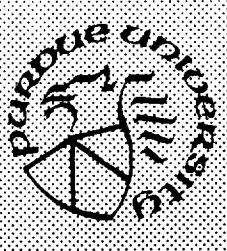


On-Line Object Feature Extraction for Multispectral Scene Representation

Hassan Ghassemian
David Landgrebe

TR-EE 88-34
August 1988

School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure page

Figure page

LIST OF NOTATION

The following notation will be generally followed in this thesis. Any exceptions will be noted at the place of occurrence.

Symbol Explanation

$p_{[i,j]}$ a pixel at the location of the i^{th} row and j^{th} column of the scene image, where i=1, 2, ..., n_x and j=1, 2, ..., n_y

$f(.)$ a functional which maps pixels' spectral attributes into a d-dimensional real space, f: {pixel spectral responses}$\rightarrow \mathbf{R}^d$

X_k a d-tuple vector called pixel-feature, $X_k = f(p_{[i,j]}) = [x_{1k}, x_{2k}, \ldots, x_{dk}]^T$

$\mathbf{P}$ the observation space called pixel-feature-set, $\mathbf{P} = \{X_1, X_2, \ldots, X_{np}\}$

n_p the number of pixels in the scene

v_p the volume of the pixel-feature-set, $v_p = d.n_p$

$\mathbf{P}_i$ a path-segment represents an object in the observation space, $\mathbf{P}_i = \{X_{1i}, X_{2i}, \ldots, X_{ni}\}$, where $\mathbf{P} = \mathbf{P}_1 U \mathbf{P}_2 U \ldots U \mathbf{P}_{no}$

n_i the number of pixels within the object $\mathbf{P}_i$

$\mathbf{L}_i$ the spatial-feature-map of object $\mathbf{P}_i$ in the scene, $\mathbf{L}_i = \{ k ; X_k \in \mathbf{P}_i \}$ where $\mathbf{L}_i \cap \mathbf{L}_j = \emptyset$ for all $i \neq j$

S_i the local spectral feature vector within the object $\mathbf{P}_i$

V_i the local contextual feature vector within the object $\mathbf{P}_i$

$\partial(.)$ a similarity measure, representing the distance between two features

$\partial_g(.)$ a normalized similarity measure, representing the distance between spectral feature normalized by their spectral gradient with respect to their spatial displacement.

$\Re(.)$ a functional denoting the unity relation, for object extraction, which is defined in the observation space by an adjacency relationship together with the similarity criterion.

$\Psi(.)$ a functional denoting the feature extraction process, $\Psi : \mathbf{P} \rightarrow \mathbf{F}$

Y_i an object-feature, $Y_i=\Psi(\mathbf{P}_i)$; e.g. $Y_i=[S_i , V_i , L_i]$

$\mathbf{F}$ the object-feature-set or the feature space, $\mathbf{F}=\{Y_1, Y_2, ..., Y_{no}\}$

$\mathbf{L}$ the spatial-feature-map of the whole scene, $\mathbf{L}=\mathbf{L}_1 U \mathbf{L}_2 U ... U \mathbf{L}_{no}$

$\mathbf{A}_k$ an index set of pixels which have an adjacency relationship with the pixel X_k , where $\mathbf{A}_k \in \mathbf{L}$

n_o the number of objects in the scene

v_f the volume of the object-feature-set, $v_f=m.n_o$

c the data compaction coefficient: $c=v_p / v_f$

ABSTRACT

Ghassemian Yazdi, M. Hassan. Ph.D., Purdue University, August 1988. On-Line Object Feature Extraction for Multispectral Scene Representation. Major Professor: David A. Landgrebe.

This thesis investigates a new on-line unsupervised object-feature extraction method that reduces the complexity and costs associated with the analysis of the multispectral image data and the data transmission, storage, archival and distribution as well. Typically in remote sensing a scene is represented by the spatially disjoint pixel-oriented features. It would appear possible to reduce data redundancy by an on-line unsupervised object-feature extraction process, where combined spatial-spectral object's features, rather than the original pixel-features, are used for multispectral scene representation.

The ambiguity in the object detection process can be reduced if the spatial dependencies, which exist among the adjacent pixels, are intelligently incorporated into the decision making process. We define the unity relation that must exist among the pixels of an object. The unity relation can be constructed with regard to the: adjacency relation, spectral-feature and spatial-feature characteristics in an object; e.g. AMICA (Automatic Multispectral Image Compaction Algorithm) uses the within object pixel-feature gradient vector as a valuable contextual information to construct the object's features, which preserve the class separability information within the data. For on-line object extraction, we introduce the path-hypothesis, and the basic mathematical tools for its realization are introduced in terms of a specific similarity measure and adjacency relation.

AMICA is an example of on-line preprocessing algorithm that uses unsupervised object feature extraction to represent the information in the multispectral image data more efficiently. As the data are read into the system sequentially, the algorithm partitions the observation space into an exhaustive set of disjoint objects simultaneously with the data acquisition process, where, pixels belonging to an object form a path-segment in the spectral space. Each path-segment is characterized by an object-feature set. Then, the set of object-features, rather than the original pixel-features, is used for data analysis and data classification.

AMICA is applied to several sets of real image data, and the performance and reliability of features is evaluated. Example results show an average compaction coefficient of more than 20/1 (this factor is data dependent). The classification performance is improved slightly by using object-features rather than the original data, and the CPU time required for classification is reduced by a factor of more than 20 as well. The feature extraction process may be implemented in real time, thus the object-feature extraction CPU time is neglectable; however, in the simulated satellite environment the CPU time for this process is less than 15% of CPU time for original data classification.

The work described in this report was supported in part by NASA Grant NAGW-925.

CHAPTER 1
INTRODUCTION

1.1 Background

The demand for a powerful system in the study of the Earth's resources (vegetation, water, minerals, etc.), monitoring of the environment, and land mapping calls for observation methods that will provide increasingly detailed information about relevant parameters with adequate resolution in time and space. In view of these needs remote sensing by imagery has rapidly gained interest in the last decade [1]. In remotely sensed data acquisition systems, observations of a scene are represented by a large set of multispectral image data taken by a variety of sensors [2,3]. The reflectance spectrum can be used to identify a large range of ground cover materials [4].

In the past, data were typically acquired in four to seven spectral bands. Recent significant developments in sensor technology make possible Earth observational remote sensing systems with unprecedented spectral resolution and data dimensionality. The value of these new sensor systems lies in their ability to acquire a nearly complete optical spectrum for each pixel in the scene. Such imaging spectrometry now makes possible the acquisition of data in hundreds of spectral bands simultaneously. For example, the High Resolution Imaging Spectrometer (HIRIS) now being developed by NASA for launch in the mid-1990's, is to have 30 meter ground

resolution, 1000 pixels per scan line, and 192 spectral bands [5]. As a result, the complexity and costs associated with the analysis of the multispectral image data and data transmission, storage, archival, and distribution are likely to increase enormously. Therefore, the search for efficient methods for scene representation to reduce the amount of data but which do not sacrifice information content increases in importance. The current work is directed at the reduction of such data redundancy in the scene representation.

On-line data redundancy reduction is especially important in data systems involving high resolution remotely sensed image data which require related powerful communication, archiving, distribution and data analysis subsystems. High resolution imaging data systems [6], e.g., AIS, AVIRIS, and HIRIS, are example systems where the application of on-line feature extraction will be important. AMICA# is an "on-line preprocessing algorithm that uses unsupervised object-feature extraction" to represent the information in the multispectral image data more efficiently, to achieve data redundancy reduction. AMICA incorporates spectral and contextual information into the object-feature extraction scheme. The algorithm uses local spectral-spatial features to describe the characteristics of objects in the scene. Examples of such features are size, shape, location, and spectral features of the objects. The local spatial features (e.g., size, shape, location and orientation of the object in the scene) of the objects are represented by a so-called spatial-feature-map. The spectral features of an object are represented by a d-dimensional vector.

For simplicity from now on the "on-line unsupervised object-feature extraction algorithm" will be referred as **AMICA** (Automatic Multispectral Image Compaction Algorithm)

1.2 Objective of the Investigation

The objective of this research is to develop a joint unsupervised object extraction and feature representation technique for removal of redundant data in high resolution multispectral image data for remotely sensed scene representation. On-line removal of redundant data is important in reducing costs and time delays in links between the sensor and the information user, or alternately on-line removal of redundancy can be used to obtain higher performance in the data analysis. For implementing this concept, the AMICA is investigated with emphasis given to practical data system considerations. The technique is intended as an on-line unsupervised object-feature extraction technique for scene representation, to achieve data redundancy reduction. In the rest of the thesis this technique, the joint unsupervised object extraction and feature representation, will be referred to as "object-feature extraction" or alternately "image compaction."

This technique is based on the fundamental assumption that the scene is segmented into objects such that all samples (pixels) from an object are members of the same class; hence, the scene's objects can each be represented by a single suitably chosen feature set. Typically the size and shape of objects in the scene vary randomly, and the sampling rate and therefore the pixel size are fixed. It is reasonable to assume that the sample data (pixels) from a simple object have a common characteristic. A complex scene consists of simple objects. Any scene can thus be described by classifying the objects in terms of their features and by recording the relative position and orientation of the objects in the scene.

The proposed image compaction can be thought of as a combined object extraction and feature representation process, where, object extraction is a process of scene segmentation that extracts similar groups of contiguous pixels in a scene as objects according to some numerical measure of similarity. Intuitively, objects have two basic characteristics: they exhibit an internal regularity, and they contrast with their surroundings. Because of the irregularities due to the noise, the objects do not exhibit these characteristics in an obvious sense. The ambiguity in the object detection process can be reduced if the spatial dependencies, which exist among the adjacent pixels, are intelligently incorporated into the decision making process.

In this work a new method for detection of objects is developed for on-line object-feature extraction. This method utilizes a new technique based on a so-called **unity relation** which must exist among the pixels within an object. This unity relation among the pixels of an object is defined with regard to an adjacency relation, spectral features, and spatial features in an object.

The technique must detect objects in real-time and represent them by means of an object-feature. The unity relation, for on-line object-feature extraction, can be realized by the path-hypothesis. The path-hypothesis is based on the fundamental assumption that pixels from an object are sequentially connected to each other by a well-defined relationship in the observation space, where the spectral variation between two consecutive points in the path follows a special rule; i.e. each pixel in an object has a certain unity relationship with the corresponding path in the observation space. The path-hypothesis is explained and illustrated in the next chapter.

By employing the path-hypothesis and using an appropriate metric for similarity measure, the scene can be segmented into objects. Viewing the scene segmentation from this perspective, a new on-line object detection process (for scene representation) is developed for unsupervised object-feature extraction in remotely sensed ground cover data.

The feature extraction process is based on the fundamental assumption that reasonably well defined objects in the scene can be represented by a suitably chosen feature set, which is extracted from the multispectral image data. An object is described by a feature-set (so-called object-feature) of parametric primitives which will be explained in chapter two. The performance of a feature extraction process is measured in terms of the information-bearing quality of the features versus the data set size. Since the noise measurement on one pixel does not influence the measurement noise for another pixel, the effect of noise is decreased substantially by averaging the spectral response of the pixels within an object. The reliability of selected object-features is investigated in chapter four.

In summary, the proposed on-line unsupervised object-feature extraction process for an efficient scene representation consists of four major steps:

object detection: Pixels from an object are sequentially connected to each other with an appropriate path. Any two points of this path satisfy a certain well defined unity relationship. This relation is defined by a specific measure in the observation space. Based on this hypothesis a functional for measuring the unity relation is defined. By using this functional the unity relation between pixels is measured, and an

appropriate path-segment is selected; i.e. pixels in an object form a path-segment in the observation space.

segmentation: The scene is partitioned into a finite, but unknown number of objects by checking the unity relation among adjacent pixels, which corresponds with growing the path-segment in the observation space.

feature extraction: Each object is represented by an object-feature set in the "feature space." The information about attributes of a given object, such as size and location, the object's spectral and contextual features can be used for representation of an object in the feature space.

measuring feature relevancy: By using an efficient measure the choice of feature relevancy in the feature space is checked. Features may then be extended or merged in the feature space. In real time, this step occurs simultaneously with the feature extraction process.

The object detection and feature extraction processes are unsupervised and realizable. After the final step the scene is represented by the relevant object-features in the "feature space," which reduces the size of the image data but preserves the useful information. Thus, the reduction of data redundancy is achieved.

It is expected that this mapping, from the pixel-feature set into the object-feature set, should generate a relevant feature set for scene representation. The spatial feature of an object often has only a weak relationship to its class. However, many classes can be distinguished reasonably well on the basis of their spectral features, using statistical pattern classification techniques. One

might expect the classification accuracy to be higher if an unknown object is classified using spectral-contextual features rather than when it is classified using only the measurement made on the pixels without context.

Spectral information of surrounding pixels is correlated with the center pixel under consideration. In object detection the spectral features of adjacent pixels are considered using substantially neighboring information; thus, the object-feature consists of both spectral and contextual features. Therefore, it is expected that the classification accuracy to be higher by using object-feature rather than the individual pixel-feature. A basic premise of object versus pixel scene representation is that since an object is usually large compared to the size of a pixel, data analysis on the basis of the object-features will be much faster and reliable.

1.3 Related Work

Basically, three different approaches have been available to reduce the redundancy in the image data representation:

- image data compression methods
- feature extraction methods
- image segmentation techniques

Image data compression techniques usually aim at an optimal trade-off between efficiency and implementation simplicity, according to the user's needs. Data compression is the science and art of processing information to obtain a simple representation with at most a tolerable loss of fidelity. Such simplification may be necessitated by storage constraints, bandwidth

limitation, or communications channel capacity. Fidelity of the eventual reproduction of the information based on the simplified or compressed representation is measured by mathematical distortion measures [7] such as error or purely subjective methods based on psychophysical tests (e.g., subjective image appearance).

Many methods and techniques are known for image data compression [8,9,10]:

(a) Methods based on image processing in the spatial domain: predictors, interpolators, delta modulation, differential pulse code modulation, variable word-length encoding, etc.

(b) Methods based on a preliminary mapping of the image data in a transformed domain and on a subsequent suitable selection of the transformed data: transformed coding, e.g., based on the Discrete Fourier, Hadamard, Walsh, Haar, Karhunen-Loeve, etc. transforms.

(c) Hybrid methods employing a combination of methods (a) and (b).

Generally these methods are used for one-dimensional (in a spectral sense) image data; e.g., television signals. But for high-dimensional (again, in a spectral sense) images, especially when the process should be on-line and unsupervised, those methods are not as practical.

The second approach to image data redundancy reduction uses spectral feature extraction methods. In data analysis the problem of choosing the relevant features for a set of multispectral image data is referred to as feature selection. Feature selection is the science or art of processing data to obtain

a simple representation with information per unit data volume maximized. In statistical pattern recognition, data analysis and signal processing feature selection reduce the information redundancy, allowing a simpler description of the system under consideration. In multispectral image data, feature selection also reduces the total computational time whenever the analysis involves a combinatorial handling of features, as in computer data classification. In most of the cases, feature extraction improves the data classification performance [11,12].

In the literature on this subject, there are many examples where the feature selection problem is handled by using linear statistical methods, such as variance and principal component analysis [13,14]. Some efforts have also been made to select features in the case of non-linear models [15]. The transformation can be parametric or nonparametric [16] but should reduce dimensionality and at the same time preserve the information necessary for identification of the spectral response.

One of the traditional feature extraction techniques is the Karhunen-Loeve, or principal component, transformation [17]. However, construction of this transformation requires estimation of the covariance matrix, and this process would take on the order of 10^{13} arithmetic operations for a typical scene 10 by 10 km such as will be acquired by AVIRIS. On the other hand, unfavorable conditions arise whenever the sample of points is so small, or the number of features is so large, that the "low of large numbers" is not applicable and the Gaussian distribution assumption is a weak hypothesis [18,19]. Or especially when the data processing is on-line and the analysis for feature extraction is in real time, in this case it should be very fast and

unsupervised. Thus, the above approaches are not as practical. Clearly, more computationally efficient transformations are needed for the analysis of multispectral image data.

An extensive bibliography of published papers and reports dealing with scene segmentation and models is presented in [20-31]. Image segmentation and clustering are both methods of grouping data. The difference between image segmentation and clustering is that in clustering, the grouping is done in the measurement space; in image segmentation, the grouping is done on the spatial domain of the image. Generally, two basic approaches to scene segmentation have appeared in the literature: the "edge detection" approach, which attempts to exploit object contrast [20,21,22], and the "region growing" approach, which uses intra-object similarity [23,24,25]. The edge detection or boundary finding approach has two steps. First, points along the boundaries of objects are found, then the complete boundaries are derived from the boundary points. Among the techniques used to detect boundary points are local gradient, template matching, two-dimensional function fitting, clustering and gradients estimated from a variable sized neighborhood.

The gradient approach, in general, is inherently noisy and produces borders that are discontinuous while also producing spurious isolated points. An algorithm developed by Wacker and Landgrebe [26] based on clustering technique is more stable and less noisy. None of these techniques find boundary points with little enough error to form complete object boundaries or guarantee closed boundaries of the objects. Thus, to form a closed boundary for each object, a heuristic procedure using a-priori information is

used to fill in missing boundary points and eliminate irrelevant boundary points. Several heuristic approaches to the problem of edge detection are given which are effective only when the object shape is restricted.

Previously investigated region growing methods have followed two steps: first the image is divided into elementary regions, then regions are merged according to a set of merging rules [25]. In [22] elementary regions are regions of constant gray levels, and the merging rules are heuristics based on what the final objects should look like. This method is difficult to use if objects do not have constant gray levels, as is the case, for multispectral image data.

Most studies have dealt with one-dimensional (in a spectral sense) image data only, e.g. television signals, and rarely has attention been devoted to the multispectral image data. The importance of multispectral image segmentation, in relation to data classification, has been recognized by Landgrebe [31]. It has been shown that more accurate and efficient classification of LANDSAT data for Earth scenes can be achieved by classifying a whole segment at a time rather than by pixel by pixel classification.

Robertson et al [23] developed a partitioning algorithm to partition an image into rectangular objects. Robertson used a recursive image segmentation algorithm by approximating each object by one or more rectangular blocks of image points, where an image has been divided into successively smaller blocks until certain stopping criteria are met. The objects represented by elementary regions are defined by a regular rectangular grid superimposed

upon image. The merging rule is to merge statistically similar, adjacent regions. This method works well only if elementary regions can be made both small enough to allow good boundary approximation and large enough to preserve the spatial characteristics of the objects of which they are a part.

Gupta and Wintz [24] developed an algorithm to find closed field boundaries in a multispectral image by partitioning the images into blobs. Gupta used hypothesis testing by first dividing the image into elementary regions, e.g. 2x2 arrays. The algorithm compares first and second order statistics of adjacent subsets. Adjacent subsets having similar first and second order statistics are merged into blobs. In this manner the entire picture is separated into blobs such that the image elements within each blob have similar characteristics. By varying either of two parameters the amount of consideration given to grey level and texture can be adjusted. The algorithm guarantees closed boundaries, where the boundaries are defined to be the edges between blobs. The algorithms, RIMPAR [23], BLOB [24] and ECHO [25], guarantee closed boundaries.

Some of the earlier studies cited above suffered from open boundary problems, some from the sensitivity to the noise, and mostly from the limitation due to the necessary conditions requiring large amounts of CPU time, iterative processing, and some from the dependency on manual initiation or supervision for threshold assignment. In those approaches that the extraction of an object is based on a per-defined elementary regions, even they may guarantee closed boundaries, the extraction of an object with a complex and irregular boundary is not accurate or it will be very time consuming. Some object extraction process requires a priori information

about the scene (e.g., statistical properties of the scene, distribution, shape and size of object, or number objects in the scene) and require threshold assignment and supervised initiation when the number of training sample is so small such that the "low of large numbers" is not applicable and Gaussian assumption is a weak hypothesis for making a correct decision.

1.4 Thesis Organization

This thesis is organized into five chapters. Following an introduction to the general concept of numerical scene observation model and methods for scene description, chapter two introduces basic mathematical tools to realize the unity relation for object extraction, where the path-hypothesis is explained and the properties of the unity relation are investigated in several theorems with their proofs. Based on the unity relation and the path-hypothesis, the Automatic Multispectral Image Compaction Algorithm (AMICA) is developed. The flow chart and the details about AMICA are presented in chapter three. In the fourth chapter, the reliability of object-features versus pixel-features is investigated and the results of the object-feature extraction, applied to several sets of real data, are presented. Finally, in the fifth chapter, the thesis is concluded with a discussion about the object-feature performance versus the pixel-features.

CHAPTER 2
SCENE MODEL AND FEATURE EXTRACTION

2.1 Introduction

The objective of this chapter is to introduce the basic components that make up the structures of an analytical model for scene representation in an efficient measure space. This process is carried out through a specific feature extraction method which maps the original data (pixel observation) into an efficient feature space, called the object-feature-space. The efficiency criterion for this mapping is the amount of useful information that can be extracted from these features, versus the data set size.

One of the fundamental tenets of modern science is that a phenomenon cannot be claimed to be well understood until it can be characterized in quantitative terms. Viewed in this perspective, much of what constitutes the core of scientific knowledge may be regarded as a reservoir of concepts and techniques which can be drawn upon to construct mathematical models of various types of systems and thereby yield quantitative information concerning their behavior. In essence a model is something whose structure, and hence behavior, corresponds in some sense to that of a particular reality of phenomena. In scene modeling, the structures we use will usually be mathematical in notation and will have elements of physical attributes in their nature.

In this chapter we present a model for a scene containing objects. In this model the variations which carry the desired information (object features) are represented in the data, but those variations which do not (noise) are minimized by averaging: each object can be characterized by the expected value of spectral responses of pixels within the object together with the object contextual features. It is assumed that two adjacent objects differ in a measurable way relative to the spectral features S_i or contextual features V_i. Objects with small area, for which the number of the pixels within the object is not sufficient for contextual feature extraction, will be represented only by the spectral feature. However, in general an object can be described by a set of three parametric primitives (S_i , V_i , $\mathbf{L}_i$), where $\mathbf{L}_i$ is the spatial-feature-map.

Following a survey of the general concept of numerical scene representation and methods for scene description, section four introduces the proposed technique for unsupervised object-feature extraction. The basics of feature selection and object-feature reliability are described in section five. In section six object detection based on the unity relation among the pixels in the observation-space is explained, and the basic mathematical properties of the unity relation are determined. After an introduction to similarity measure in section seven, for realization of the unity relation the path-hypothesis is introduced; then objects in both observation-space and feature-space are illustrated. Section seven concludes with construction of the functional for unity relation test. The equations for feature extraction are derived and presented in section eight. After a summary about the proposed feature extraction procedure for scene representation in section nine, this chapter concludes with a brief discussion of object-feature classification.

2.2 Remotely Sensed Scene Observation

In remote sensing systems, properties of an unknown scene are determined at a distant location, through noncontact measurement, based upon the interaction of the electromagnetic radiation with the scene [4,32]. Remote sensing has greatly extended man's perception of the world's resources and interaction of nature and man made influences. Remote sensing has grown from simple photography and photo interpretation to satellite borne sensors and sophisticated machine aided analysis. Important developments in sensor technology, computer systems, pattern recognition theory, and image processing techniques have brought the remote sensing state of art to the point where it is a powerful tool for studying the Earth's resources remotely. In the modern remotely sensed Earth Observation Systems there are two main distinct activities: scene representation by numerical data or *data acquisition*; and *information extraction*, by means of which the acquired numerical data is converted into useful information.

The scene (in this work this is assumed it to be part of the Earth's surface) is the target of the remote sensing system, which is under investigation and the interest is to extract information about the scene's structure and content. The desired information is assumed to be contained in the spectral, spatial, and temporal variation of electromagnetic energy coming from the scene which is gathered by the sensors.

Typically a complex scene is composed of relatively simple objects of different sizes and shapes, each object of which contains only one class of surface cover type. The scene is often described by classifying the objects

and recording their relative positions and orientations in the scene in terms of tabulated results and/or a thematic-map (class-map), Fig. 2.1.

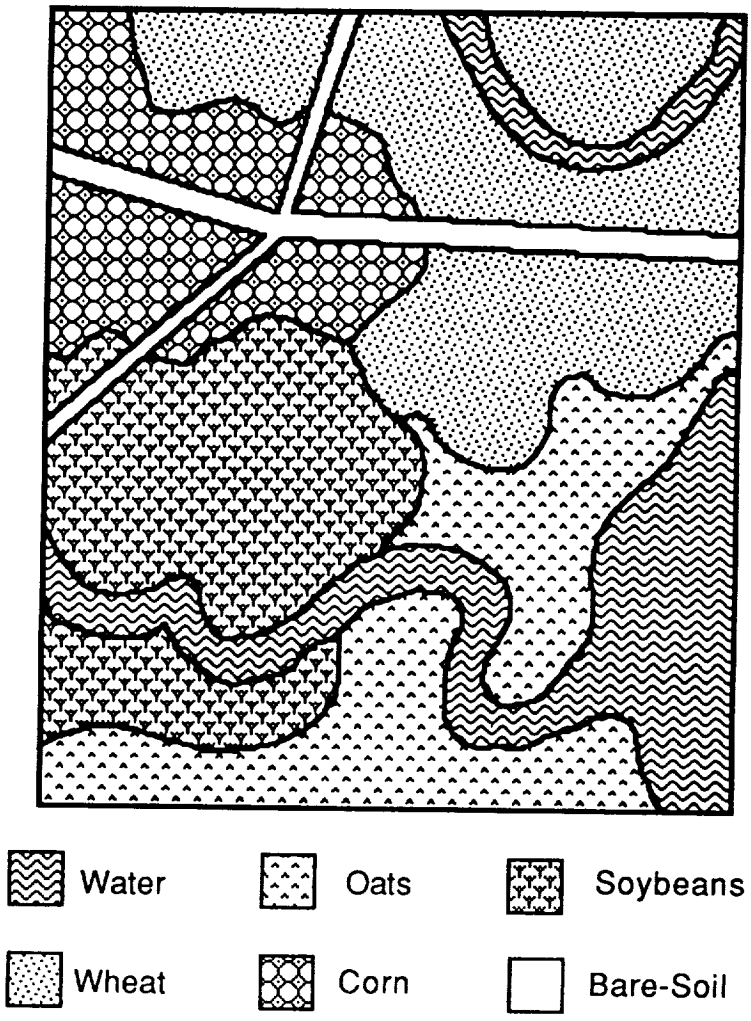


Fig. 2.1. A typical thematic-map (class-map)

A critical portion of the data acquisition system is a multispectral sensor. Multispectral sensor systems employ sensors to observe portions of the electromagnetic spectrum typically ranging from the visible region to the reflective infrared regions. Also the thermal portion of the spectrum and other portions of the electromagnetic spectrum such as microwave region have important uses in remote sensing.

In a remote sensing system, primary features of a scene are formed by multispectral observations, which are accomplished by spatially and spectrally sampling the scene.

A typical multispectral sensor samples several spectral dimensions and one spatial dimension from the scene at a given instant of time. The second spatial dimension is provided by the motion of the platform which carries the scanner over the region of interest, generating a raster scan; alternately, the raster can be provided by area array detector [6]. Thus, through the data acquisition system, the scene may be viewed in an image form taken at each of a number of electromagnetic wavelengths. This image can be thought of as a multi-layer matrix whose elements are called pixels. The physical model of a scene after spatially and spectrally sampling is illustrated in Fig. 2.2.

Let $p_{[i,j]}$ be representative of a pixel at the location of the i^{th} row and j^{th} column of the scanned image of the scene. As shown in Fig.2.2, the number of pixels in each scan line is n_x, and the number of lines in each scene frame is n_y, which both are finite and assumed known.

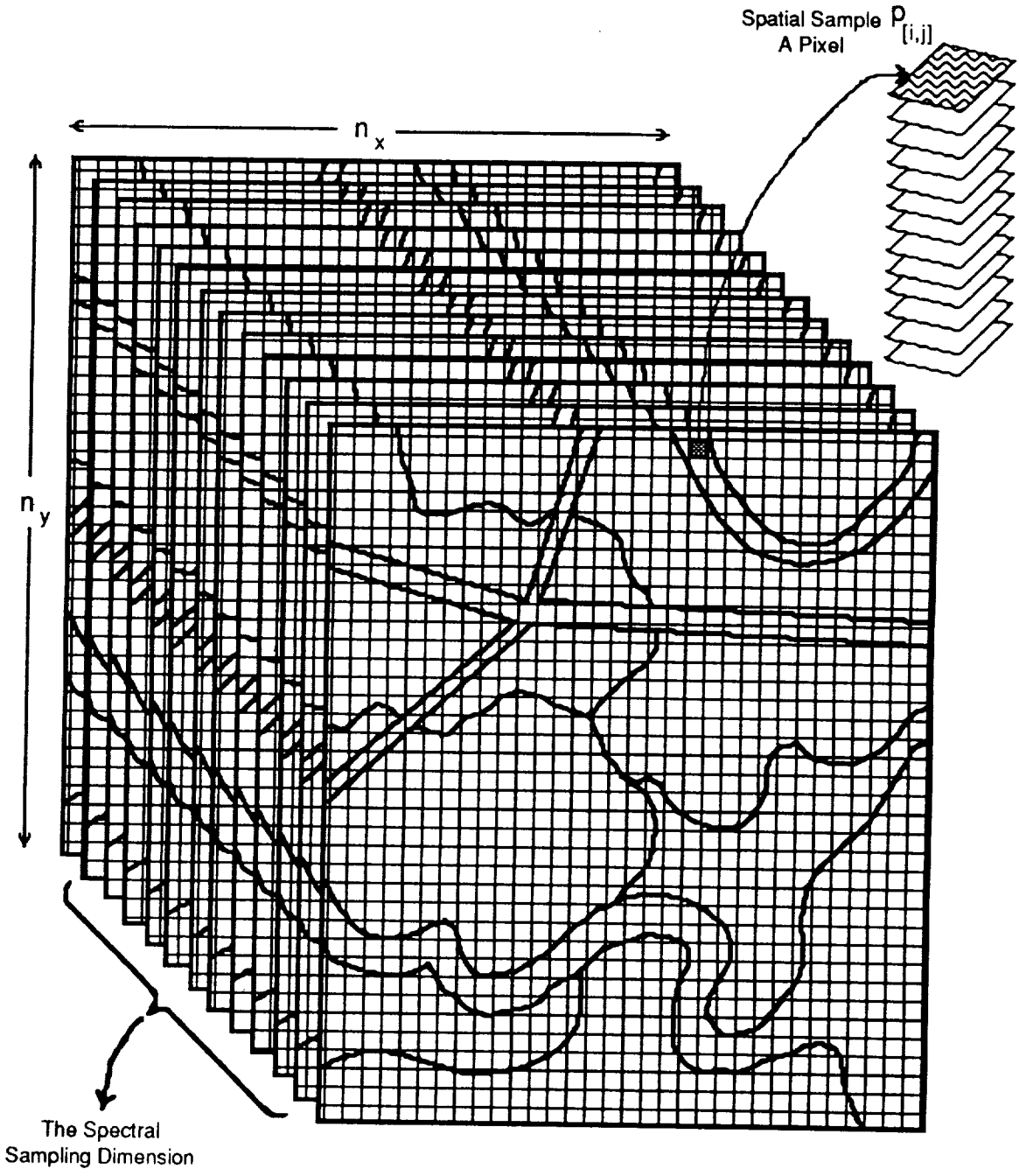


Fig. 2.2. Multispectral scene pixel-description

The instantaneous field of view of the data acquisition system defines the pixel size (spatial resolution) and is the projection of the detector back through the optical system onto the scene. The spectral resolution of the data acquisition system defined the spectral dimensionality **d** of the data. Figure 2.3 illustrates the spectral and spatial resolution of a pixel-feature.

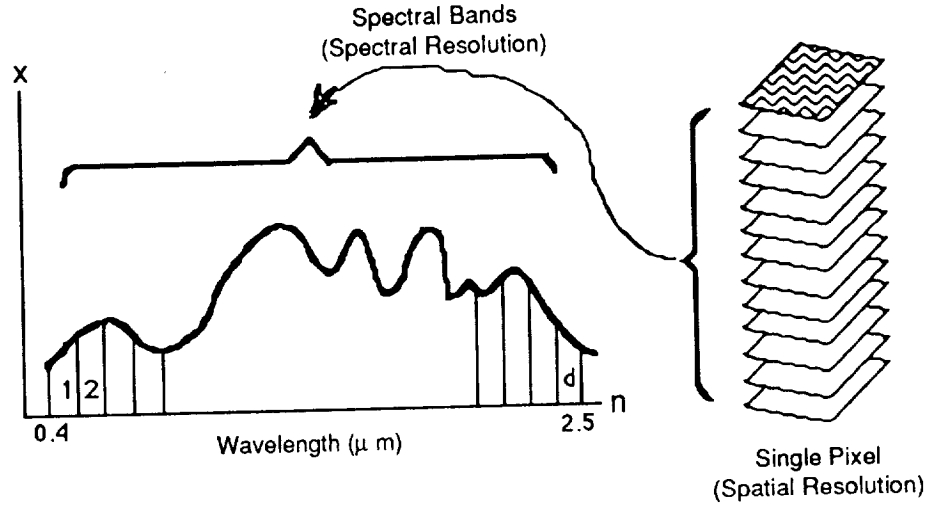


Fig.2.3. A typical pixel-feature

The output of the data acquisition system is multi-dimensional data, represented numerically by an ordered set of **d**-dimensional vectors. We represent the multispectral sensor response by a functional $f(.)$, where this functional maps the pixel's spectral attributes into a d-dimensional real space.

$$f: \{\text{pixel spectral responses}\} \rightarrow \mathbf{R}^d$$

It is assumed that d different spectral observations of each single pixel are obtained by the multispectral sensor. Each pixel is represented numerically, then, by an d-dimensional set of measurements, denoted by vector X_k in the observation space, where the vector components are a set of measurement at each spectral interval; i.e. each value in this set, x_{ik}, is a real number proportional to the energy received from the pixel by the sensor in that particular band of the electromagnetic spectrum.

The vector X_k is then referred to as the feature vector of $p_{[i,j]}$; i.e., each pixel $p_{[i,j]}$ can be identified by knowing its feature X_k. Mathematically this set is represented by a d-tuple vector and is called a *pixel-feature*.

$$X_k=[x_{1k}, x_{2k}, \dots , x_{dk}]^t=f(p_{[i,j]})$$

The set of pixel-features, which represents the whole scene by a subset of $\mathbf{R}^d$ space, is called the *observation-space* and represented by:

$$\mathbf{P}=\{X_1, X_2 , \dots , X_{np}\}$$

where n_p is the number of spatial observation (pixels).

$$n_p=n_x.n_y$$

The volume of $\mathbf{P}$ is defined for scene representation by the number of spatial and spectral observations, and is equal to:

$$v_p=d.n_p$$

Notice that there is a relation between the index of the pixel-feature "k" and the pixel spatial coordinates in the scene [i,j]. The spatial location of each

pixel in the scene can be determined uniquely by knowing its row number, i, and its column number, j, where the index address of a pixel is represented by a two-tuple [i,j]. An index operator is defined, such that by knowing the pixel's index address, [i,j], the index of the corresponding pixel-feature in the observation space, the scalar k, can be determined; i.e., it represents the *index mapping* from the spatial domain into the spectral domain. The well ordered sequence of indices, which represents the whole scene, is called the *spatial-feature-map*, **L**; where **L** is a subset of natural numbers.

$$\mathbf{L}=\{k;\ X_k=f(p_{[i,j]})\in \mathbf{P}\}$$

$$i=1, 2, ..., n_x \quad \text{and} \quad j=1, 2, ..., n_y$$

The spatial-feature-map is needed to identify the spectral feature of a pixel, by knowing its spatial coordinates in the scene, in the observation space. The scene is represented numerically by the pixel-feature-set **P** together with the corresponding spatial-feature-map **L**, of which all elements, for the scene, are fixed and deterministic.

One of the distinctive characteristics of the spatial dependence in multispectral data is that the spectral separation between two adjacent pixels is less than two non-adjacent pixels, because the sampling interval tend to be generally smaller than the size of an object; i.e., two pixels in spatial proximity to one another are unconditionally correlated with the degree of correlation decreasing as the distance between them increases [33].

The results of study on measurement of different order statistical spatial dependency in image data, specifically the measurement of first, second and

third order amplitude statistics along a image scan line show considerable correlation between adjacent pixels [34]. Seyler [35] concluded, from the measurement of the distribution of the difference between adjacent pixels, that the probability that two adjacent pixels have the same gray level is about 10^6 times the probability that they differ by the maximum possible amplitude difference. Kettig [33] by measuring the spatial correlation of multispectral data, showed that the correlation between adjacent pixel is much less when conditional upon being with an object, as compared to unconditional correlation.

High correlation among adjacent pixels in the observation space represents redundancy in scene data. When such redundancy occurs, reducing the size of the observation space should be possible without loss of information.

2.3 Object vs. Pixel Scene Description

As previously stated the scene is assumed to consist of relatively simple objects of different sizes and shapes. The resolution of the spatial representation depends on both pixel size and the interval between samples, which are usually equal, Fig. 2.2. By under-sampling information is lost; however, over-sampling will cause increased redundancy.

