when fitted with the second-order MRF model, parameter c_1 and c_2 for these images are quite small comparing to the other ones in (8), hence all the binary texture classes are modeled as first-order BMRF's. The class parameter vector for each class is estimated from training data and shown in Table 4.3-3a with corresponding classification results in Table 4-3.4a. All the subimages have been correctly classified.

D.) Unsupervised Classification for Natural Texture

The results for this part is obtained in the same way as in C.) except the class model vectors are obtained through clustering, assuming the number of classes is known to be four. The centroids of the clusters and classification results are shown in Table 4.3-3b and 4.3-4b. All the data are correctly classified.

4.3.3 Texture Discrimination:

Unlike texture classification, which assigns an entire image to a specific class, the interest now is to discriminate between different texture classes within the image. The ML approach of [4] is adapted, under a MRF modeling assumption, to develop a new likelihood functional using the information provided by the MRF model. Discrimination experiments have been performed on test images containing synthetic textures and natural textures.

Assume that textured image, $\mathbf{f} = \{f(i,j),(i,j)\epsilon\mathbf{L}\}$, is a realization of a random field denoted by $\mathbf{F} = \{F(i,j),(i,j)\epsilon\mathbf{L}\}$ and the lattice can be decomposed into regions $\mathbf{R}_1, ..., \mathbf{R}_Q$ of K different textures as shown in Fig. 4.3-7. That is

$$\mathbf{L} = \cup_{q=1}^{Q}\mathbf{R}_q, \tag{17}$$

where $K \leq Q$.

We will model the texture classes within each region as a MRF defined over that region. Regions belonging to the same texture class will have the same MRF model vector. Suppose each pixel (i,j) of the image belongs to one of K texture classes denoted by the hypothesis H_k, $k = 0, 1, ..., K-1$. Texture discrimination is the process in which each pixel is assigned to a particular class. Suppose a window of size $(2M+1) \times (2M+1)$ is constructed for each pixel position (i,j) and the pixels within this window are denoted by $\mathcal{F}_{i,j} = \{f(k,\ell),(k,\ell)\epsilon\mathcal{W}_{i,j}\}$, where $\mathcal{W}_{i,j} = \{(k,\ell), i - M \leq k \leq i + M, j - M \leq \ell \leq j + M\}$, with M is much less than N and the periodic condition is imposed. The likelihood functional in this case, given that the *k'th* hypothesis is acting, is defined as

$$L_k\{\mathcal{F}_{i,j}\} = P\{\mathcal{F}_{i,j}|H_k\}; \quad k = 0, 1, ..., K - 1. \tag{18}$$

Pixel (i,j) will be assigned to texture class k_0, if

$$L_{k_0}\{\mathcal{F}_{i,j}\} = \max_{0 \leq k \leq K-1} L_k\{\mathcal{F}_{i,j}\}. \tag{19}$$

After this procedure is performed for all the pixels, the image will be segmented into disjoint regions which belong to different texture classes in such a way as to minimize the classification error [4].

Although this method is theoretically optimal, the joint probability of the pixels in the window is hard to evaluate and complicates the evaluation of $L_k\{\cdot\}$. For example, assuming the texture in each regions $\mathbf{R}_q$ is modeled as a MRF, the likelihood functional in (17) can be written as

$$L_k\{\mathcal{F}_{i,j}\} = \sum_{(k,\ell)\notin\mathcal{W}_{i,j}} \sum_{f(k,\ell)} P[f(k,\ell),(k,\ell)\epsilon\mathbf{R}_{i,j}|H_k]; \quad k = 0, 1, ..., K - 1, \tag{20}$$

where $P[\cdot|H_k]$ is the class-conditional joint probability distribution of the MRF and $\mathbf{R}_{i,j}$ is one of the $\mathbf{R}_q'$s in (17). The difficulty of evaluating (20) is that region $\mathbf{R}_{i,j}$ is unknown before the discrimination process. Again, as in the case of texture classification, a suboptimal approach is desired which preserves some optimality of the previous likelihood functional and yet is easy to compute. The ML approach developed in this section is such an approach which uses the coding structure. In this approach, the likelihood is the coding-likelihood evaluated only within the decision window centered at the pixel to be classified. More specifically, we chose a coding which contains the center point in the window, denoted by $C_{i,j}$. Now, the ML approach can be described as follows: Define the class-conditional coding-likelihood as

$$
\begin{aligned}
L_{c,k}\{\mathcal{F}_{i,j}\} &= P[f(k,\ell),(k,\ell)\epsilon C_{i,j} \cap \mathcal{W}_{i,j} | f(m,n),(m,n)\epsilon \mathbf{R}_{i,j},(m,n) \notin C_{i,j}, H_k] \\
&= \prod_{(k,\ell)\epsilon C_{i,j} \cap \mathcal{W}_{i,j}} P[f(k,\ell) | f(m,n),(m,n)\epsilon n(k,\ell), H_k]; \\
& k = 0,1,...,K-1. \qquad (21)
\end{aligned}
$$

This likelihood can be easily computed from the local characteristics and the discriminator will assign a pixel (i,j) to texture class k_0 if

$$L_{c,k_0}\{\mathcal{F}_{i,j}\} = \max_{0 \leq k \leq K-1} L_{c,k}\{\mathcal{F}_{i,j}\}. \qquad (22)$$

Due to its simplicity, the algorithm above is quite efficient. For an $N \times N$ image with K texture class types it requires approximately N^2K computations to process the image. Note that the previously described MAP algorithm of Derin, et al. requires about N^2K^D computations using dynamic programming where D is an integer and $D \geq 2$. In addition, the ML algorithm proposed here can be implemented through parallel computation since the assignment of a pixel in the image does not depend on that of others.

When training data for different types of textures are available, the MRF model class vectors can be estimated from them, resulting in supervised texture discrimination. When the training data is not available, a clustering scheme, similar to the one described in the last section for unsupervised texture classification can be used. In particular, consider a sliding window of size $M_1 \times M_2$ on the observed image as shown in Fig. 4.3-8 where we assume $M_1, M_2 << N$. At each position of the window, a MRF parameter vector is estimated from the data within the window. The parameter vectors obtained are then used as the sample vectors for the clustering algorithm. We have chosen the K-means algorithm as the clustering algorithm. Finally the centroids of the clusters are taken as class model vectors as if they were estimated already from training data. Here, again, we assume knowledge of the number of clusters. In this clustering approach the choice of size of the window is quite important. If the size is too large, a single window might contain data from several different texture classes whereas if the size is too small, a single window might not contain enough data for reliable estimation. Both result in unreliable sample vectors which will effect the accuracy of the results of the clustering algorithm. The size of the window should be related to the expected size of the texture regions in the image. At this point, we make the choice heuristically trying different windows and selecting the one which provides reasonably good segmentations. Notice here, the sliding window described above is used for unsupervised model parameter estimation before segmentation whereas the decision window described previously is used during the segmentation.

Texture discrimination experiments have been performed on images containing synthetic and natural textures. Each test image used in these experiments is 128 × 128 and contains two different textures distributed in the image according to the "region-map" shown in Fig. 4.3-9. After the ML discrimination, each pixel in the resulting image is assigned one of two gray levels, depending on which texture class it belongs to. The results for both supervised and unsupervised discrimination are presented below. While the results are for binary and two class images, the extension of the method to non-binary and multi-class problem is straightforward.

A.) Supervised Discrimination of Synthetic Textures:

The two-class test image is shown in Fig. 4.3-10b along with the region map. The two synthetic textures are generated using the first-order isotropic BMRF's. The model parameters are estimated from training data which are different realization of the above BRMF's and are listed in Table 4.3-5a. The results of applying the ML discriminator in (22) with different decision window sizes are shown in Fig. 4.3-10c and 4.3-10d. As can be seen, the 3 × 3 window provides very good discrimination. More extensive experimental results of the same nature can be found in [19], [21].

B.) Unsupervised Discrimination of Synthetic Textures:

The test image used in this experiment is the same as that in A.). The model vectors for the two different texture classes are obtained from clustering using a nonoverlapping 32 × 32 sliding window.That is, we take $N_i = M_i, i = 1,2$, here and in all experimental results to follow. The estimated values are shown in Table 4.3-5b and the results of discrimination using these model vectors is shown in Fig.'s 4.3-10e and 4.3-10f. It can be seen the clustering scheme is quite effective in the ideal case when the textures are from MRF's.

C.) Supervised Discrimination for Natural Textures:

The test image shown in Fig. 4.3-11b, along with the region map, contains two binary quantized natural textures from Brodatz' photo album. These binary textures are modeled here by the first-order BMRF model. The model parameters are estimated from training data and are shown in Table 4.3-6a. The discrimination results for selected decision windows are shown in Fig. 4.3-11c and 4.3-11d. Notice now that a larger decision window is needed to obtain results comparible to those of A.). This might be caused by the model mismatch. However, the discrimination results are still quite good.

D.) Unsupervised Discrimination for Natural Textures:

The experiment in C.) is repeated with the model parameter vectors obtained from clustering with a nonoverlapping 16 × 16 sliding window. The resulted cluster centroids (model vectors) are shown in Table 4.3-6b and the texture discrimination results with different decision window size are shown in Fig. 4.3-11e, and f. Again, the clustering approach worked well.

4.3.4 Summary

In this paper, we have developed a MRF model-based ML approach to texture classification and discrimination problems. Under the MRF texture modeling assumption, they were formulated as statistical decision problems. To make computation feasible, the likelihood functional originally derived based on the joint probability distribution of the MRF model is approximated using Besag's coding method. Most of the statistical model-based approaches proposed previously are supervised. That is, they require a training data set for model parameter estimation. Unlike these approaches, we also consider unsupervised schemes which do not require the training data. For the latter, a novel clustering technique is proposed to estimate the model parameters directly from the observed image. Experimental results on texture classification and discrimination using these two schemes are shown to be quite promising.

However, there are limitations to the MRF model-based approach. For example, in the unsupervised clustering scheme we assume the number of different texture classes is known which is generally not the case. Although a number of methods exist which can be used to determine the number of classes, they are mostly ad hoc and there is little known as to how well they work in general. In some cases, a reasonable assumption might be made about the number of classes based on a priori knowledge of the situation. However, to make the unsupervised scheme work in general, it is desired to develop more reliable methods to estimate this number. One such approach is described in [21].