Typically the size and shape of objects in the scene vary randomly, Fig.2.1, and the sampling rate, and therefore the pixel size, is fixed; it is inherent in image data that data-dimensionality (the number of spatial-spectral observations for scene representation) increases faster than its intrinsic-dimensionality (the size of the smallest set which can represent the same scene, numerically, with no loss of information). Because the spatial sampling interval is usually comparable to the object size, it follows that each object is represented by an array of similar pixels. Therefore, scene segmentation into pixels is not an efficient approach for scene representation; however, a scene can be segmented into objects, and since the shape and size of objects match the scene variation, scene representation by simple-objects is more efficient.

An *object* consists of contiguous pixels from a common class which have a unity relationship with each other; i.e., their features are statistically similar, and since the recognition of patterns is based on pattern-feature comparison (although there is a fluctuation in the spectral response of pixels within an object) these pixels have the same common characteristic and carry equivalent "useful" information about the scene. The similarity among the

pixels of an object occurs in part in the form of redundancy in the measurement (observation) space. Redundancy in the object data implies that a scene can be segmented into objects, instead of pixels, with no loss of useful information. Thus the method of scene description by object extraction will be directed at the reduction of data redundancy in the scene representation, if the relevant object-feature is selected.

Object detection refers to finding the natural groups among the contiguous pixels. In other words, the data is sorted into objects such that the "Unity Relation" holds among members of the same object and not between members of different adjacent objects. Essentially object detection might be viewed as assigning appropriate meaning to the terms "natural unions" and "natural association" where "natural" usually refers to homogeneous and "well separated structures;" which makes object extraction similar to clustering process [31].

Object extraction and clustering are similar in the sense that they both are methods of grouping data; however, spatial considerations make clustering and object extraction different. Because an object can be textured, the pixels within an object might not form a compact cluster in the measurement (observation) space. Also, because there can be several instances of a particular class of entities in a single image, Fig. 2.1, nonadjacent objects might be nearly identical in observation space. Another difference is that in object extraction the existence of a partition that completely separates objects is guaranteed. However, in clustering, if we allow underlying classes with overlapping density functions, the classes can never be completely separated in the observation space.

Object extraction can be thought of as transforming the original image, which is a pixel-description of a scene, Fig. 2.2, into an arrangement of object-description, Fig. 2.4.

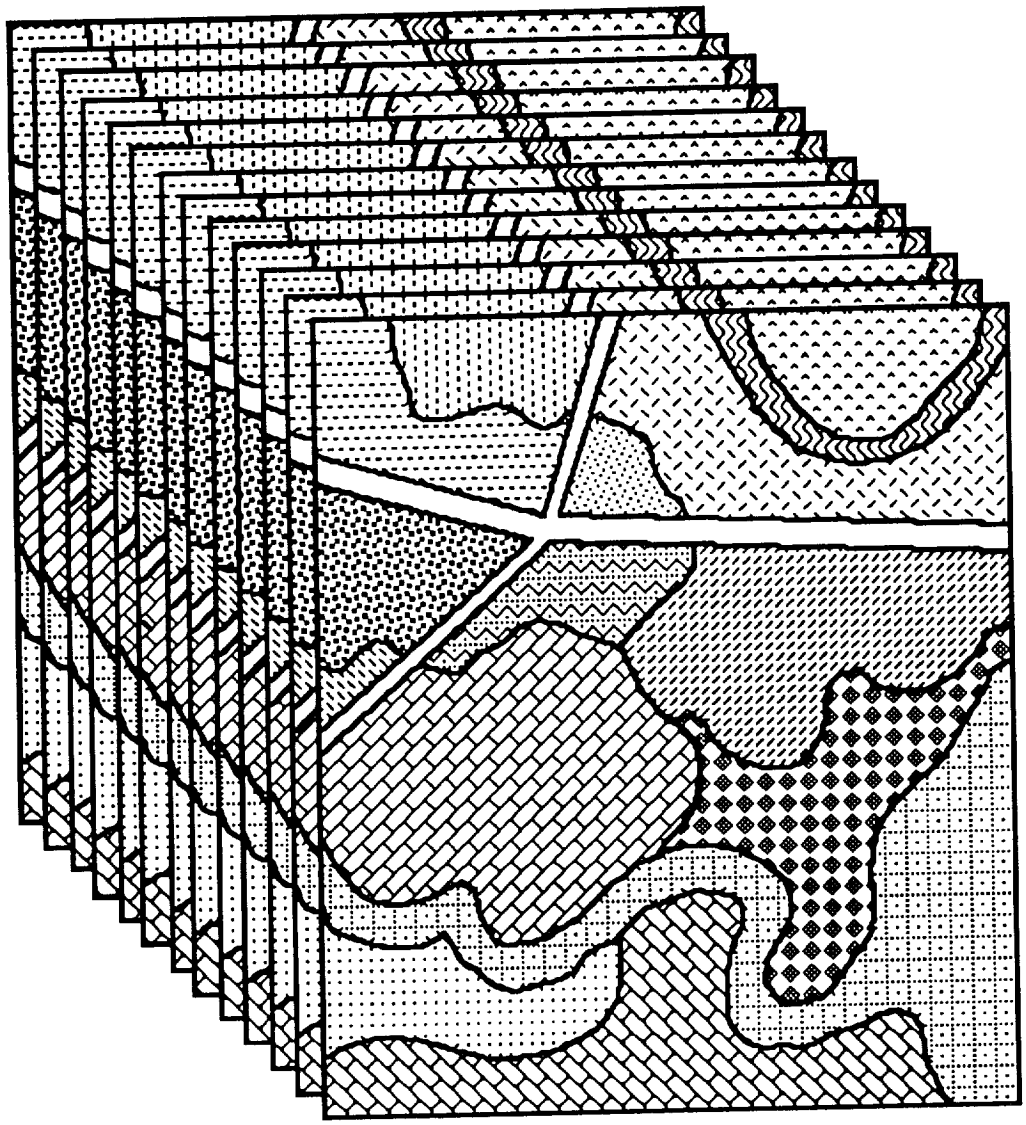


Fig. 2.4 Multispectral scene object-description

An object-description is often better than a pixel-description, for two basic reasons:

- First, more information about the scene entity is available from a collection of pixels associated with the object than from an individual pixel associated with the scene. This fact has been exploited by "object" classification algorithms that make a classification decision for each group of image points, for example by sequential classification [11]. The potential advantages of object classification are especially great when class probability densities differ in shape but exhibit a high degree of overlap. Classifying objects instead of pixels also allows the measurement and use of spatial characteristics such as size, shape and texture, which have been found to be useful in classification.

- Second, an object representation is often more compact than a pixel description. This savings in storage space or transmission speed occurs if objects contain enough points so that specifying the locations and essential properties of the objects takes fewer bits than specifying the collection of individual pixel properties.

In this thesis a method for detection of objects, based on a specific region growing method, is developed for automatic scene representation by object-features versus pixel-features. This method utilizes a new technique, based on the unity relation and the path hypothesis, which detects objects and represents them by means of an object-feature in the feature-space

2.4 Object Model

In the analysis and processing of multispectral images one encounters a large amount of data. To enable efficient processing of this data, it would be preferable to have an underlying model that explains the dominant characteristics of the given image data. Subsequent processing of the images can be efficiently accomplished by using the models fitted to the data. With the above scheme, a scene is segmented into spatially disjoint objects: the image described by the objects' features and by recording the relative position and orientation of the objects in the scene.

Let $\mathbf{L}_i$ be the spatial-feature-map of an object $\mathbf{P}_i$ in the scene. In other words, $\mathbf{P}_i$ is a set composed of contiguous pixels belonging to the same class, and $\mathbf{L}_i$ is the index-set of the pixels in the corresponding object.

$$\mathbf{P}_i=\{X_{1i}, X_{2i}, \ldots, X_{ni}\}$$

$$\mathbf{L}_i=\{k: X_k \in \mathbf{P}_i\}$$

The number of objects in the scene, n_o, is unknown, but, the scene's spatial-feature-map consists of union of finite disjoint closed-sets, represented by $\mathbf{L}_j$'s.

$$\mathbf{L}=\mathbf{L}_1 \cup \mathbf{L}_2 \cup \ldots \cup \mathbf{L}_{no}$$

$$\mathbf{L}_i \cap \mathbf{L}_j=\emptyset \text{ for } i \neq j$$

Objects in imaged scenes are describable by sets of relevant attributes or features. These features represent distinct measurements or observable properties. The object's initial measurements, which are encoded as pixel-

features, are subsequently subjected to a object-feature transformation: the relevant object-feature can be the spatial-feature-map of the scene and the object spectral-contextual features, which are explained in the next section in this chapter. Figure 2.5 illustrates an example of object-feature.

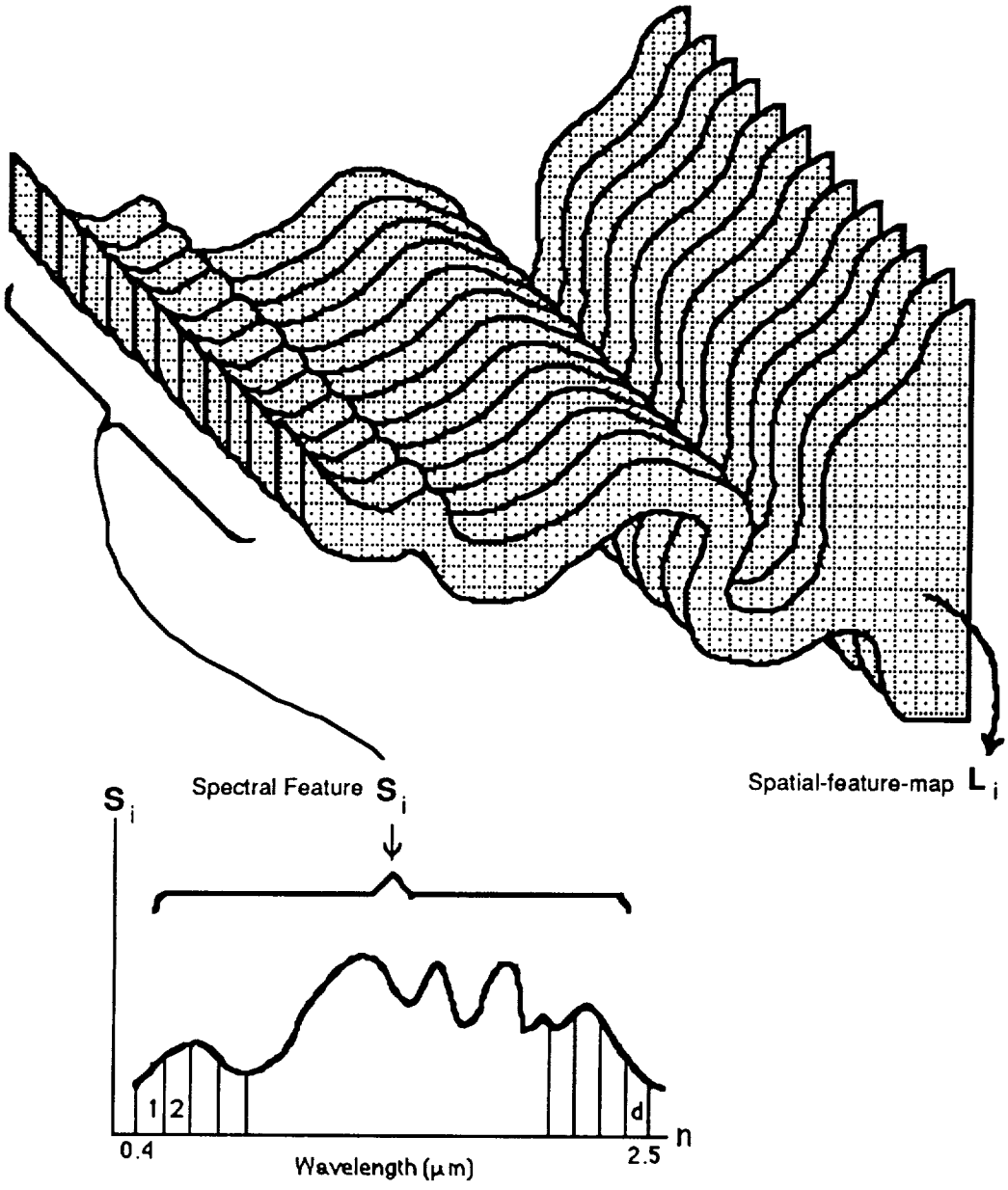


Fig.2.5. A typical object-feature

Each of the objects contains a union of similar pixels, and the union of the simple objects represents the whole scene. All pixels of an object, whose pixels satisfy the unity relation, can be represented by an object-feature set: a complex scene consists of objects, any scene can thus be described by classifying the objects in terms of their relevant object-features Y_i. The object-feature extraction process is denoted by a nonlinear function $\Psi(.)$ which maps the observation-space (pixel-feature-set) into a more efficient feature-space (object-feature-set). The procedure for object feature extraction is explained in the "Object-Feature Extraction" section of this chapter. Here we represent the feature extraction only by notation.

$$\Psi : \mathbf{P} \rightarrow \mathbf{F}$$

The operator $\Psi(.)$ is actually the feature extractor which maps the object's pixel-features $\mathbf{P}_i$ into a point, called object-feature Y_i, in the feature space. It is expected that this mapping, from the pixel-feature space into the object-feature space, should generate a relevant feature set for representation of the objects' attributes.

$$Y_i = \Psi(\mathbf{P}_i)$$

The set of object-features, $\mathbf{F}$, which represents the whole scene, is called the *feature-space*: the scene can be represented by the object-feature-set $\mathbf{F}$. It is desirable that the amount of useful information conveyed by object-feature-set, $\mathbf{F}$, be equivalent to the amount of useful information conveyed by the original data set, $\mathbf{P}$, while at the same time v_f, the volume of the feature set, should be much smaller than v_p, the volume of the original data set.

$$v_f << v_p$$

A system that maps a sequential source set $\mathbf{P}=\mathbf{P}_1 U \mathbf{P}_2 U \ldots U \mathbf{P}_{no}$ into an object-feature set $\mathbf{F}=\{Y_1, Y_2, \ldots, Y_{no}\}$ defines the ***compaction system.*** The allowed size of the reproduction set is much smaller than the total size of source set and hence compaction is achieved; i.e., it takes fewer bytes to specify the reproduction set than the original source set. The compaction coefficient c is defined as the ratio of the volume of pixel-feature-set, v_p, divided by the volume of the object-feature-set, v_f.

$$c=v_p / v_f$$

The accuracy of this system (the information content in the object-feature-set) is dependent on the parametric primitives which are used in object-feature construction; however, this accuracy has an upper bound which is controlled by the level of noise which exists in the acquired data. In the analysis of a set of data points in multidimensional space, the need to choose the most relevant features frequently arises. Feature selection techniques are used to find properties of objects in the scene which can serve as the strongest clues to their identification. This subject is investigated in the next section where the appropriate feature will be selected for object representation.

2.5 Feature Selection

Typically, objects are classified (grouped) by an experimental study of some phenomenon. One basic property all objects possess is that they are describable by some set of relevant attributes (features). These features represent distinct measurements or observable properties. In the case of physical objects attributes can include, for example, size, shape, texture, average spectral response, etc. The initial encoding of the attributes is dictated by the measurement devices in use or by an established convention. Features may be measured on different scales: qualitative scales; e.g., nominal; or quantitative scales; e.g., ordinal, interval, ratio and absolute. These initial measurements, which are encoded as numerical variables, may be subsequently subjected to a problem-dependent transform. Qualitative properties such as small, medium, large, etc. can be replaced by relative numbers. In this thesis we restrict consideration to a case in which all the attributes making up the object feature are quantitative. The choice of relevant features is very task-dependent and may depend more on the judgment of the system designer or operator than on any other system components.

In theory, decisions about class membership for a noisy object should be based upon as many observations of the object as possible, and preliminary decisions concerning subsets of object-features can provide less than maximally reliability recognition. Thus theoretically, the most reliable decision should be based upon all the pixels in the object. Also in theory,

every result achievable with d variables can also be achieved with d+1 variables, but the converse is not true.

Thus one might expect that by increasing the number of features the object recognition error rate should decrease or at least stay the same, but in practice quite often the performance of the features will improve up to a point, then begin to deteriorate as further attributes are added. This is referred to as the Hughes' phenomenon [36]. Kovalevsky [37] points out that though in complicated cases of interdependent variables it is not easy to estimate the error probability, it is always true that increasing the number of variables used for taking statistical decisions may only improve the reliability, however the decision procedure error may become worse.

Duda and Hart [12] comment that for the Bayesian decision procedure beyond a certain point the inclusion of additional parameters leads to higher probabilities of error, if the number of training samples is finite. Trunk [19] shows that the probability of error of the Bayes detector approaches zero as the dimensionality increases and all parameters are known, but the probability of error approaches 0.50 (for the two class case) as the dimensionality increases and the parameters are estimated. Thus, even though theoretically as many parameters should be used as possible, a specific decision procedure may become worse by using too many parameters.

The existence of an optimal set of features is indicated for the representation of the objects, relative to feature selection and the feature reliability problem. An object can be described by a set of parametric primitives. Such primitives

may be based on observation as well as knowledge about the object. Typically in remote sensing the important primitives, for recognition of an object, are spectral feature and/or contextual features. But since it is usually presumed that the shape and size of natural objects in a scene (ground cover types) are random and unrelated to the ground cover classes, these features are often ignored in feature extraction and pattern recognition of the ground cover types. However in this work, the objects' geographical features are preserved in the spatial-feature-map and can be used by an appropriate pattern recognition system, if it be necessary.

It is assumed that two adjacent objects differ in a measurable way relative to the spectral or contextual features. In this system, a set of points representing similar patterns are represented with the same features. Thus, the attributes of $\mathbf{P}_i$ can be refined by observation which are given by a set of three parametric primitives:

$$Y_i = (S_i, V_i, \mathbf{L}_i)$$

where S_i is the estimated within-object spectral feature representation, V_i is the estimated contextual feature, and $\mathbf{L}_i$ is the spatial-feature-map or the object geographical shape and location in the scene.

Let n_i be the number of pixels in the object $\mathbf{P}_i$, and $\mathbf{L}_i$ be the corresponding spatial-feature-map, then the object spectral feature S_i is estimated by averaging the spectral response of pixels within the object $\mathbf{P}_i$ (it seems plausible to assume that a set of points representing similar patterns could be represented with the same features)

$$S_i = \frac{1}{n_i} \sum_{k \in L_i} X_k$$

The intra-object spatial variation (contextual feature) is object-class dependent [31] and can be used for object recognition. In this work the object contextual feature is represented by the within-object spectral variation vector, V_i. The spectral variation vector is defined by the gradient of the pixel-feature with respect to the spatial displacement in the spatial direction δ. Let u_i be the unit-vector in the direction of i^{th} axis in a $\mathbf{R}^d$ space, the gradient of a pixel-feature, $X \in \mathbf{P}$, is introduced by the following equation:

$$\nabla_\delta X = \vec{u}_1 \frac{\partial x_1}{\partial \delta} + \vec{u}_2 \frac{\partial x_2}{\partial \delta} + \ldots + \vec{u}_d \frac{\partial x_d}{\partial \delta}$$

Then the contextual feature, V_i, is estimated by averaging the spectral variation of pixels within the object $\mathbf{P}_i$.

$$V_i = \frac{1}{n_i - 1} \sum_{k \in L_i} \nabla_\delta X_k$$

Notice that the spatial direction δ can be horizontal, vertical, diagonal, and any other possible spatial direction. This averaging reduces the effects of noise on the responses of a difference operator, ∇. The objects with small area, whose number of pixels within the object is not sufficient for contextual feature estimation, will be represented only by the spectral feature. This is done by adjusting the degree of uncertainty in the feature extraction process: the uncertainty about the feature is inversely dependent on the number of pixels that are contained within the object $\mathbf{P}_i$.

In summary, an object can be represented by a set of three parametric primitives (S_i, V_i , L_i). Although, the contextual feature is dependent on the sensor resolution as well as the sensor altitude from the scene, the intra-object spatial variation between adjacent pixels can be a significant factor for on-line object extraction (see path-hypothesis in section 2.7). It is expected that this mapping, from pixel-feature space into object-feature space should generate a relevant feature set for scene representation. The performance of an object-feature extraction process is measured in terms of the information-bearing quality of the features versus the data set size.

By using the unsupervised object-feature extraction the scene is represented in the feature space. Once each object-feature is classified, the membership of pixels which belong to that object are determined simultaneously regardless of their size and location in the scene. Classification accuracy is an important quantitative measure of feature quality in the remote sensing system. Feature quality can be measured in terms of:

- overall misplacement error
- the feature classification performance
- subjective objects appearance.

The first and second measures are quantitative criteria which have convenient mathematical forms, where the first one measures the number of pixels assigned to the incorrect neighboring object (based on object class type) relative to the total number of pixels in the scene. The second one

measures the number of pixels classified into the correct class relative to the total number of pixels in that particular class.

The subjective objects' appearance is an appropriate criterion when some objects in the scene become more important than others regardless of the size of the objects, or when the ground-truth-map for the first two evaluations is not available. In such cases it is often too difficult to define a mathematical expression for adequately quantifying feature quality. The visual assessment will be used for this kind of qualification. The feature evaluation is performed and explained in detail in chapter four.

2.6 Object Detection and Unity Relation

An object consists of the union of pixels which have a unity relation with each other. The unity relation is the important tool for construction of objects. This is explained in this section. Intuitively, objects have two basic characteristics: they exhibit an internal consistency, and they contrast with their surroundings. Because of the irregularities due to the noise, the objects may not exhibit these characteristics in an absolute sense. The ambiguity in the object detection process can be reduced if the spatial dependencies, which exist among the adjacent pixels, are intelligently incorporated into the decision making process. In this work unity relation among the pixels of an object is constructed with regard to the adjacency relation, the spectral-features and the spatial-feature characteristics in an object.

Image data is represented by a two-dimensional rectangular array of pixels. One of the important characteristics of such data is the special nature of the dependence of the feature at a lattice point to that of its neighbors. The unconditional correlation between two pixels in spatial proximity to one another is often high, and such correlation usually decreases as the distance between pixels increases. One way of characterizing this dependency among the neighboring pixels is to represent it by a ***unity relation*** which exists among the pixels of an object, meaning that an object consists of ***contiguous pixels*** from a common class where their features are ***statistically similar***.

The keys to the unity relation among the pixels of an object are the adjacency relation and the similarity criterion. Pixels from an object carry common characteristic information about the scene, and since all information

extraction processes are based on the patterns' feature comparison, mathematically it can then be said that the unity relation exists between two pixels X_k and X_r if and only if they satisfy two criteria simultaneously:

1) They have an ***adjacency relation*** with each other, in the sense that they are spatially contiguous or their spatial distance is filled by a sequence of contiguous pixels from the same class. The subset of **L** (spatial-feature-map) whom their corresponding pixels having an adjacency relation with the pixel X_k is represented by the set $\mathbf{A}_k$, called neighborhood set. The adjacency relation is illustrated in chapter three.

2) They have the same attributes, or they carry equivalent useful information about the scene, in the sense that their features are ***similar*** to each other. This means that the distance between these attributes, in an appropriate metric-space, is less than unity.

$$\partial_s(X_r, X_k)<1$$

This metric (the similarity measure) is explained in detail in the next section; in this section only the properties of a unity relation, regardless of the selected metric, are investigated. Let $\Re(.)$ be a relation on set **P**. When the relation exists it is represented by $\Re(.)=1$, otherwise by $\Re(.)=0$.

The $\Re(.)$ is a unity relation provided that $\Re(.)$ satisfies the following properties for all X_k, X_r, X_h belonging to pixel-feature-set **P**:

i) Similarity and Adjacency Properties:

$$\{ \partial_s(X_k, X_r)<1 \text{ and } r\in \mathbf{A}_k \} \Leftrightarrow \Re(X_k, X_r)=1$$

ii) Reflexive Property: $\Re(X_k, X_k)=1$

iii) Symmetric Property: $\Re(X_k, X_r) = \Re(X_r, X_k)$

iv) Transitive Property: The unity relation has a transitive property, that is, if the unity relation exists between X_k and X_r, $\Re(X_k, X_r)=1$, and at the same time it exists between X_h and X_r, $\Re(X_h, X_r)=1$, that is sufficient for existence of the unity relation between X_k and X_h, $\Re(X_k, X_h)=1$. Since the unity relation exists among the pixels in an object, we say that ***the unity relation exists between each pixel of an object and the object itself.***

Notice that the unity relation is defined by a property between two individual pixels in an object, but it is extended to the property between a pixel and an object (based on the transitive properties). The unity relation properties are very important tools for object extraction.

Proposition: The unity relation is said to exist between the pixel X_k and the object $\mathbf{P}_i$ if and only if for some X_r belonging to $\mathbf{P}_i$ the relation $\Re(X_r, X_k)=1$ is true.

Proof of the sufficient condition: since X_r belongs to the object $\mathbf{P}_i$, the unity relation exists between X_r and all other pixels from $\mathbf{P}_i$; i.e.,

$$\Re(X_r, X_h)=1 \text{ for all } X_h \in \mathbf{P}_i$$

but $\Re(X_r, X_k)=1$, and from the unity relation transitive property, this implies that

$$\Re(X_k, X_h)=1 \text{ for all } X_h \in \mathbf{P}_i$$

which it is equivalent to saying that the unity relation exists between the pixel X_k and the object $\mathbf{P}_i$; thus $X_k \in \mathbf{P}_i$.

Proof of necessity condition: if the unity relation exists between X_k and the object $\mathbf{P}_i$, we should have

$$\Re(X_k, X_h)=1 \text{ for all } X_h \in \mathbf{P}_i$$

But the set $\mathbf{P}_i$ is a finite closed set, and since $X_r \in \mathbf{P}_i$, it is a necessary condition to have

$$\Re(X_k, X_r)=1$$

We had pointed out that, the unity relation in the observation space is defined by an adjacency relationship together with a similarity criterion among the pixels' attributes. The similarity between the pixels' attributes is of basic importance in attempting to test the existence of the unity relation. This is evident since the existence of two adjacent objects is a consequence of the dissimilarity of features from neighboring pixels where two adjacent objects differ in at least one of the spectral or contextual features. But the accuracy of the similarity measure is dependent on the selected metric space used for functional construction and has an upper bound which is controlled by the amount of noise in the system. The uncertainty in the similarity measure is significantly reduced using the within object regularities (contextual feature). This property is used in the path-hypothesis for unity relation construction. In the next section the main concept of similarity measure based on path-hypothesis is presented.

2.7 Similarity Measure and Path Hypothesis

The principles for evaluation of the unity relation are:

a) the measure of similarity

b) membership in the same neighboring set, characterized by a single concept (adjacency relation).

The traditional principle for comparing features utilizes some measure of pattern similarity, which is the reciprocal of a distance measure.

To define a distance, we need a functional which maps all pairs of elements from the set of features into the real line:

$$\partial : \{ X , Y \} \rightarrow \mathbf{R}^{+}$$

Such a functional is called a metric if it possesses the following properties:

Positive definite:

$$\partial(X,Y)>0$$

$$\partial(X,Y)=0 \Leftrightarrow X=Y$$

Symmetric:

$$\partial(X,Y)=\partial(Y,X)$$

Triangle inequality:

$$\partial(X,Z) \leq \partial(X,Y) + \partial(Y,Z)$$

These requirements are merely formalized statements reflecting the intuitive properties of a distance. The set of features **P** together with functional $\partial(.)$ is called a metric space and is represented by a two-tuple $(\mathbf{P}, \partial)$.

An interesting class of metric spaces [38], which is called **Minkowsky Spaces** or $\mathbf{L}^p$ spaces, is obtain when a metric $\partial(.)$ is defined in a $\mathbf{R}^d$ space by the following formula

$$\partial_p(X,Y) = \left[\sum_{k=1}^{d} (x_k - y_k)^p \right]^{\frac{1}{p}}$$

Another well-known metric space is called **Chebyshev Space**, which is defined in a $\mathbf{R}^d$ space by the following equation

$$\partial_c(X,Y) = \operatorname*{Sup}_{k=1}^{d} \{ | x_k - y_k | \}$$

One can show that

$$\partial_c(X,Y) = \lim_{p \to \infty} \partial_p(X,Y)$$

This shows that the Chebyshev space is a special case in the Minkowsky spaces (∂_∞). The metric ∂_1 is called Diamond space, ∂_2 is called Euclidean space or sphere space, and ∂_∞ is called a square space (Fig.2.6).

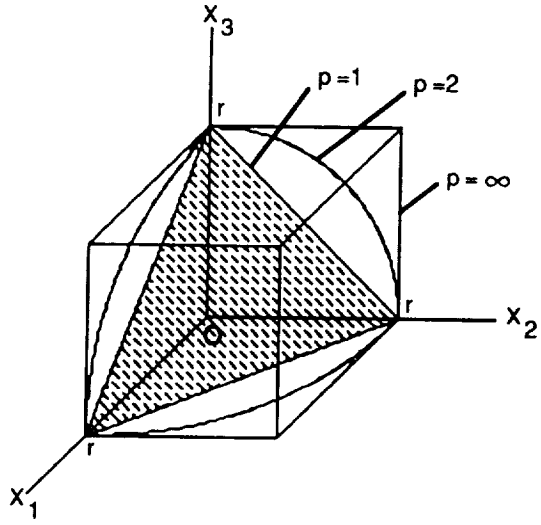


Fig. 2.6 Equidistant surface $\partial(O,X)=r$ in the Minkowsky spaces (d=3)

In view of various definitions of metric spaces the question arises as to which metric space can best explain the similarity and differences. For example, let X be an unknown pixel and Y_1 ,Y_2 , Y_3 be spectral feature-vectors of three different neighboring objects. Figure 2.7 illustrates these feature-vectors in a $\mathbf{R}^d$ space (d=32) where the vertical axis shows the value of each elements, x_n, in the feature vector, and the horizontal axis shows the position of that particular element in the feature vector.

$$X=[x_1, x_2, \ldots, x_n, \ldots, x_{32}]^t$$

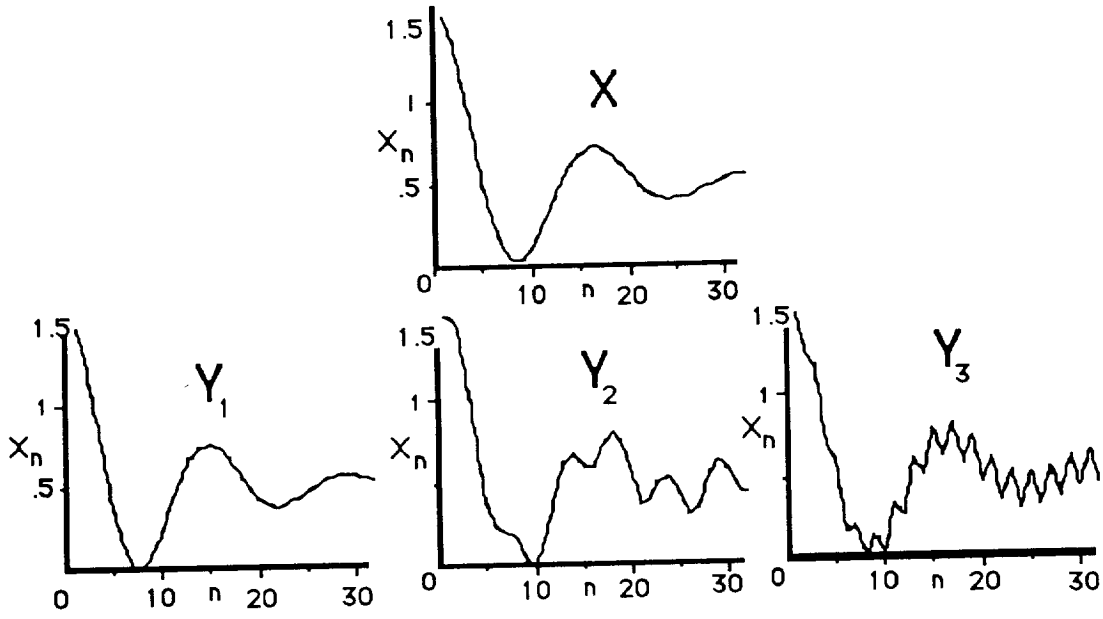


Fig. 2.7 Which of Y_1, Y_2, Y_3 is more similar to X ?

Let the decision rule for unity relation be the minimum distance criterion between the spectral features (similarity measure). Then, table 2.1 gives different answers for the existence of the unity relation among X and Y_1 ,Y_2 , Y_3, where in the Diamond space Y_1, in the Euclidean space Y_2, and in the Chebyshev space Y_3 is selected to have a unity relation with X.

Table 2.1 Distance between features

Distance in the Measure Space			
X from	Diamond	Euclidean	Chebyshev
$\partial_p(X,Y_1)$	**2.56**	0.54	0.18
$\partial_p(X,Y_2)$	2.62	**0.52**	0.13
$\partial_p(X,Y_3)$	3.16	0.56	**0.10**

There is not any analytical approach to find the optimum metric space for similarity measure in the general sense for multispectral image processing; however, the superiority of the Diamond metric over the other metric spaces can be determined experimentally [39,41]; also, notice that the following property always holds for any pair of features (X,Y) in a $\mathbf{R}^d$ space:

$$\partial_c(X,Y) = \partial_\infty(X,Y) \leq \partial_2(X,Y) \leq \partial_1(X,Y)$$

Thus in this work we restrict the consideration to the metrics that measure the similarity distance in the Diamond space.

In order to characterize the unity relation among the pixels, consider pixel X_k and a neighbor, X_r. If they are sufficiently similar according to some suitable measure, then X_k and X_r should be merged to a single simple object $\mathbf{P}_i$; otherwise they should be assigned to different objects. To implement this scheme the definition of a suitable functional for measuring the distance is necessary. Assume this functional exists and is defined by $\partial_s(.)$, such that if

$$\partial_s(X_k, X_r)<1$$

we say that X_k is similar to X_r, then the unity relation exists between them, which means X_k and X_r carry common characteristic information about the scene, and locally belong to the same object $\mathbf{P}_i$.

One disadvantage of the above approach is that it is quite local. There is always uncertainty in applying a similarity measure to two adjacent pixels (uncertainty is explained in appendix A at the end of this thesis). It is logical that employing more pixels as evidence would tend to reduce this uncertainty in distance measurement. On the other hand, increasing the amount of evidence increases the complexity of the unsupervised process and possibly the error of the feature extraction procedures [19,36].

It would seem that the optimum approach for designing the functional $\partial_s(.)$ is that it should be based on the totality of pixel features in that object, $\mathbf{P}_i$. Since $\mathbf{P}_i$ is not pre-determined and the features of $\mathbf{P}_i$ are unknown, a hypothesis we shall refer to as the path hypothesis will be useful for **on-line** functional assignment.

Path Hypothesis: As pixels from an object are sequentially received, the object accumulates in a fashion such that the object is increasingly well defined. The path of sequential association which the pixels follow in the spectral space form a continually evolving hypothesis regarding the object definition. Segments in this path are determined on a spectral basis relative to the then current status of all other adjacent objects by the spectral variation between two consecutive points in the path using a specific metric to be defined presently. Segments in the path are also determined based upon the spectral separation between the current and the most recently preceding pixel of that object in spatial space, thus incorporating both spectral and spatial information in the association of pixels with objects.

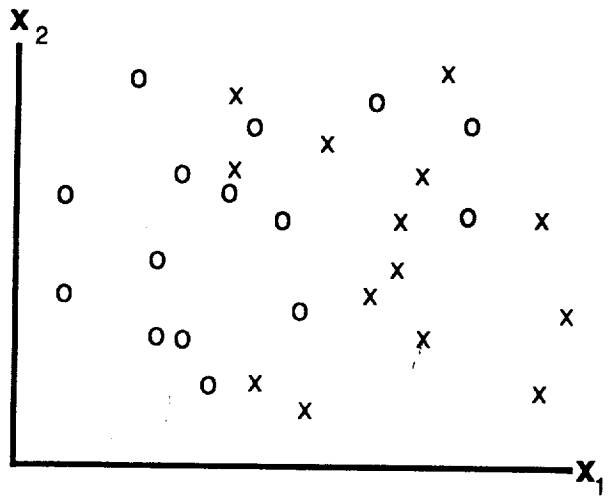


Fig. 2.8. Two adjacent objects in the observation space

Notice that the sequential nature of this hypothesis is significant. For example, consider the hypothetical distribution of the points of two object in

spectral space as shown in Fig.2.8. The path by which they might have accumulated is illustrated in Fig.2.9 and is such as to explain how such an overlapping class structure is feasible.

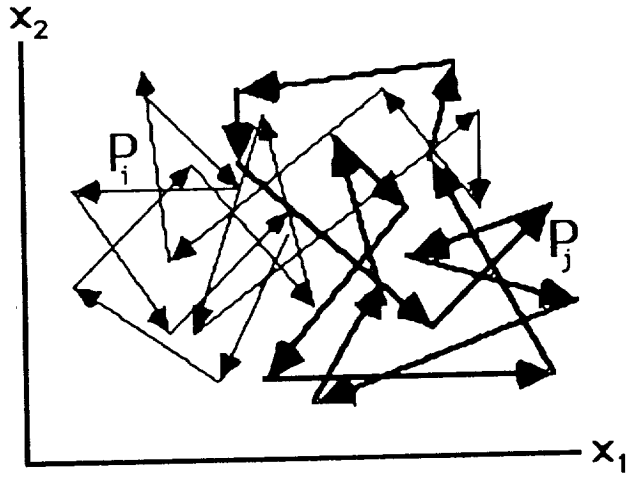


Fig. 2.9. Two objects are represented by the corresponding path-segments

Notice theoretically, there is only one specific path-segment, $\mathbf{P}_i$, in the spectral space defined by an object, and the path sequence cannot pass through another portion of the observation space, Fig.2.9. The separation of the two objects based upon the paths are made possible by the spatial dependency in the path definition.

$$\mathbf{P}_i \cap \mathbf{P}_j = \emptyset \text{ for all } i \neq j$$

In other words, the existence of a partition that completely separates objects is substantially increased by the path-hypothesis; and also, non-adjacent

objects which belong to the same class, but are spatially separated by the other objects, have different path-segments and even might be nearly identical in the observation space.

It should be realized that the path-segment $\mathbf{P}_i$ is defined in the spectral space and it is different from a spatial path in the scene. A path-segment $\mathbf{P}_i$ is represented by its spectral-feature S_i, spectral variation regularity V_i, and the path end point X_{ki-1}.

The path hypothesis thus determines a possible sequence of points in the observation space for each object, which implies that each object forms a well-defined sequence in observation space, called the **path-segment**. The succession of consecutive observations describes a particular trajectory in the observation space. Any change in the behavior of two consecutive points (the end point of the path-segment X_{ki-1} and the current pixel X_k) in this trajectory can define a start point of a new path-segment. i.e. the detection of a new object follows that of the detection of the end of a path-segment ($X_{ki}=\emptyset$).

The sequential nature of this method and its on-line implementation cannot wait for the complete trajectory for decision making about a small path-segment. We thus propose to determine the apparition of a new path-segment (new object) by $n_w/2$ past observations and $n_w/2$ following observations, which these observations supposed to represent the initial intra-object rule. Then the process will continue based on the intra-object rule until a new object be detected.

In this work a functional for testing the unity relationship between the pixel-feature X_r and object-feature Y_i based on the Ghassemi-distance in the Diamond space is introduced. Where *the Ghassemi-distance is a metric that normalize the spectral distance by their spectral gradient vector.*

$$\partial_g (Y_i, X_r) = (| S_i - X_r |)^T (\alpha V_i + \beta V_n)^{-1}$$

$$\text{where} \quad \alpha = \frac{d\, n_i}{n_i + d} \quad \text{and} \quad \beta = \frac{d^2}{n_i + d}$$

V_i , S_i , and n_i are the same parameters as defined in section 2.5. As previously stated when the number of pixels within an object is not sufficient, we incorporate V_n into the normalization factors. Where V_n is the spectral variation-vector built on $n_w/2$ past observations and $n_w/2$ following observations (see also chapter three). The inverse of a vector, V^{-1}, is defined by the following relationship between their corresponding elements:

$$V^{-1} = [\, v_1^{-1}, v_2^{-1}, \ldots, v_d^{-1} \,]^t$$

Notice the functional is normalized and the distance is compared with unity, i.e. this functional maps the pair of object-feature and pixel-feature into two disjoint exclusive regions.

$$\partial_g(.) : F \times P \rightarrow [\, 0, 1 \,) \cup [\, 1, \infty \,)$$

The first region is the interval [0, 1) with the property that any pair that have been mapped into this interval satisfies the similarity criterion. The second region is [0, ∞) that indicates the unity relation test is failed.

2.8 Object-Feature Extraction

In this work we restrict consideration to a case in which all pixels making up an object have quantitative and conceptual similar attributes. It is assumed that two adjacent objects differ in a measurable way relative to the spectral or contextual features, based on the Ghassemi-distance. Each object is represented by a set of three parametric primitives (S_i, V_i , $\mathbf{L}_i$), which are explained in section 2.5 in this chapter.

Based on the path-hypothesis, the data read sequentially into the system. A present pixel is compare with its adjacent objects (path-segments in the observation space) for the unity relation test. If the distance between pixel-feature X_k and object-feature Y_i is less than unity , $\partial_g(Y_i, X_k)<1$, then this pixel will be annexed to this object, and the object-feature will be updated. This comparison will be done for other possible adjacent objects, and if there is another object which has a unity relation with the pixel, then this object and the former one will be merged. If there is no adjacent object to have the unity relation with the pixel, then this pixel will be a singular object. This procedure is illustrated in chapter three.

Let X_r be the feature of an unknown pixel $p_{[i,j]}$ which has a neighboring set $\mathbf{A}_i$. And let Y_i be the object-feature estimated from $\mathbf{P}_i$. If X_r has the unity relation with the path-segment $\mathbf{P}_i$:

$$\partial_g(Y_i, X_r) < 1 \quad \text{and} \quad r \in \mathbf{A}_i \iff \Re(\mathbf{P}_i , X_r) = 1$$

this implies that X_r belongs to the object $\mathbf{P}_i$; hence, the index k should annex to $\mathbf{L}_i$; $\mathbf{P}_i$, and X_r can be represented by the same feature Y_i. In practice the

object $\mathbf{P}_i$ and the pixel X_r are merge to $\mathbf{P}_i$ in the feature space simultaneously with the object detection process.

The pixel annexation process, $\mathbf{P}_i = \mathbf{P}_i \cup X_r$, in the observation space is equivalent with the object-feature updating. As it had been pointed out in the section 2.5 in this chapter, an object is represented by a set of three parametric primitives $Y_i = (S_i, V_i, \mathbf{L}_i)$. Thus, the object-feature updating can be performed by the following computations:

$$S_i = \frac{n_i S_i + X_i}{n_i + 1}$$

$$V_i = \frac{n_i V_i + | X_k - X_{k+1} |}{n_i + 1}$$

$$\mathbf{L}_i = \mathbf{L}_i \cup k$$

After object-feature extraction, the next step is to measure the feature reliability in feature space (these two steps occurring simultaneously in this case). This process tests the object validity and fusion (merging) tendency of the object in feature space. The object may be merged consistent with the objects' features in feature space.

For a fusion tendency test, the transitive property of the unity relation is used. Two objects in the feature space can be merged if they have the unity relation.

Lemma: If a single pixel has a unity relation with two different objects simultaneously, then both objects have a unity relation with

each other, they have a fusion tendency, and they should merge to a single object.

Proof: Let X_r be the pixel that has the unity relation with the objects $\mathbf{P}_i$ and $\mathbf{P}_j$ simultaneously. Then, the following relations hold simultaneously:

$$\Re(X_r, X_k)=1 \text{ for all } X_k \in \mathbf{P}_i \text{ ; and } \Re(X_r, X_h)=1 \text{ for all } X_h \in \mathbf{P}_j$$

by the unity relation transitive property. This implies that

$$\Re(X_r, X_h)=1 \text{ for all } X_h \in \mathbf{P}_i \mathbf{U} \mathbf{P}_j$$

This implies that X_r belongs to both objects $\mathbf{P}_i$ and $\mathbf{P}_j$ simultaneously, or belong to a new object, $\mathbf{P}_l$, which is the union of the objects $\mathbf{P}_i$ and $\mathbf{P}_j$ including the pixel X_r itself

$$\mathbf{P}_l=\mathbf{P}_i \mathbf{U} \mathbf{P}_j \mathbf{U} X_r$$

In practice the objects $\mathbf{P}_i$ and $\mathbf{P}_j$ are merged to $\mathbf{P}_l$ in the feature space simultaneously with the object detection process, where it is equivalent with object-feature updating in the feature space. Where object-feature updating are performed by the following computations (see section 2.4):

$$\mathbf{L}_l=\mathbf{L}_i \mathbf{U} \mathbf{L}_j \mathbf{U}\, r$$

$$S_l = \frac{n_i S_i + n_j S_j + X_r}{n_i + n_j + 1}$$

$$V_l = \frac{n_i V_i + n_j V_j + | X_r - X_{r-1} |}{n_i + n_j + 1}$$

Since during the object detection the information of the pixel-features is compacted into the object-features, the fusion tendency test depends only on the object-features; e.g., Y_i and Y_j. Thus this operation is very fast and is equivalent with the processing of a single pixel, as measured by the computation time required.

2.9 Summary

A scene can be represented by features of spatially disjoint objects, rather than the pixels' features. The pixels of an object have a unity relationship, which can be realized by a path hypothesis by means of an efficient functional $\partial_g(.)$. This functional tests the unity relation between pixels. Based on such a scheme, an object is realized as a path-segment in the observation space $\mathbf{P}_i$, where the path's elements, $\mathbf{P}_i=\{X_{1i},X_{2i}, \ldots X_{ni}\}$, carry equivalent useful information about the scene. The sequential nature of the method let the process implemented on-line. Each path-segment is represented by a relevant object-feature set $Y_i=\Psi(\mathbf{P}_i)$ in the feature-space. Finally, the set of object-features $\mathbf{F}=\{Y_1, Y_2, \ldots, Y_{no}\}$, rather than pixel-features set $\mathbf{P}=\mathbf{P}_1 U \mathbf{P}_2 U \ldots U \mathbf{P}_{no}$, is used for data transmission and for the classification process as well. The next chapter explains the object detection and feature extraction algorithm.

Let $\mathbf{P}\cong\mathbf{F}$ be the basic relation between observation space and feature space. Then $\mathbf{P}_i\cong Y_i$ implies that: Y_i represent equivalent useful information about the scene to that the object $\mathbf{P}_i$. Once each object-feature, Y_i , is classified, the

memberships of pixels which belong to the corresponding object are determined simultaneously, regardless of the pixels' location in the scene.

The spectral information of surrounding pixels is correlated with the corresponding pixel. This dependency appears in the form of contextual features that are incorporated in the object feature extraction. Because both contextual and spectral features are used in the feature extraction process, one might expect classification accuracy to be higher in object-feature classification using the object-feature rather than pixel by pixel classification, using the pixel-feature. In other words, it is more likely that the classification of ground cover fields in the feature space based on the object-feature can be more accurate and efficient than the pixel by pixel classification in the original pixel-features. Since the classification process is performed in the feature space rather than in the observation space, the algorithm has the potential to be faster than a conventional one. This results from the fact that the size of the object-feature-set is much smaller than the size of the pixel-feature-set. The performance of object-features is presented in chapter four.

CHAPTER 3
ON-LINE UNSUPERVISED OBJECT-FEATURE EXTRACTION ALGORITHM

3.1 Introduction

As has been pointed out, the reduction in complexity and costs associated with the analysis of multispectral image data, data transmission, storage, archival and distribution is an important task of on-line unsupervised object-feature extraction. The ambiguity in the object detection process can be reduced if the spatial dependencies, which exist among the adjacent pixels, are intelligently incorporated into the decision making process.

AMICA uses unity relation for object detection. In this work, the unity relation among the pixels of an object is constructed with regard to the: adjacency relation, spectral-feature and spatial-feature characteristics in an object; i.e. AMICA uses the within object pixel-feature gradient vector as a valuable contextual information to construct the object's features, which preserve the class separability information within the data.

Based on the path-hypothesis the data read sequentially into the system and the unity relation between a current pixel and the path-segments (objects in the observation space) will be examined. This pixel may then be merged into

an appropriate object or it may initiate a new object. As explained in section 2.5, an object will be represented by the object-feature set.

Typically the performance of an object-feature extraction algorithm is measured by the performance of the object-features with respect to the pixel-features, as well as by its implementation complexity. The complexity of object-feature extraction algorithm is a particularly important consideration in the hardware implementation and required computation time. The performance of object-feature is presented in the chapter four.

In this chapter we focus on the reduction in complexity of AMICA with no degradation in the accuracy of the process. This is a challenging and sophisticated task especially when it is to be implemented on-line.

AMICA consists of four distinct activities, Fig.3.1:

1- functional estimation
2- unity relation test
3- feature extraction and feature reliability test
4- feature transmission

All of these activities can be performed and implemented by AMICA as explained in the previous chapter. In this chapter we try to optimize performance of the AMICA, in the sense of CPU time and memory requirements for implementing the feature extraction process. This can be achieved by optimization of the number of past and following observations, n_w, for initiation of a new object, as explained in section 2.7, to minimize the required memory. Also, it can be achieved by redundancy reduction in the

number of unity relation and feature reliability tests, to minimize the computation time for decision making.

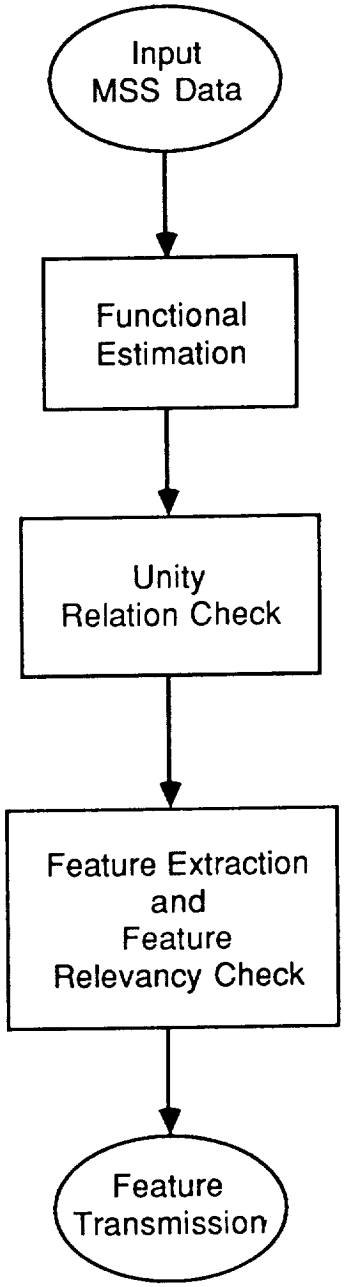


Fig. 3.1. Real-time object-feature extraction activities

In the next section we introduce the necessity of an assignment of a window for removal of the feature extraction dependence on the scanning direction which may occur in the case of initiation of a new object. Then section three will explain how one can reduce the number of operations per node for the unity relation and feature reliability test, by using the unity relation properties. Finally, section four explains the algorithm procedure and the flow chart of an optimized AMICA.

3.2 Optimal Window Size Assignment

To apply the unity relationship test to a pixel we must first estimate the within-object spatial parameters, V_i. The estimated parameters are used to generate a normalized metric (functional) for evaluation of unity relation within each object. It is assumed that the object is large enough to obtain a reliable estimate of the model parameters. The object is assumed "large" with respect to the "d" dimensionality. To estimate the parameters at each point we proceed as follows. Suppose the spatial parameters evolve slowly over each object. As pointed out in the previous chapter, because of the sequential nature of AMICA and its on-line implementation the system cannot wait for the complete trajectory for decision making about a small path-segment. Thus, generally there are two cases that should be considered for functional assignment and feature selection (see section 2.5):

1) In the first case, the number of pixels within the object is sufficient for contextual feature extraction. In this case the object will be represented

by both spectral and contextual features, and the dominant factor to normalize the functional is the V_i extracted within the object $\mathbf{P}_i$.

$$V_i = \frac{1}{n_i - 1} \sum_{k \in L_i} \nabla_\delta X_k$$

2) In the second case, the object is small, or the number of pixels within the object is not sufficient for contextual feature extraction. In this case the object will be represented only by the spectral features, and we determine the validity of a new path-segment (new object) by $n_w/2$ past observations and $n_w/2$ following observations accomplished within a window. These observations are presumed to represent the initial intra-object characteristic. In this case the dominant factor to normalize the functional is the V_n extracted within an appropriate window.

$$V_n = \frac{1}{n_w - 1} \sum_{l = \frac{-n_w}{2}}^{\frac{n_w}{2}} \nabla_\delta X_{k-l}$$

In this work the spatial direction δ is defined to be in all of three horizontal h, vertical v, and diagonal vh spatial directions (see appendix C). In the second case there is a potential problem with the feature extraction (if V_n is not incorporated into the functional); that is, the object-feature inherent dependence on the order in which pixels and the small objects (if there is some) are examined for unity relation. Figure 3.2 illustrates eight of the possible ways that data can be read into the system.

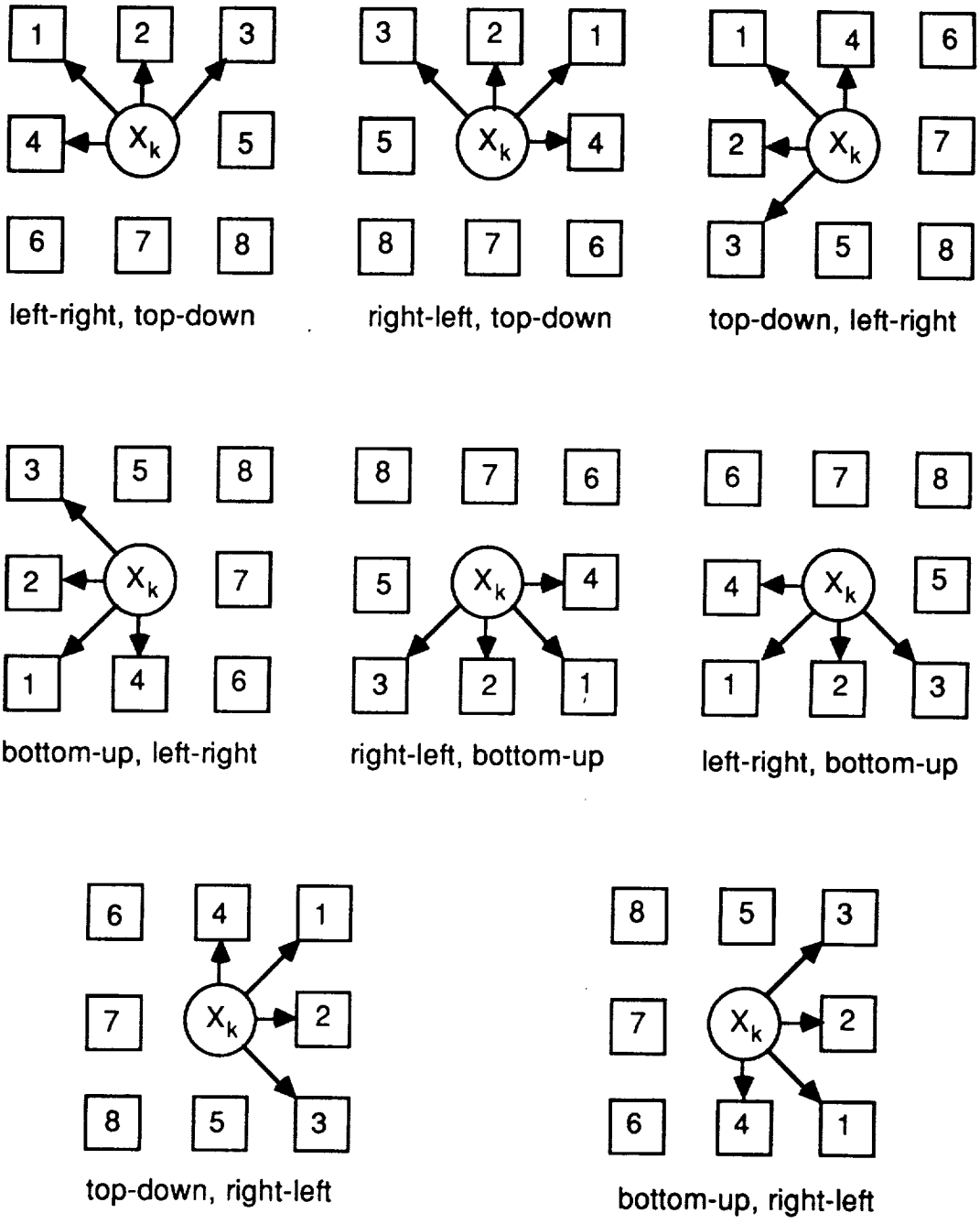


Fig. 3.2. Eight possible scanning directions for reading data into the system

Thus there are eight approaches that a pixel can be examined with the adjacent small objects (if there is some) sequentially. For example, a left-right, top-down scan may not yield the same initial objects extraction as a right-left, bottom-up scan does. In other words, on-line object detection outcome is dependent on scanning direction when the objects are small.

The unity relation based on the path-hypothesis provides that the scanning direction dependency can be minimized by selection of an efficient window size, n_w, and appropriate position for the window, Fig. 3.3. This dependency can be significantly reduced by measuring the feature reliability among the adjacent objects.

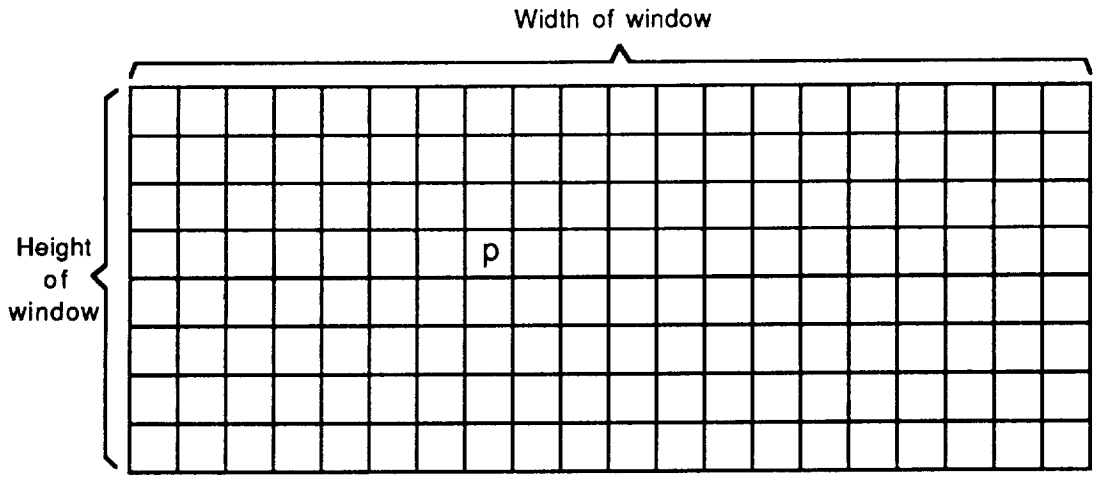


Fig. 3.3. Position of the pixel in the window for functional estimation

Now, the detection of a new object is built upon the $n_w/2$ past observations and $n_w/2$ following observations. In this way the scanning direction dependency is minimized; objects will be grown until the number of pixels within an object is sufficient for contextual feature estimation. Then the local spectral gradient will be the important factor for functional assignment. As it is shown in chapter two, in practice, object growth will be accomplished by adjustment of the coefficients α and β in the functional.

$$\partial_g (Y_i, X_r) = (| S_i - X_r |)^T (\alpha V_i + \beta V_n)^{-1}$$

The memory requirement and CPU time for functional assignment are proportional to the size of window, n_w. Since the data is read into the system line by line, we try to minimize the number of lines per window to optimize the CPU time and storage for functional normalization. On the other hand, reduction in the number of lines in the window, n_l, increases scanning direction dependence, which reduces the quality and performance of the object-features.

In an experiment several types of multispectral image data (see chapter four) are used. As a criterion for evaluation of the effect of the window size, the performance of the object-features versus the original pixel-features is presented. Figure 3.4 illustrates the effect of the window size on performance of the object-features where the performance is represented by the percentage of correct feature classification.

In this figure the width of the window is fixed and is equal to the number of pixels in each scan-line, n_x, thus the height of window, n_l (number of lines in the window), is used as the reference for window size.

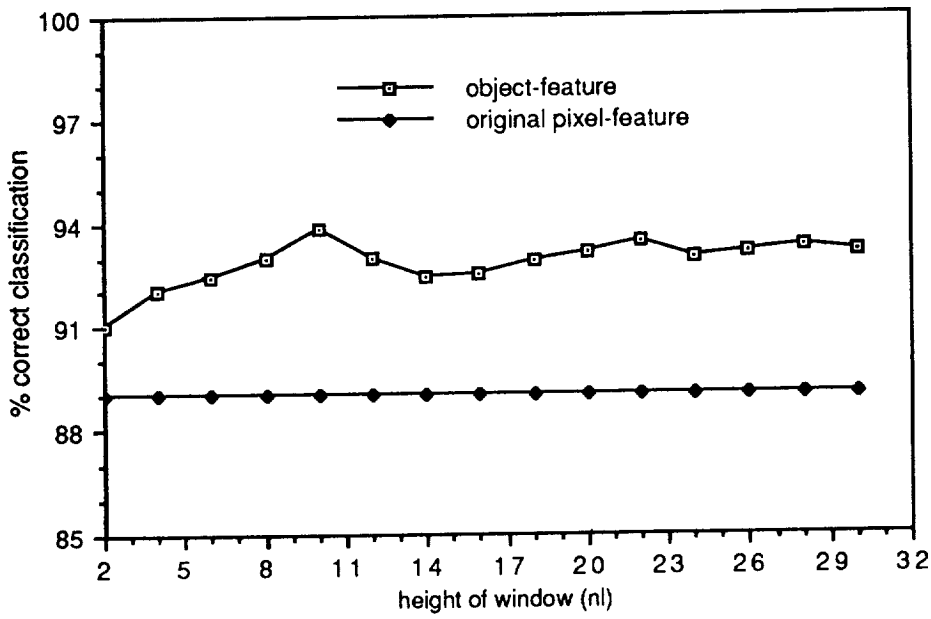


Fig. 3.4. Effect of window size on the feature performance

The results of this experiment show that, for the data set used, if the width of the window be equal to n_x or greater than 64, practically, the optimum height for the window is eight or ten lines per window.

3.3 Reducing the Unity Relation and Feature Reliability Tests

The multispectral image data are read into the system sequentially in the raster scan oriented format, where each pixel can have at most eight adjacent neighboring objects. Let p be a pixel under consideration and O_{nw}, O_{ne}, O_n, ..., O_s, O_{se} be the eight adjacent neighbors of the pixel p, Fig.3.5. If the the unity relation exists between the pixel and any of those objects, the pixel-feature will be combined (feature fusion) with the corresponding object-feature, and the pixel location will be annexed to the spatial-feature-map.

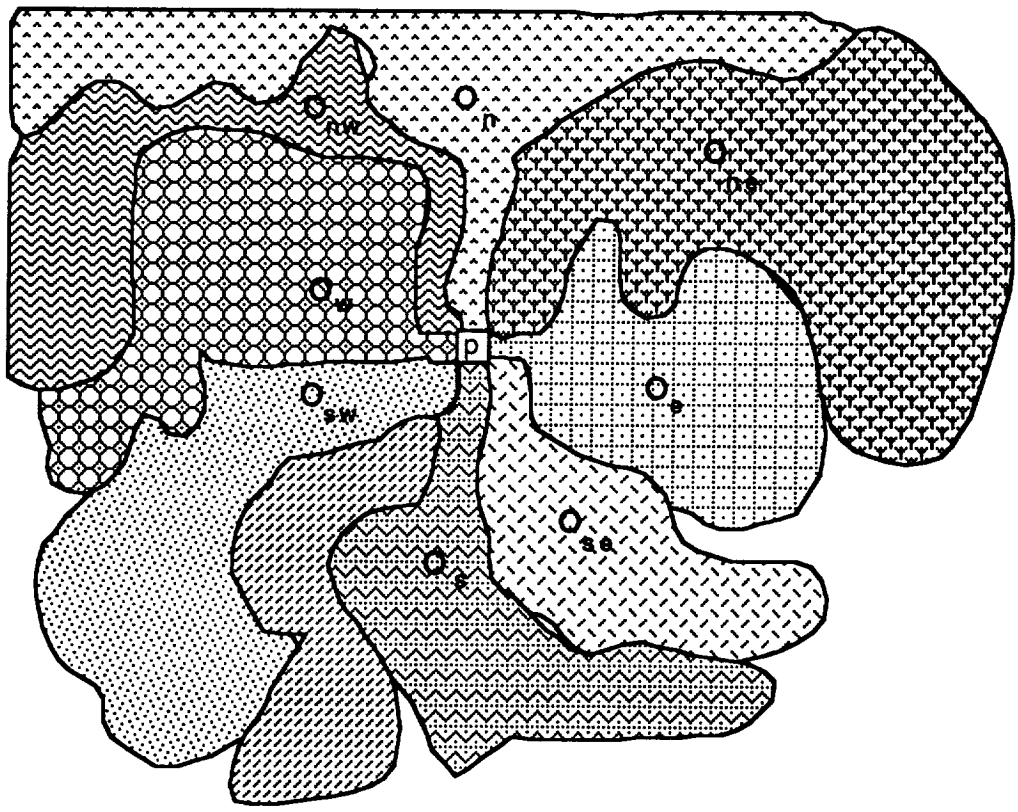


Fig. 3.5. Eight adjacent neighboring objects

As it is explained in chapter two, pixel p is represented numerically by the vector X_k, and its eight adjacent neighbors O_{nw} , O_n , O_{ne}, ..., O_{se} can be represented by corresponding path-segments $\mathbf{P}_{k1}$, $\mathbf{P}_{k2}$, $\mathbf{P}_{k3}$,... , $\mathbf{P}_{k8}$, or by their features Y_{k1} , Y_{k2} , Y_{k3} ,... , Y_{k8} respectively. Notice as pointed out in section 2.7, in practice a path-segment $\mathbf{P}_{ki}$ is represented only by its spectral feature S_{kl} , spatial feature V_{kl}, and its end point X_{k-1}. AMICA tests for the existence of the unity relation between X_k and its eight neighbors. Simultaneously the system measures the feature reliability of the eight neighboring objects and tests for shrinkage tendency and for all possible combinations between the eight neighboring objects' features. In other words, given a pixel X_k in the observation space, regardless of the existence of the unity relation, AMICA should test for feature reliability and any other possibility of existence of the unity relation among X_k and $\mathbf{P}_{k1}$, $\mathbf{P}_{k2}$, $\mathbf{P}_{k3}$,... , $\mathbf{P}_{k8}$. For example, if the unity relation exists between X_k and $\mathbf{P}_{k2}$, after annexation of p to O_n in the feature space (connection of X_k to the path-segment $\mathbf{P}_{k2}$) the algorithm should continue for testing the existence of the unity relation and shrinkage tendency between $\mathbf{P}_{k2}$ and other neighboring features and continue measuring the object-feature reliability.

For any incoming pixel observations, X_k, there are $256=2^8$ possible cases in the observation-space that the algorithm should check for unity relation and feature reliability: C_1^8 cases if all of the 8 neighbors belong to 8 different classes, C_2^8 cases if all of the 8 neighbors are from only 7 different classes, C_3^8 cases if all of the 8 neighbors belong to only 6 different classes, C_4^8 cases if all of the 8 neighbors belong to only 5 different classes, C_5^8 cases if all of the 8 neighbors belong to only 4 different classes, C_6^8 cases if all of the

8 neighbors belong to only 3 different classes, C_7^8 cases if all of the 8 neighbors belong to only 2 different classes and C_8^8 case if all of the 8 neighbors are from the class, where:

$$C_n^k = \frac{k!}{n!(k-n)!}$$

The symmetric property of the unity relation provides for AMICA to reduce the number of check points per each pixel by using only the first four adjacent neighboring objects, Fig.3.6, instead of eight adjacent neighboring objects.

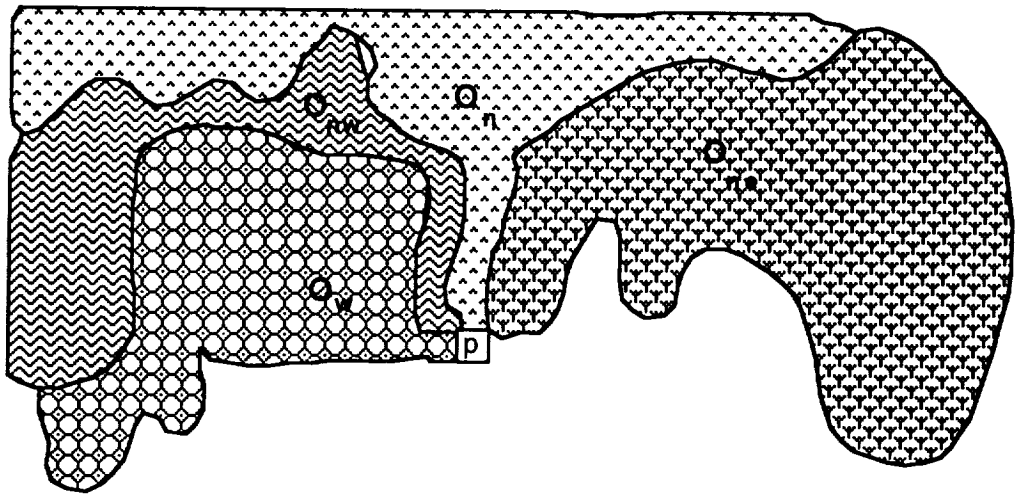


Fig. 3.6. Four adjacent neighboring objects

Proposition 1: Based on the unity relation properties, the results of AMICA, using the first four adjacent neighboring objects, are equivalent to the results

of AMICA, using eight adjacent neighboring objects when the input data is read by of the system sequentially in the raster scan oriented format.

Proof: Assume that AMICA checks the unity relation and feature reliability only among X_k , $\mathbf{P}_{k1}$, $\mathbf{P}_{k2}$, $\mathbf{P}_{k3}$, and $\mathbf{P}_{k4}$, and then starts repeating the same procedure among X_r , $\mathbf{P}_{r1}$, $\mathbf{P}_{r2}$, $\mathbf{P}_{r3}$ and $\mathbf{P}_{r4}$, where X_r is the first East adjacent pixel to X_k. But in this new coordinate, Y_{r1} is the same as Y_{k2} , and Y_{r2} is the same as Y_{k3} , also Y_{r4} is equivalent to X_k in the feature space. This implies that, mathematically, AMICA will check the unity relation and feature reliability for Y_{k5} in the next incoming pixel X_r , and since the unity relation is transitive and symmetric, the outcome will be the same as when X_k and Y_{k5} are considered together for the unity relation and feature reliability test.

The same situation exists for $\mathbf{P}_{k6}$, $\mathbf{P}_{k7}$ and $\mathbf{P}_{k8}$ in the sequence of the next line scan, where the unity relation and feature reliability of at least one pixel from each of these path-segments ($\mathbf{P}_{k6}$, $\mathbf{P}_{k7}$, $\mathbf{P}_{k8}$) with X_k will be checked in the feature space. In other words, since the unity relation is symmetric, mathematically the last four adjacent neighbors of X_k ($\mathbf{P}_{k5}$, $\mathbf{P}_{k6}$, $\mathbf{P}_{k7}$ and $\mathbf{P}_{k8}$) will be considered in the next incoming pixel data and the next sequence of scan line data. This approach for testing the unity relation is much easier to implement because there are fewer objects to compare simultaneously and fewer object-features to be tested for feature reliability.

Using the first four-adjacent neighbors, AMICA should still check $16=2^4$ possible cases per each pixel for existence of the unity relation and feature reliability in the feature-space.

Proposition 2: For any given pixel X_k, the unity relation and feature reliability test is redundant for all of the cases in which two of the neighboring objects have an adjacency relation: $[Y_{k1}, Y_{k2}]$, $[Y_{k1}, Y_{k4}]$, $[Y_{k2}, Y_{k3}]$, $[Y_{k2}, Y_{k4}]$, $[Y_{k1}, Y_{k2}, Y_{k3}]$, $[Y_{k1}, Y_{k2}, Y_{k4}]$, $[Y_{k2}, Y_{k3}, Y_{k4}]$, $[Y_{k1}, Y_{k3}, Y_{k4}]$ and $[Y_{k1}, Y_{k2}, Y_{k3}, Y_{k4}]$. In other words, the only relevant cases that should be considered by AMICA for unity relation and feature reliability test, among X_k and its adjacent neighboring objects, are: $[\mathbf{P}_{k1}]$, $[\mathbf{P}_{k1}, \mathbf{P}_{k3}]$, $[\mathbf{P}_{k2}]$, $[\mathbf{P}_{k3}]$, $[\mathbf{P}_{k3}, \mathbf{P}_{k4}]$ and $[\mathbf{P}_{k4}]$.

Proof: Among the eight neighboring objects of X_k, for any two adjacent objects there exist at least one pixel at their boundary, X_b, that has an adjacency relation with X_k. Using the first four adjacent neighbors, the pixel X_b has been checked for the unity relation sequentially before pixel X_k, and since X_k and X_b are adjacent, the transitive property of the unity relation provides us that checking for the unity relation and feature reliability for those adjacent objects is redundant, where

$$\Re(X_k, X_b)=1 \iff \Re(X_k, \mathbf{P}_i)=1 \text{ for any } b \in \mathbf{L}_i$$

Therefore, the unity relation properties provide that an efficient algorithm for each pixel should check at most the first four neighboring objects, and, as shown in the previous proposition, there are only two possible cases, $[Y_{k1}, Y_{k3}]$ and $[Y_{k3}, Y_{k4}]$, that feature reliability of object-feature need to be checked. These two cases are the same as those for which the shrinkage tendency among four neighboring objects should be checked, $[\mathbf{P}_{k1}, \mathbf{P}_{k3}]$ and $[\mathbf{P}_{k3}, \mathbf{P}_{k4}]$.

Based on path hypothesis and the unity relation properties at each node there is only one of the following cases that two objects can merge together. In the first case X_k has a unity relation with $\mathbf{P}_{k1}$ and $\mathbf{P}_{k3}$ simultaneously. Then both objects should be represented by a single feature. Figure 3.7. illustrates this activity in the feature map. The features will be updated as it is explained in chapter three. The second case for merging two objects occurs when X_k has a unity relation with $\mathbf{P}_{k4}$ and $\mathbf{P}_{k3}$ simultaneously. Then the feature map should be updated, as it is illustrated in the Figure 3.8, and both objects will be represented by the same object-feature.