References

1. R.M. Haralick, "Statistical and structural approaches to texture", Proc. IEEE, Vol. 67, pp. 786-804, May 1979.
2. S.W. Zucker, "Toward a model of texture", COMPUTER GRAPHICS AND IMAGE PROCESSING, Vol. 5, pp. 190-202, 1976.
3. S.Y. Lu and K.S. Fu, "A syntactic approach to texture analysis", COMPUTER GRAPHICS AND IMAGE PROCESSING, Vol. 7, pp. 303-330, 1978.
4. J.W. Modestino, R.W. Fries, A.L. Vickers, "Texture discrimination based upon an assumed stochastic texture model", IEEE Trans. Pattern Anal. Machine Intel., Vol. PAMI-3, pp. 557-580, Sept. 1981.
5. J.W. Modestino and A.L. Vickers, "A maximum likelihood approach to texture classification", IEEE Trans. Pattern Anal. and Machine Intel., Vol. PAM-4, pp. 61-68, Jan. 1982.
6. M. Hassner and J. Sklansky, "The use of Markov random fields as models of texture", COMPUTER GRAPHICS AND IMAGE PROCESSING, Vol. 12, pp. 357- 370, 1980.
7. G.C. Cross and A.K. Jain, "Markov random field texture models", IEEE Trans. Pattern Anal. Machine Intel., Vol. PAMI-5, pp. 25-39, Jan. 1983.
8. H. Derin and H. Elliot, "Modeling and segmentation of noisy and textured images using Gibbs random fields", IEEE Trans. Pattern Anal. Machine Intel., Vol. PAMI-9, pp. 39-55, Jan. 1987.
9. S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images", IEEE Trans. Pattern Anal. Machine Intel., Vol. PAMI-6, pp. 721-741, Nov. 1984.
10. K. Abend, T. Harley and L.N. Kanal, "Classification of binary random patterns", IEEE Trans. Inform. Theory, Vol. IT-11, pp. 538-544, Oct. 1965.
11. H. Derin, "The use of Gibbs distributions in image processing", COMMUNICATIONS AND NETWORKS, I.F. Blake and V.H. Poor, ed. New York, Springer- Verlag, 1986.
12. R. Chellappa and S. Chatterjee, "Classification of textures using Gaussian Markov random fields", IEEE Trans. on Acoust., Speech and Sig. Proc., Vol. ASSP-33, pp. 959-963, Aug. 1985.

13. C.W. Therrien, T.F. Quatieri and D.E. Dudgeon, "Statistical model- based algorithms for image analysis", Proc. IEEE, Vol. 74, pp. 532-551, Apr. 1986.

14. J. Besag, "Spatial iteration and the statistical analysis of lattice systems (with discussion)", J. of Royal Statist. Soc., Series B, Vol. 36, pp. 192-326, 1974.

15. H.L. Van Trees, DETECTION, ESTIMATION AND MODULATION THEORY: Part I, John WIley & Sons, New York, 1968.

16. K. Fukunaga, INTRODUCTION TO STATISTICAL PATTERN RECOGNITION, Academic Press, 1972.

17. J.T. Tou and R.C. Gonzalez, PATTERN RECOGNITION PRINCIPLE, Addison-Wesley Publishing Company, 1974.

18. B. Everitt, CLUSTER ANALYSIS, published by Heinemann Education Books Ltd., (for) Social Science Research Council, London, 1974.

19. J. Zhang, "Markov random fields with applications to texture classification and discrimination", M.S. Thesis, Rensselaer Polytechnic Institute, Troy, New York, Dec. 1985.

20. P. Broadatz, TEXTURE: A PHOTO ALBUM FOR ARTISTS AND DESIGNERS, New York: Dover, 1956.

21. J. Zhang, "Stochastic model-based image analysis", Ph.D. Thesis, Rensselaer Polytechnic Institute, Troy, New York, in preparation.

22. F.S. Cohen and D.B. Cooper, "Simple, parallel, hierarchical and relaxation algorithms for segmenting non-causal Markovian random fields", IEEE Trans. Pattern Anal. Machine Intel., Vol. PAMI-9, pp. 195-219, March 1987.

23. L. Onsager, "Crystal statistics, I. A two-dimensional model with an order-disorder transition", PHYS. REVIEW, Vol. 65, pp. 117-149, 1944.

Table 4.3-1

	Parameters			
	a		b	
Texture Class	Actual	Estimated	Actual	Estimated
Class 1	-2.0	-2.18	1.0	1.07
Class 2	-6.0	-5.55	3.0	2.74
Class 3	2.0	2.01	-1.0	-1.03
Class 4	6.0	5.72	-3.0	-2.90

a.) Estimated from training data

	Parameters			
	a		b	
Texture Class	Actual	Estimated	Actual	Estimated
Class 1	-2.0	-2.02	1.0	1.01
Class 2	-6.0	-5.15	3.0	2.63
Class 3	2.0	1.96	-1.0	-0.94
Class 4	6.0	4.80	-3.0	-2.45

b.) Estimated by clustering

Estimated Model Parameters for the First-Order BMRF.

	Assigned Class			
True Class	1	2	3	4
1	9	0	0	0
2	0	9	0	0
3	0	0	9	0
4	0	0	0	9

No. of Subimages=36
No. of correct Assignments=36
% of correct classification=100%

a.) Supervised Classification

	Assigned Class			
True Class	1	2	3	4
1	9	0	0	0
2	0	9	0	0
3	0	0	9	0
4	0	0	0	9

No. of Subimages=36
No. of correct Assignments=36
% of correct classification=100%

b.) Unsupervised Classification

Table 4.3-2

Classification Results for the First-Order BMRF.

Table 4.3-3

	Estimated Model Parameters			
Parameters	Grass	Wood	Bark	Sand
a	-1.94	-4.56	-2.95	-2.26
b_1	0.75	0.36	1.21	1.23
b_2	1.14	4.07	1.71	1.17

a.) Estimated from training data

	Estimated Model Parameters			
Parameters	Grass	Wood	Bark	Sand
a	-1.80	-3.92	-2.77	-2.35
b_1	0.67	0.32	1.09	1.23
b_2	1.11	3.62	1.71	1.12

b.) Estimated by clustering

Estimated Parameters for Natural Texture
Samples Modeled as First-Order BMRF's.

	Assigned Class			
True Class	Grass	Wood	Bark	Sand
Grass	4	0	0	0
Wood	0	4	0	0
Bark	0	0	4	0
Sand	0	0	0	4

a.) Supervised Classification

	Assigned Class			
True Class	Grass	Wood	Bark	Sand
Grass	4	0	0	0
Wood	0	4	0	0
Bark	0	0	4	0
Sand	0	0	0	4

b.) Unsupervised Classification

Table 4.3-4

Classification Results for the Natural Texture Samples.

Table 4.3-5

	Est. Parameter	
Texture Class	a	b
Class 1	-5.55	2.74
Class 2	5.72	-2.90

a.) Estimated from training data

	Est. Parameter	
Texture Class	a	b
Class 1	-4.67	2.32
Class 2	3.56	-1.87

b.) Estimated by clustering

Estimated Parameters from the first-order BMRF For Synthetic Texture Discrimination.

Table 4.3-6

	Estimated Parameters		
Texture	a	b_1	b_2
Grass	-1.94	.75	1.14
Ripple	-3.03	.343	2.53

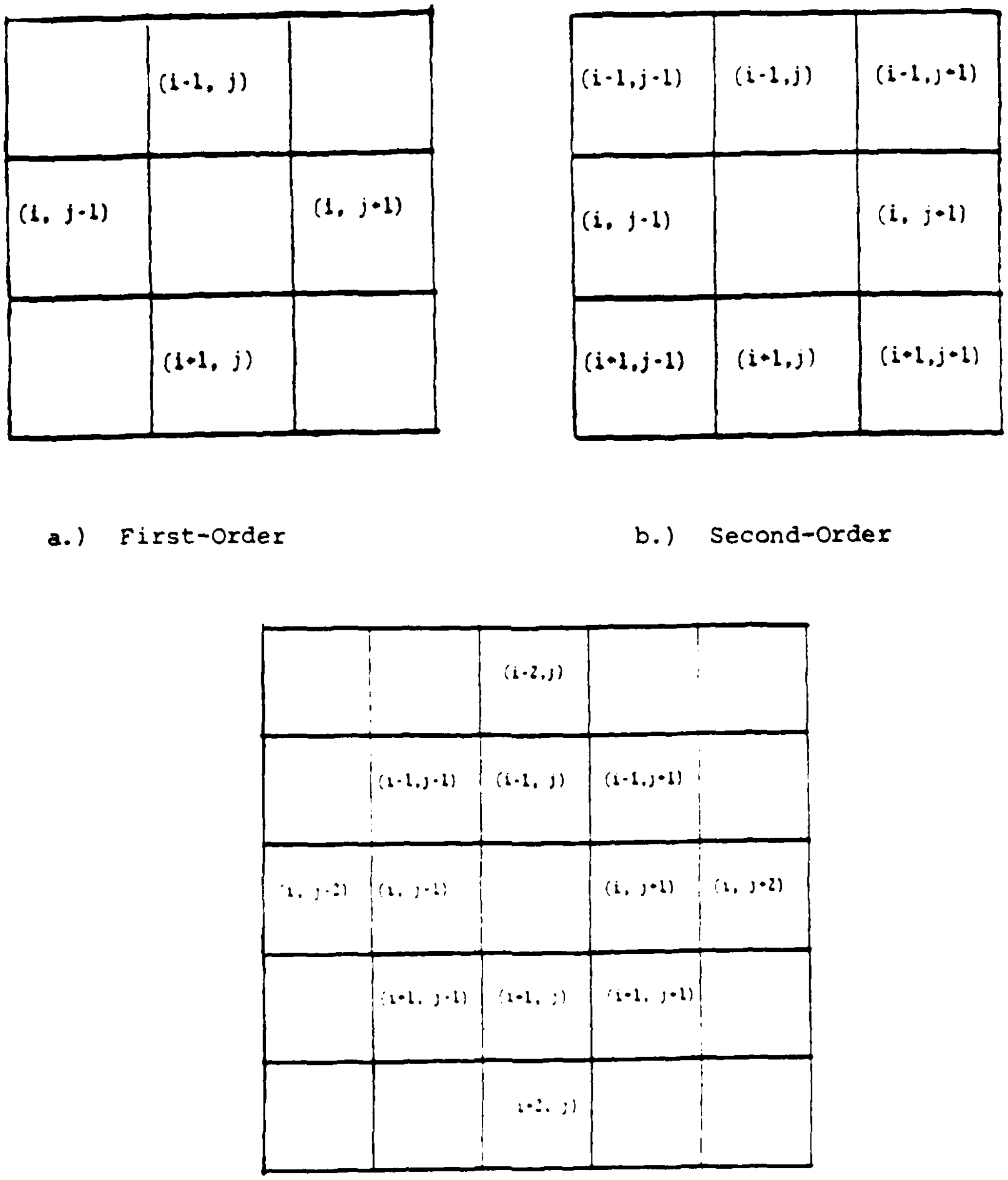
a.) Estimated from training data

	Estimated Parameters		
Texture	a	b_1	b_2
Grass	-1.82	.53	1.31
Ripple	-2.88	-.15	3.05

b.) Estimated by clustering

Estimated First-order MRF Model Parameters for the Discrimination of Natural Textured Image.