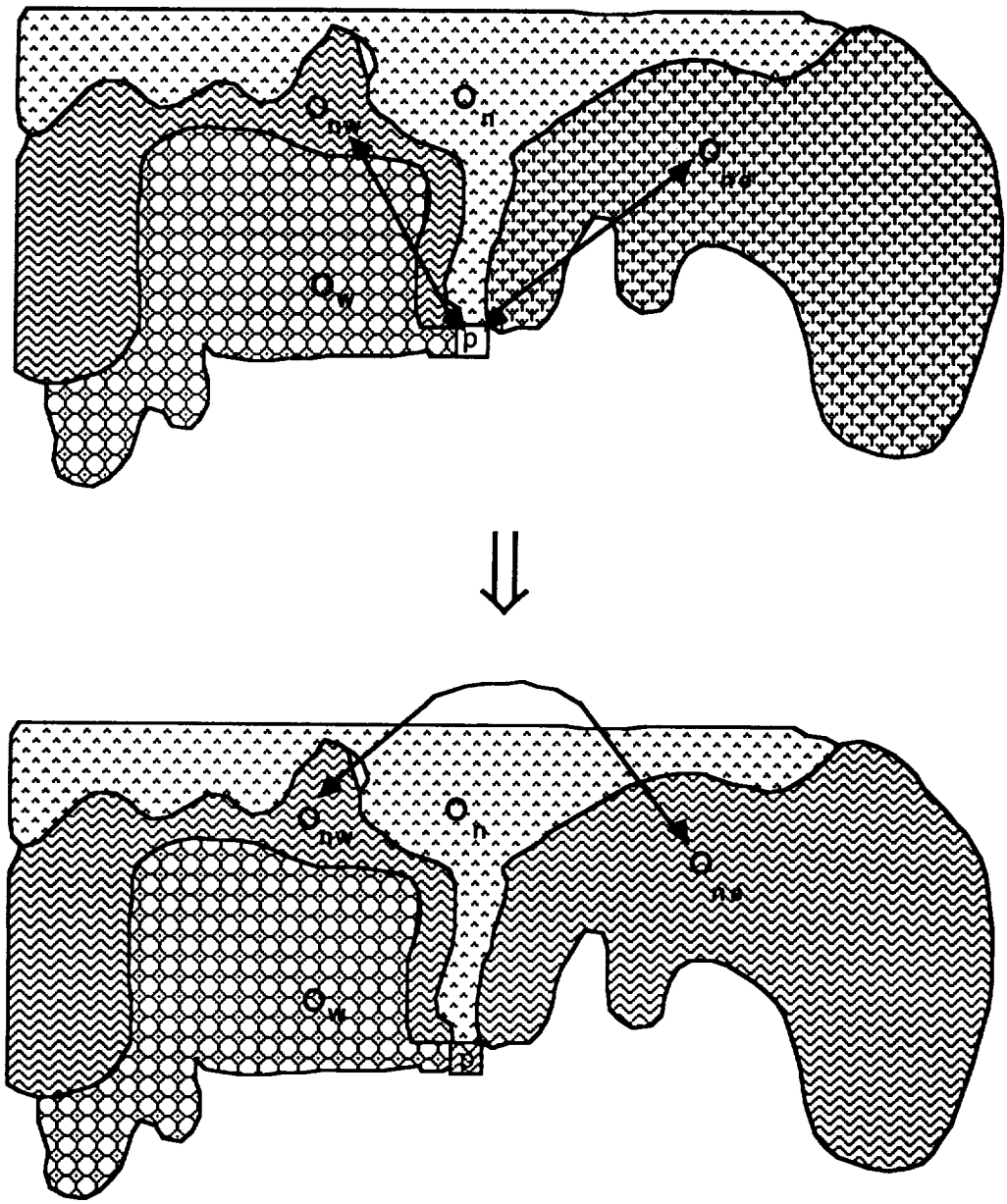


Fig. 3.7. The first possible case for merging two objects in the spatial-feature-map

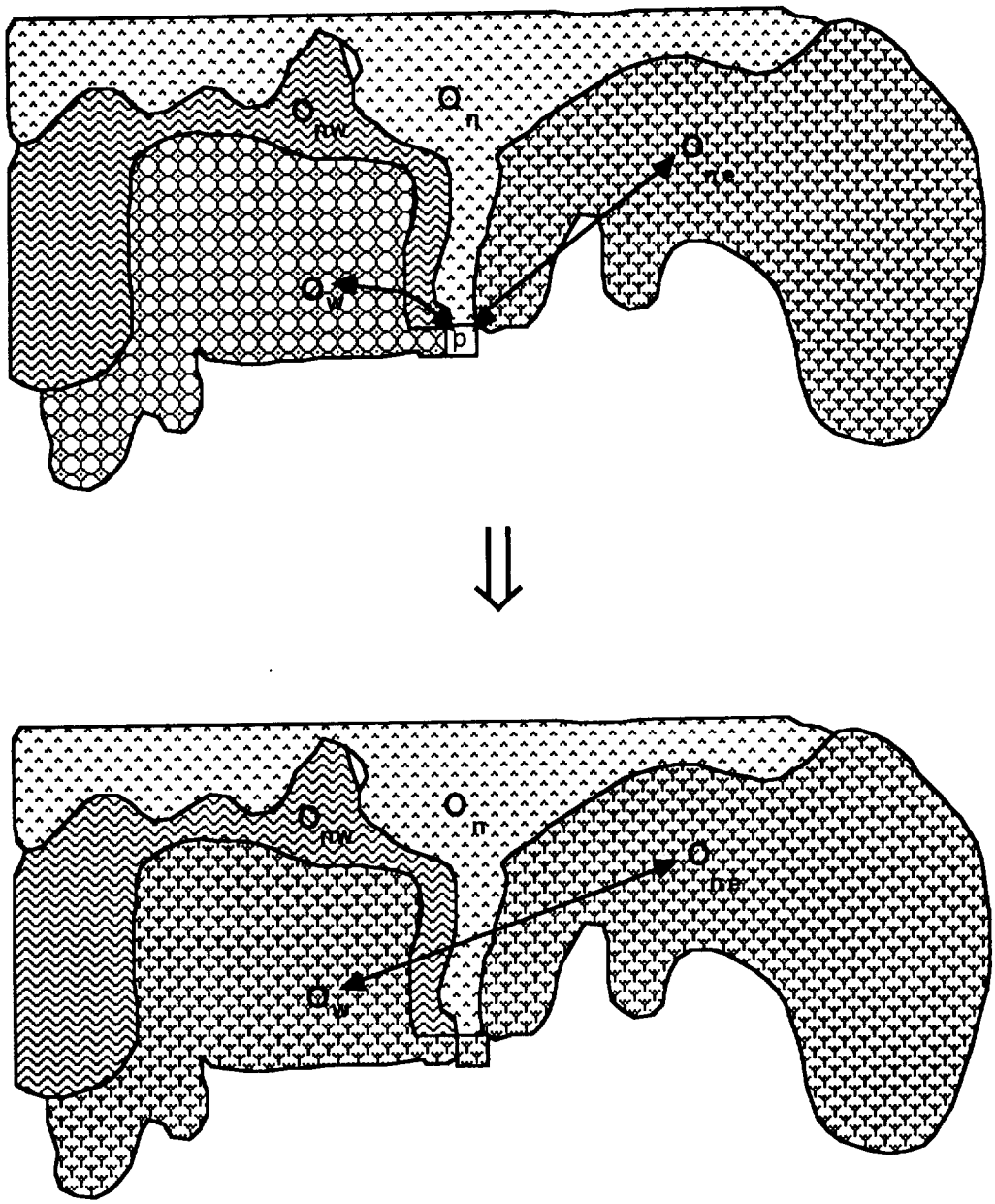


Fig. 3.8. The second possible case for merging two objects in the spatial-feature-map

3.4 AMICA Explanation

The flow chart of AMICA is illustrated in figure 3.9. The multispectral image data is read into the system in the left-right, top-down raster scan format. To begin the process the local pixel-feature gradient is estimated and normalized within the window. After a start period, the window is shifted such that the pixel under consideration is always in the middle line of the window.

An unknown pixel $p_{[i,j]}$ is compared with its four adjacent objects by testing the unity relation between the pixel-feature X_k and corresponding path-segments ($\mathbf{P}_{k1}$, $\mathbf{P}_{k2}$, $\mathbf{P}_{k3}$, $\mathbf{P}_{k4}$). If the unity relation exists then the pixel will be connected to the corresponding path-segment. If more than one path-segment were selected then the corresponding objects will be merged ($\mathbf{P}_i=\mathbf{P}_{k1}U\mathbf{P}_{k3}$ or $\mathbf{P}_i=\mathbf{P}_{k3}U\mathbf{P}_{k4}$). If the unity relation does not exist among any of those four adjacent path-segments compared with X_k, then X_k will be initialized as a new path-segment, $\mathbf{P}_i=X_k$. After each decision making and feature reliability check, the features of the corresponding object (Y_i) will be updated.

The boundary of object (in the spatial-feature-map) is checked for closeness, and then the feature of the closed object will be transmitted to the earth-station. At each node, for unity relation and feature reliability test, the system needs only the information about the adjacent objects, and the coefficients of the functional are necessary only for the open objects. This implies that the compaction system required only information about the open objects in the last two lines.

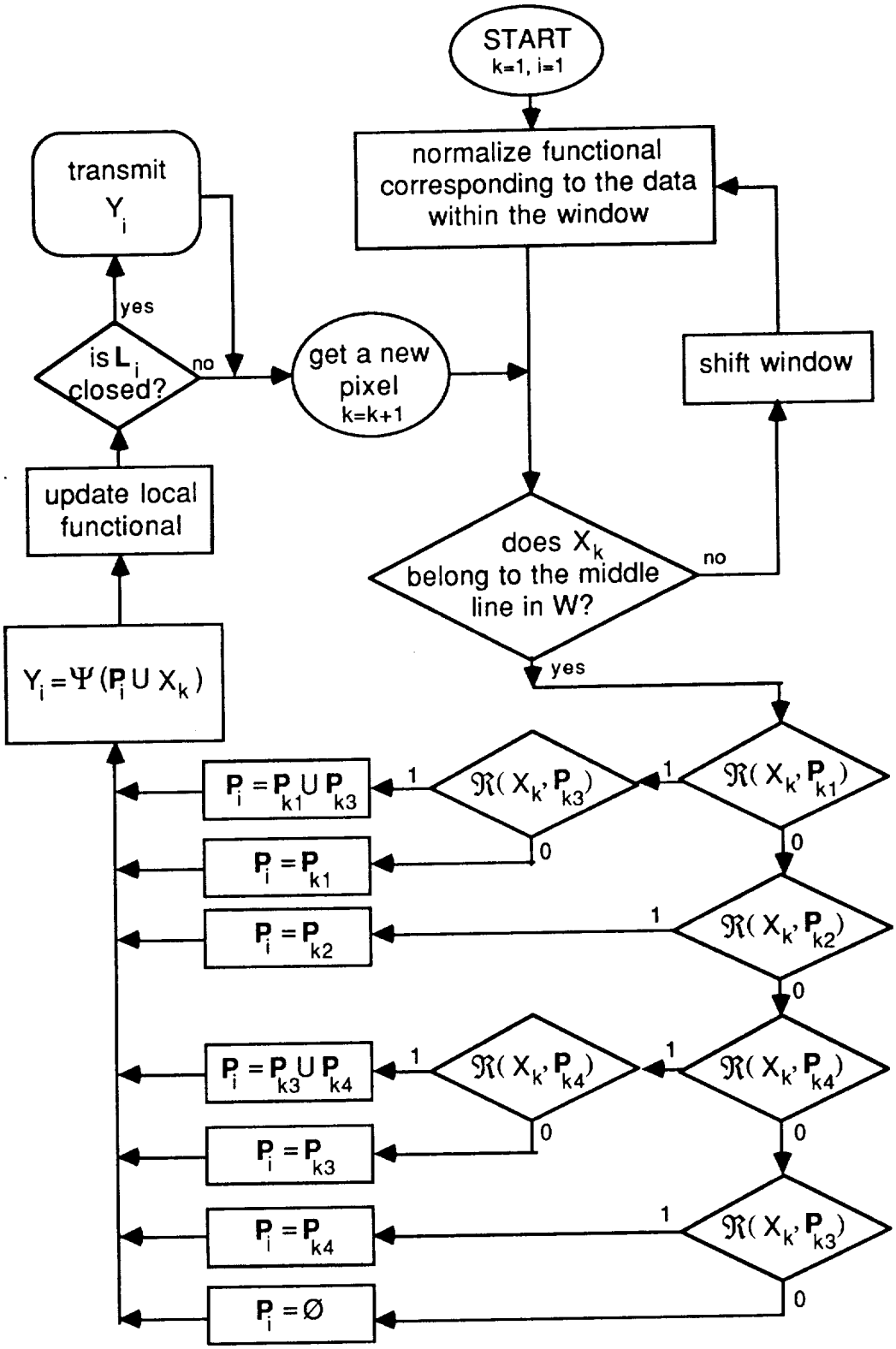


Fig. 3.9. Flow chart of AMICA

Thus, the system should store at most only $2n_x$ object-features. When the object is closed and its feature transmitted, the corresponding buffers for functional assignment will be reset and ready for a new coefficients assignment. In chapter four, results of implementation of AMICA to the real data are presented.

In summary AMICA consists of four main activities:

1- **functional assignment**: the coefficients for functional normalization are measured statistically within the window **W**, where X_k belongs to the middle line in this window (see page 51 and 64).

2- **unity relation check**: Given any pixel X_k the unity relation between X_k and all four adjacent neighboring objects, $\mathbf{P}_{k1}$, $\mathbf{P}_{k2}$, $\mathbf{P}_{k3}$ and $\mathbf{P}_{k4}$ are checked and X_k will be assigned to appropriate path-segment, $\mathbf{P}_i$.

3- **feature extraction and feature reliability test**: the spectral and contextual feature of an object will be extracted and the feature reliability will be measured by using unity relation properties, where two objects may be merged and will be represented by the same object-feature in the feature space.

4- **feature transmission**: by using the feature map, the boundary of each object is checked. Any time that AMICA finds that the boundary of an object is closed, the features of the corresponding object will be transmitted to the earth-station.

CHAPTER 4

FEATURE EVALUATION

4.1 Introduction

The objective of this chapter is to demonstrate the validity of the unity relation and path-hypothesis for on line unsupervised object-feature extraction and to show that performance of the object-features is better than the pixel-features. The performance of a feature extraction process is measured in terms of the information-bearing quality of the features versus the size of the data set. Classification accuracy is an important quantitative measure of feature quality in applications where the data is automatically interpreted (e.g., remotely sensed image data). However, classification accuracy is dependent on the classification algorithm as well as the feature extraction technique, and often significant accuracy improvements can be obtained by tailoring the classification algorithm to the specific feature extraction technique. Therefore, it is important to investigate jointly feature-extraction and classification when the feature quality (relevancy of the features for scene representation) is based on classification accuracy.

In this chapter several real image data sets are used to provide comparative performance results for the various feature configurations between the original pixel-features X_k and compacted object-features Y_i. The features'

reliability and quality are measured in terms of overall misplacement error in the scene (OME), feature classification performance (FCP), and subjective objects appearance (SOA). The same training samples and decision rule are used for each comparison, Fig. 4.1.

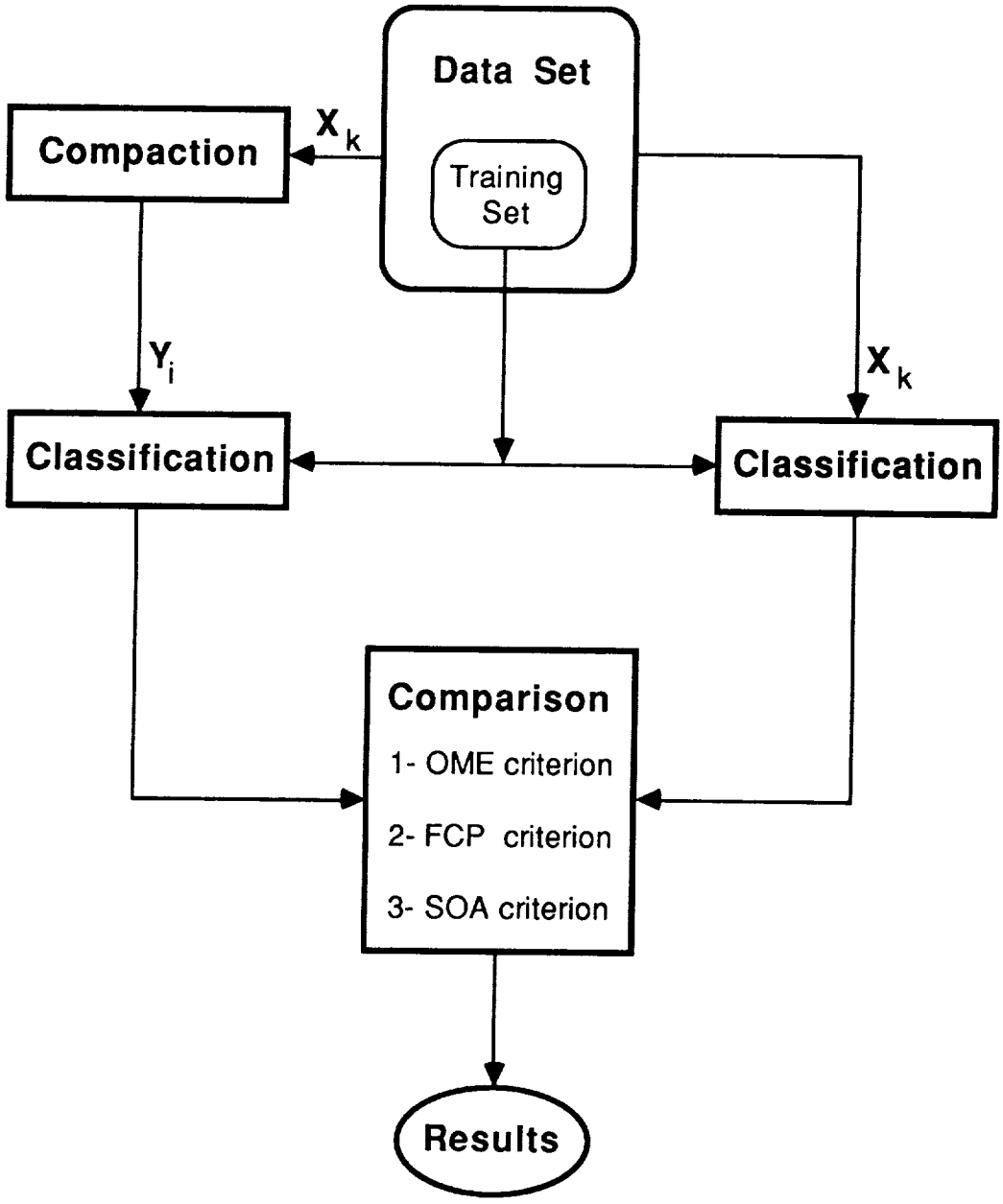


Fig. 4.1 Feature reliability evaluation

The first evaluation is a simple quantitative criterion which has a convenient mathematical form to measure the number of pixels assigned to an incorrect neighboring object based on the object classification, relative to the total number of pixels in the scene (overall misplacement error). Let GTM={r_1,r_2,..., r_{nt}} represent the ground-truth-map of the original data, and let CPM={c_1,c_2,..., c_{nt}} represent the classification-pixel-map result of feature classification. Then the overall misplacement error can be computed by comparison of the CPM and the GTM:

$$OME = 100 \frac{n_m}{n_t}$$

where n_m is the total number of pixels misplaced into incorrect neighboring objects (based on the object class), and n_t is the total number of pixels in the scene. The first features' evaluator, also, measures the quality of object's spatial shape and boundary accuracy in terms of overall misplacement error. The OME is not generally meaningful for measuring the features quality in an absolute sense. For example, it would not usually be useful for comparing feature quality across different feature selections. However, the OME can be very useful for comparing the performance of the options (e.g. window size) of the feature extraction technique on the same multispectral image data using the same classification algorithm with the same training sample sets.

The feature classification performance (FCP) measures the number of pixels classified into the correct class relative to the total number of pixels in that particular class. This criterion is used to evaluate the object-feature performance when the effects of classifier decision rule and training samples on the class feature performance should be considered.

Good ground truth information is a very important parameter in feature evaluation to minimize the unrelated error in the feature extraction. However, obtaining a valid ground-truth-map (GTM) and registering the multispectral image data with this map is often costly and very time consuming. Thus, among the available real data those subsets which have a relatively reliable ground-truth-map should be selected and used for the OME and FCP feature evaluations.

The subjective appearance is an appropriate criterion when the ground-truth-map is not accurate enough to be used by other feature evaluators, or when some objects in the scene are more important than the others regardless of the size of the objects. In such cases it is often too difficult to define a mathematical expression for a feature quality adequate for quantitative evaluation. In tis case visual assessment will be used for this kind of qualification. This criterion is used to evaluate the spatial quality of the spatial-feature-map, for prediction of more information about the scene, by using more complex features, which should be extracted from the training samples. In other words, by incorporating the object appearance in the spatial-feature-map into the feature selection strategy, more complex objects in the scene can be detected. For example some significant within-class variation shows that more information about the complex objects (perhaps soil type covered by vegetation) in the scene might be extracted by using even more complex features.

In this chapter after an introduction to classification decision rules in section 4.2, the experimental results will be presented in section 4.3.

4.2 Classification

Classification is the procedure most often used for quantitative analysis of remote sensing image data. It rests upon using suitable algorithms to label the pixels (or objects) in the scene as representing particular ground cover types, or classes. In classification of the compacted image data, once each object-feature is classified, the memberships of pixels which belong to the corresponding object are determined simultaneously regardless of their size and location in the scene. Irrespective of the classification algorithm, this procedure consists of the following essential practical steps [32]:

- Choose representative or prototype pixels from each of the desired set of classes (for example water, urban regions, cropland, rangelands, etc.). These pixels form a training set. Training sets for each class can be established using site visits, maps, air photographs or even photointerpretation of a color composite product formed from the image data. Often the training pixels for a given class will lie in a common region enclosed in a border; that region is then called a training field (the contiguous training fields are used for within class context parameters estimation).

- Use the training data to estimate the parameters of the particular classifier algorithm to be used; these parameters will be the properties of the probability model used, or will be the equation that defines partitions in the multispectral space. The set of parameters for a given class is called the *class-feature* set of that class.

- Using the trained classifier, label or classify every pixel (or object) in the scene into one of the desired ground cover types (information classes). Here the whole scene segment of interest is typically classified.

- Produce tabular summaries or *thematic* (class) maps which summarize the results of the classification.

In this chapter, to evaluate object-feature reliability two different classifiers are considered: the Maximum Likelihood classifier and the Minimum Distance classifier [4].

4.2.1 Maximum Likelihood Decision Rule

Maximum likelihood classification is the most common supervised classification method used with remote sensing image data; this is developed in the following in a statistically acceptable manner [11]. Let the ground cover classes for a scene be represented by: ω_i , i = 1, 2, ..., m, where m is the total number of desired classes in the scene. Suppose that sufficient training data is available for each ground cover type. This can be used to estimate a probability distribution for ground cover type that describes the likelihood of finding a feature from class ω_i , at the position of X. The maximum likelihood decision rule decides that X belongs to ω_i if and only if

$$p(X|\omega_i)p(\omega_i) > p(X|\omega_j)p(\omega_j) \quad \text{for all } j \neq i$$

where $p(X|\omega_j)$ is the class ω_j probability density function evaluated at X , and $p(\omega_j)$ (so-called a-priori probability of class ω_j) is the probability that class ω_j

occurs in the scene. Computational convenience results in definition of the discriminant function $g_i(X)$ [11]

$$g_i(X) = \ln\{p(X|\omega_i)p(\omega_i)\} = \ln\{p(X|\omega_i)\}+\ln\{p(\omega_i)\}$$

where **ln** is the natural logarithm, so the maximum likelihood decision rule becomes

$$X \in \omega_i \Leftrightarrow g_i(X) > g_j(X) \text{ for all } j \neq i$$

A very common classification approach in multispectral image data application is the maximum likelihood Gaussian parametric classifier on a per vector basis. This classifier is often used because of its relatively simple implementation, especially when the spectral features are the only features that are used for object representation. Then the discriminant function for maximum likelihood classification, based upon the assumption of Gaussian distribution, is:

$$g_i(X) = \ln\{p(\omega_i)\} - 0.5\ln|\Sigma_i| - 0.5(X-M_i)^t \Sigma_i^{-1}(X-M_i)$$

where M_i and Σ_i are the mean vector and covariance matrix of the data in class ω_i, which is estimated from the training samples. Let n_i samples be from class ω_i, then the mean and covariance of the each class are estimated by the following equations:

$$M_i = \frac{1}{n_i}\sum_{k=1}^{n_i} X_k$$

$$\Sigma_i = \frac{1}{n_i - 1}\sum_{k=1}^{n_i} (X_k - M_i)(X_k - M_i)^t$$

Often the analyst has no useful information about the $p(\omega_i)$. In this case a situation of equal prior probabilities is assumed. As a result $\ln\{p(\omega_i)\}$ can be removed from the discriminant function, since it is the same for i=1, 2, ..., m. In that case the 0.5 common factor can also be removed. Thus the discriminant function can be simplified into:

$$g_i(X) = - \ln|\Sigma_i| - (X-M_i)^t \Sigma_i^{-1} (X-M_i)$$

Sufficient training samples for each ground cover class must be available to allow reasonable estimates of the elements of the mean vector and the covariance matrix to be determined. For a d dimensional multispectral space at least d+1 samples are required to avoid the covariance matrix being singular. Apart from this condition it is clearly important to have as many training pixels as possible, particularly as the dimensionality of feature space increases, since in higher dimensional spaces there is an increased chance of having some individual dimensions poorly represented [19]. Swain [4] recommends as a practical minimum that 10d samples per class be obtained for training, with 100d as being highly desirable, if it can be attained.

The effectiveness of maximum likelihood classification depends upon a reasonably accurate estimation of the mean vector M and the covariance matrix Σ for each spectral class. This in turn is dependent upon having a sufficient number of training pixels for each of those classes. In cases where this is not so, inaccurate estimates of elements of Σ results, leading to poor classification. When the number of training samples per class is limited it can be more effective to resort to a classifier that does not make use of covariance information.

4.2.2 Minimum Distance Decision Rule

When the number of training samples per class is limited to have a reasonably accurate estimation of the covariance matrix Σ for each spectral class, the minimum distance (M.D.) classifier [11], is an alternative which to solves the problem of the M.L. Gaussian classifier. Also M.D. classifier can be useful when the other object's parameters such as contextual features (local spectral gradient) together with the spectral feature are used for data classification.

The assignment of an unknown pixel to a class is based on the minimum distance decision rule, where the degree of assignment of pixel to each object would depend on the relative distance between the object-feature and each class-feature, estimated from the training samples [11,40]. With M.D. classifier, training data is used only to determine class-features [32]: ***Classification is then performed by placing an object-feature (or pixel-feature) in the class of the nearest class-feature***. Let Y be an unknown feature and $\partial_i(Y)$ be such a distance from class-feature of ω_i then:

$$Y \in \omega_i \Leftrightarrow \partial_i(Y) < \partial_j(Y) \quad \text{for all } j \neq i$$

As it was explained in chapter two, an object-feature can be represented by a three-tuple (S_y, V_y, $\mathbf{L}_y$), where S_y is the spectral vector, V_y is the within object gradient vector, $\mathbf{L}_y$ is the spatial-feature-map, and n_y will be the number of pixels in the object. Let n_i be the number of pixels within the training field ($\mathbf{L}_i$) used for class-feature estimation from class ω_i. Then the class-feature of ω_i

can be represented by three parameters (S_i , V_i , n_i) where S_i is the estimated class spectral vector and V_i is the estimated within class gradient vector.

To incorporate the within-class contextual information into the minimum distance classifier decision rule, a discriminant function based on a new *distance* is defined, where in L^p space [38] the general form of this minimum distance classifier is defined by the following equation:

$$\partial_i^p(Y) = (S_i - S_y)_p^t (g_i V_i + g_y V_y)^{-1} + \ln(V_i^{g_i} \otimes V_y^{g_y})$$

The coefficients g_i and g_y are defined by:

$$g_i = \frac{n_i}{n_i + n_y} \quad \text{and} \quad g_y = \frac{n_y}{n_i + n_y}$$

Again S and V are d-dimensional vectors and g is a scalar, **ln** is the natural logarithm, and the superscript "t" represents that the vector is transposed. The subscript "p" denotes that the distance is measured in the L^p space. The operation $\otimes$ between two vectors V_1 and V_2 is defined by:

$$V_1 \otimes V_2 = \prod_{k=1}^{d} v_{1k} v_{2k}$$

where the scalars v_{1k} and v_{2k} are elements of vectors V_1 and V_2 respectively. The power "p" of a vector is defined by the following relationship between their corresponding elements:

$$\begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_{d-1} \\ V_d \end{bmatrix}^p = \begin{bmatrix} V_1^p \\ V_2^p \\ \vdots \\ V_{d-1}^p \\ V_d^p \end{bmatrix}$$

Then the inverse of a vector based on the above definition is unique, where:

$$V \otimes V^{-1} = 1$$

This minimum distance classifier may be attractive (specially when the CPU time for classification of data be a significant parameter) since it is a faster technique than the maximum likelihood classification: the speed of this classifier is order O(d), however the speed of the M.L. Gaussian decision rule is of order $O(\mathbf{d}^2)$. In the maximum likelihood classification each class is modeled by a multivariate normal class model; however, the Gaussian assumption is not used in this minimum distance technique. Notice that though several ground cover classes can be classified by this M.D. classifier, it might be more suitable to use the M.L. classifier (the M.L. classifier is usually more accurate than the M.D. classifier).

In the next section the experimental test results are presented, where both decision rules (the Maximum Likelihood Gaussian classifier and the Minimum Distance classifier) are used for feature classification.

4.3 Experimental Results

In this section the proposed feature extraction technique is applied to several set of image data. As previously stated, the objective of these experiment is to demonstrate the validity of the unity relationship and the path-hypothesis, and to show that the performance of object-feature is better than the performance of pixel-feature regardless of the choice of classification decision rule and the training set.

The original of the first MSS data set (called Flight-line 210) is contained on LARS tape number 165, file number 1, rune number 71053900. This file contains Indiana agricultural data. The data has 12 spectral bands (0.46 μm to 11.70 μm, Table 4.1) and was collected by the University of Michigan Scanner. Corn Blight Watch Flight-line 210 was overflown at about noon on August 13, 1971 from an altitude of 5,000 feet. The area covered was a 1.4x9.7 mile strip of farmland. There were 228 samples/scan line and 1161 scan lines for a total of 264,708 pixels.

Table 4.1 Flight-line 210 spectral bands

Channel	Spectral Band, μm
1	0.46 - 0.49
2	0.48 - 0.51
3	0.50 - 0.54
4	0.52 - 0.57
5	0.54 - 0.60
6	0.58 - 0.65
7	0.61 - 0.70
8	0.72 - 0.92
9	1.00 - 1.40
10	1.50 - 1.80
11	2.00 - 2.60
12	9.30 - 11.70

The original of the second MSS data set (called Flight-line C-1) was obtained from the sampled and quantized output of a 12-channel (0.40 μm to 1.00 μm, Table 4.2) airborne scanner flown over predominantly agricultural regions in Indiana. Flight-line C-1 was obtained on 28 June, 1966 at 12:30 P.M. The area covered was approximately 4 miles long and 1 mile wide of farmland. There were 228 samples/scan line and 950 scan lines for a total of 216,600 pixels.

Table 4.2 Flight line C-1 spectral bands

Channel	Spectral Band, μm
1	0.40 - 0.44
2	0.44 - 0.46
3	0.46 - 0.48
4	0.48 - 0.50
5	0.50 - 0.52
6	0.52 - 0.55
7	0.55 - 0.58
8	0.58 - 0.62
9	0.62 - 0.66
10	0.66 - 0.72
11	0.72 - 0.80
12	0.80 - 1.00

The third MSS data set has 7-channel spectral bands (three visible spectral bands and four infrared spectral bands) which have been obtained from an urban area and includes the O'Hare Airport. This area has 256 samples/scan line and 256 scan lines for a total of 65,536 pixels.

In addition, three single-band images with different complexity are used for evaluation of feature extraction process by spatial-feature-map comparison. These three are: "campus-512", which contains a portion of the the Purdue campus in the visible spectral band; and the girl; and the space shuttle

image. The former contained 512 x 512 pixels and the two latter contained 256 x 256 pixels.

As has pointed out, the reliability and quality of feature extraction process are measured in terms of overall classification error, feature classification performance, and subjective objects appearance. The performance of the first two evaluators is highly dependent on the accuracy of the ground-truth-map. Thus in order to minimize the effect of unrelated error in classification performances, the first two evaluations (OME and FCP) are applied only to the test areas which have a relatively accurate ground-truth-map (Fig.4.3 see also appendix B). However the SOA evaluation is applied to the whole area in all of the data sets.

4.3.1 Feature Classification Performance

The original pixel-features and the compacted object-features are used separately to determine the classification accuracy in each space. This evaluation is done by comparing classification performance in these two particular spaces. The results of each trial can be presented in comparison tables, where in each table three different parameters have been considered:

- classification performance
- compaction coefficient
- CPU time for classification.

The set No.1 contained 9 different ground cover classes (appendix B), which the 12 channels spectral imagery of this area are presented by Fig.4.2.a and Fig.4.2.b. Tables 4.3 and 4.4 are examples of feature evaluation using two

different agricultural data sets, when the M.L. Bayes Gaussian decision rule is selected.

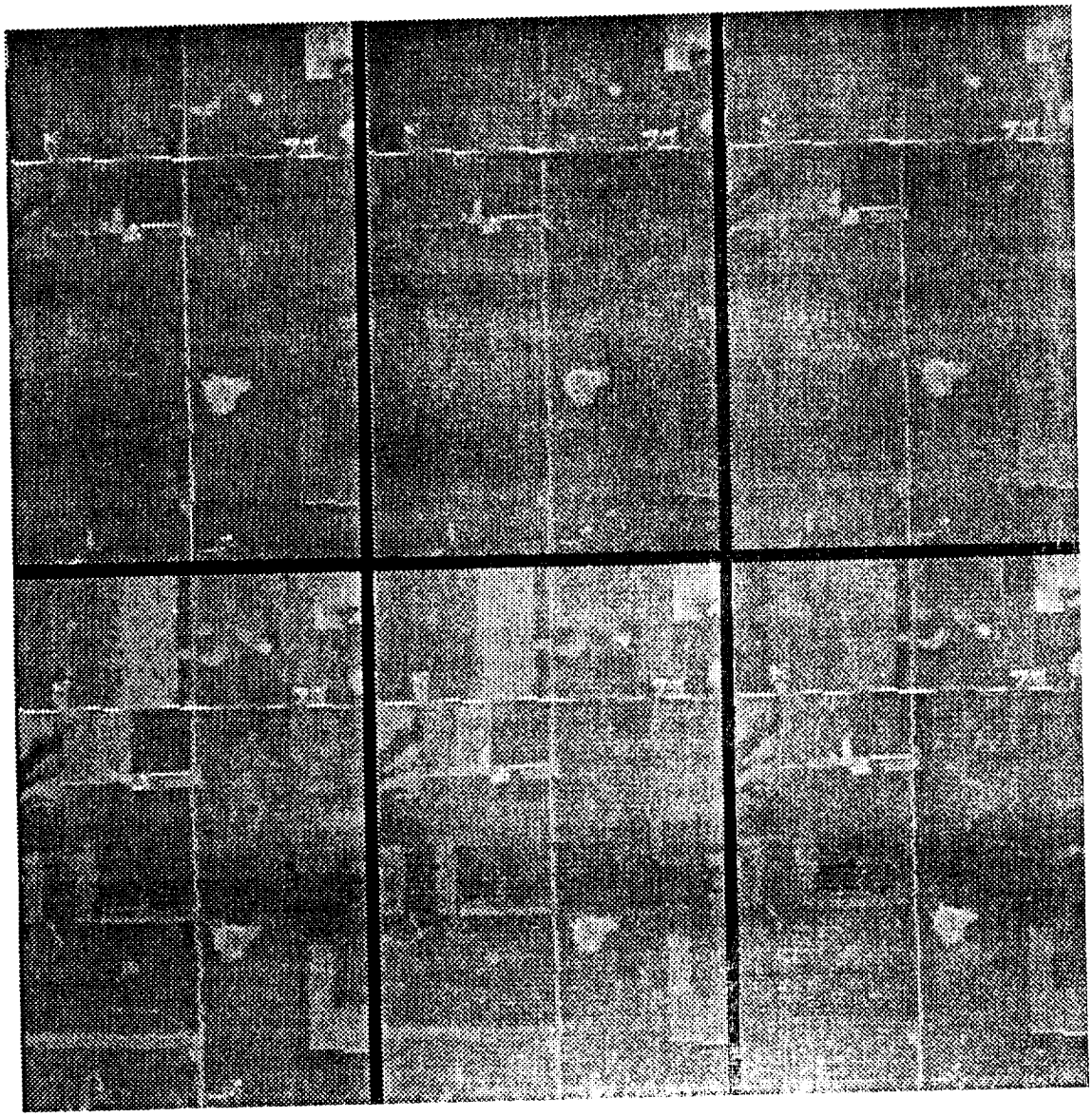


Fig.4.2.a Spectral imagery of test area set NO.1 channel-1 to channel-6

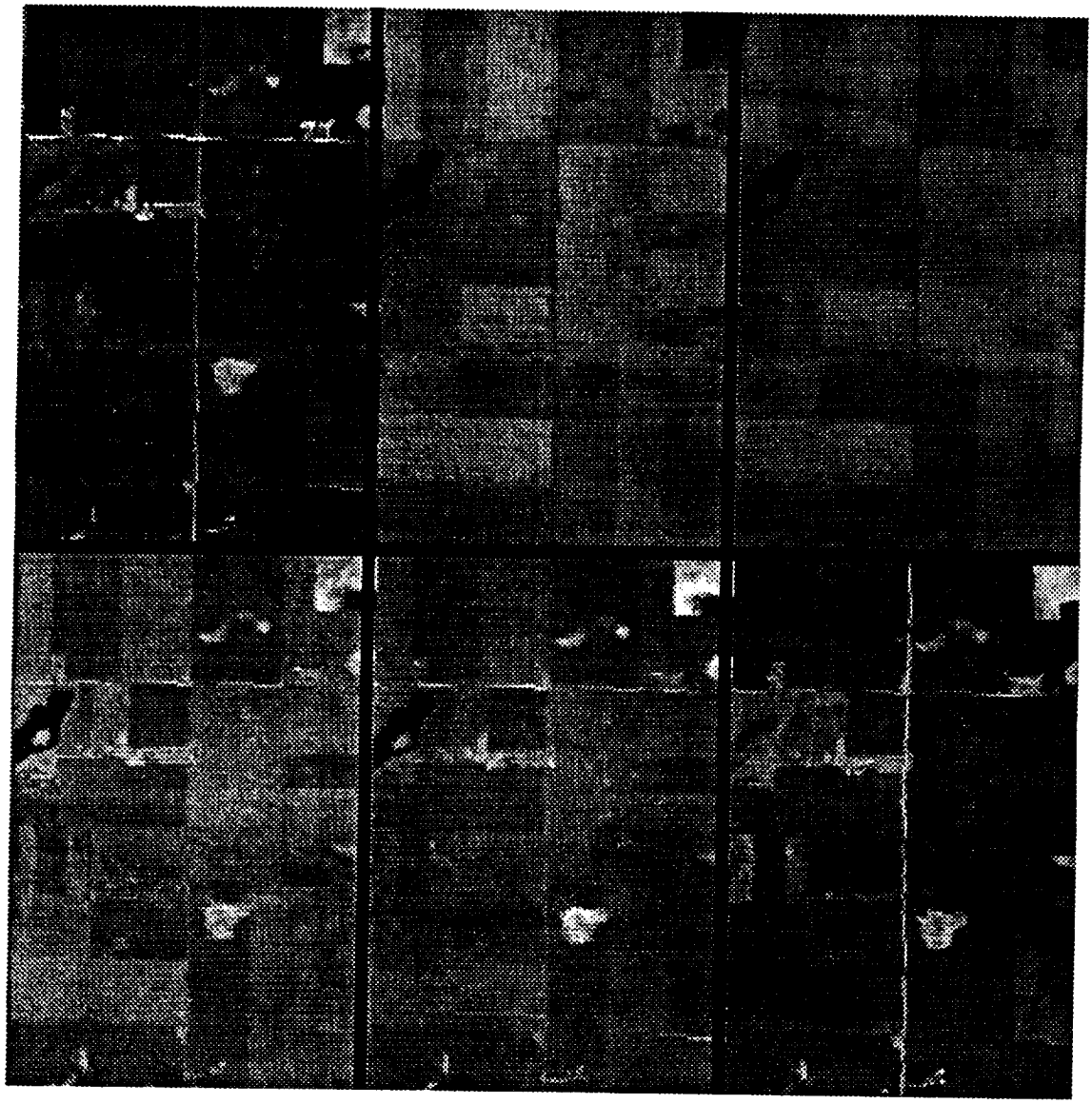


Fig.4.2.b Spectral imagery of test area set NO.1 channel-7 to channel-12

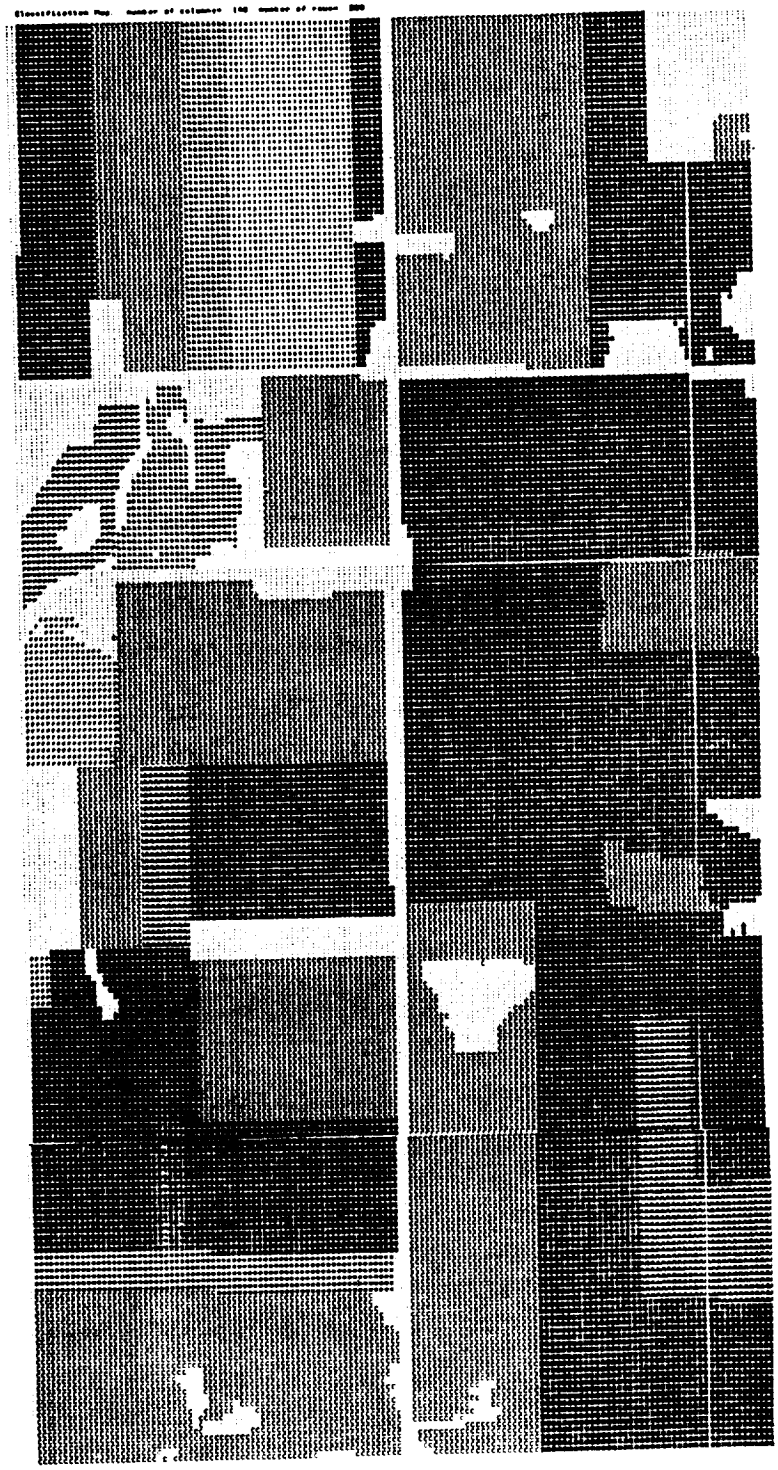


Fig.4.3 Ground-truth-map of data set No.1

Table 4.3. Feature performance using Bayes-ML classifier (set No.1)

Performance of Pixel-Features											
Number of Features=369,600 Bytes						**Compaction Coefficient =1**					
True Class	Number of Samples Classified										
	Total#	Corn	Soy.	Wood	Whe.	Sud.	Oats	Pas.	Hay	Non	%Correct
Corn	10104	8942	102	145	149	1	22	0	22	721	88.5%
Soybeans	12910	6	11717	482	108	8	87	0	14	488	90.8%
Woods	389	4	10	328	3	0	1	2	0	41	84.3%
Wheat	944	0	8	8	732	0	24	0	9	163	77.5%
Sudex	1219	0	17	0	0	1175	21	0	2	4	96.4%
Oats	603	1	12	0	8	3	508	0	28	43	84.2%
Pasture	339	0	0	0	0	0	0	307	0	32	90.6%
Hay	746	22	1	1	21	3	52	0	592	54	79.4%
Nonfarm	3546	17	69	14	68	1	111	9	81	3176	89.6%
Totals	30800	8992	11936	978	1089	1191	826	318	748	4722	89.2%
Overall Performance = 89.2%						CPU Time = 515.2 seconds					

Performance of Object-Features											
Number of Features=13,692 Bytes						**Compaction Coefficient =27**					
True Class	Number of Samples Classified										
	Total#	Corn	Soy.	Wood	Whe.	Sud.	Oats	Pas.	Hay	Non	%Correct.
Corn	10104	9592	123	17	67	0	6	0	66	233	94.9%
Soybeans	12910	24	12409	209	74	1	27	0	11	155	96.1%
Woods	389	0	4	385	0	0	0	0	0	0	99.0%
Wheat	944	6	11	12	824	0	11	0	0	80	87.3%
Sudex	1219	0	9	0	0	1193	13	0	3	1	97.9%
Oats	603	4	1	0	2	0	588	0	0	8	97.5%
Pasture	339	0	0	0	0	0	0	339	0	0	100.0%
Hay	746	45	0	0	0	9	1	0	691	0	92.6%
Nonfarm	3546	69	136	12	94	8	244	0	118	2865	80.8%
Totals	30800	9740	12693	635	1061	1202	890	339	889	3342	93.8%
Overall Performance = 93.8%						CPU Time = 18.8 seconds					

Comparison:

Compaction Coefficient.....369,600/13,692 = **27.0**

Performance Improved.....from **89.2%** to **93.8%**

CPU time speed up factor ..... 515/18.8 = **27.4**

Table 4.4. Feature performance using Bayes-ML classifier (set No.2)

Performance of Pixel-Features											
Number of Features=115200 Bytes						**Compaction Coefficient =1**					
True Class	Number of Samples Classified										
	Total#	Corn	Soy.	Wood	Whe.	Sud.	Oats	Pat.	Hay	Non	%Correct
Corn	2400	2250	0	146	0	0	1	3	0	0	94%
Soybeans	2400	0	2322	76	0	0	1	1	0	0	97%
Woods	800	3	1	785	5	0	0	0	6	0	98%
Wheat	800	17	0	11	767	0	5	0	0	0	96%
Sudex	800	0	0	0	0	791	5	0	4	0	99%
Oats	800	0	0	9	7	0	719	63	2	0	90%
Pasture	800	0	0	0	0	0	98	701	1	0	87%
Hay	800	0	2	11	2	0	2	15	768	0	96%
Nonfarm	0	0	0	0	0	0	0	0	0	0	100%
Totals	9600	2270	2325	1038	781	791	831	783	781	0	94.8%
Overall Performance = 94.8%						CPU Time = 22.20 seconds					

Performance of Object-Features											
Number of Features=4236 Bytes						**Compaction Coefficient =27**					
True Class	Number of Samples Classified										
	Total#	Corn	Soy.	Wood	Whe.	Sud.	Oats	Pat.	Hay	Non	%Correct
Corn	2400	2391	1	6	0	0	0	2	0	0	100%
Soybeans	2400	0	2399	0	0	0	0	1	0	0	100%
Woods	800	14	1	778	7	0	0	0	0	0	97%
Wheat	800	33	0	0	767	0	0	0	0	0	96%
Sudex	800	0	0	0	0	792	5	0	0	3	99%
Oats	800	0	0	7	2	0	767	18	6	0	96%
Pasture	800	0	0	0	0	0	12	788	0	0	99%
Hay	800	0	7	3	0	0	0	7	783	0	98%
Nonfarm	0	0	0	0	0	0	0	0	0	0	100%
Totals	9600	2403	2328	1240	729	792	851	770	487	0	98.6%
Overall Performance = 98.6%						CPU Time = 0.67 seconds					

Comparison:

Compaction Coefficient..... 115200/4236 = **27.2**

Performance Improved.....from **94.8%** to **98.6%**

CPU time speed up factor.....22.20/0.67 = **33.13**

To establish the unity relation, the system learns about the functional coefficients simultaneously with the data acquisition process by measuring the object spectral gradient which, is then, normalized within a window. Classification accuracy is dependent on both the classification algorithm and the training sample set, furthermore, it is slightly dependent on the window size, which will be investigated in this section. Various multispectral image data are used to measure object misplacement error versus size of window for functional assignment. This corresponds to determining feature performance as a function of window size n_w. The performance of the compacted object-feature, extracted from multispectral image data, is plotted in Fig.4.4 (using data set No.1) and Fig.4.5 (using data set No.2), and compared with the performance of the original pixel-features from the same scene.

The general form of the functional is defined by (see section 2.7):

$$\partial_g (Y_i, X_r) = (| S_i - X_r |)^T (\alpha V_i + \beta V_n)^{-1}$$

In the second trial as Figure 4.5 shows two different functionals Fun.1 and Fun.2 are defined for object-feature extraction. Where in the first functional (Fun.1) the coefficient α and β are defined by

$$\alpha = 0 \text{ and } \beta = 1 \quad \text{if} \quad n_i \leq 5.d$$

$$\alpha = 1 \text{ and } \beta = 0 \quad \text{otherwise}$$

and in the second functional (Fun.2) the coefficient α and β are defined by

$$\alpha = \frac{d\, n_i}{n_i + d} \quad \text{and} \quad \beta = \frac{d^2}{n_i + d}$$

As Fig.4.4 and Fig.4.5 show the feature performance for window size more than 3 lines is almost constant for a particular functional.

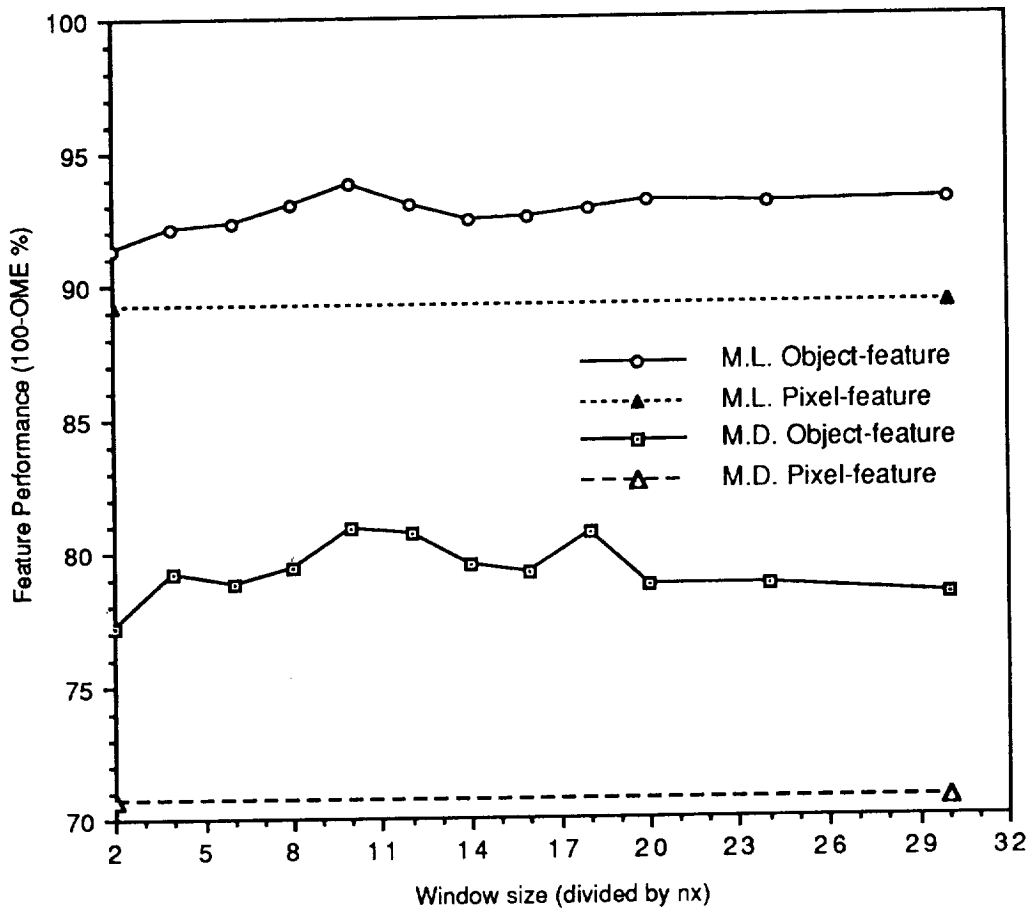


Fig. 4.4. Effect of window size on feature performance using data set No.1

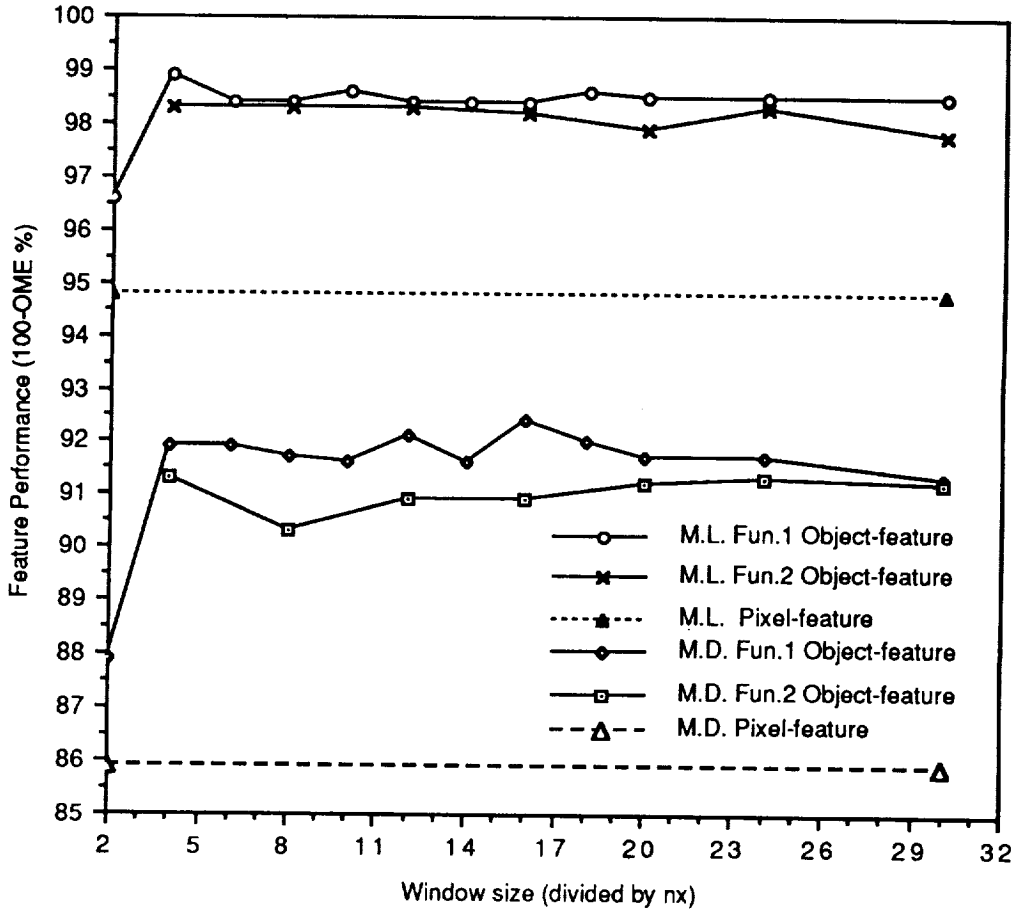


Fig. 4.5. Effect of window size on feature performance using data set No.2

The compaction coefficient is dependent on the data type, where the metric coefficient for feature extraction is adapted within the window. Thus it is expected that, regardless of the data type and the functional space, the compaction coefficient also be a function of the window size. Three different data sets are used and the performance of the compaction coefficient as a function of window size is plotted in Fig.4.6. Figure 4.6. shows the compaction coefficient is almost constant for window height larger than three lines.

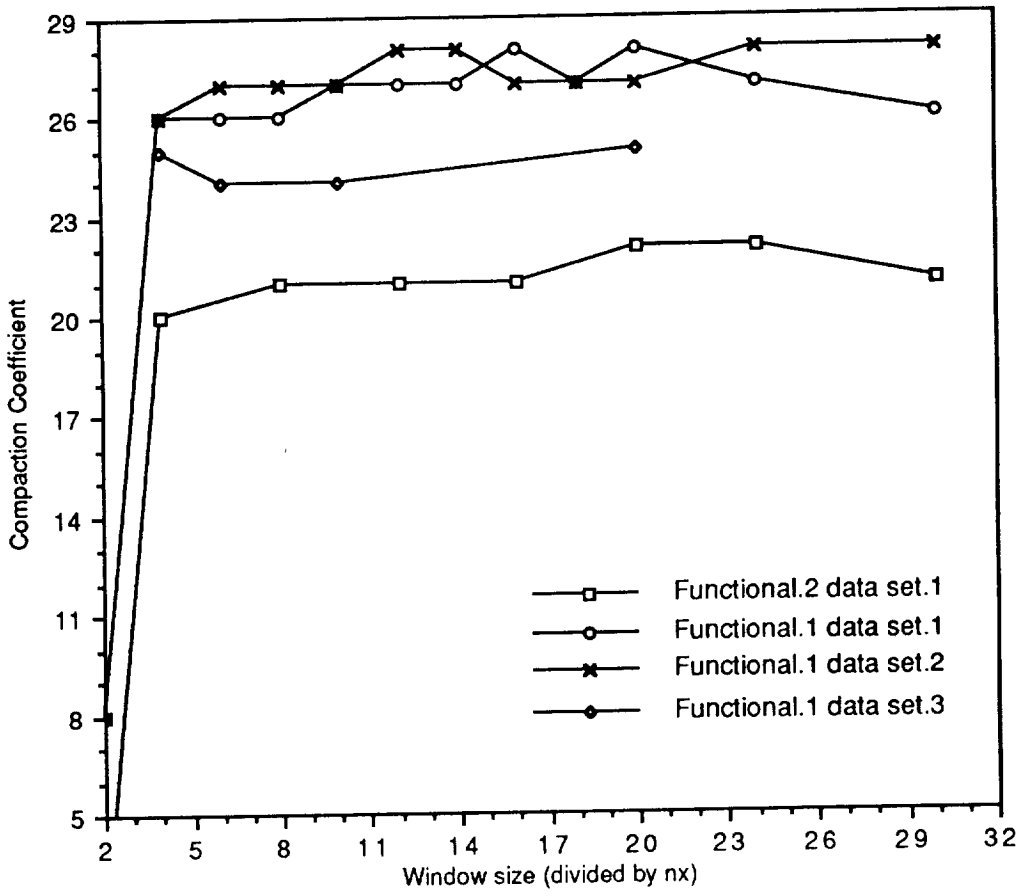


Fig. 4.6. Effect of window size on the compaction coefficient using three different data sets

Classification accuracy is dependent on both the classification algorithm and the training sample set. It is only slightly dependent on the window size, as seen earlier. The objective of this section is to demonstrate the validity of the path hypothesis for unsupervised object-feature extraction and to show that performance of the object-features is better than the pixel-features, independent of the choice of classifier decision rule. Once an object-feature is classified the membership of pixels which belong to that object are determined simultaneously, regardless the pixels' location in the scene.

Three different spaces (Diamond space, Euclidean space, and Chebyshev space) are used separately for minimum distance functional construction to illustrate the degree of metric dependence in the feature classification performance relative to a maximum likelihood Gaussian decision rule.

The M.D. discriminant function in Diamond space represented by the following equation:

$$\partial_i^1(Y) = \sum_{k=1}^{d} \left\{ \frac{| s_{ik} - s_{yk} |}{g_i v_{ik} + g_y v_{yk}} + g_i \ln(v_{ik}) + g_y \ln(v_{yk}) \right\}$$

In the Euclidean space the M.D. discriminant function is defined by:

$$\partial_i^2(Y) = \sum_{k=1}^{d} \left\{ \frac{(s_{ik} - s_{yk})^2}{g_i v_{ik} + g_y v_{yk}} + g_i \ln(v_{ik}) + g_y \ln(v_{yk}) \right\}$$

And in the Chebyshev space the discriminant function is represented by:

$$\partial_i^c(Y) = \sup_{k=1}^{d} \{ \frac{| s_{ik} - s_{yk} |}{g_i v_{ik} + g_y v_{yk}} + g_i \ln(v_{ik}) + g_y \ln(v_{yk}) \}$$

The results of overall feature performance (100 - OME %) versus selected classification decision rule in the metric spaces are plotted in figure 4.7 and Fig.4.8.

Figure 4.9 shows the comparative results of feature classification performance (FCP) maximum likelihood Gaussian decision rule, where, in this test only the spectral features of objects are used for their classification. Fig.4.10 shows FCP of data set No.1 using minimum distance decision rule in Diamond space. Fig.4.11 shows FCP of data set No.1 using M.D. decision rule in the Euclidean space. Fig.4.12 shows the feature classification performance of data set No.1 using M.D. decision rule in the Chebyshev space for classification, where the performance of the pixel-feature and the object-features are presented separately.

The same experiment is done by using data set No.2 for classification of object-features (compacted data) and pixel-features (original data), and the comparison results are presented in the figures Fig.4.13, Fig.4.14, Fig.4.15 and Fig.4.16.

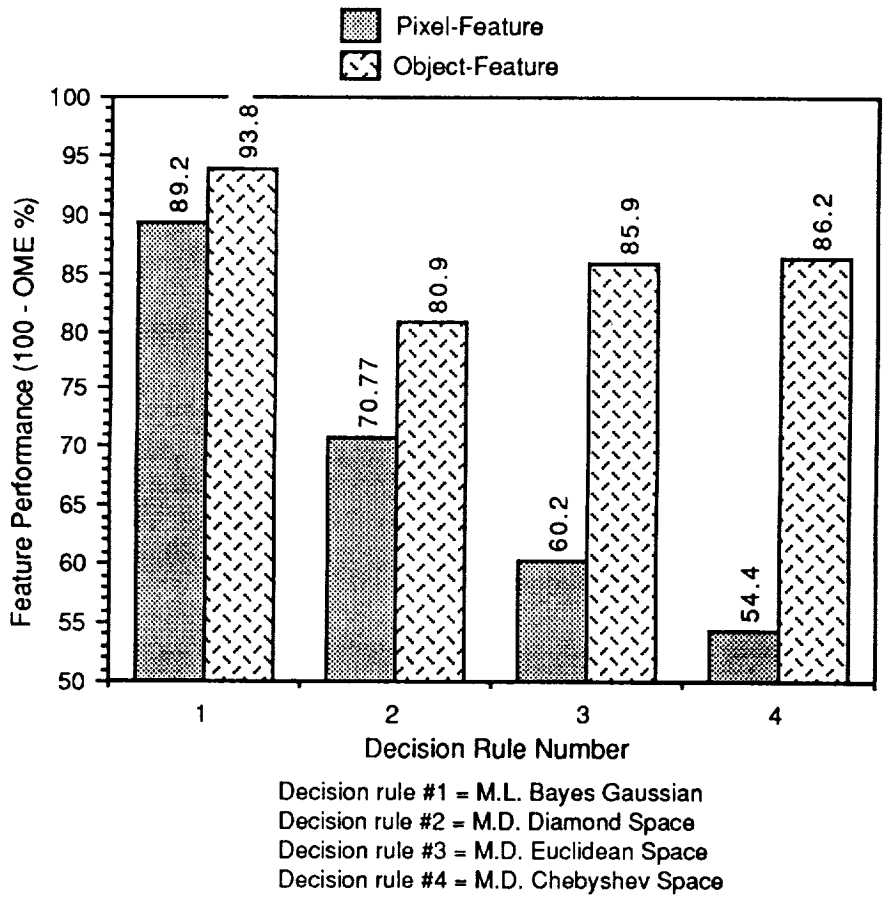


Fig. 4.7 Overall Feature Performance using different classification rules (data set No.1)

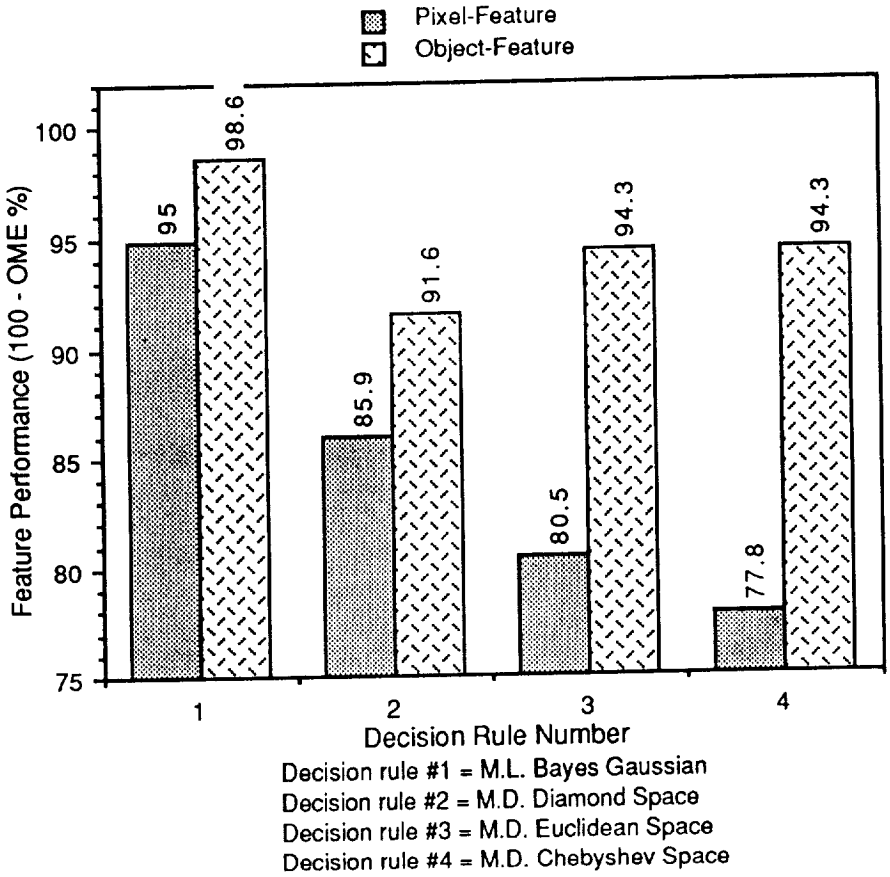


Fig. 4.8 Overall Feature Performance using different classification rules (data set No.2)

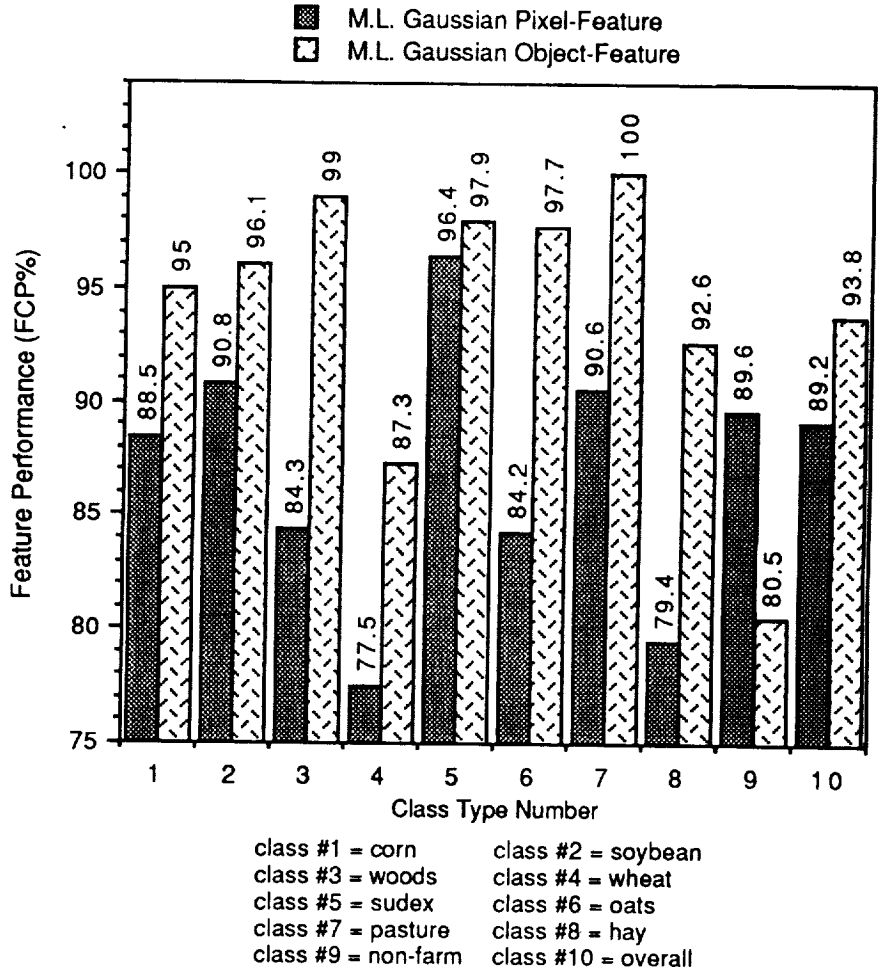


Fig. 4.9 Feature performance using M.L. Bayes Gaussian decision rule (data set No.1)

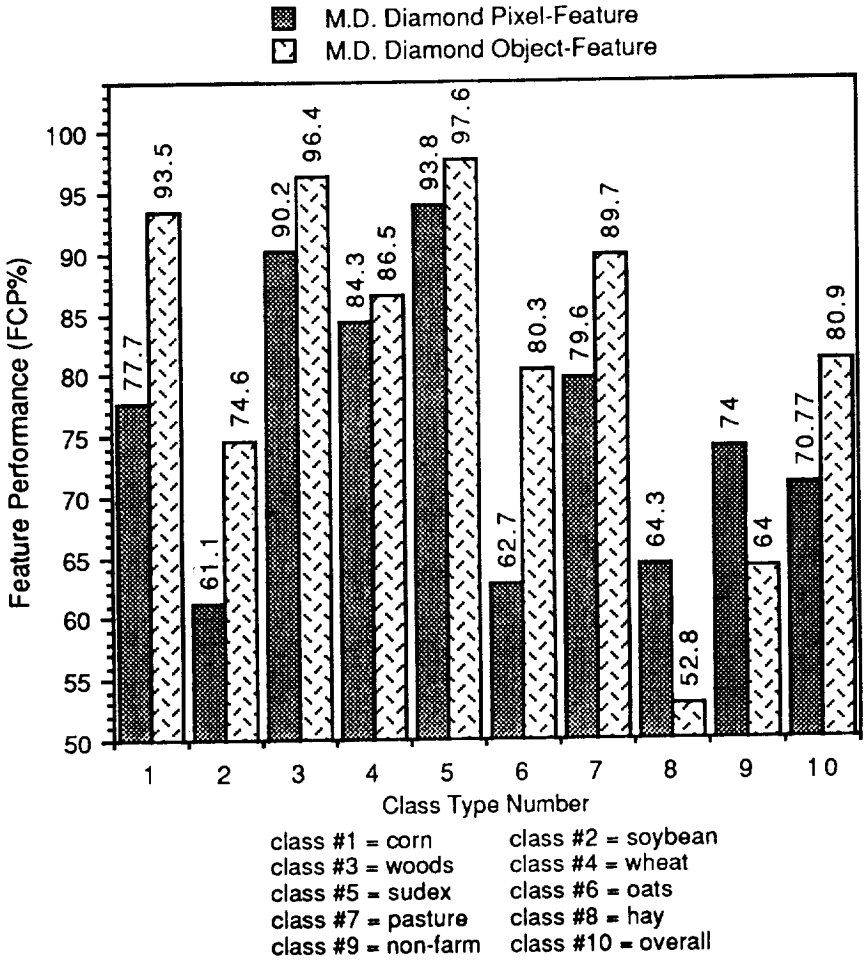


Fig. 4.10 Feature performance using M.D. Diamond space decision rule (data set No.1)