Figure 4.3-1



	(i-1, j)	
(i, j-1)		(i, j+1)
	(i+1, j)	

a.) First-Order

(i-1,j-1)	(i-1,j)	(i-1,j+1)
(i, j-1)		(i, j+1)
(i+1,j-1)	(i+1,j)	(i+1,j+1)

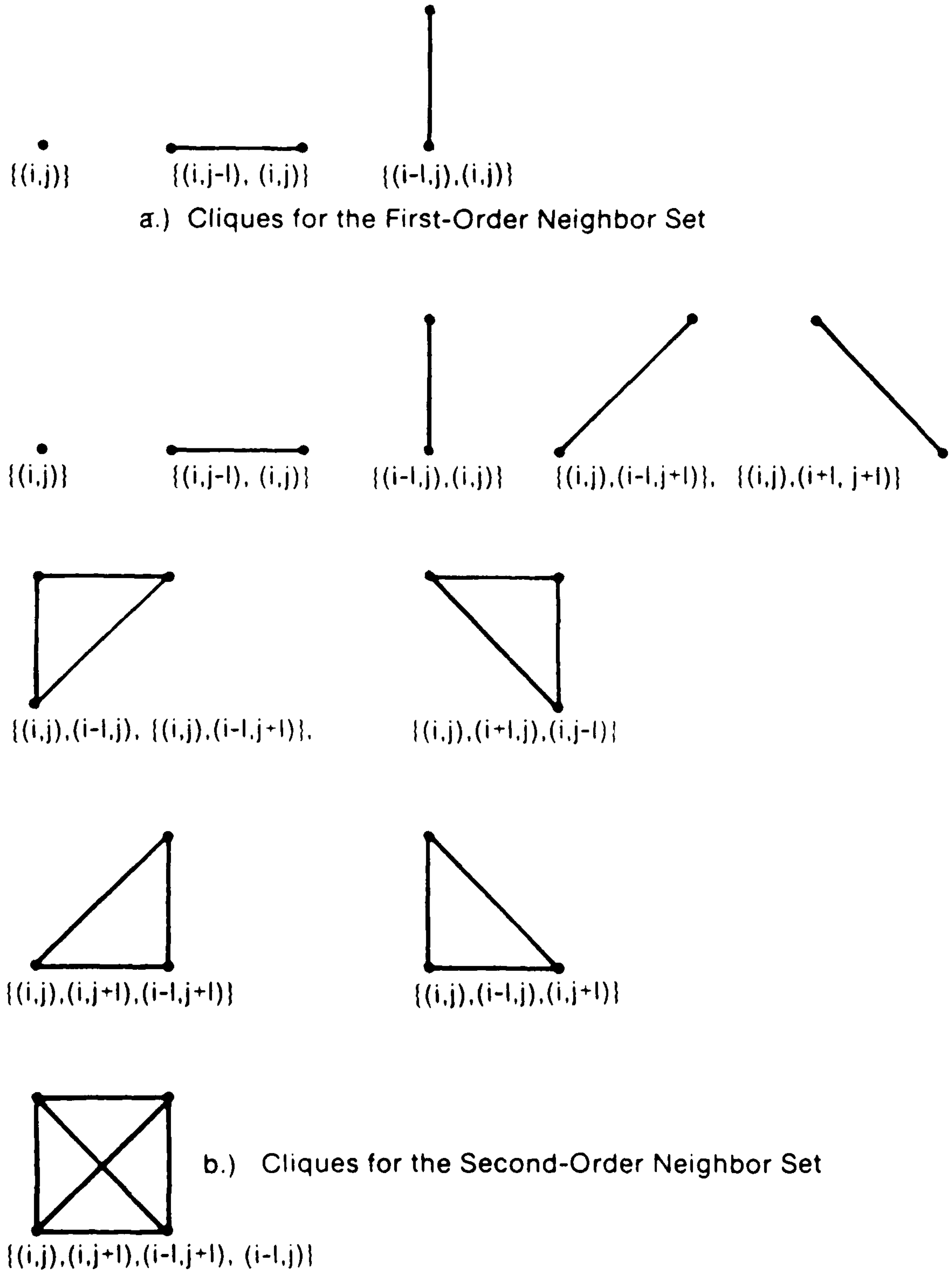
b.) Second-Order

		(i-2,j)		
	(i-1,j-1)	(i-1, j)	(i-1,j+1)	
(i, j-2)	(i, j-1)		(i, j+1)	(i, j+2)
	(i+1, j-1)	(i+1, j)	(i+1, j+1)	
		(i+2, j)		

c.) Third-Order

Examples of Neighborhood Systems for MRF's.

Figure 4.3-2



a.) Cliques for the First-Order Neighbor Set

b.) Cliques for the Second-Order Neighbor Set

Examples of Cliques for First-Order and Second-Order Neighborhood Systems

Figure 4.3-3

1	2	1	2	1	2
2	1	2	1	2	1
1	2	1	2	1	2
2	1	2	1	2	1
1	2	1	2	1	2
2	1	2	1	2	1

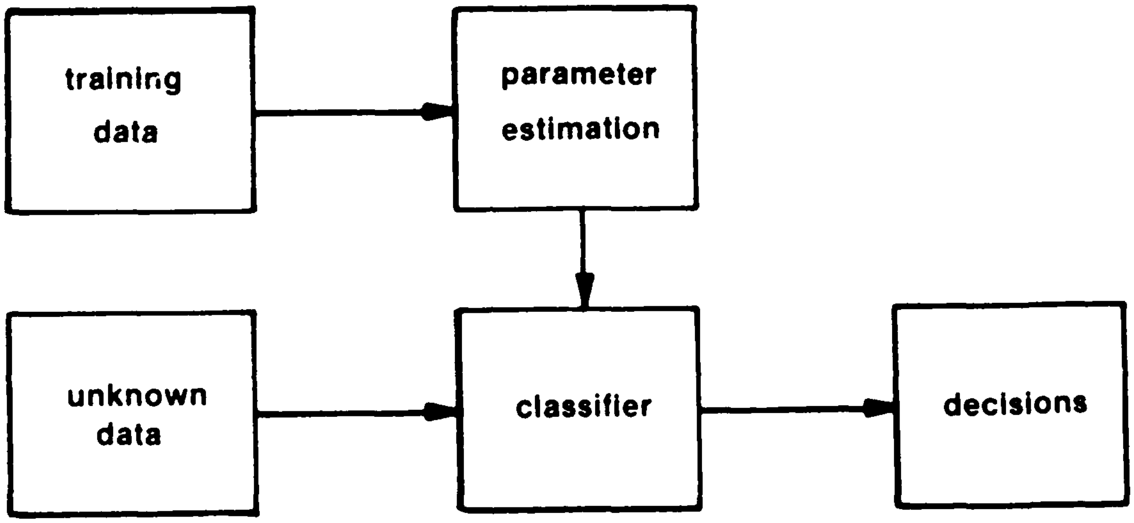
a.) Codings for the First-Order Neighbor Set

1	2	1	2	1	2
3	4	3	4	3	4
1	2	1	2	1	2
3	4	3	4	3	4
1	2	1	2	1	2
3	4	3	4	3	4

b.) Codings for the Second-Order Neighbor Set

Examples of Different Codings.

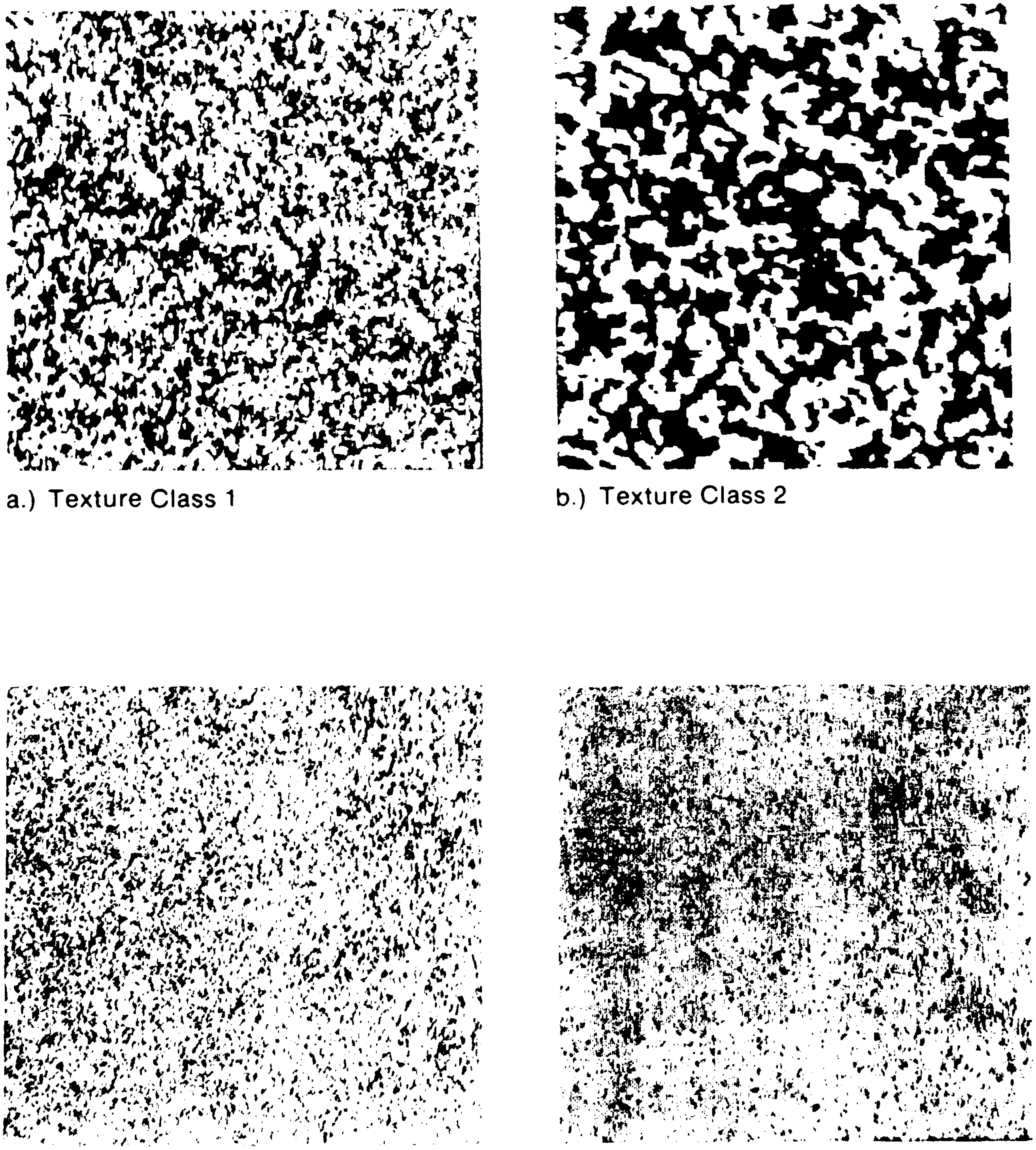
Figure 4.3-4



A Texture Classification System

Figure 4.3-5

Synthetic Textures Modeled as First-Order BMRF's



a.) Texture Class 1

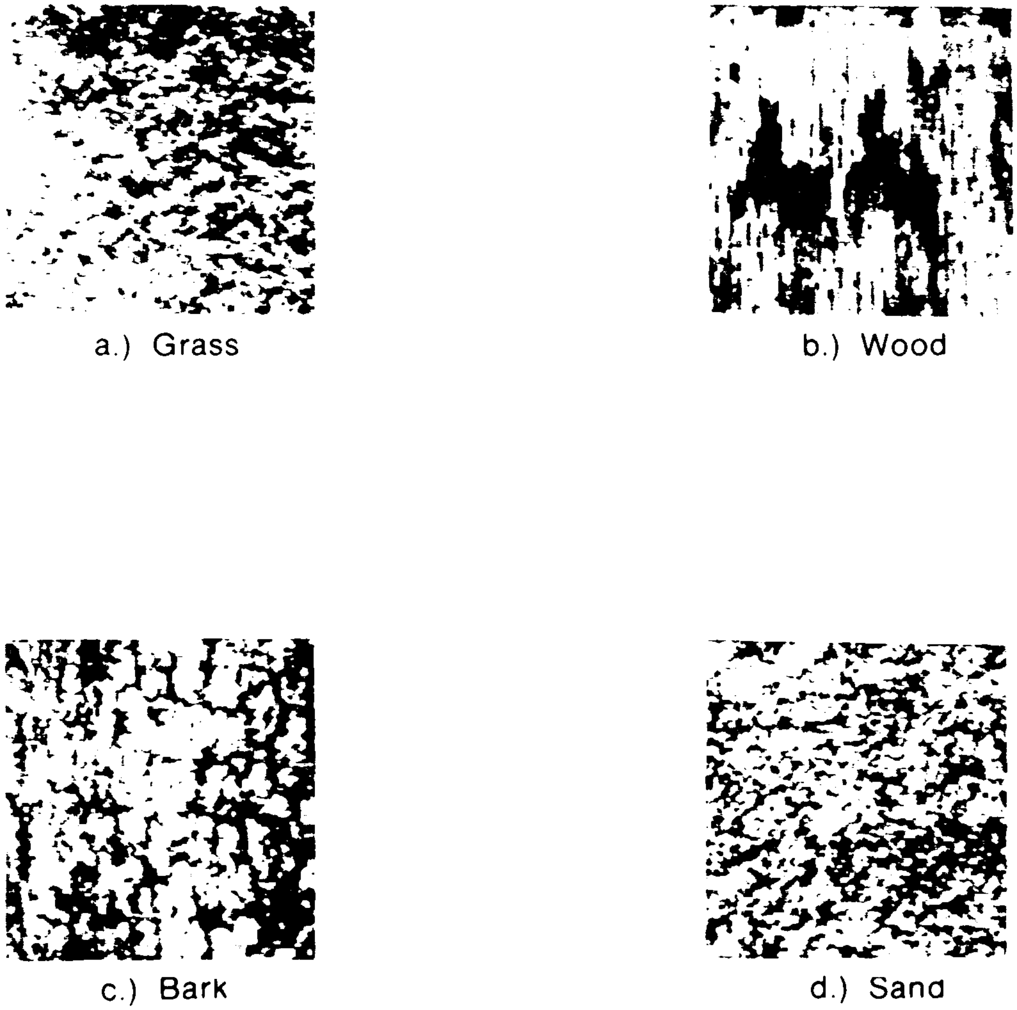
b.) Texture Class 2

c.) Texture Class 3

d.) Texture Class 4

Figure 4.3-6

Binary Quantized Samples of Natural Textures.



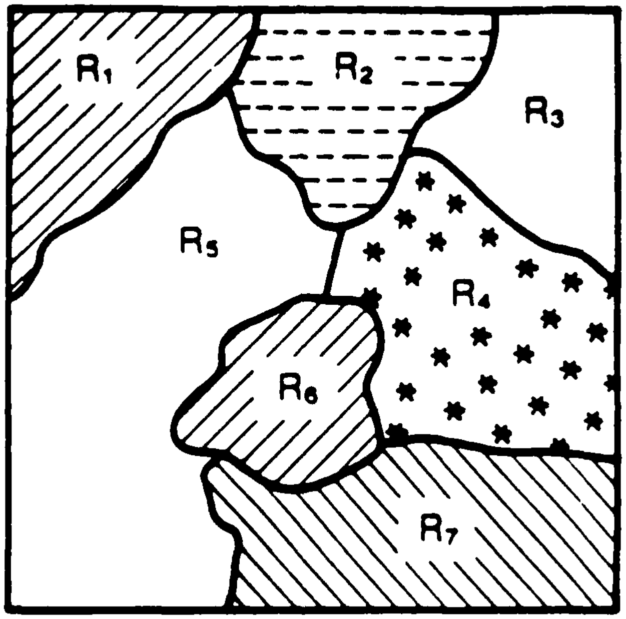
a.) Grass

b.) Wood

c.) Bark

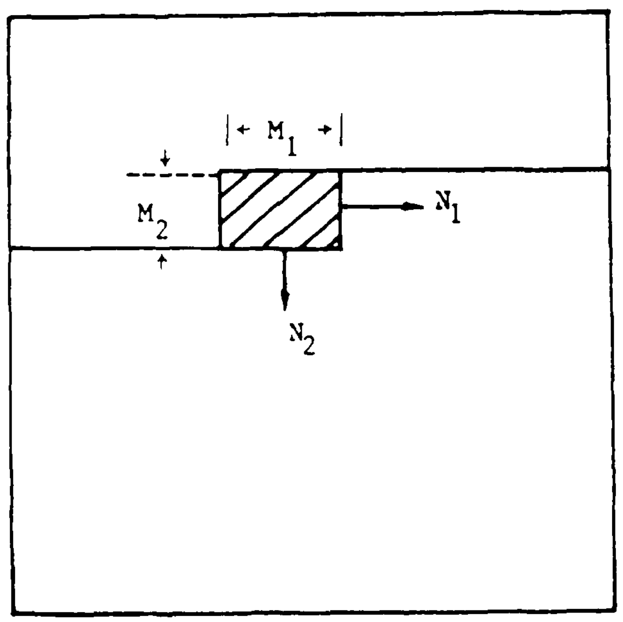
d.) Sana

Figure 4.3-7



An Image Containing Multiple Texture Regions

Figure 4.3-8



A Sliding Window on the Image Plane .

Figure 4.3-9

A Region Map for Texture Discrimination Experiments.

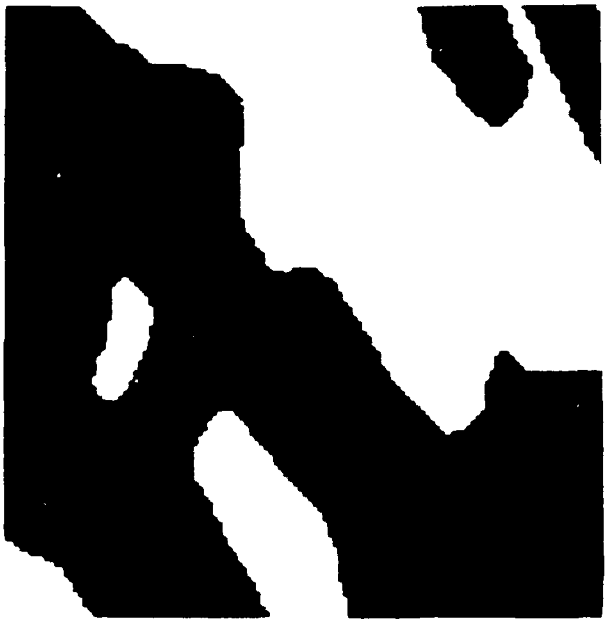
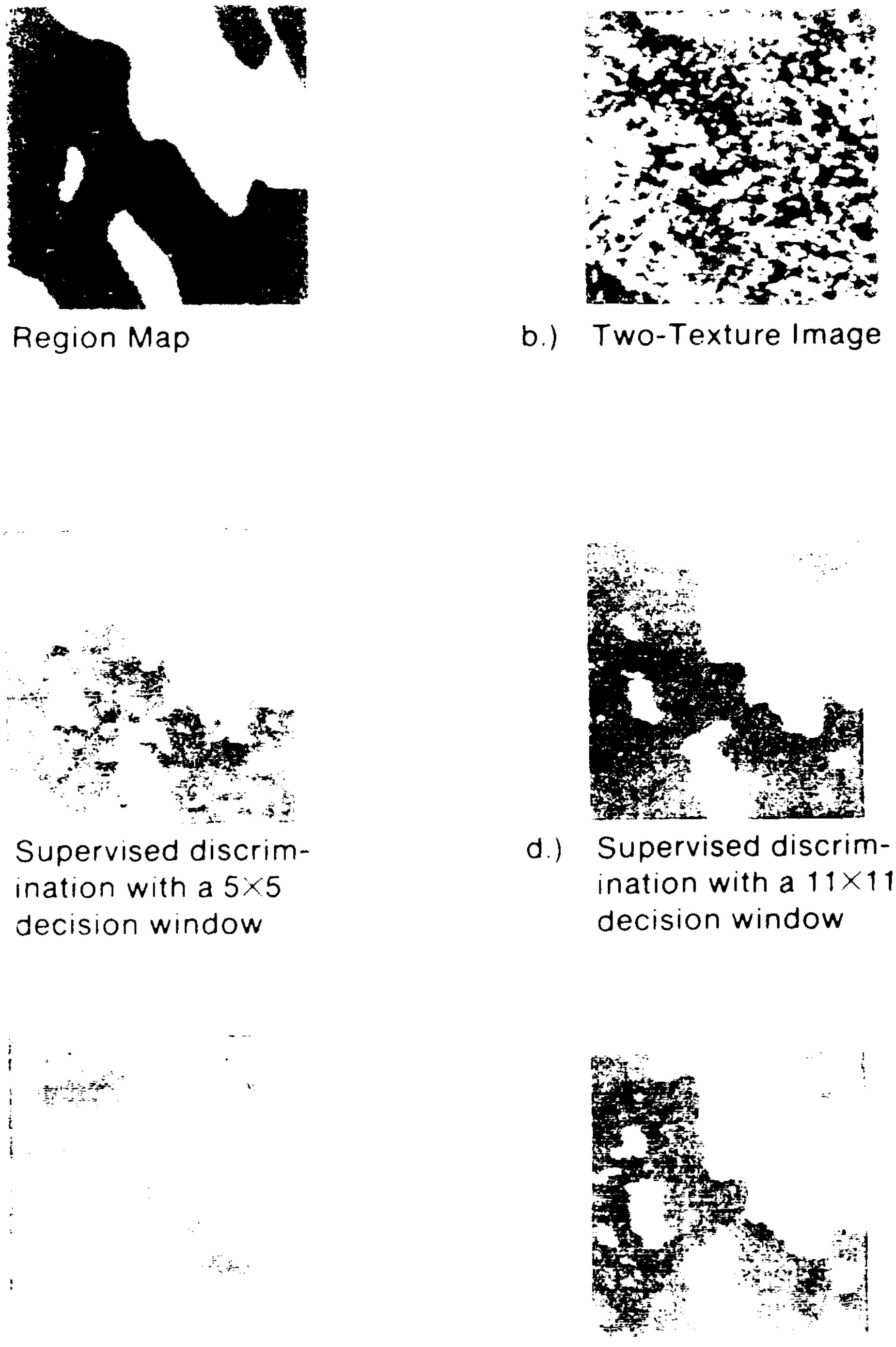


Figure 4.3-10

An Example of Synthetic Texture Discrimination.