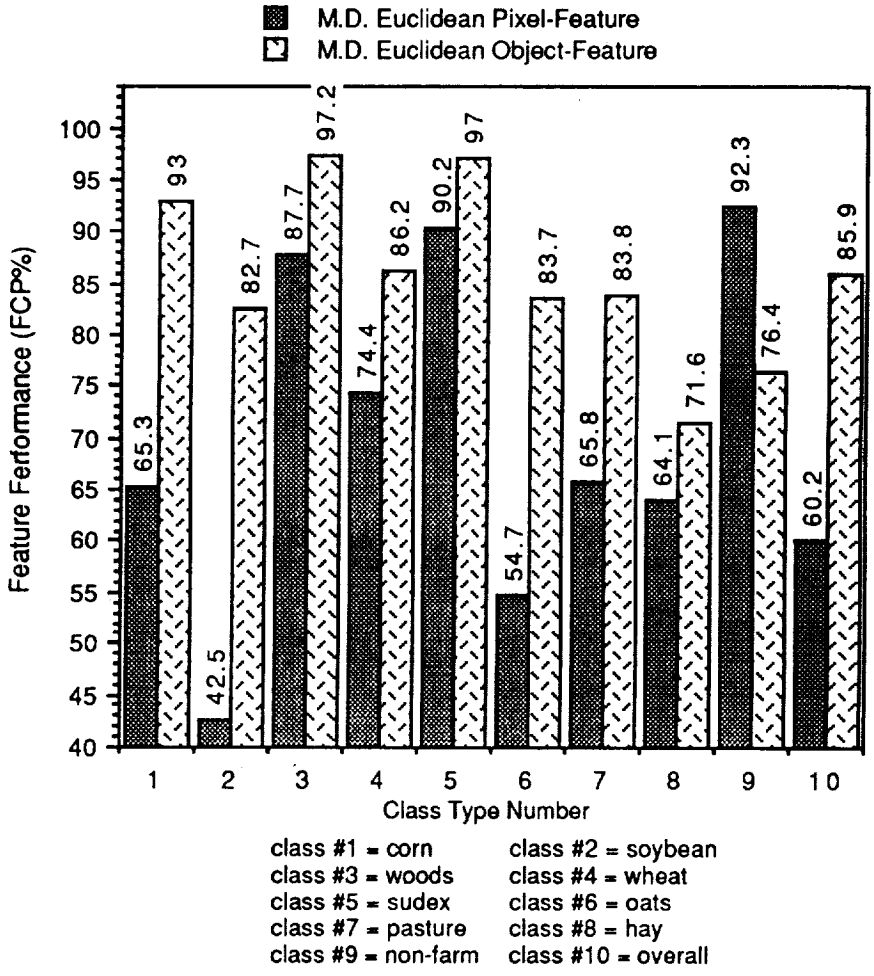


Fig. 4.11 Feature performance using M.D. Euclidean space decision rule (data set No.1)

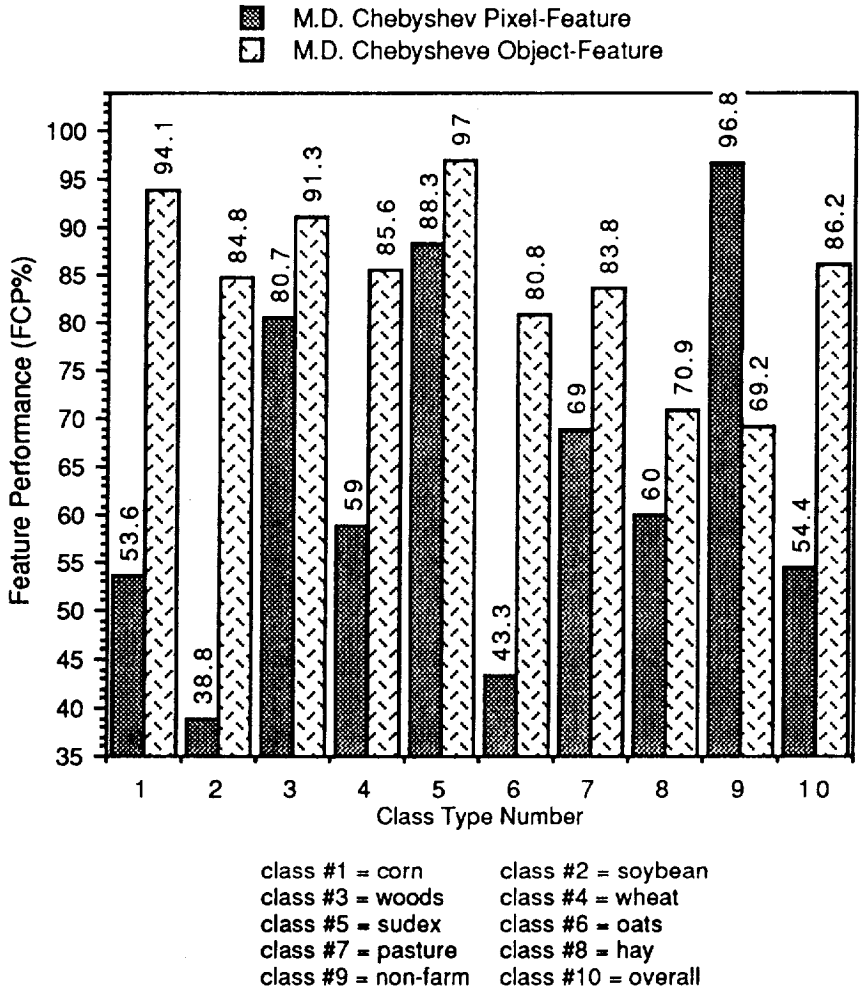


Fig. 4.12 Feature performance using M.D. Chebyshev space decision rule (data set No.1)

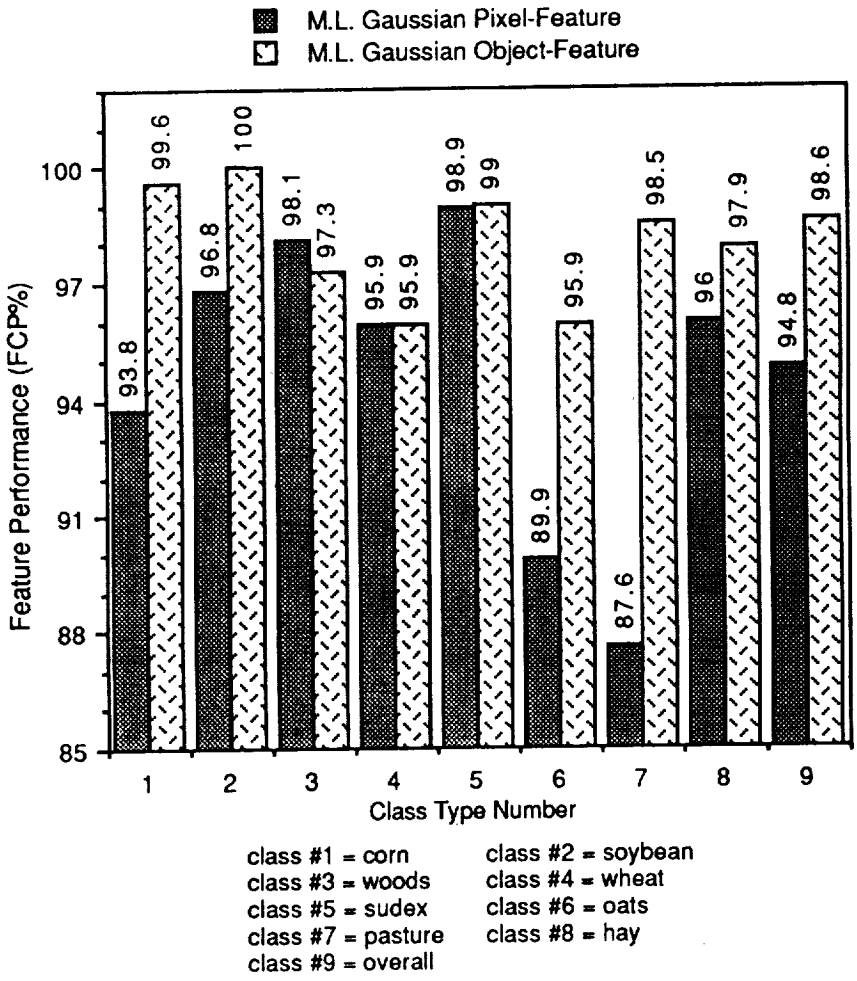


Fig. 4.13 Feature performance using M.L. Bayes Gaussian decision rule (data set No.2)

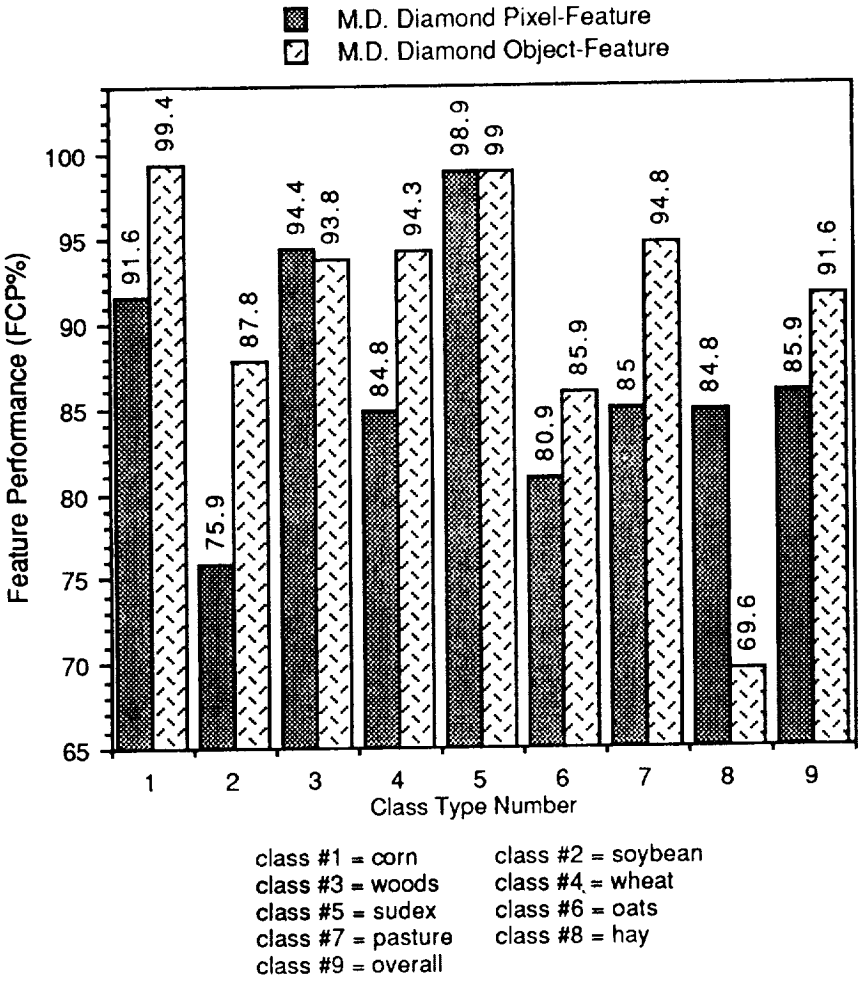


Fig. 4.14 Feature performance using M.D. Diamond space decision rule (data set No.2)

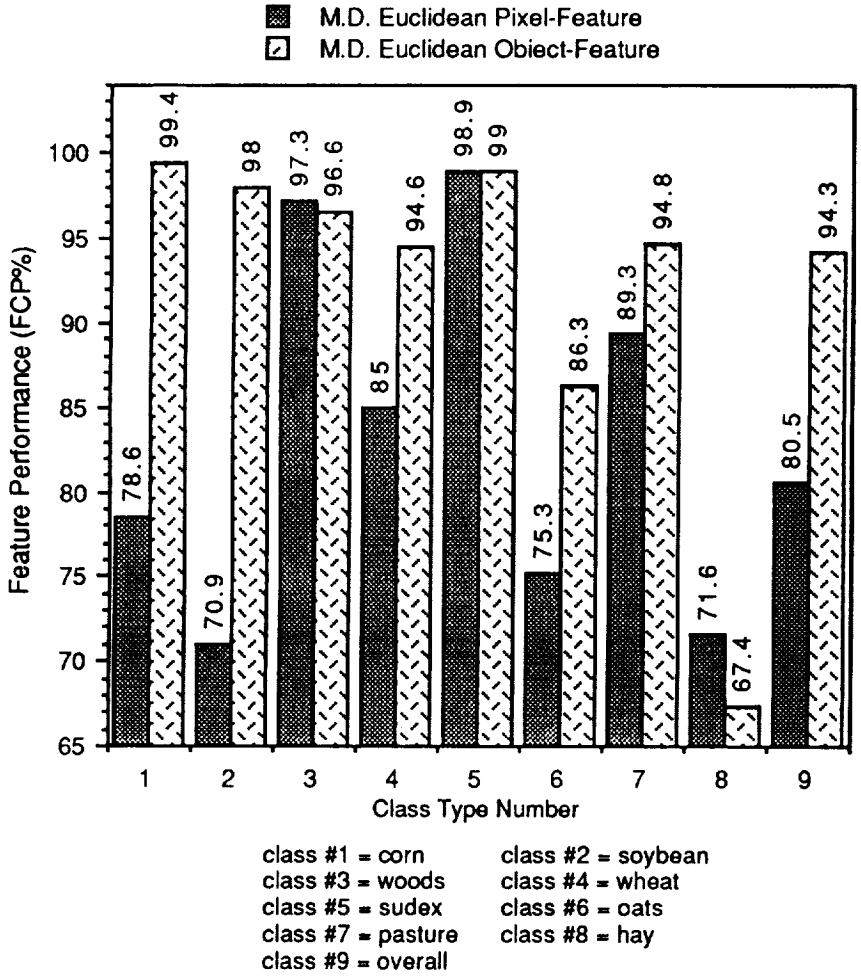


Fig. 4.15 Feature performance using M.D. Euclidean space decision rule (data set No.2)

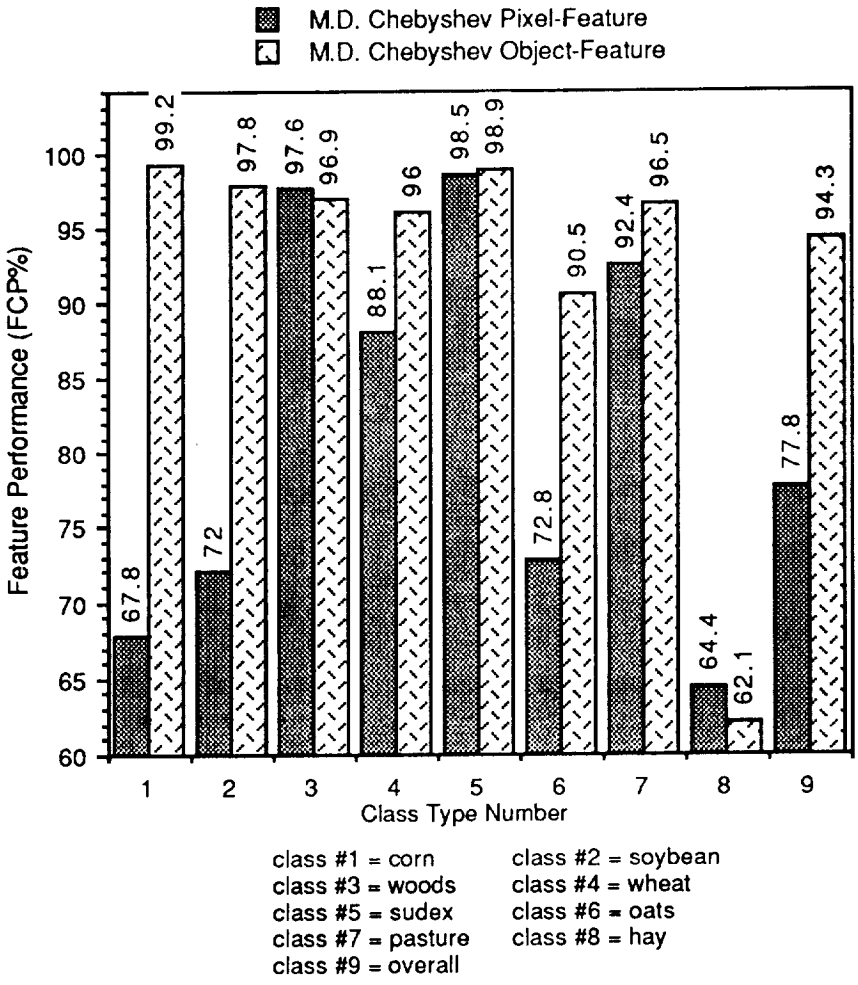


Fig. 4.16 Feature performance using M.D. Chebyshev space decision rule (data set No.2)

The spectral feature for a given class is a function of sun angle, ground slope, ground moisture, atmospheric absorption, MSS instrument noise, and many other parameters. Therefore, the classification performance is dependent on the training samples, even if they come from the same ground cover type. By training the classifier using samples from the data set to be analyzed the effects of many of these parameters are normalized out. Typically, a total training set for a given class consists of several subsets of data, which are selected from various locations throughout the image. However for obtaining contextual features those pixels should be contiguous in the square area (training field) with a size comparative to the data dimensionality (number of spectral channels). Also, contextual information is highly dependent on the sensor altitude and spatial resolution. So the need for training samples from the same Flight-line is significant.

Two training sets are selected from different areas in different sizes. The statistics of the training sets are presented in the appendix B. The performance of pixel-features (original data) and object-feature is presented in Table 4.3, using the first training set. In this experiment a standard maximum likelihood Gaussian classifier is used for classification. Figure 4.17 shows the comparative feature classification performance (FCP) using the first training set, where only the spectral features of objects are used for their identification. The same experiment is done by using the second training set for classification of the object-features (compacted data) and pixel-features (original data), and the comparison results are presented by Fig.4.18.

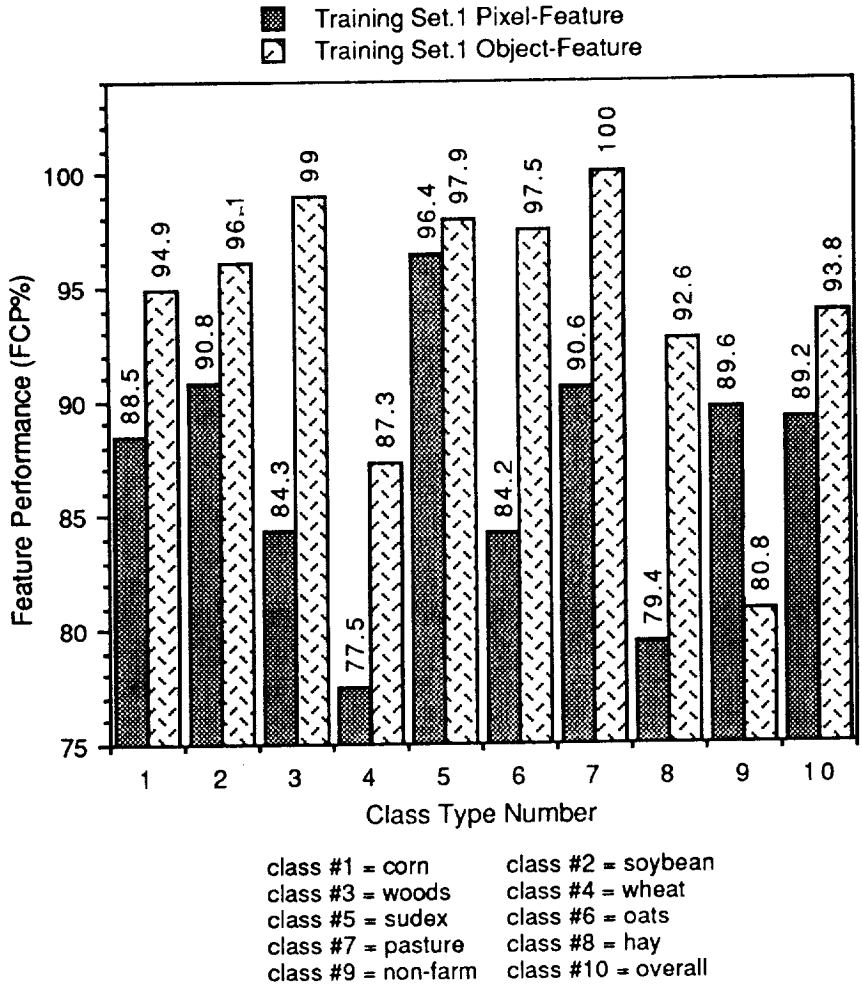


Fig.4.17 Feature performance using training set No.1

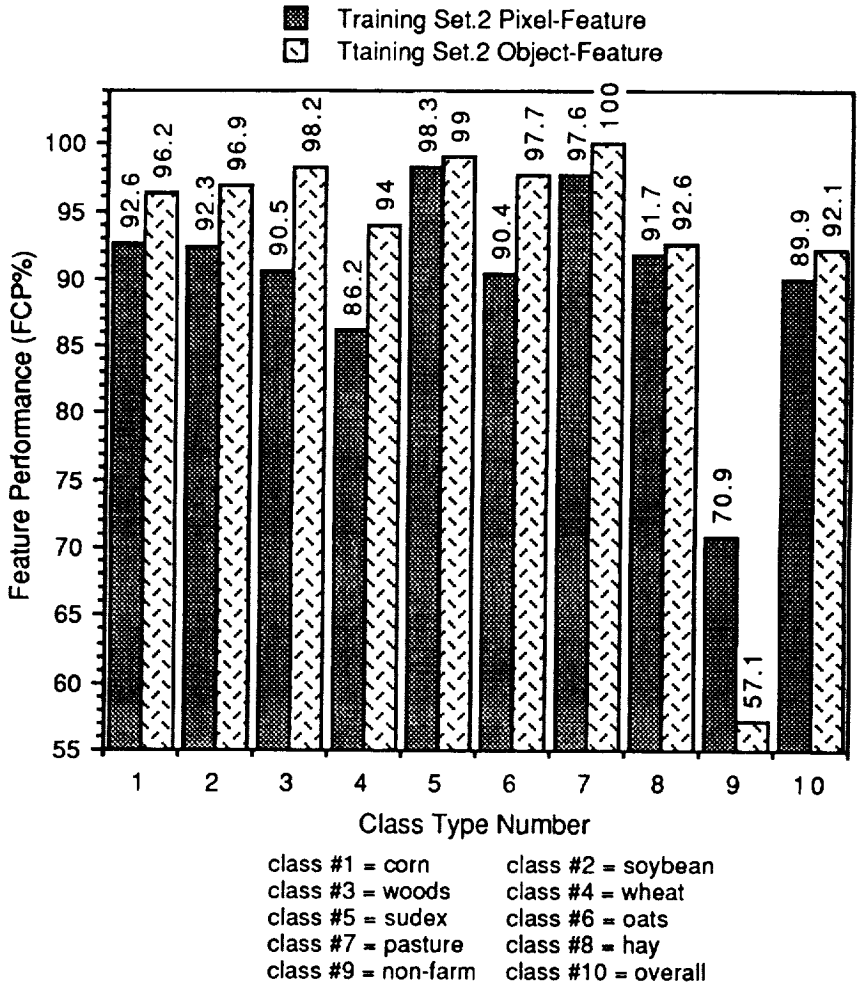


Fig.4.18 Feature performance using training set No.2

Since the classification process is performed in the feature-space rather than in the observation-space, the algorithm is much faster than conventional ones. This results from the fact that the size of feature-space is much smaller than the size of pixel-feature-space. These examples' results show that data redundancy is reduced by a significant amount (in this case, the size of the feature-space for scene representation is reduced by a factor of 27). In addition, the accuracy of information extracted from the object-features (as measured by classification accuracy) is slightly greater than that obtained when using the original pixel-features. It is believed that the classification of ground cover fields using the object-features based on the proposed approach is more accurate and efficient than the point by point classification in the original pixel-features.

4.3.2 Spatial-Feature-Map Appearance

Subjective appearance is an appropriate criterion when some objects in the scene become more important than the others regardless of the size of objects. In such cases it is often difficult to define a mathematical expression for adequately quantifying feature quality. The visual assessment will be used for this kind of qualification and for evaluation of the accuracy of the compaction process in preserving the features of small objects or objects with the complex boundaries. The obvious disadvantage of this criterion is that it is subjective rather than quantitative. The subjective feature evaluation is performed by visually comparing the map of compacted object-feature (called

spatial-feature-map), with the spatial map of the original pixels (called ground-truth-map).

As it pointed out, since the classification performance is dependent on the training samples and the ground-truth-map, the spatial-feature-map appearance is a valuable criterion for feature evaluation. Also, in the feature-map there is a significant within-class information which can be used for even ground-truth-map evaluation.

Spectral information of surrounding pixels is correlated with the center pixel under consideration. In object detection the spectral features of adjacent pixels are considered using neighboring information; thus the object-feature which we represent them in this experiment only by (S_i , L_i) built upon both spectral and contextual information. Therefore, it is expected that the classification accuracy to be higher by using object-feature rather than the individual pixel-feature (notice that we did not consider effect of V_i in the classification of object-feature using M.L. decision rule). Fig.4.19 shows by using the object-feature, for example, the wheat field, which is circled, classified better than when the pixel-features are used for its classification, Fig.20.

A test for robustness of the path hypothesis and accuracy of the unity relation shows that the functional based on path-hypothesis, can detect a single randomly selected pixel in a relatively large soybean field which is replaced by a pixel from some other ground cover types; this pixel is shown in a triangle in Fig.4.19 and Fig.20.

Fig. 4.19 Classification map of object-features (data set N0.1)

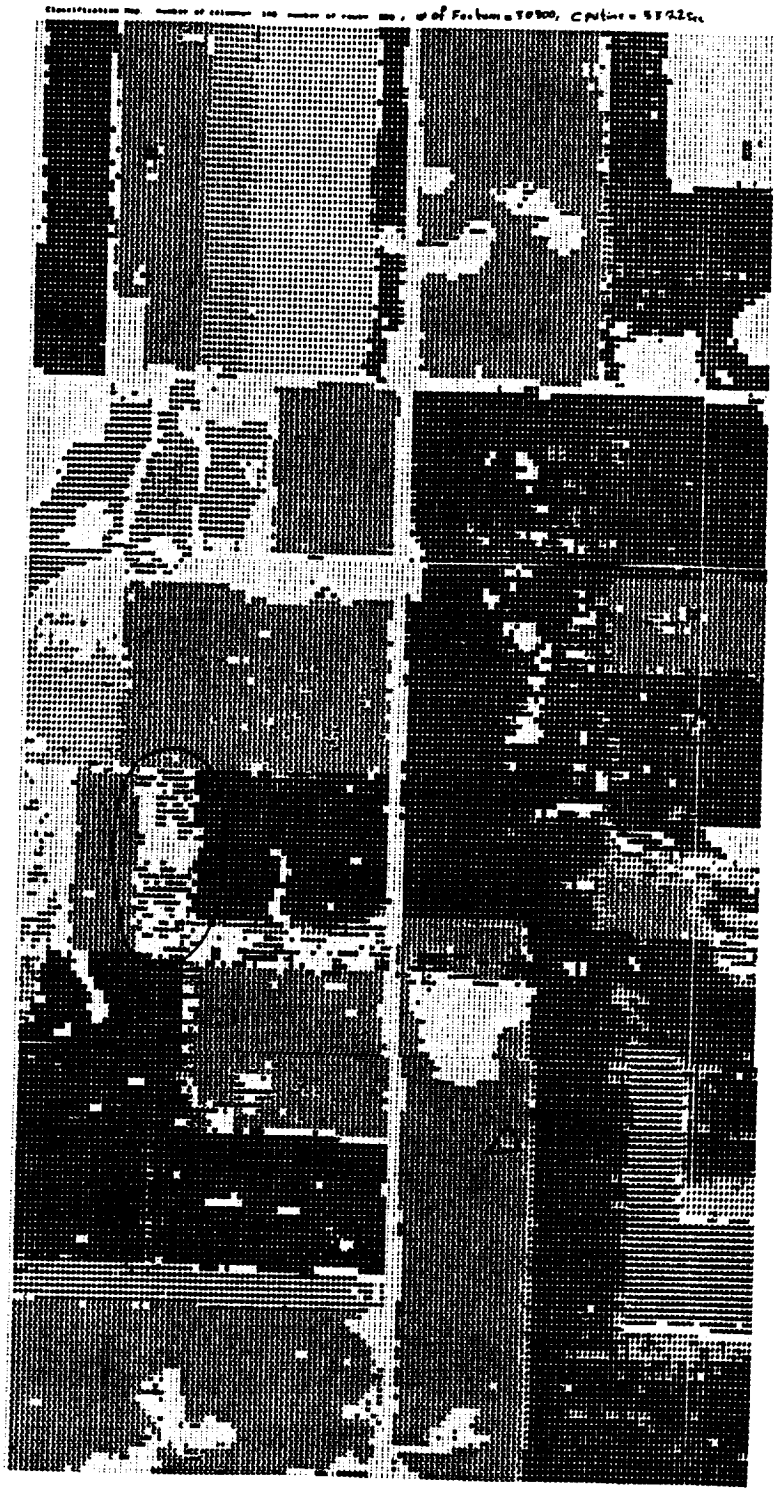


Fig. 4.20 Classification map of pixel-features (data set N0.1)

The appearance of an object in the spatial-feature-map can be intelligently incorporated into the feature selection strategy for extraction of more complex classes in the scene. Figure 4.3 shows the original data ground truth-map data set No.1 and Fig.4.21 is the corresponding spatial-feature-map of data set No.1 after object-feature extraction, which contained 1141 different objects. Fig.4.21 shows that there is significant within-class variation, and thus more information about the scene (e.g., soil type and vegetation condition) might be extracted than will be attempted here, perhaps using even more complex features.

It is often desirable to define boundaries sharply which separate a relatively limited number of objects with different spectral features, but it is not important to preserve the interclass scatter information within the boundaries. The loss of interclass scatter information is roughly equivalent to contouring within the scene and ignoring the contextual features. This kind of illustration is a good tool for evaluation of image data which do not have a ground-truth-map. To illustrate the validity of the unity relation and the path-hypothesis in definition of correct boundaries and accuracy of the spatial-feature-map, several images with the different complexity are used.

The campus-image with the size of 512 by 512 is shown in figure 4.22 which is more complex than the later images. The results of compaction of the campus-image by the factor of 41 is presented in the figure 4.23. Figure 4.24 is the original girl-image with the size of 256 by 256 pixels, and Figure 4.25 shows the spatial-feature-map of the same image after compaction by a factor of 45. Figure 4.26 shows results of image compaction (spatial-feature-

map) by the factor of 36 using the original shuttle-image with the size of 256 by 256 pixels which is illustrated in figure 4.27.

By comparison of the spatial-feature-maps of compacted images by their corresponding original images, the accuracy of the object-feature-extraction technique will be clear.

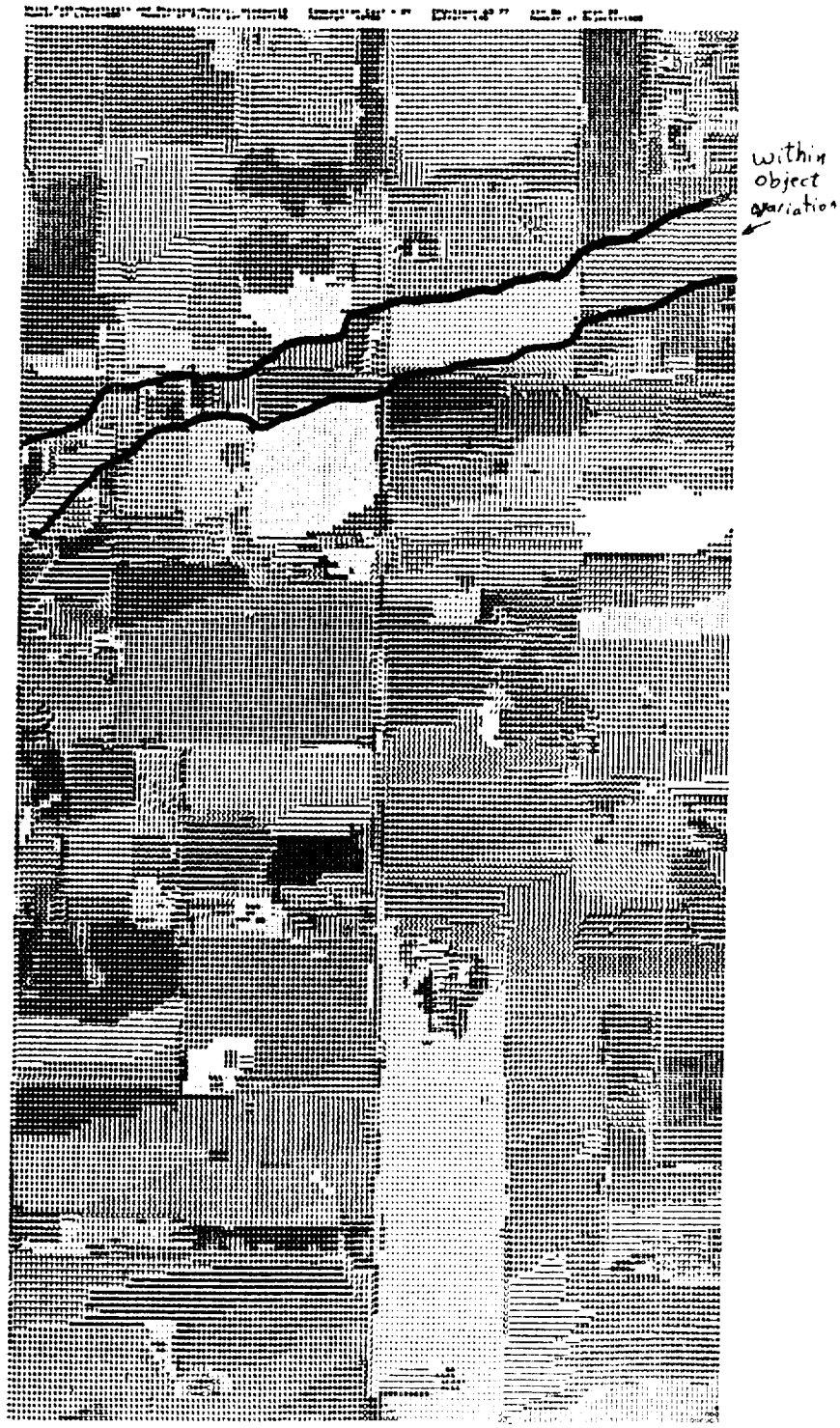


Fig. 4.21 The spatial-feature-map of data set N0.1

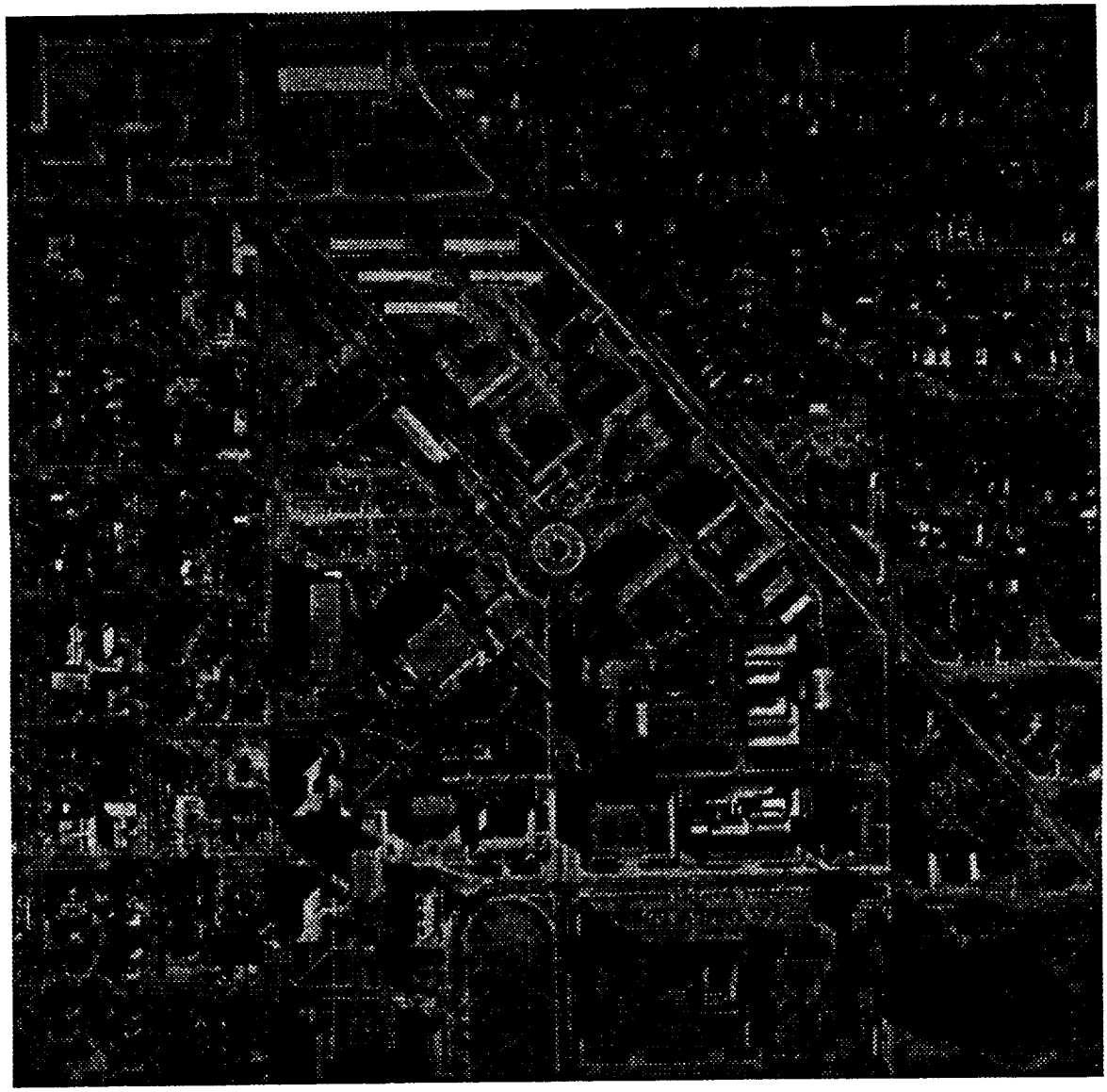


Fig. 4.22 The original 512 x 512 campus-image

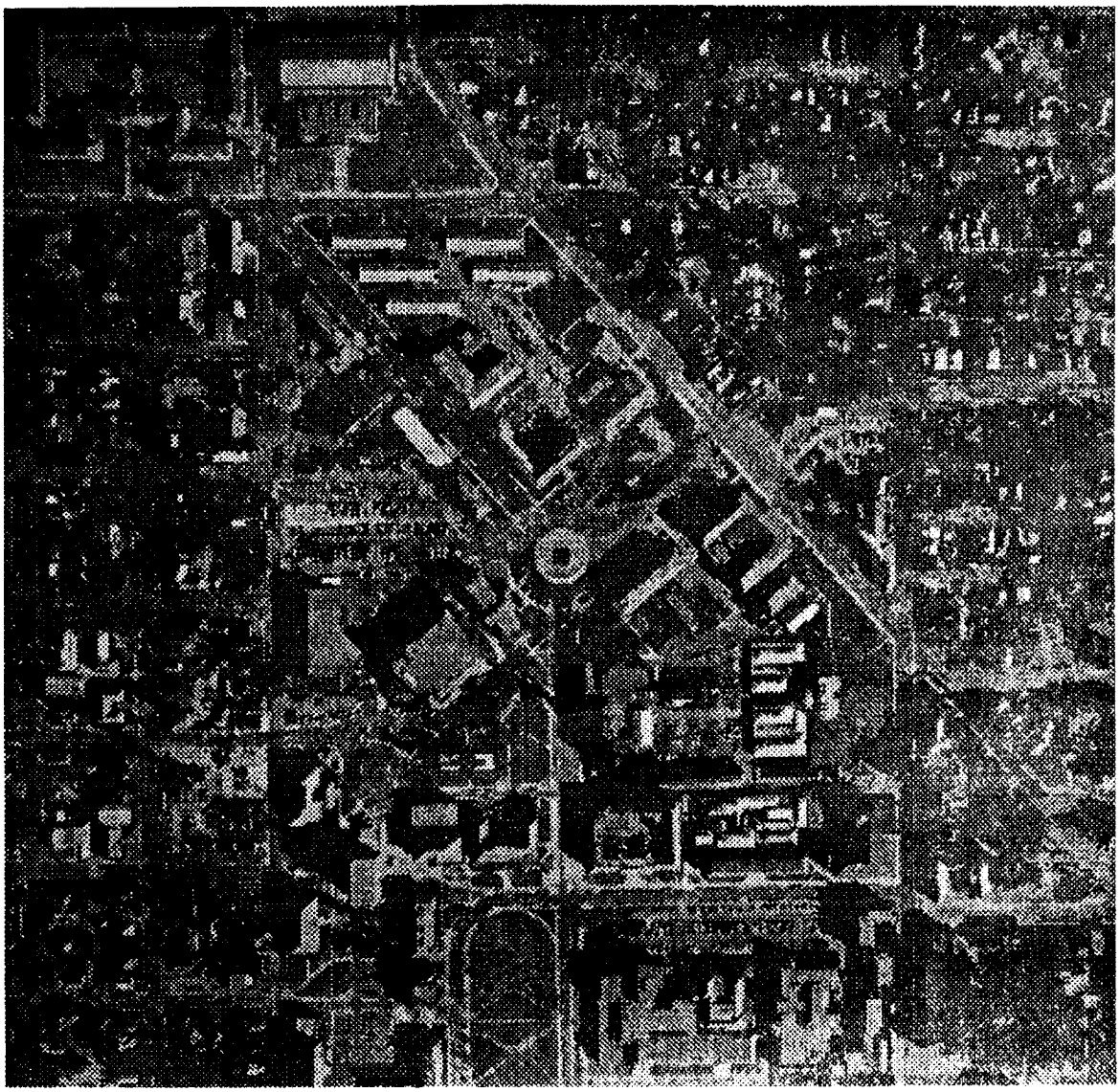


Fig. 4.23 The spatial-feature-map of campus-image after compaction

Fig. 4.24 The original 256 x 256 girl-image

Fig. 4.25 The spatial-feature-map of girl-image after compaction

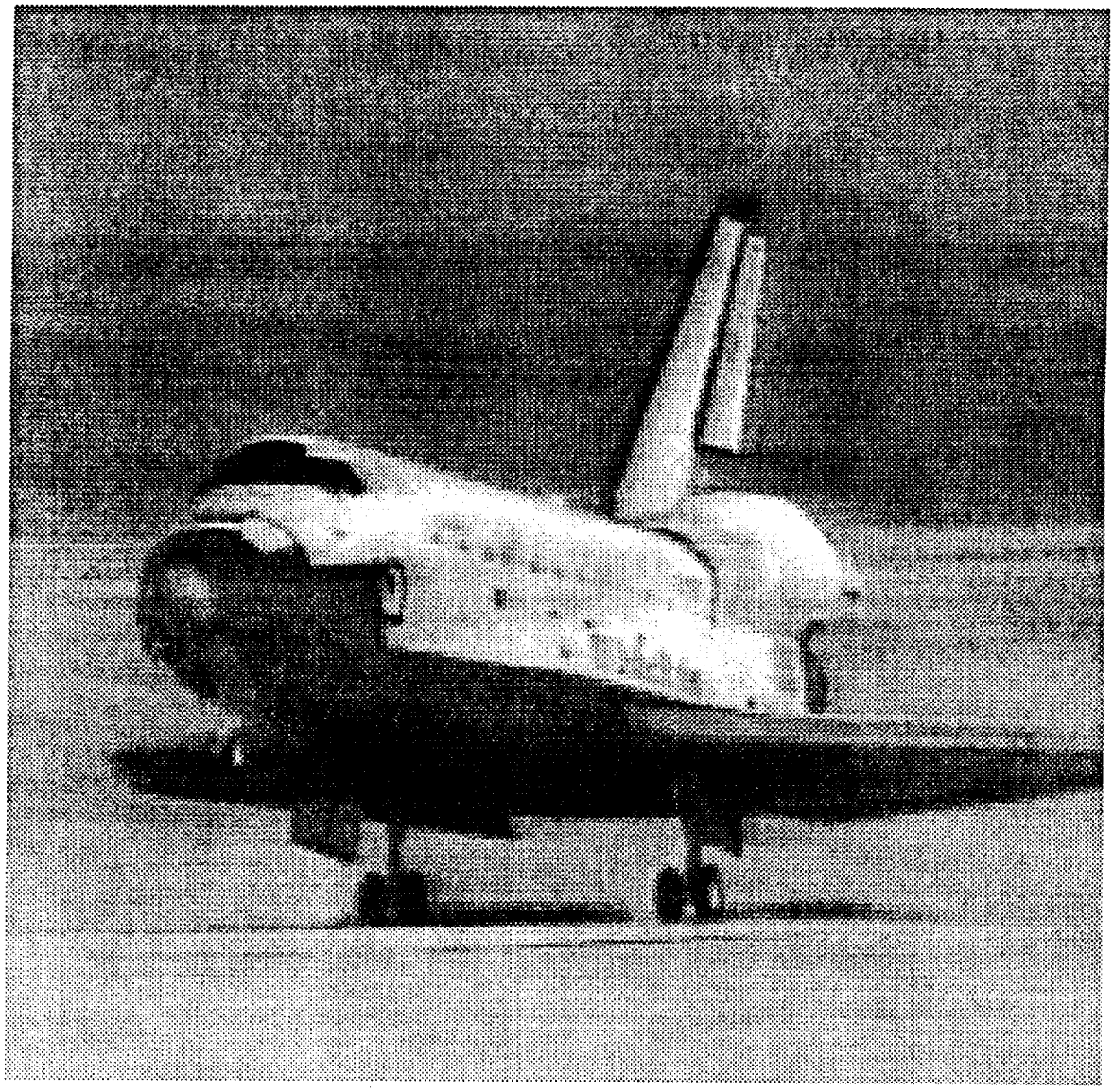


Fig. 4.26 The original 256 x 256 shuttle-image

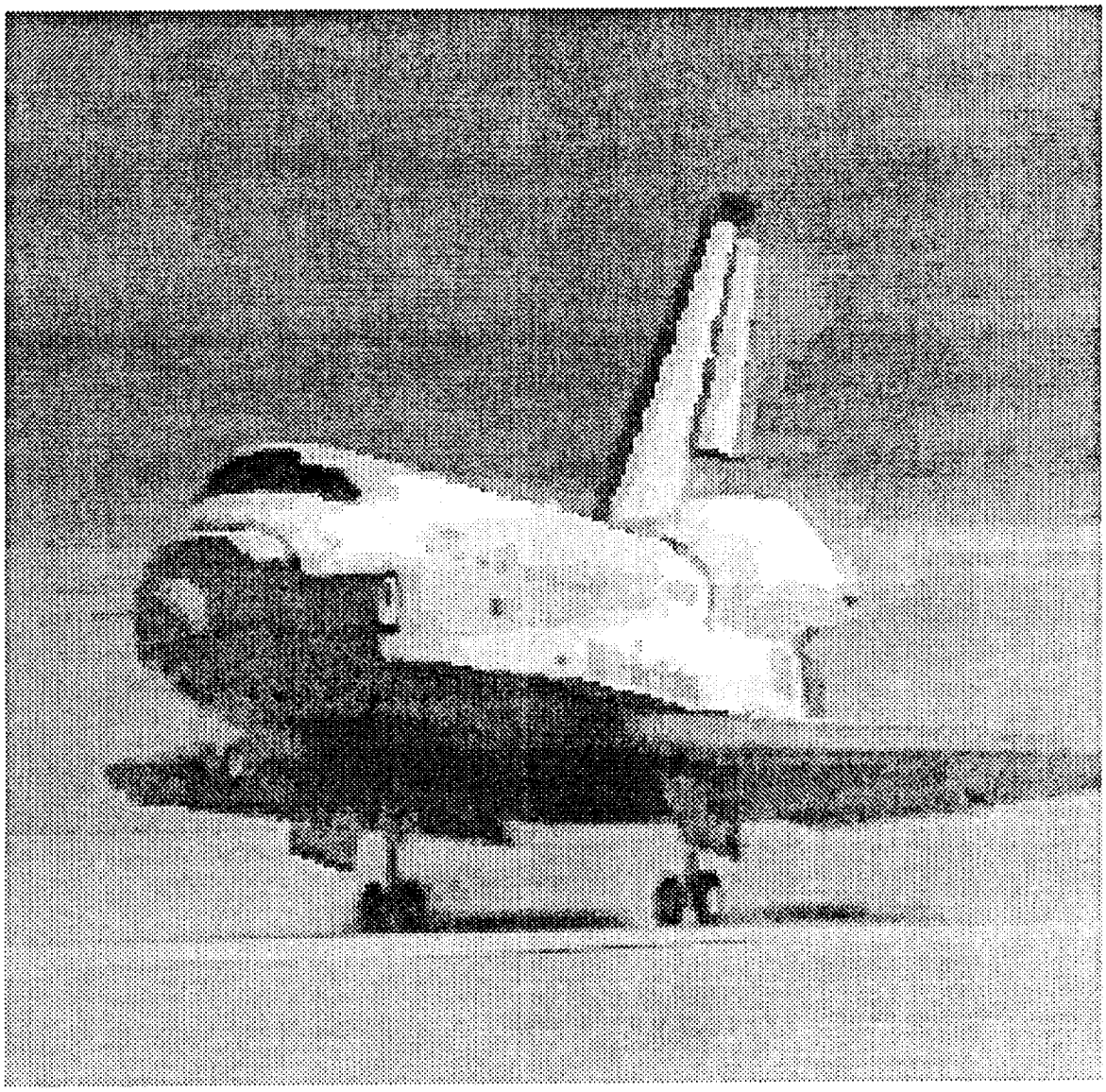


Fig. 4.27 The spatial-feature-map of shuttle-image after compaction

CHAPTER 5
SUMMARY AND CONCLUSION

In order to reduce data redundancy in multispectral imagery we have proposed a model, based on a scene object-description, for multispectral image representation. We have developed an on-line unsupervised object-feature extraction algorithm (called AMICA) which detects the objects by using the unity relation based on the path-hypothesis. The unity relation among the pixels of an object can be defined with regard to the: adjacency relation, spectral-feature and spatial-feature characteristics in an object. AMICA uses the within object pixel-feature gradient vector as a valuable contextual information to construct the object's features, which preserve the class separability information within the data. Based on the path-hypothesis the data read sequentially into the system and the unity relation between a current pixel and the path-segments (objects in the observation space) are examined, the current pixel may be merged into an appropriate object or it will initiate a new object. An object is represented by a relevant object-feature set.

AMICA is implemented to real multispectral image data. The performance of the object-features is compared with the performance of the original pixel-feature. The effect of metric spaces, classifier decision rules, training sets on

the performance of the features are studied. Three different evaluation strategies (overall misplacement error, feature classification performance and subjective object appearance) are selected for comparative feature evaluation using the pixel-features and the object-features.

The experimental results indicate that data volume is reduced by a significant amount (the size of the feature-space for scene representation is reduced by a factor between 20 to 50 which is data dependent). In addition, the accuracy of information extracted from the object-features (as measured by classification accuracy) is greater than that obtained when using the original pixel-features. It is believed that the classification of ground cover fields in the feature-space based on the proposed approach, is more accurate and efficient than the point by point classification in the original space.

The correlation among the adjacent pixels in the image data appears in the form of redundancy in the spectral-spatial features. Spectral information of surrounding pixels is correlated with the center pixel under consideration. In object detection the spectral features of adjacent pixels are considered using neighboring information. Therefore, it is expected that the classification accuracy to be higher by using object-feature rather than the individual pixel-feature. The improvement of the classification performance is a consequence of incorporation of the spatial information in the object-feature extraction decision rule; however, in addition to that, it could be also a consequence of complexity reduction by data compaction (Hughes' phenomenon).

Since the classification process is performed in the feature-space rather than in the observation-space, the algorithm is much faster than conventional ones.

The object appearance in the feature-map can be incorporated (by visual assessment) into the feature selection strategy for extraction of more complex objects in the scene.

AMICA can be used as a boundary finding algorithm by using the spatial-feature-map. It has been implemented to find boundaries in the pictures (e.g., figure of shuttle) as well as for multispectral imagery. In this case the feature-map can be used for further practical application such as detection and recognition of objects to replace the human vision system, in the automatic object recognition.

In summary, AMICA has been used advantageously on two problems concerning feature extraction and compression of multispectral remotely sensed image data. It appears that the proposed object-feature extraction process, for on-line redundancy reduction in the scene representation, has several advantages over most of the conventional techniques.

- The performance with the object-features can provide an improvement in classification accuracy instead of any degradation.

- The process is substantially capable for tracking the complex boundaries.

- The feature extraction process does not require any prior information such as statistical properties of the scene, distribution, shape and size of the object, or number and type of classes in the scene: the process is in this sense completely unsupervised. There is only a very weak dependence on the window size.

- The process allows substantial flexibility for choosing features. Depending on the desired information, different attributes can be emphasized or de-emphasized by selection of an appropriate functional in the measure-space.

- The information extraction process does not need to be preceded by a data de-compaction.

- The compaction process is not iterative and may be implemented on board the sensor platform. i.e. the proposed object-feature extraction could be carried out before transmitting the image data to the receiving terminal. In this case, a reduction of the transmission rate, and, consequently, the required transmitting bandwidth is also achieved. Of course, this solution requires an increase in the processing capabilities at the satellite borne terminal, but the present trend of VLSI digital microelectronics and development in this technology suggests this approach as a technically feasible solution in the future. In this case, the unsupervised object-feature extraction process can be illustrated in Fig. 5.1 as an interface between the data acquisition system and the telecommunication system, which object-

features, rather than pixel-features, are used for data transmission as well as data archiving, distribution and information extraction.

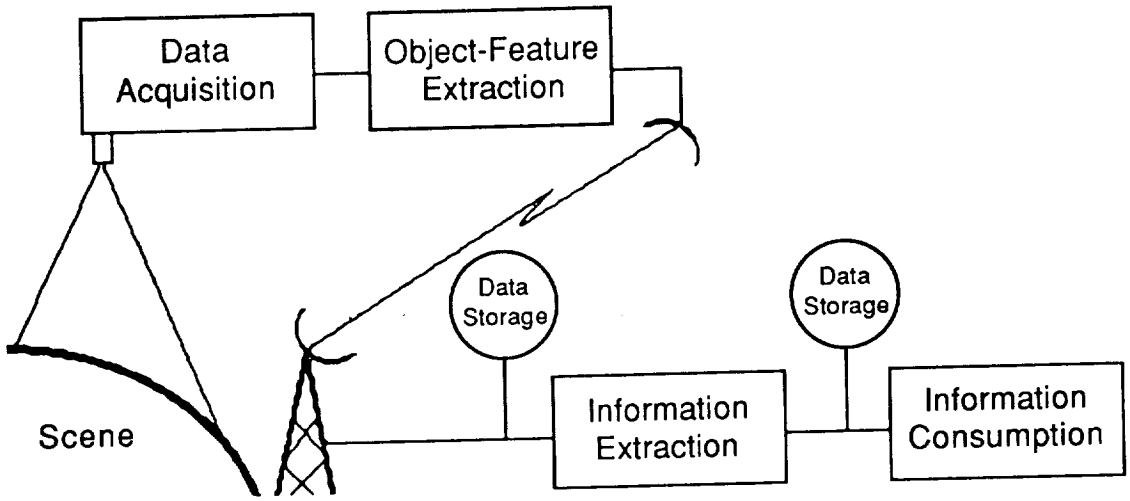


Fig.5.1 A typical application of object-feature extraction

BIBLIOGRAPHY

BIBLIOGRAPHY

List of References

[1] Americal Society of Photogrammetry, "*Manual of Remote Sensing*," second edition, 1983

[2] D.A. Landgrebe "Analysis Technology for Land Remote Sensing," Proc. IEEE, vol.69, no.5, pp.628-642, May 1981

[3] P.H. Swain "Advanced Interpretation Techniques for Earth Data Information Systems," Proc. IEEE, vol 73, no.6, pp 1031-1039, June 1985

[4] P.H. Swain and S.M. Davis, "*Remote Sensing: The Quantitative Approach*," McGraw-Hill, 1978

[5] NASA "HIRIS, the High-Resolution Imaging Spectrometer: Science Opportunities for the 1990s," Earth Observation System, vol IIc, NASA Headquarters instrument panel report, 1987

[6] A.F. Goetz et al "Optical Remote Sensing of the Earth," Proc. of the IEEE, vol.73, no.6, pp.950-969, June 1985

[7] R.G. Gray and L.D. Davisson "A Mathematical Theory of Data Compression," Proc. 1974 Intern. Conf. Commun., pp.40A-1-5, 1974

[8] A.K. Jain "Image Data Compression: a Review," Proc. IEEE, vol 69, March 1981

[9] V. Cappellini "*Data Compression and Control Techniques with Applications*" , Academic Press, 1985

[10] P.A. Wintz "Transform Picture Coding," Proc. IEEE, vol.60, no.7, pp.807-820, July 1972

[11] K. Fukunaga "*Introduction to Statistical Pattern Recognition*" Academic Press, 1972

[12] R.O. Duda and P.E. Hart "*Pattern Classification and Scene Analysis*," Wiley, N.Y. 1973

[13] D.J. Wiersma "Analytical Design of Multispectral Sensors," PhD Thesis, Purdue University, May 1979

[14] K. Fukunaga and W.L.G. Koonts, "Application of the Karhunen-Loeve Expansion to Feature Selection and Ordering," IEEE Trans. on Computers, vol.19, no.4, pp.311-318, April 1970

[15] K. Fukunaga and R.D. Short "Nonlinear Feature Extraction with a General Criterion Function" IEEE Trans. IT, vol.24, pp.600-607, 1978

[16] J.V. Ness "On the Dominance of Non-parametric Bayes Rule Discriminant Algorithms in High Dimensions" Pattern Recognition, vol.12,pp.355-368, 1980

[17] Devijver and J. Kittler "*Pattern Recognition a Statistical Approach*" Prentic Hall, 1982

[18] A.K. Jain and B. Chandrasekaran "Dimensionality and Sample Size Considerations in Pattern Recognition Practice" N-Holland, Handbook of Statistics, vol.2, pp.835-855, 1982

[19] G.V. Trunk "A Problem of Dimensionality; a Simple Example" IEEE Trans. PAMI , vol.1, pp 306-307, 1979

[20] R. M. Haralick and L.G. Shapiro "Image Segmentation Techniques" Computer Vision, Graphics, and Image Precessing 29, pp.100-132, 1985

[21] R. Nevatia "Image Segmentation" Handbook of Pattern Recognition and Image Processing, pp.215-245, 1986

[22] C.R. Brice and C.L. Fennema "Scene Analysis Using Regions" Artificial Intelligence 1, pp.205-226, 1970

[23] T.V. Robertson , P.H. Swain and K.S. Fu, "Multispectral Image Partitioning," Technical Report TR-EE 73-26, School of Electrical Engineering, Purdue University, 1973

[24] J.N. Gupta and P.A. Wintz, "A Boundary Finding Algorithm and its Applications," IEEE Trans. on Circuits and Systems, vol.22, no.4, pp.351-362, April 1975

[25] R.L. Kettig and D.A. Landgrebe "Classification of Multispectral Image Data by Extraction and Classification of Homogeneous Objects" IEEE Trans. GEO., vol. GE-14, pp. 19-26, 1976

[26] A.G. Wacker and D.A. Landgrebe, "Boundaries in Multispectral Imagery by Clustering," IEEE Symp. on Adaptive Processes, University of Texas at Austin, Dec. 1970

[27] T.L. Huntsberger, C.L. Jacobs, and R.L. Cannon "Iterative Fuzzy Image Segmentation" Pattern Recognition. vol.18, pp.131-137, 1985

[28] J. Sklansky "Image Segmentation and Feature Extraction," IEEE Trans. SMC vol. 8, no.4, pp 237-246, April 1978

[29] F.S. Cohen and D.B. Cooper "Simple Parallel Hierarchical and Relaxation Algorithms for Segmenting Noncausal Markovian Random Fields," IEEE Trans. on PAMI, vol.9,no.2, March 1987

[30] H. Samet "The Quadtree and Related Hierarchical Data Structuctures," Computing Surveys, vol.16, no.2, June 1984

[31] D.A. Landgrebe "The Development of a Spectral-Spatial Classifier for Earth Observational Data," Pattern Recognition vol.12, pp.165-175, 1980

[32] John A. Richards "*Remote Sensing Digital Image Analysis*," Springer Verlag, Berlin, Heidelberg, 1986

[33] R.L. Kettig "(Appendix A, Spatial Correlation) Computer Classification of Remotely Sensed Multispectral Image Data by Extraction of Homogeneous Objects," PhD thesis, Purdue University, May 1975

[34] W.F. Schreber, "The Measurement of Third Order Probability Distribution of T.V. Signals," IRE Trans. on Information Theory, vol.IT2, pp.94-105, September 1952

[35] A.J. Seyler, "The Coding of Visual Signals to Reduce Channel Capacity Requirements," Proc. IEE, vol.109, pp.676-684 July 1962

[36] G.F. Hughes "On the Mean Accuracy of Statistical Pattern Recognizers" IEEE Trans. on Information Theory, vol. IT-14, no.1, pp.55-63, 1968

[37] V.A. Kovalevsky "*Image Pattern Recognition*," Springer Verlay, New York, Berlin, 1980

[38] Y.A. Shreider "*What Is Distance?*" The University of Chicago Press, 1974

[39] F. H. Clarke "*Optimization and Nonsmooth Analysis*," Wiley Interscience Publication, 1983

[40] A.G. Wacker, "Minimum Distance Approach to Classification," Doctoral Thesis, Purdue University, West Lafayette, IN, January 1972

[41] M.R. Anderberg "*Cluster Analysis for Application*" Academic press, N.Y., 1971

General References

[42] Broder, A.Z. et al "On the Performance of Edited Nearest Neighbor Rules in High Dimensions," IEEE Trans. SMC, vol.15, no.1, January 1985

[43] Chellappa, R. "Two-Dimensional Discrete Gaussian Markov Random Field Models for Image Processing" Proc. of Int. Conf. on SMC 1983

[44] Clarke, D.A. "*Fundation of Analysis with an Introduction to Logic and Set Theory*" 1971

[45] Conners, R.W. and Harlow, C.A. "A Theoretical Comparison of Texture Algorithms" IEEE Trans. on PAMI, vol.2, NO.3, May 1980

[46] Cushnie, J.L. "The Interactive Effect of Spatial Resolution and Degree of Internal Variability within Land-cover Types on Classification Accuracies" INT.J. Remote Sensing, vol.8, no.1, pp15-29, 1987

[47] Davission, L.D., "The Theoretical Analysis od Data Compression Systems," Proc. IEEE, vol.56, pp.176-186, 1968

[48] Day, W.E., and R.S. Wells "Extremes in the Complexity of Computing Metric Distances Between Patterns" IEEE Trans. on PAMI, vol.6, n0.1, January 1984

[49] DiGesu and Maccarne "Feature Selection and Possibility Theory" P.R. vol.19, pp.63-72, 1986

[50] Fletcher, R. "*Optimaization*," Academic Press, 1969

[51] Fu, K.S. and D.A. Landgrebe and Phillips "Information Processing of Remotely Sensed Agricultural Data" Proc. IEEE, vol57, pp.639-653, 1969

[52] Fukunaga, K. and Flick "Classification Error for a Very Large Number of Classes" IEEE Trans. PAMI, vol.6, pp.779-788, 1984

[53] Fukunaga, K. and R.D. Short "A Class of Feature Extraction Criterion and its Relation to the Bayes Risk Estimate" IEEE Trans. IT, vol.26, 1980

[54] Fukunaga, K. and Filck "A Test of the Gaussian-ness of a Data Set Using Clustering" IEEE Trans. PAMI vol.8, pp.240-247, 1986

[55] Gilchrist, W. "*Statistical Modeling*," Wiley, 1984

[56] Gupta, M.M "*Fuzzy Information, Knowledge Representation and Decision Analysis*," 1983

[57] Habibi, A., P.A. Wintz, "Image Coding by Linear Transformation and Block Quantization," IEEE Trans. Commun. Tech., vol.Com-19, pp.50-62, 1971

[58] Haralick, R.M. and Hyonam "A Context Classifier" IEEE Trans. on GRS, vol GE.24, NO.6 , Nov. 1986

[59] Kalayeh, H.M., Muasher and D.A. Landgrebe "Feature Selection with Limited Training Samples" IEEE Trans. Geo., vol.GE-21, pp.434-438, 1983

[60] Kandel, A. "*Fuzzy Mathematical Techniques with Application" Addison-Wesley*, 1986

[61] Kanefsky and Fong "Predictive Source Code Techniques Using Maximum Likelihood Prediction for Compression of Digitized Images" IEEE Trans. on IT, vol. IT-30, NO 5, Sept. 1984

[62] Kittler, J. "A Method of Feature Selection for High Dimensional Problems" conf. on P.R. 1974

[63] Kovalevsky, V.A. "Local Versus Global Decisions in Image Recognition" Proc. IEEE vol. 67, pp 745-752, 1979

[64] Lierop, M.P. "Geometric Transformation on Pictures Represented by Leafcodes" Computer Vision, Graphics, and Image Procesing 33, pp.81-98, 1986

[65] Lowitz, G.E. "What the Fourier Transform Can Really Bring to Clustering" P.R. vol.17, pp.657-665, 1985

[66] Mero, L. and T. Vamos "Medium Level Vision" Progress in Pattern Recognition, 1981

[67] Morgera, S.D "Linear, structured covariance estimation: An application to pattern classification for remote sensing," Patten Recognition Letters, pp.1-7, February 1986

[68] Neustadt,Lucien W. "*Optimization, A Theory of Necessary Conditions*," Princeton University Press, 1976

[69] Parthasarathy, C. and B.N. Chatterji "The Use of Data Windows in Feature Extraction for High Dimensional PR Problems" Pattern Recognition Letters 4, February 1986

[70] Plastria, F. "Two Hierarchies Associated with each Clustering Scheme" P.R. vol.19, pp.193-196, 1986

[71] Rao, M.M. "*Real and Stochastic Analysis*," Wiley, 1986

[72] Richards, J.A., D.A. Landgrebe, and P.H. Swain "Pixel Labeling by Supervised Probabilistic Relaxation" IEEE Trans. on PAMI, vol.3, NO.2, March 1981

[73] Ritter, G.X., and P.D. Gader "Image Algebra Implementation on Cellular Array Computers" IEEE, 1985

[74] Short, R.D. and K. Fukunaga "The Optimal Distance Measure for Nearest Neighbor Classification" IEEE Trans. IT, vol.27, pp.622-627, 1981

[75] Talmon, J.T. "A Multiclass Nonparametric Partitioning Algorithm" P.R. Feb. 1986

[76] Tilton, J.C., and P.H. Swain "Incorporating Spatial Context into Statistical Classification of Multidimensional Image Data" Agristars, Purdue University LARS, August 1981

[77] Wang, D.C. and A.H. Vagnucci "Gradient Inverse Weight Smoothing Scheme and the Evaluation of its Performance," Comuter Graphics and Image Processing 15, 167-181, 1981

[78] Weszka, J.S., Dyer, C.R. and Rosenfeld, A. "A Comparative Study of Texture Measure for Terrain Classification" IEEE Trans. on SMC, vol.6, NO.4, April 1976

[79] Wharton, S.W. "An Algorithm for Computing the Number of Distinct Spectral Vector in Thematic Mapper Data" IEEE Trans. on GRS, vol. GE.23, NO 1, Jan. 1985

[80] Watada, Montonami, Tanaka, and Asai "Discriminant Analysis Based on Fuzzy Distance" IFAC Fuzzy Information Marseille, France, (1983)

[81] Yamashita, M. "Distances Defined by Neighborhood Sequences" P.R, vol.19, pp.237-246, 1986

[82] Zhang and Haralick "Contextual Classification by Stochastic Relaxation" IEEE conf. pp.399-407, (1985)

APPENDICES

Appendix A

Uncertainty

It is important to realize, regardless of the similarity measure selected, that the measurements are not precise and there is always some uncertainty in the observations. The effect of this uncertainty should be considered in the selection of a system of mathematical models. There are several sources of uncertainty, and since the type of uncertainty involved in our approach will effect the choice of the system model, it is important to distinguish between them. The main sources of uncertainty in systems are as follows:

Inaccurate measurements: Inexact measurements can cause uncertainty in models of physical processes which are absolutely deterministic. For example, measurement accuracy of a certain amount of electromagnetic radiation energy by the sensor, regardless of the quality of the sensor system, depends on the number of digitized gray-levels.

Random occurrences: If the outcome of a physical process is believed to be random, regardless of the measurement accuracy, there is another type of uncertainty. There is an element of concern about the evolution of the process which is unaffected by environmental imprecision. For example, withdrawing only one seed from a box containing a mixture of 50% wheat and 50% oat has two equally likely, mutually exclusive outcomes (Oat or Wheat) per each trial. Thus the evolution of a sequence of identical and independent trials of this experiment can not be predicted with certainty. Models of processes which exhibit this kind of uncertainty are called stochastic or probabilistic models. For a

fair mixture of the oat and wheat, we have the a prior idea that the probability of each outcome is equal to 0.5; however, this value will change to zero or one after the trial was determined.

Vague descriptions: There is an element of uncertainty which is not caused by measure error nor by random occurrence. Assume that all members of a set **P** are deterministic and fixed. Let X_k be a deterministic member of P. There is uncertainty in the determination of the set of all members from **P** which are similar to X_k because of vagueness in the similarity criterion. Neither a deterministic model nor a stochastic model is suitable for this physical situation, which manifests a source of non-statistical uncertainty or fuzziness. In a deterministic set every element has a precisely defined criterion of membership equal to either zero or one. In the case of a fuzzy set the class of objects need not necessarily either belong or not belong to this class. Here each object may have intermediate grades of membership, ranging from zero to one. It is important to realize that the class membership is fixed and will not change even after the trial was determined.