a.) Region Map

b.) Two-Texture Image

c.) Supervised discrimination with a 5×5 decision window

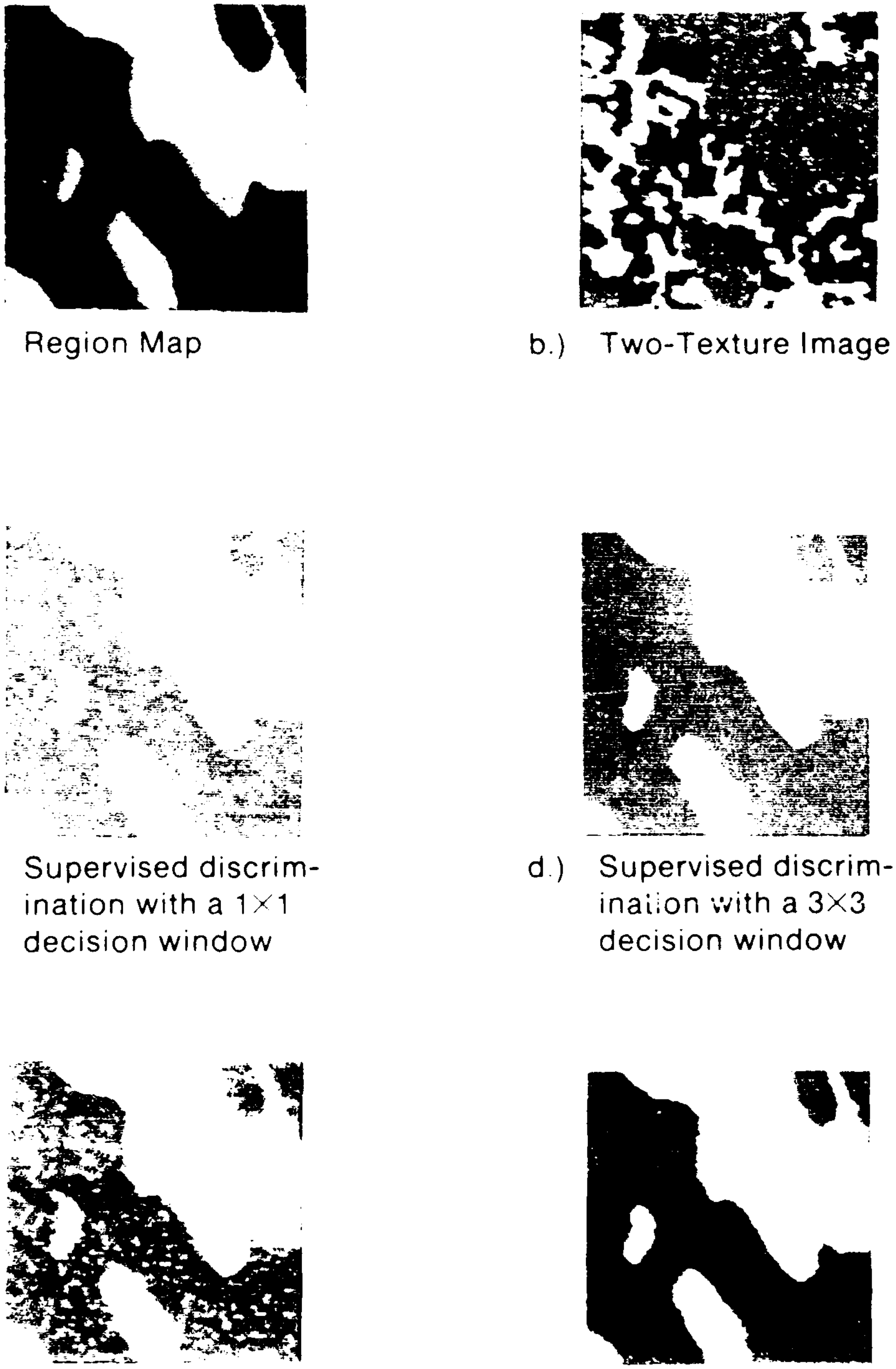
d.) Supervised discrimination with a 11×11 decision window

e.) Unsupervised discrimination with a 5 · 5 decision window

f.) Unsupervised discrimination with a 11 · 11 decision window

Figure 4.3-11

An Example of Natural Texture Discrimination.



a.) Region Map

b.) Two-Texture Image

c.) Supervised discrimination with a 1×1 decision window

d.) Supervised discrimination with a 3×3 decision window

e.) Unsupervised discrimination with a 1×1 decision window

f.) Unsupervised discrimination with a 3×3 decision window

4.4 Cluster Validation With Application to Image Segmentation:

Clustering procedures have found wide application in statistical data analysis and processing. The application of specific interest here is stochastic model-based image segmentation where a clustering algorithm is used to estimate the model parameters for the various image classes in an observed image. In this, and similar applications, it's generally the case that the clustering algorithm requires prior knowledge of the number of clusters or data classes. For many applications, however, the number of clusters is not known a priori and we would like to determine it directly from the data. This is known as the cluster validation problem. For stochastic model-based image segmentation, the solution of this problem directly affects the quality of the segmentation. In this work we propose a model fitting approach to the cluster validation problem based upon Akaike's Information Criterion (AIC). The explicit evaluation of the AIC for the image segmentation problem is achieved through an approximate maximum-likelihood (ML) estimation algorithm. We demonstrate the efficacy of the proposed approach through experimental results for both synthetic mixture data, where the number of clusters is known, and to stochastic model-based image segmentation operating on real-world images, for which the number of clusters is unknown. This approach is shown to correctly identify the known number of clusters in the synthetically generated data and to result in good subjective segmentations in aerial photographs.

4.4.1 Background:

Clustering procedures are widely used in various applications of pattern classification and statistical data analysis. In a clustering procedure, the observed data or entities are grouped together to form a number of clusters in such a way that the entities within a cluster are more similar to each other than to those in other clusters. The measure of similarity, usually heuristically defined, is called the cluster criterion.

For the past three decades, many clustering algorithms have been developed by researchers in such diverse fields as biology, statistical data analysis and pattern recognition, using very different cluster criteria [1]. In some previous work [2]-[4] on stochastic model-based image segmentation, clustering algorithms have been used to estimate the model parameter vectors for different image classes directly from the observed image. Since the nature of this work is related to statistical pattern recognition, the clustering algorithm used was selected from those developed within the pattern recognition community. One of the most successful clustering algorithms in this respect is the K-means algorithm [5],[6]. This algorithm is optimum in the sense that it minimizes the variance within each cluster and has been widely used in unsupervised pattern recognition. However, an important problem existing with most clustering algorithms, including the K-means algorithm, is that the *number* of clusters in the data must be specified a priori before using the clustering algorithm.

In some situations this number can be derived from prior knowledge about the data, or sometimes can even be determined from visual inspection of the two- dimensional projection of the data. However, in many applications, such as our previous work on image segmentation, it is desired to estimate this number directly from the observed data since a priori knowledge is generally not available and the data are often vectors of dimension higher than two such that the projection method is not satisfactory. Furthermore, even when the data is two dimensional, visual inspection may not be successful if the data clusters cannot be decided by observation. This problem is of great practical importance for many clustering algorithms and is known as the cluster validation problem [7]. For stochastic model-based image segmentation, such as the schemes described in [2]-[4], the solution of this problem directly affects the quality of the resulting segmentation. If the estimated number of clusters, or data classes, is smaller than the true value, the objects in the image will not be well separated. Likewise, if this estimated number is too large, a single object may be separated into a number of smaller regions. Both of these situations are to be avoided.

Most of the previously proposed solutions to the cluster validation problem can be classified into two approaches: a heuristic approach and a statistical hypothesis testing approach. In the heuristic approach, the number of clusters are determined by using some ad hoc criteria. For example, for the K-means algorithm it has been proposed to look at the plot of the average of the variances within the clusters under assumptions of different K, the number of clusters. The value of K corresponding to the point where the curve begins to saturate can then be taken as the estimated number of classes. Many ad hoc variations of the K-means algorithms have been proposed based on similar ideas. In these algorithms, the number K is increased or decreased according to criteria such as intra-cluster variance and distance between clusters. While some practical problems can be solved using the heuristic approach, it does not provide a general solution to the cluster validation problem and, even when applied to specific problems, the criteria have to be fine-tuned through trial-and-error. This, in part, reflects the difficult nature of the problem. More specifically, as pointed out by Everitt [1] and Jain [7], clusters are generally very difficult to define precisely.

To find generally applicable and mathematically rigorous solutions to cluster validation, many researchers have tried to formulate the problem as a statistical hypothesis testing problem [8],[9]. For example, hypothesis tests have been proposed to test whether a given cluster should be divided into two. More general likelihood tests have been attempted with the data modeled in terms of finite mixture distributions [9]. However, due to the structure of the mixture distribution, the parameters, which characterize one hypothesis (for example, the null hypothesis) are at the boundary of the parameter space of the other hypothesis. This, in turn, violates the regularity conditions (cf. [9]) which are required for the validity of the asymptotic distribution theory for the generalized likelihood ratio (GLR) test which exists for many simple hypothesis testing situations where each of the hypotheses can be described in terms of a single probability distribution. As a result, no GLR test is available at this point to determine the number of clusters directly from observation data.

On the other hand, 'he problem we face is not unlike the one faced in developing a theory to fit an autoregressive (AR) model to real-world data in which the order of the model has to be decided before the model parameters can be estimated from the data. Having observed that neither heuristic nor hypothesis testing approaches alone would provide a satisfactory solution to determining the order of the model, hence the practical fitting of a model to observation data, Akaike [10] suggested that the problem should be viewed as a *multiple decision* problem. That is, rather than asking which hypothesis is acting (which order is correct), we should ask which model best fits the data. The goodness of fit, as pointed out later by Akaike [11], should be a properly defined entropy function and the best fit should be obtained by maximizing this quantity. Based on this maximum-entropy principle, Akaike proposed a criterion, called the AIC (Akaike's information criterion), to determine both the order and the parameters of an AR model for observed data. Although there have been some criticisms of the AIC as being inconsistent, Akaike showed that the AIC is robust and optimal in a minimax sense. That is, it is optimal when there is no a priori knowledge about the distribution of the model parameters. In addition, Akaike and others also extended the AIC to several Bayesian variations called the BIC (Bayesian Information Criterion)[13],[14]. This class of criteria can be shown to be AIC's averaged with respect to various a priori distributions for the model parameters. Although the AIC criterion and its variations have achieved substantial success, mostly in AR model fitting, their application is, of course, not limited to AR time series modeling.

In this work, we have applied the AIC to the problem of cluster validation. The solution is then used to find the number of distinct image classes in an observed image. There has been little previous work on the application of the AIC to cluster validation. Sclove [17] demonstrated a way to use the AIC to verify image segmentation results. After segmenting a synthetic image under the assumption of two and three classes, the AIC was used to verify that the segmentation with three classes is a better segmentation. Our results differ from Sclove's work in that we apply the AIC explicitly to the cluster validation problem and, in the application to image segmentation, we use the AIC to decide the proper number of classes in an image before segmentation. The explicit evaluation of the AIC is obtained by an approximate maximum- likelihood (ML) estimation algorithm to be described. We demonstrate the effective application of this procedure to both synthetic data, where the true number of classes is known, and to real-world aerial photographs, in which case the number of classes is unknown and can be assessed only subjectively.

In the next section, we will formulate the cluster validation problem as a mixture model-fitting problem and describe how to determine the number of clusters by using the AIC. Then, in Section 4.4.3, we will show some experimental results in which the number of clusters is determined from synthetic data or real image data using the AIC criterion. We will also show real-world image segmentation results obtained with the number of classes determined by the AIC. Finally, a summary and conclusions are provided in Section 4.4.4.

4.4.2 The Model Fitting Approach:

In this work, we determine the number of clusters from the observed data by finding the best-fitting random mixture model for the data using the AIC criterion. Assume that the sample data is represented by N independent and identically distributed (i.i.d.) m-dimensional vectors, $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N\}$. Furthermore, assume that a mixture distribution can be used to model the probability distribution of $\mathbf{y} \epsilon \mathbf{Y}$. That is,

$$p(\mathbf{y}) = \sum_{k=1}^{K} \pi_k p_k(\mathbf{y});\ \mathbf{y} \epsilon \mathbf{Y}, \tag{1}$$

where the $p_k(\mathbf{y})$'s are individual m-dimensional component pdf's with π_k's as the weights such that

$$\pi_k > 0, \text{for } k = 1, 2, \ldots, K \tag{2a}$$

and

$$\sum_{k=1}^{K} \pi_k = 1. \tag{2b}$$

The number K is the number of mixture components and is used as an indicator of the number of clusters. That is, we consider each cluster in the data as a component of the mixture distribution with K the number of clusters.

A special case of the mixture distribution is the *Gaussian* mixture where the individual pdf's, $p_k(\mathbf{y}), k = 1, 2, \ldots, K$, are all Gaussian [9]. For example, suppose $K = 2$ and $m = 2$ and suppose the components of the individual sample vectors are independent. In this case the Gaussian mixture is completely defined by the parameter vector[1] $\mathbf{a} = (\mathbf{m}_1, \mathbf{m}_2, \sigma_1^2, \sigma_2^2, \pi_1)$ where $\mathbf{m}_k$ and σ_k^2 are each of dimension $m = 2$ representing, respectively, the mean-value vector $\mathbf{m}_k = (m_{k1}, m_{k2})$ and variance vector $\sigma_k^2 = (\sigma_{k1}^2, \sigma_{k2}^2)$ associated with $p_k(\mathbf{y}), k = 1, 2$. The parameter vector, $\mathbf{a}$, is of dimension $K' = 9$ in this case.

[1] We will make use of this notation in describing some experimental results in the next section.

Under the mixture distribution modeling assumption, the problem of determining the number of clusters for the observation $\mathbf{Y}$ becomes that of finding the best-fitting mixture model for $\mathbf{Y}$. The resulting K in that model would then be taken as a good estimate for the number of clusters. According to Akaike, the best fit should be the one that maximizes a generalized entropy or minimizes the AIC criterion defined as

$$AIC(K) = -2log[\text{maximum-likelihood of the model}(K)] + 2K', \tag{3a}$$

where

$$\text{maximum-likelihood of the model}(K) = p_K(\mathbf{Y} \mid \hat{\mathbf{a}}_{ML}^{(K)}), \tag{3b}$$

Here, $\hat{\mathbf{a}}_{ML}^{(K)}$ is the maximum-likelihood (ML) estimate of the model parameter vector, $\mathbf{a}$, of the mixture model given K, the number of components, and K' is the number of *independent* parameters of the K-component mixture model. In the case of the Gaussian mixture model, the vector $\hat{\mathbf{a}}_{ML}^{(K)}$ consists of ML estimates for the parameters of the Gaussian component pdf's and the first K-1 weights, $\pi_1, \pi_2, ..., \pi_{K-1}$. Now, for a given set of sample data vectors, $\mathbf{Y}$, the optimal estimate of the number of clusters is

$$K_0 = \arg \min_{1 \leq K \leq K_{max}} AIC(K), \tag{4}$$

where K_{max} is a prespecified upper limit for K. Rigorous justification of the AIC for model fitting can be found in a series of papers by Akaike [10]-[14]. This method can be easily implemented provided we can find the ML estimate of the mixture model parameters which is known in statistical data analysis as the mixture estimation problem [9].

The ML estimation approach has been a very successful method in stochastic model parameter estimation for the pdf's which contain only one component. Explicit solution can often be found by solving the likelihood equation and the ML estimate in many cases is consistent [18]. Even in the case where the true distribution of the data is not the same as the model, the consistency property often still holds under mild regularity conditions [19]. This result is especially important since, when we try to use a model to approximate an unknown probability distribution using ML estimation, we hope tht the estimates are consistent. Unfortunately, these results do not readily extend to the mixture distributions [9]. First of all, explicit solution is impossible even for the two-component case. Secondly, the likelihood surface often has singularity points which makes numerical solution difficult. A major reason for this is that the data is incomplete in the sense that we do not know a priori to which cluster a data vector belongs. However, a number of approximate ML algorithms do exist. One of the more popular methods is the so-called EM (expected maximum) algorithm [9]. It has been shown that under mild regularity conditions it does provide local maxima that are consistent. However, a disadvantage of the EM algorithm is its relatively slow convergence.

In this work we use an approximate ML estimation scheme using a clustering algorithm. First of all, the K-means clustering algorithm is applied to the data to divide the data into K groups. Then each group is assumed to correspond to the sample data for one and only one mixture component. A ML estimate is then evaluated on each group separately to estimate the parameters for the corresponding mixture component. Finally, a component weight i.e., the π_k's can be estimated as the ratio of the number of samples in a group tc the total number of samples. This approximation transforms the problem of ML estimation of a mixture to that of ML estimation of several individual p.d.f.'s. It will be shown in the next section, through experimental results, that it provides reasonably good estimates. This scheme also converges fast since the clustering algorithm is known to possess fast convergence properties.

The Gaussian mixture model is the most studied mixture model because it is a realistic model for many applications and it is mathematically tractable. In this work, we make explicit use of the Gaussian mixture model. To further simplify the mathematics, we assume the components of the individual observation vectors are independent.[2] Under these assumptions, the procedure of determining the number of clusters in a set of observed data can be stated as follows:

1.) For a given $K = 1, 2, ..., K_{max}$, apply the K-means clustering algorithm with the number of classes preset to K.

[2] The component p.d.f.'s are then completely described by their mean value vectors $\mathbf{m}_k = (m_{k1}, m_{k2}, ..., m_{km})$ and variance vectors $\sigma_k^2 = (\sigma_{k1}^2, \sigma_{k2}^2, ..., \sigma_{km}^2), k = 1, 2, ..., K$.

2.) Estimate the mean and variance vectors for each cluster and the weight of each cluster.

3.) Compute $AIC(K)$.

4.) Select K_0 as the estimate of the number of clusters in the data if it minimizes $AIC(K)$ for all $K = 1, 2, ..., K_{max}$.

There are two applicable expressions for the likelihood functional in the use of the AIC criterion. If we consider the data vectors to be *incomplete*, that is, the class status of the samples is unknown, we will have the standard likelihood expression for the mixture which, from (1), becomes

$$p_K(\mathbf{Y}|\mathbf{a}) = \prod_{i=1}^{N} \sum_{k=1}^{K} \pi_k p_k(\mathbf{y}_i). \tag{5}$$

On the other hand, if we first classify the data by applying the K-means algorithm, we in effect assign data vectors to hypotheses classes. In this case a data vector assigned to class k can be considered coming from a particular class and has a probability π_k of occurring. The corresponding expression for the likelihood functional for *correctly* classified samples then becomes

$$p_K(\mathbf{Y}|\mathbf{a}) = \prod_{k=1}^{K} \pi_k^{N_k} \prod_{j=1}^{N_k} p_k(\mathbf{y}_{k_j}), \tag{6}$$

where $N_k \leq N$ is the number of samples in the k^{th} cluster and $\mathbf{y}_{k_j}, j = 1, 2, ..., N_k$ are data vectors associated with this cluster. Since we have used the K-means algorithm for approximate ML estimation, each sample vector is assigned to a unique class. In what follows, we will make use of the second likelihood functional as expressed by (6). The ML estimate, $\hat{\mathbf{a}}_{ML}^{(K)}$, to be used in (3) in computing AIC(K) is then formed from the resulting K class-conditional parameter estimates together with the estimated weights as described above.

The method proposed here is quite general in that we can use assumptions for the single mixture components other than Gaussian. Furthermore, other AIC related criteria, such as BIC's, can be properly adapted to it. Finally, we note that the AIC criterion has a useful intuitive appeal. More specifically, when two or more models are almost equally likely, in the sense they have approximately the same maximum likelihoods, the AIC criterion selects the one with the smaller number of parameters or the least complex.

4.4.3 Experimental Results:

In this section we demonstrate the effectiveness of the model-fitting approach to determining the number of clusters from observed data. First, we perform an experiment on synthetic data where the sample vectors are indeed generated from a Gaussian mixture distribution as described in the last section. Then, we will apply the same approach to the image segmentation problem to identify the number of image classes present in an observed image. The synthetic data set is used to study the ideal performance of this approach, while the image data is used to assess its application to a particular real-world problem. We now present the results for these two cases separately.

A.) Synthetic Data:

In this experiment, three two-dimensional ($m = 2$) Gaussian mixture data sets with two, three and four components, or clusters, are generated as shown in Fig. 4.4-1. We choose the data to be two-dimensional since it's then easy to display on a plane. There are two objectives of this experiment: first, to see if the approximate ML estimates provide a reasonable estimate of the true model parameters and, secondly, to see whether the AIC provides correct estimates of the number of clusters, even in the ideal case. The results of the parameter estimates for all the test data sets under the correct assumptions on the number of clusters are shown in Table 4.4-1. It can be observed that when the assumption of the number of classes acting corresponds to the true but unknown value, the parameter estimates are quite accurate. This indicates that the approximate ML estimation scheme using clustering is quite effective. In Table 4.4-2, we have shown the AIC's computed for all the test data under the assumptions of different number of clusters, with $K_{max} = 8$. We find that the AIC does make correct decisions each time. This indicates that when the data is indeed a Gaussian mixture, the method proposed here tends to estimate the number of clusters correctly. Additional examples are given for a much larger variety of Gaussian mixtures in [19] with similar results.

B.) Application to Image Data:

In this experiment, we attempt to apply the method proposed in the previous section to a stochastic model-based image segmentation procedure developed previously in [2]-[4]. In our work, we take the point of view that image segmentation is the process of assigning the pixels of the image to a finite, and usually small, number of image model classes. In a stochastic model-based approach, different image classes are modeled by random field models. For simplicity, we consider the Gaussian model used in [2] for tonal properties of the image classes. That is, each image class is modeled by an i.i.d. Gaussian random field. Then, each image class can be characterized in terms of a model parameter vector consisting of only two components: the mean and variance.

In the segmentation process, the pixels are assigned to model classes through a likelihood test based on the Gaussian model. In particular, a likelihood test for each pixel is performed on the data contained in a *decision* window of a fixed size centered at that pixel position. Before the image can be segmented, however, the model vectors corresponding to different image classes have to be estimated from the image. It was suggested in [2] that this can be realized by a clustering approach on the sample model vectors estimated from a sliding *estimation* window on different spatial locations in the image and the resulting cluster centers can then be taken as the model vectors for the image classes. The clustering algorithm used was the K-means algorithm in which the number of image classes, or clusters, needs to be specified beforehand. The method proposed here provides an *objective* way to determine the number of image classes.

In Fig.'s 4.4-2a and 4.4-3a, we show two aerial photographs. The first contains a building, roads and vegetation while the second contains an oil tank complex surrounded by vegetation. The computed AIC's for different numbers of clusters are shown in Table 4.4-3 with $K_{max} = 10$. The sliding estimation window is of size 16 x 16 pixels. The results suggest that in the first image there are four tonal classes while for the second image five tonal classes best fits the data. The images are segmented using the corresponding model vectors estimated according to that suggested by the AIC criterion and are shown in Fig.'s 4.4-2 and 4.4-3, along with the original images. In these segmentations different tonal areas are well separated. For comparison purpose we have also shown the results of the segmentation using from two up to six classes. It can be seen from the results for both images that, when the assumed number of classes is smaller than that determined by the AIC, a number of significant regions of reasonably large size are missing from the segmentation. On the other hand, when the number of classes is larger than that suggested by the AIC, no significant change in segmentation will result from the increase of the number of classes except the appearance of some noisy regions with small size. This suggests that the AIC model-fitting approach is a reasonable objective approach for practical applications such as image segmentation.

4.4.4 Summary:

In this paper we described a model-fitting approach for determining the number of clusters in observed random data and its applications to stochastic model- based image segmentation. The problem, also known as cluster validation, is solved by finding a best-fitting mixture distribution model to the observed data. The goodness of fit is determined by the AIC criterion. An approximate ML parameter estimation scheme using clustering is proposed to compute the AIC. Experimental results are also described to demonstrate the ideal performance and practical applicability of this method. In the experiments, the AIC correctly determines the number of clusters in the synthetic mixture data and provides a subjectively reasonable number of classes for a number of real-world images. These results indicate that the proposed approach is quite general and effective.

This work also brings up several interesting issues which need further investigation. First of all, it would be of interest to apply the BIC criteria to cluster validation and compare the results with that of the AIC. To do so we need to decide on what parameter set the averaging of the likelihood is to be performed and how to implement the numerical integration involved in the averaging. It would also be of interest to use the EM algorithm as the estimation method for computing the AIC and compare the results on Gaussian mixture model-fitting with those described in this paper. Finally, work is underway in applying the model-fitting approach to image segmentation where the image classes are modeled as autoregressive random fields [3]. This work will be reported on at some later time.

References

[1] B. S. Everitt, *Cluster Analysis*, published by Heinemann Education Books (for) Social Science Research Council, London.

[2] J. Zhang and J. W. Modestino, "Image Segmentation Using a Gaussian Model", unpublished RPI report, March 1987.

[3] J. Zhang and J. W. Modestino, " Unsupervised AR Random Field Model-Based Image Segmentation", unpublished RPI report, March 1987.

[4] J. Zhang and J. W. Modestino, "Texture Classification and Discrimination Using the Markov Random Field Model", submitted to Pattern Anal. and Mach. Intell., Sept. 1987.

[5] K. Fukunaka, *Introduction to Statistical Pattern Recognition*, Academic Press, New York, 1972.

[6] J. T. Tou and R. C. Gonzalez, *Pattern Recognition Principles*, Addison-Wesley Publishing Company. Reading, MA, 1974.

[7] A. K. Jain,"Cluster Analysis", *Handbook of Pattern Recognition and Image Processing*, Ed. by T.Y. Young and K.S. Fu, Academic Press, New York, 1986.

[8] R. Dubes and A. K. Jain, "Validity Studies in Clustering Methodologies", *Pattern Recognition*, Vol. 11, pp. 235-254, 1979.

[9] D. M. Titterington, A.F.M. Smith and U.E. Makov, *Statistical Analysis of Finite Mixture Distributions*, John Wiley & Sons, New York, 1985.

[10] H. Akaike, "A New Look at the Statistical Model Identification", IEEE Trans. Automatic Control, AC-19, pp 716-722, Dec. 1974.

[11] H. Akaike, "An Entropy Maximization Principle", Proc. Symp. on *Applied Statistics*, Ed. P. R. Krishnaiah, pp. 27-41, North Holland, Amsterdam, 1977.

[12] H. Akaike, "A Bayesian Extension of the Minimum AIC Procedure", Biometrika, Vol. 66, pp. 237-242, 1979.

[13] H. Akaike, "A Bayesian Analysis of the Minimum AIC Procedure", Ann. Inst. Statis. Math., Vol. 30A, pp. 9-14, 1979.

14 H. Akaike, "Canonical Correlation Analysis of Time Series and the Use of an Information Criterion", in *System Identification Advances and Case Studies*, Ed. by R. K. Mehra and D. G. Lainiotis, Academic Press, New York, 1976.

15 G. Schwarz, "Estimating the Dimension of a Model", The Ann. of Statistics, Vol. 6, No. 2, pp. 461-464, 1978.

16 R. L. Kashyap, R. Chellappa and N. Ahuja, "Decision Rules for Choice of Neighbors in Random Field Models of Images",*Computer Graphics and Image Processing*, Vol. 15, pp. 301-318, 1981.

17 S. L. Sclove, "Application of the Conditional Population-Mixture Model to Image Segmentation", IEEE Trans. Pattern Anal. Machine Intel., Vol. PAMI-5, pp. 428-433, July 1983.

18 H. L. Van Trees, *Detection, Estimation, and Modulation Theory: Part I*, John Wiley & Sons, New York, 1968.

19 P. J. Huber, "The Behavior of Maximum Likelihood Estimates Under Non-Standard Conditions", Proc. 5th Berkeley Symp. on Math. Stat. and Probability, Vol. 1, pp. 221-233, 1967.

20 J. Zhang, "Statistical Model-Based Image Analysis", Ph.D. Thesis, Rensselaer Polytechnic Institute, Troy, New York, in preparation.

Table 4.4-1

Parameter Estimates for the Synthetic
Gaussian Mixture Data: No. of samples=500

Parameters	Values	
	Real	Estimated
$\underline{m}_1$	(4.00,4.00)	(3.99,4.03)
$\underline{m}_2$	(9.00,9.00)	(8.89,9.12)
$\underline{\sigma}_1^2$	(1.00,1.00)	(1.04,1.06)
$\underline{\sigma}_2^2$	(1.00,1.00)	(1.01,1.02)
$\underline{\pi}$	(0.50,0.50)	(0.49,0.51)

a.) *Two-Component Gaussian Mixture*

Parameters	Values	
	Real	Estimated
$\underline{m}_1$	(4.00,4.09)	(3.88,3.99)
$\underline{m}_2$	(9.00,9.00)	(8.89,9.11)
$\underline{m}_3$	(9.00,4.00)	(9.02,4.06)
$\underline{\sigma}_1^2$	(1.00,1.00)	(0.01,1.05)
$\underline{\sigma}_2^2$	(1.00,1.00)	(1.00,1.00)
$\underline{\sigma}_3^2$	(1.00,1.00)	(1.03,1.05)
$\underline{\pi}$	(0.33,0.33,0.33)	(0.32,0.34,0.34)

b.) *Three-Component Gaussian Mixture*

Parameters	Values	
	Real	Estimated
$\underline{m}_1$	(4.00,4.00)	(3.81,4.04)
$\underline{m}_2$	(4.00,9.00)	(3.87,9.16)
$\underline{m}_3$	(9.00,9.00)	(8.92,9.09)
$\underline{m}_4$	(9.00,4.00)	(9.03,4.00)
$\underline{\sigma}_1^2$	(1.00,1.00)	(0.99,0.98)
$\underline{\sigma}_2^2$	(1.00,1.00)	(0.99,1.10)
$\underline{\sigma}_3^2$	(1.00,1.00)	(1.00,0.90)
$\underline{\sigma}_4^2$	(1.00,1.00)	(1.03,1.11)
$\underline{\pi}$	(0.25,0.25, 0.25,0.25)	(0.23,0.26, 0.25,0.26)

c.) *Four-Component Gaussian Mixture*

Table 4.4-2

Computed AIC's for the Synthetic Data with $K_{max} = 8$.

K	AIC(K)
1	998
2	388(min)
3	463
4	472
5	490
6	549
7	580
8	557

a) Two-component Gaussian mixture

K	AIC(K)
1	960
2	712
3	586 (min)
4	617
5	681
6	703
7	692
8	709

b) Three-component Gaussian mixture

K	AIC(K)
1	988
2	852
3	805
4	717 (min)
5	753
6	782
7	806
8	803

c) Four-component Gaussian mixture

Table 4.4-3

Computed AIC's for the Real Image Data with $K_{max} = 10$.

K	AIC(K)
1	526
2	534
3	511
4	506(min)
5	515
6	518
7	519
8	512
9	518
10	526

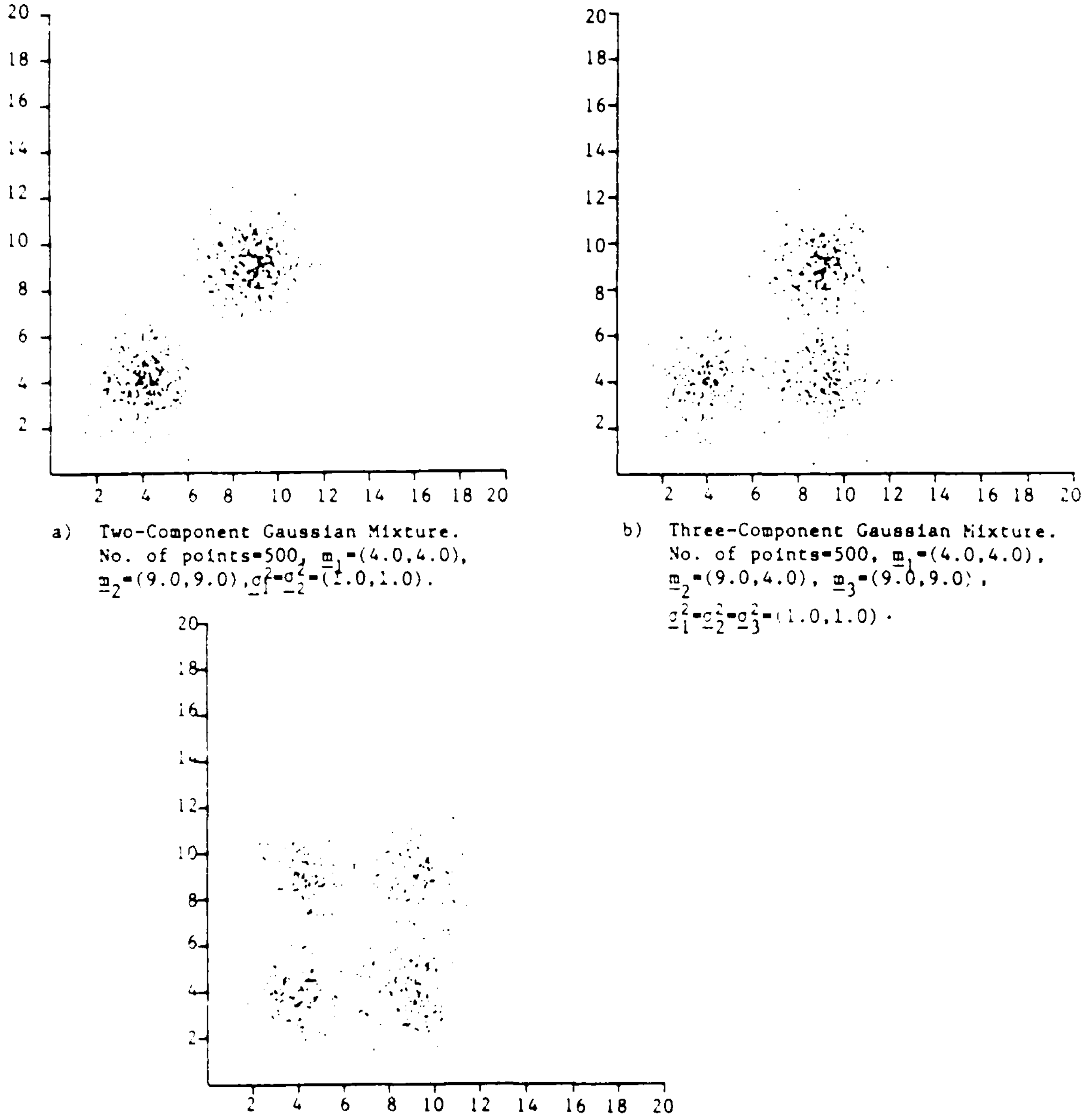
a) Road Scene

K	AIC(K)
1	882
2	768
3	705
4	679
5	664(min)
6	682
7	674
8	677
9	670
10	672

b) Oil Tank Scene

Figure 4.4-1

Examples of Synthetic Gaussian Mixture Data

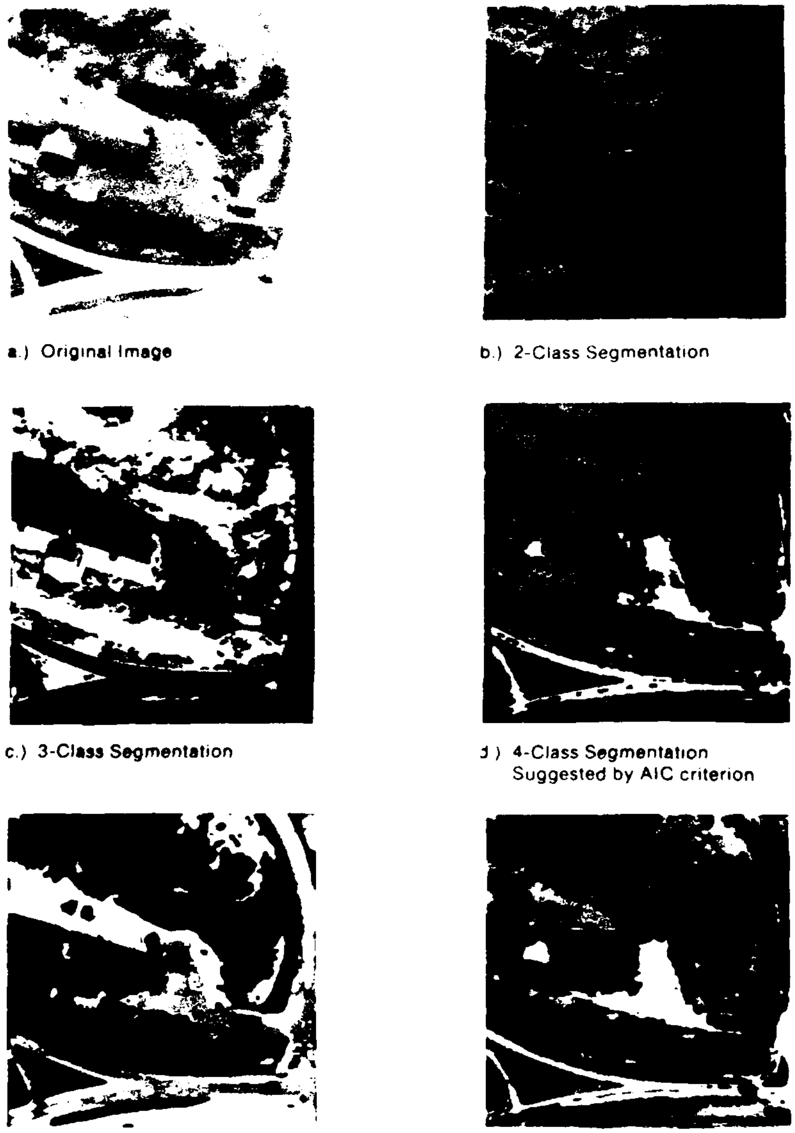


a) Two-Component Gaussian Mixture. No. of points=500, $\underline{m}_1$=(4.0,4.0), $\underline{m}_2$=(9.0,9.0), $\underline{\sigma}_1^2$=$\underline{\sigma}_2^2$=(1.0,1.0).

b) Three-Component Gaussian Mixture. No. of points=500, $\underline{m}_1$=(4.0,4.0), $\underline{m}_2$=(9.0,4.0), $\underline{m}_3$=(9.0,9.0), $\underline{\sigma}_1^2$=$\underline{\sigma}_2^2$=$\underline{\sigma}_3^2$=(1.0,1.0).

c) Four-Component Gaussian Mixture. No. of points=500, $\underline{m}_1$=(4.0,4.0), $\underline{m}_2$=(9.0,4.0), $\underline{m}_3$=(4.0,9.0), $\underline{m}_4$=(9.0,9.0), $\underline{\sigma}_1^2$=$\underline{\sigma}_2^2$=$\underline{\sigma}_3^2$=$\underline{\sigma}_4^2$=(1.0,1.0).

Fig. 4.4-2 Segmented Road Scene According to the AIC criterion.



a.) Original Image

b.) 2-Class Segmentation

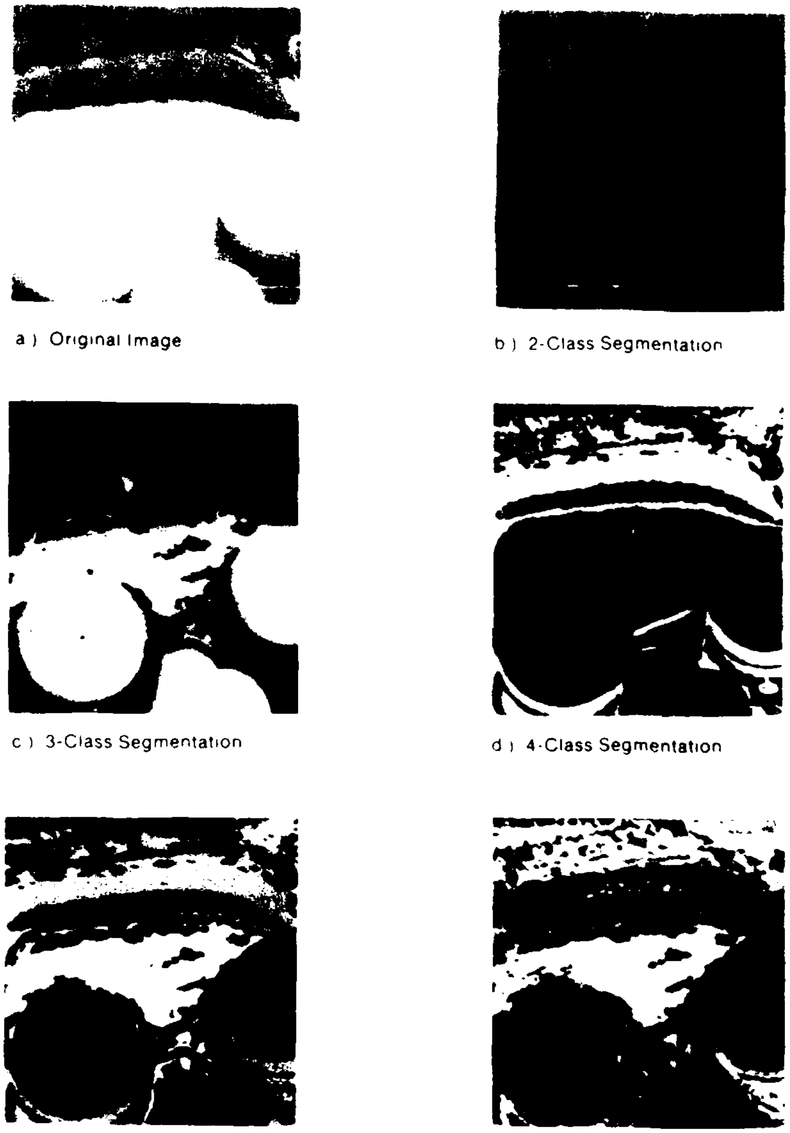
c.) 3-Class Segmentation

d) 4-Class Segmentation
Suggested by AIC criterion

e) 5-Class Segmentation

f) 6 Class Segmentation

Fig. 4.4-3 Segmented Oil Tank Scene According to the AIC criterion.



a) Original Image

b) 2-Class Segmentation

c) 3-Class Segmentation

d) 4-Class Segmentation

e) 5-Class Segmentation Suggested by AIC

f) 6-Class Segmentation

4.5 Overall Summary and Conclusions:

We have been very active over the last year in further refining our concept of a region-based hierarchical approach to image interpretation. A major thrust has been in the development and implementation of improved image segmentation schemes. We expect to continue this effort into FY'88 and, in particular, to concentrate more on improvements in the interpretation process. Issues to be investigated will include:

1. Investigate techniques for fusing information from different image segmentation schemes to provide pertormance improvements over that achievable with any single scheme.
2. Investigate improvements in the information theoretic criteria for unsupervised determination of the number of different image classes present.
3. Develop and investigate techniques for choosing the appropriate model type for stochastic model-based image segmentation schemes.
4. Devise and investigate techniques for incorporating knowledge information into image segmentation schemes. In particular, investigate techniques for incorporating feedback from the interpretation process.
5. Additional, and perhaps more powerful, features have to be incorporated into the image segmentation procedure.
6. Object detection and boundary extraction procedures need to be incorporated into the image segmentation process.
7. More comprehensive region and mutual attributes need to be employed in the image interpretation process.
8. The manual image segmentation procedure needs to be improved and interfaces with knowledge database worked out.
9. Our raw image database needs to be expanded.
10. More general procedures for designing the clique functions need to be worked out.
11. Optimum annealing schedules for effecting the simulated annealing search procedure need to be developed.

12. Propagation of interpretations from one region to the next needs to be investigated.

13. We need to provide feedback from the interpretation process to the segmentation process to improve its performance.

14. We have to investigate how map data and/or archival, previously interpreted, image data can be utilized to improve the photointerpretation process or to implement change detection/interpretation procedures.

MISSION
of
Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control, Communications and Intelligence (C^3I) activities. Technical and engineering support within areas of competence is provided to ESD Program Offices (POs) and other ESD elements to perform effective acquisition of C^3I systems. The areas of technical competence include communications, command and control, battle management, information processing, surveillance sensors, intelligence data collection and handling, solid state sciences, electromagnetics, and propagation, and electronic, maintainability, and compatibility.