Appendix B

Data Statistic

Test area Data set NO.1 containing:

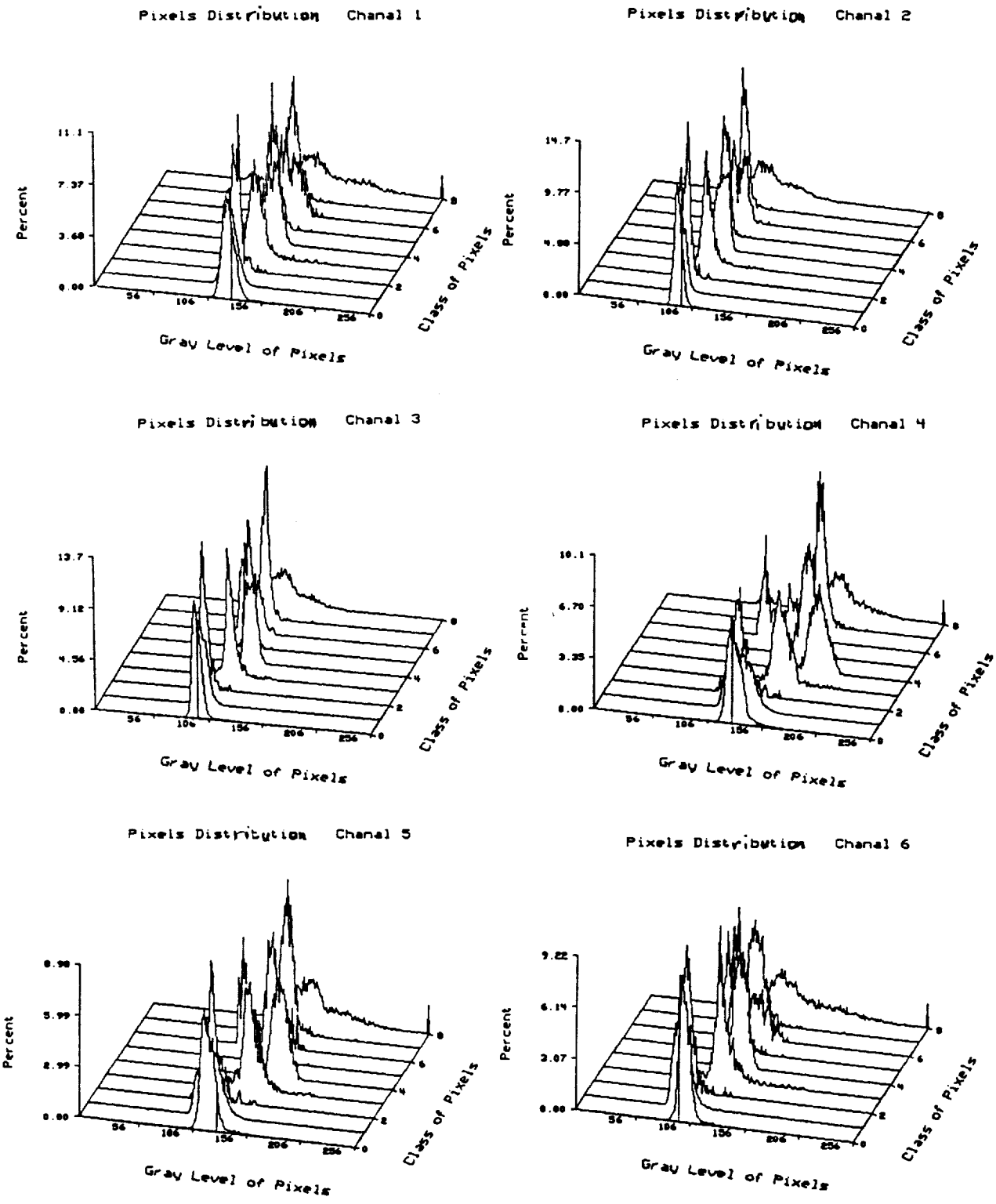
Number of lines/frame = 220

Number of pixels/line = 140

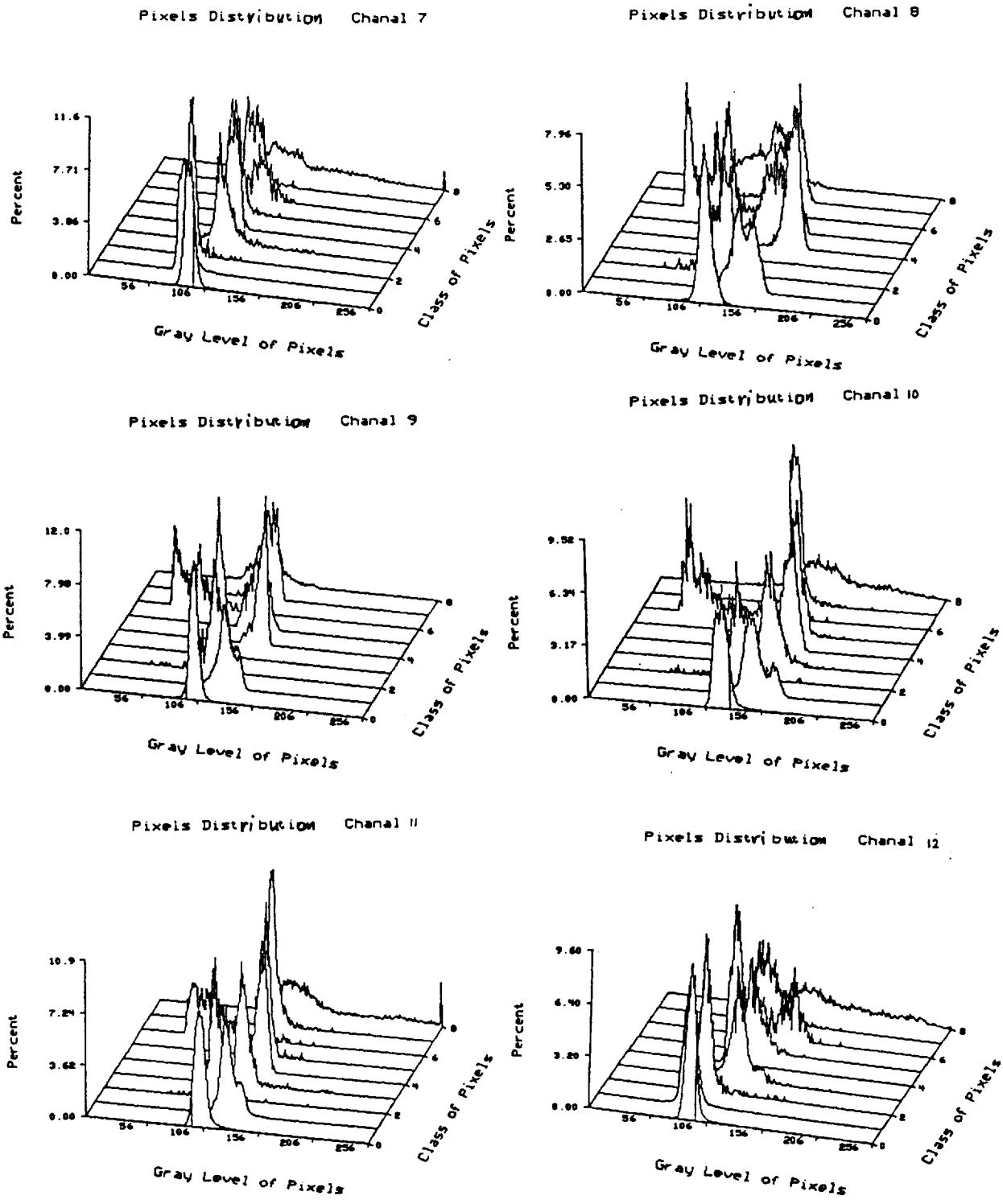
Number of channels = 12

Data set NO.1 test area class statistic

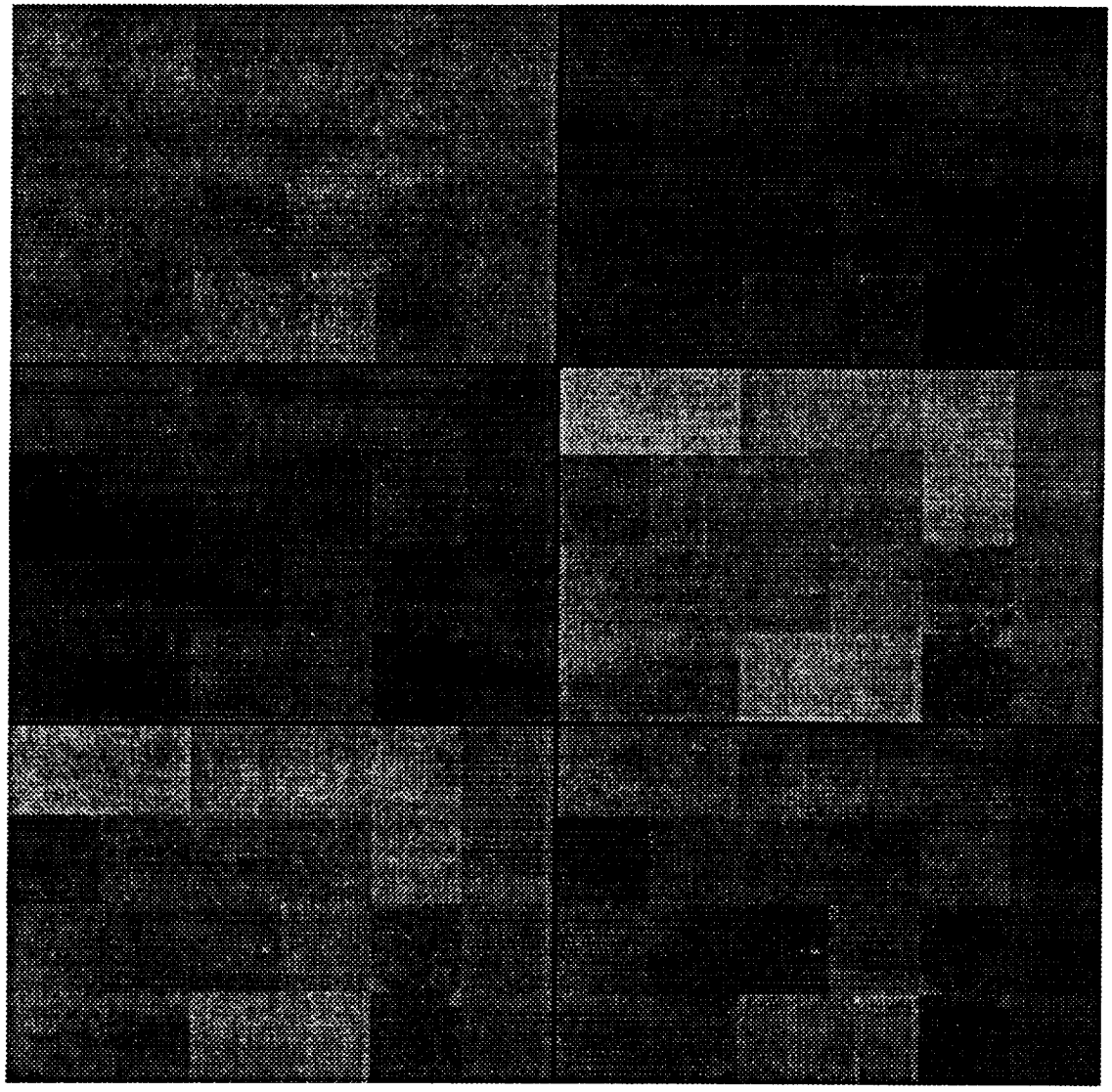
Class type		Description	# of pixels	Percentage
C	1	Corn	10104	33%
S	2	Soybeans	12910	42%
f	3	Woods	389	1.3%
W	4	Wheat	944	3.1%
x	5	Sudex	1219	3.9%
o	6	Oats	603	2%
P	7	Pasture	339	1.1%
h	8	Hay	746	2.4%
:	9	Non-Farm	3546	11.5%



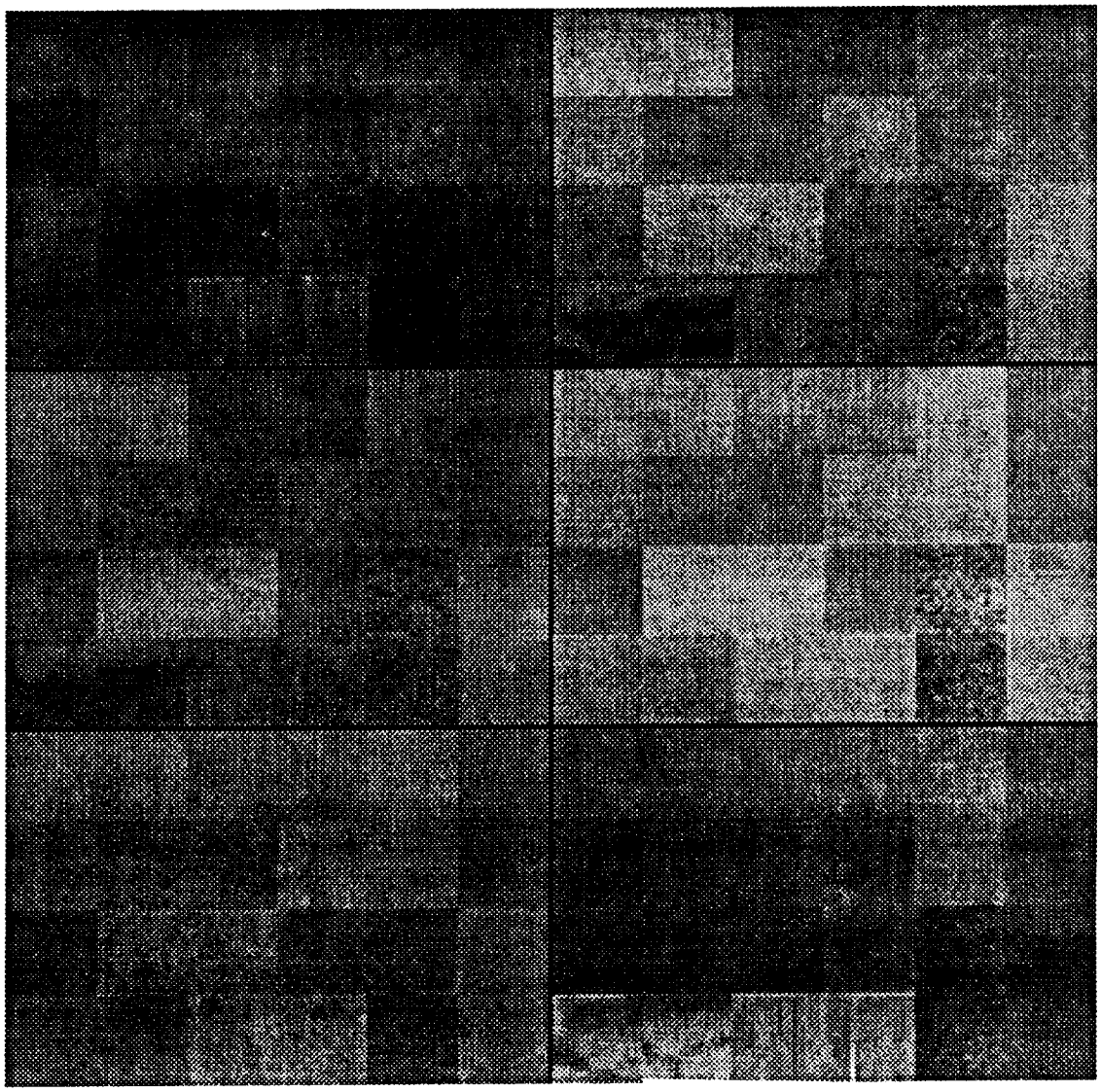
Data set NO.1 class distribution



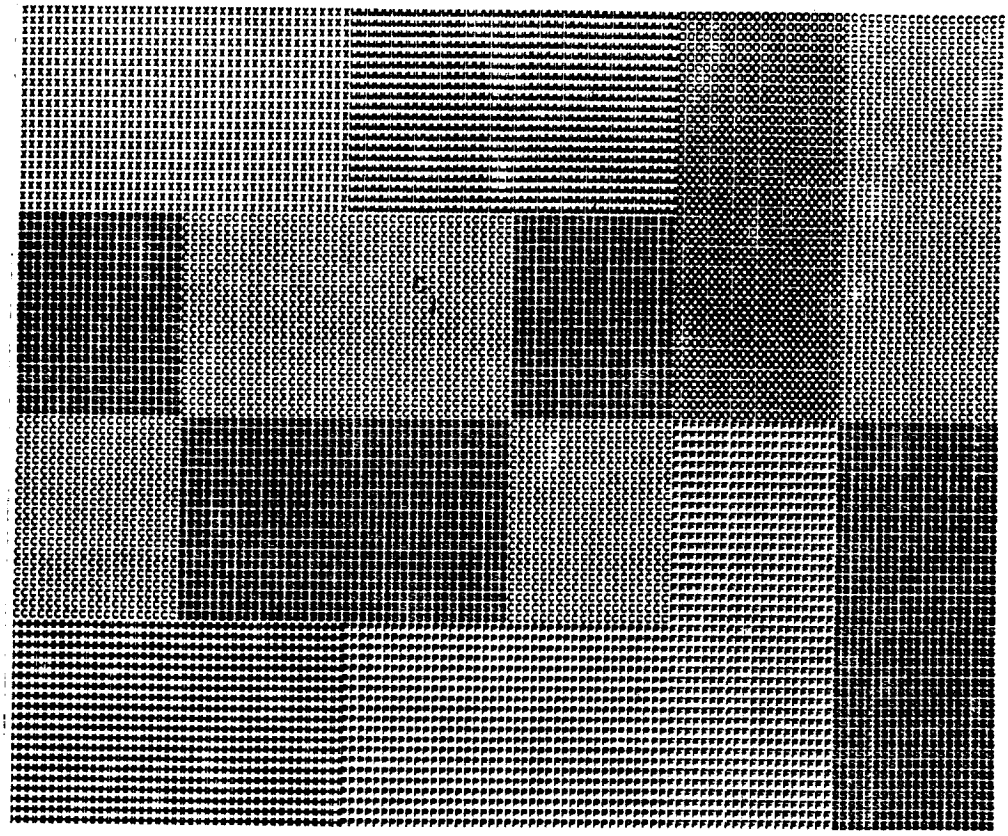
Data set NO.1 class distribution



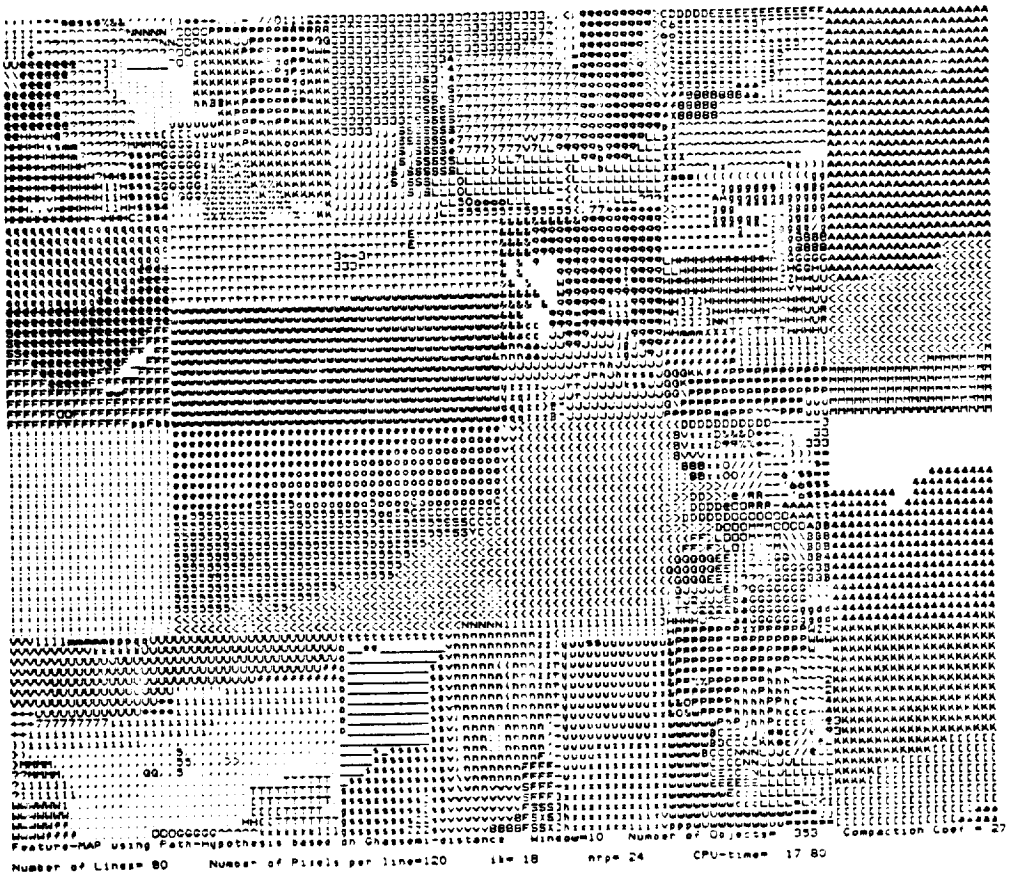
Spectral imagery of test area data set NO.2 from channel-1 to channel-6



Spectral imagery of test area data set NO.2 from channel-7 to channel-12



Ground-truth-map of data set NO.2



The spatial-feature-map of data set N0.2

Class-mean Vector is:

Class-covariance Matrix is:

Samples are from Sudex-field. Number of samples is 800.

130.69	41.40	18.66	17.53	19.88	18.56	14.34	18.13	24.90	12.69	21.21	15.37	13.21
97.89	18.66	18.96	12.90	16.56	15.84	14.19	14.59	18.69	10.88	17.87	12.74	10.57
110.74	17.53	12.90	17.12	18.48	17.67	12.48	13.58	17.31	8.71	15.58	10.68	8.52
171.58	19.88	16.56	18.48	44.67	29.24	22.09	18.79	22.54	14.61	23.84	15.55	7.74
150.09	18.56	15.84	17.67	29.24	34.63	19.36	18.94	21.28	10.06	20.13	12.56	8.64
114.98	14.34	14.19	12.48	22.09	19.36	25.58	14.89	15.41	13.62	18.98	14.89	9.14
95.75	18.13	14.59	13.58	18.79	18.94	14.89	18.43	17.62	9.58	17.55	11.63	9.48
156.73	24.90	18.69	17.31	22.54	21.28	15.41	17.62	39.80	18.22	29.58	20.35	10.48
129.88	12.69	10.88	8.71	14.61	10.06	13.62	9.58	18.22	26.63	24.21	19.21	2.89
148.17	21.21	17.87	15.58	23.84	20.15	18.98	17.55	29.58	24.21	41.98	24.80	9.66
123.97	15.37	12.74	10.68	15.85	13.56	14.89	11.83	20.35	19.21	24.80	24.22	7.25
95.08	13.21	10.57	8.52	7.74	8.69	9.14	9.48	10.48	2.89	9.66	7.25	19.32

Samples are from Oats-field. Number of samples is 800.

130.50	28.97	6.95	4.84	1.82	5.87	9.20	14.92	-9.83	-0.48	10.85	7.61	15.78
96.94	6.95	9.41	3.39	1.69	4.98	8.16	11.07	-7.32	-1.54	6.70	4.31	11.92
103.59	4.84	3.29	6.52	4.51	4.93	5.77	6.25	-1.99	0.22	3.58	1.98	6.59
150.03	1.82	1.69	4.51	16.37	8.82	5.11	2.51	5.29	2.06	-0.31	0.01	2.70
135.44	5.87	4.98	4.93	8.82	16.40	9.29	9.62	-1.30	0.44	3.45	3.17	11.82
110.59	9.20	8.16	5.77	5.11	9.29	19.13	17.03	-11.80	-6.37	7.36	6.82	21.79
98.29	14.92	11.07	6.25	2.51	9.62	17.03	27.73	-20.18	-6.55	13.07	10.66	29.48
118.20	-9.83	-7.32	-1.99	5.29	-1.30	-11.80	-20.18	45.64	21.93	-7.59	-11.13	-31.03
110.31	-0.48	-1.54	0.22	2.06	0.44	-6.37	-6.55	21.93	24.21	1.18	-3.51	-15.73
148.93	10.85	6.70	3.58	-0.31	3.45	7.36	13.07	-7.59	1.18	17.66	11.90	15.06
123.47	7.61	4.31	1.98	0.01	3.17	6.82	10.66	-11.11	-3.51	11.90	25.33	19.30
113.17	15.78	11.92	6.59	2.70	11.82	21.79	29.48	-31.03	-15.73	15.06	19.30	58.36

Samples are from Pasture-field. Number of samples is 800.

132.73	44.88	19.78	13.41	12.16	17.68	25.76	37.43	-13.06	-4.12	18.96	15.51	33.04
99.44	19.78	30.49	9.50	8.37	13.32	23.42	29.39	-16.93	-3.48	15.93	14.11	26.96
106.14	13.41	9.50	11.42	9.09	11.70	13.64	18.49	-4.96	-1.61	8.66	6.01	15.27
153.48	12.16	8.37	9.09	18.66	13.14	14.06	15.88	0.83	1.64	6.90	3.44	5.79
140.15	17.68	13.32	11.70	13.14	31.42	21.35	26.36	-4.64	-2.56	10.98	7.42	20.58
117.59	25.76	23.42	13.64	14.06	21.35	46.53	42.61	-16.16	-6.84	22.58	19.16	45.08
105.18	37.43	29.39	18.49	15.60	26.36	42.61	61.47	-26.04	-10.42	28.45	24.88	61.11
109.42	-13.06	-16.93	-4.96	0.83	-4.64	-16.16	-26.04	35.96	14.40	-3.87	-9.10	-45.73
103.64	-4.12	-3.48	-1.61	1.64	-2.56	-6.84	-10.42	14.40	16.70	4.07	-4.60	-32.10
151.49	18.96	15.93	8.66	6.90	10.98	22.58	28.45	-3.87	4.07	43.55	29.31	20.09
130.06	15.51	14.11	6.01	3.44	7.42	19.16	24.88	-9.10	-4.60	29.31	37.56	31.54
132.27	33.04	26.96	15.27	5.79	20.58	45.08	61.11	-45.73	-32.10	20.09	31.54	126.13

Samples are from Hay-field. Number of samples is 800.

120.15	28.43	10.32	4.48	-3.70	0.45	6.36	14.26	-24.78	-19.32	4.38	6.38	38.37
88.85	10.32	10.83	3.67	-1.58	1.51	7.11	11.07	-14.59	-12.31	3.05	4.94	23.82
92.27	4.48	3.67	14.56	30.21	22.35	8.14	0.90	48.05	25.08	12.32	7.68	-31.70
126.63	-3.70	-1.58	30.21	114.64	79.53	28.85	-15.37	190.68	112.70	38.03	18.99	-153.20
116.19	0.45	1.51	22.35	79.53	58.48	20.24	-0.93	132.04	67.75	24.38	14.01	-96.93
97.85	6.36	7.11	8.14	28.85	20.24	27.96	18.79	14.20	-0.28	6.71	10.12	-3.32
89.89	14.24	11.07	0.90	-15.37	-0.93	18.79	38.51	-65.99	-50.10	-5.67	5.33	64.20
93.13	-24.78	-14.59	48.05	190.68	132.04	14.20	-65.99	406.02	241.20	67.54	25.64	-343.74
94.54	-19.32	-12.31	25.08	112.70	67.75	-0.28	-50.10	241.20	183.18	43.33	14.53	-217.00
131.32	4.38	3.05	12.32	38.03	24.38	6.71	-5.67	67.54	43.33	52.17	21.23	-31.54
111.71	6.38	4.94	7.68	18.99	14.01	10.12	5.33	25.64	14.53	21.23	23.63	2.25
130.87	38.37	23.82	-31.70	-153.20	-96.45	-3.32	64.20	-343.74	-217.00	-31.54	2.25	402.02

Class-mean Vector is:

Class-covariance Matrix is:

Samples are from Corn-field. Number of samples is 3400.

123.79	31.53	5.63	6.47	10.80	10.33	7.07	7.05	5.48	5.50	9.78	5.23	-0.02
92.01	5.63	8.32	5.25	9.39	9.51	6.76	5.55	4.80	4.45	8.00	4.02	0.57
94.48	6.47	5.25	9.52	12.34	11.16	8.33	6.47	5.77	4.57	7.36	3.39	0.15
138.43	10.80	9.39	12.34	38.89	22.17	16.09	11.42	11.28	7.31	9.69	6.08	-0.04
119.03	10.33	9.51	11.16	22.17	23.73	14.83	10.76	9.83	6.61	9.59	3.96	0.41
99.86	7.07	6.76	8.33	16.03	14.83	14.44	8.51	7.38	5.73	9.61	4.93	1.31
91.58	7.05	5.55	6.47	11.42	10.76	8.51	10.28	6.63	4.71	10.24	5.14	1.57
111.46	5.48	4.80	5.77	11.28	9.83	7.38	6.63	12.56	5.53	6.55	3.25	-0.54
103.56	5.50	4.45	4.57	7.31	6.61	5.73	4.71	5.53	9.01	13.71	7.97	-0.83
121.06	9.78	8.00	7.38	9.69	9.59	9.61	10.24	6.55	13.71	40.42	22.48	6.96
102.80	5.23	4.02	3.39	4.08	3.96	4.93	5.14	3.25	7.97	22.48	20.38	5.51
90.63	-0.02	0.57	0.15	-0.04	0.41	1.31	1.57	-0.54	-0.83	6.96	5.51	16.59

Samples are from Soybeans-field. Number of samples is 3400.

118.05	24.57	7.88	9.68	14.16	13.63	10.24	11.09	6.23	7.46	21.77	16.59	8.30
86.32	7.88	10.22	6.47	11.45	11.18	9.38	9.53	4.25	6.88	16.85	13.53	7.83
88.98	8.68	6.47	11.19	16.59	14.55	10.34	10.04	8.97	8.03	22.72	16.67	6.79
123.02	14.16	11.45	16.55	42.64	31.58	20.33	16.49	31.92	27.39	54.34	37.87	9.96
100.86	13.63	11.18	14.55	31.58	31.68	20.14	18.38	17.39	15.92	41.00	30.51	11.38
87.15	10.24	9.38	10.34	20.33	20.14	22.26	17.58	8.03	8.88	22.66	20.98	13.64
78.48	11.09	9.53	10.04	16.49	18.38	17.58	31.69	-9.54	-4.94	15.97	16.88	13.98
138.16	6.23	4.25	8.97	31.92	17.39	-0.03	-9.54	98.63	73.59	81.38	43.32	-10.98
136.27	7.46	6.88	8.03	27.39	15.92	8.89	-4.94	73.59	69.67	76.34	43.49	-6.19
148.29	21.77	16.85	22.72	54.34	41.00	22.66	15.97	81.38	76.34	127.95	82.81	14.34
124.91	16.59	13.53	16.67	37.87	30.51	20.98	16.88	43.32	43.49	82.81	68.35	17.52
85.58	8.30	7.83	6.79	9.96	11.38	13.64	13.98	-10.98	-6.19	14.34	17.52	32.45

Samples are from Woods-field. Number of samples is 800.

109.91	39.06	20.41	20.35	52.33	50.02	35.27	34.61	37.33	18.78	47.98	29.17	16.57
80.34	20.41	20.38	21.50	40.62	40.67	31.25	28.89	30.93	16.77	39.87	25.66	14.09
82.49	20.35	21.50	34.35	61.10	57.35	38.51	37.48	45.00	18.78	52.74	32.08	17.56
110.53	52.33	40.62	61.10	132.59	113.99	78.14	71.94	96.48	44.19	110.89	65.24	35.01
100.38	50.02	40.67	57.35	113.99	114.64	74.08	71.97	98.64	46.85	105.29	59.66	33.00
91.85	35.27	31.25	38.51	78.14	74.08	78.42	54.17	69.08	39.24	69.71	53.46	30.07
79.34	34.61	28.89	37.48	71.94	71.97	54.17	53.04	90.59	26.25	66.83	41.71	23.93
100.83	37.33	30.93	45.00	96.48	98.64	69.08	90.59	133.51	55.30	136.72	69.25	45.11
97.17	18.78	16.77	18.78	44.19	46.85	39.24	26.25	55.30	102.89	78.80	71.08	24.42
110.15	47.98	39.87	52.74	110.89	105.29	69.71	66.83	136.72	78.80	208.35	102.00	65.48
93.61	29.17	25.66	32.08	65.24	59.66	53.46	41.71	69.25	71.08	102.00	94.88	34.51
84.54	16.57	14.09	17.56	35.01	33.00	30.07	23.93	45.11	24.42	65.48	34.51	63.01

Samples are from Wheats-field. Number of samples is 800.

125.07	26.50	9.44	6.14	4.48	9.36	13.16	16.35	-10.27	-4.04	11.02	8.12	10.77
93.18	9.44	12.24	4.98	4.65	8.10	11.36	14.50	-8.47	-3.46	9.01	7.68	9.96
100.94	6.14	4.98	9.64	11.34	12.02	10.67	11.29	-3.35	-0.51	12.88	9.37	10.33
146.48	4.48	4.65	11.34	33.01	25.31	15.63	13.04	5.05	4.11	36.08	18.81	14.61
132.80	9.36	8.10	12.02	25.31	29.88	19.99	19.30	-0.12	1.38	26.32	19.56	18.00
108.33	13.16	11.36	10.67	15.63	19.99	25.33	22.64	-7.09	-2.89	31.28	17.44	19.53
95.80	16.35	14.50	11.29	13.04	19.30	22.64	31.90	-14.01	-5.09	33.59	18.68	22.48
106.91	-10.27	-8.47	-3.35	5.05	-0.12	-7.09	-14.01	33.00	7.17	-4.34	-4.39	-18.47
98.75	-4.04	-3.46	-0.51	4.11	1.38	-2.89	-5.09	7.17	6.68	2.71	0.66	-2.58
136.48	11.02	9.01	12.88	36.08	26.32	31.28	33.59	-4.34	2.71	48.90	30.51	30.20
113.91	8.12	7.68	9.37	18.81	19.56	17.44	18.68	-4.39	0.66	30.51	39.20	23.39
106.53	10.77	9.96	10.33	14.61	18.00	19.53	22.48	-18.47	-2.58	30.20	23.39	34.53

Class statistic of test area data set NO.2

Appendix C

Program for AMICA

```
*****************************************************************
*       This progeram tests the Unity Relation                  *
*       to extract object-features, for scene representation.   *
*       nh      number of chenels (dimensionality).             *
*       ny      number of lines in the scene                    *
*       nx      number of pixels per scan line                  *
*       nwg     higth of window= 2 x nw                         *
*       gH      Horizontal pixel-feature Gragien                *
*       gV      Vertical pixel-feature  Gragient                *
*       gHV     Diagonal pixel-feature Gragient                 *
*       inputs :                                                *
*               Data in BIL format:                             *
*       outputs:                                                *
*               spatial-feature-map; ADRS = L_i                 *
*               object-features; Feature (X=S_i)                *
*****************************************************************

        parameter(nh=12,nx=140,ny=220,nrp=24,nw=5)
        parameter(nwg=2*nw,nxy=nwg*nx,nhmx=5*nh,nmax=50
*nh)
        integer
wH(nh,nwg),wV(nh,nwg),gH(nh),gV(nh),gHV(nh),sumH,sumV
        integer
X(nh),W(nh,nx,nw),ADRS(nx,nrp),adk(nx,3),bufk(nx)
        integer dmin(nx,3),DIS,dis1,dis2,dis3,dis4,dis5
        integer bufa(3*nx),a1,a2,a3,a4,a5,armn,adlt,a, fdr,uopen
        integer Y(nh,nx),V(nh,nx),S(nh,nx),N(nx)
        character*1 scen(nx)
        real X1(nh),tim(2)
        fdr=uopen('F210',384)
**      fdr=uopen('data/FLC1',384)
**      fdr=uopen('data/FLC2',384)
**      fdr=uopen('data/FLC3',384)
**      fdr=uopen('data/FLC4',384)
        open(10,file='FEATURE')
        open(11,file='ADRS')
        open(12,file='FEATUR1')
        open(13,file='FEATUR2')
*****************************************************
*       Initiation of the Functional Coefficients   *
*****************************************************
        do 1000 iw=1,nw
        iw1=iw-1
        do 1100 ih=1,nh
        wH(ih,iw)=0
        wV(ih,iw)=0
        call uread(fdr,scen,nx)
                do 1110 ix=1,nx
                W(ih,ix,iw)=ichar(scen(ix))
                if(W(ih,ix,iw).lt.0)
W(ih,ix,iw)=W(ih,ix,iw)+256
                ix1=ix-1
        if(ix.ne.1) wH(ih,iw)=wH(ih,iw)+iabs(W(ih,ix,iw)-
W(ih,ix1,iw))
        if(iw.ne.1) wV(ih,iw)=wV(ih,iw)+iabs(W(ih,ix,iw)-
W(ih,ix,iw1))
1110            continue
                wH(ih,iw+nw)=wH(ih,iw)
                wV(ih,iw+nw)=wV(ih,iw)
1100            continue
1000    continue
        do 1200 ih=1,nh
        wV(ih,1)=wV(ih,2)
        wV(ih,nw+1)=wV(ih,nw)
        gH(ih)=0
        gV(ih)=0
                do 1210 iwg=1,nwg
                gH(ih)=gH(ih)+wH(ih,iwg)
                gV(ih)=gV(ih)+wV(ih,iwg)
1210            continue
1200    continue
        sumH=0
        sumV=0
        do 1220 ih=1,nh
        sumH=sumH+gH(ih)
        sumV=sumV+gV(ih)
1220    continue
        if(sumH.lt.sumV) then
                do 1230 ih=1,nh
                gHV(ih)=gH(ih)
1230            continue
        else
                do 1240 ih=1,nh
                gHV(ih)=gV(ih)
1240            continue
        endif
*****************************************************
*       Segmenatation of the First Scan Line        *
*****************************************************
        do 1300 ik=1,nx
        bufk(ik)=0
1300    continue
        ik=1
        k5=ik
        bufk(ik)=ik
        ik=ik+1
                ADRS(1,1)=k5
                adk(1,1)=k5
                N(k5)=1
                do 1400 ih=1,nh
                S(ih,k5)=W(ih,1,1)
                V(ih,k5)=0
                Y(ih,k5)=W(ih,1,1)
1400            continue
        do 1500 ix=2,nx
        do 1510 ih=1,nh
        X(ih)=W(ih,ix,1)
1510    continue
        dis1=DIS(nxy,V(1,k5),S(1,k5),N(k5),X,gH,nh)
        if(dis1.le.nh) then
            ADRS(ix,1)=k5
                adk(ix,1)=k5
            N(k5)=N(k5)+1
            do 1520 ih=1,nh
                S(ih,k5)=S(ih,k5)+X(ih)
                V(ih,k5)=V(ih,k5)+iabs(Y(ih,k5)-X(ih))
                Y(ih,k5)=X(ih)
1520        continue
        else
        k5=ik
        bufk(ik)=ik
        ik=ik+1
            ADRS(ix,1)=k5
                adk(ix,1)=k5
            N(k5)=1
            do 1530 ih=1,nh
                S(ih,k5)=X(ih)
```

```
                    V(ih,k5)=0
                    Y(ih,k5)=X(ih)
1530            continue
        endif
1500    continue
        a=k5
        ia=0
        iy=2
        iy1=1
        iy2=nrp
        iy3=nrp
        ibf=2
        ibf1=1
        ibf2=3
*****************************************************
*       Segmentation of the Rest of Scene           *
*****************************************************
        i=1
        iw=nw
        iw1=nw-1
        iwg=nwg
        nnp=ny-nw+1
        t1=etime(tim)
        t1=tim(1)
        do 2999 ip=2,ny
        i=1+mod(i,nw)
        if(ip.gt.nnp) goto 101
*****************************************************
*       Computing the Functional Coefficients       *
*****************************************************
        iw=1+mod(iw,nw)
        iw1=1+mod(iw1,nw)
        iwg=1+mod(iwg,nwg)
        do 2100 ih=1,nh
        call uread(fdr,scen,nx)
                    do 2110 ix=1,nx
                    W(ih,ix,iw)=ichar(scen(ix))
                    if(W(ih,ix,iw).lt.0)
W(ih,ix,iw)=W(ih,ix,iw)+256
2110                continue
        wH(ih,iwg)=0
        wV(ih,iwg)=0
                    do 2120 ix=2,nx
                    ix1=ix-1
                    wH(ih,iwg)=wH(ih,iwg)+iabs(W(ih,ix,iw)-
W(ih,ix1,iw))
                    wV(ih,iwg)=wV(ih,iwg)+iabs(W(ih,ix,iw)-
W(ih,ix,iw1))
2120                continue
        gH(ih)=0
        gV(ih)=0
                    do 2130 itg=1,nwg
                    gH(ih)=gH(ih)+wH(ih,itg)
                    gV(ih)=gV(ih)+wV(ih,itg)
2130                continue
2100    continue
        sumH=0
        sumV=0
        do 2140 ih=1,nh
        sumH=sumH+gH(ih)
        sumV=sumV+gV(ih)
2140    continue
        if(sumH.lt.sumV) then
                    do 2150 ih=1,nh
                    gHV(ih)=gH(ih)
2150                continue
        else
                    do 2160 ih=1,nh
                    gHV(ih)=gV(ih)
2160                continue
        endif
*****************************************************
*       Labeling th First Column of the Scene       *
*****************************************************
101     ix=1
        ix2=ix+1
        do 110 ih=1,nh
        X(ih)=W(ih,ix,i)
110     continue
*************************
*       Case #2         *
*************************
        a2=ADRS(ix,iy1)
        k2=adk(ix,ibf1)
        dis2=DIS(nxy,V(1,k2),S(1,k2),N(k2),X,gV,nh)
        if(dis2.gt.nh) goto 103
                    ADRS(ix,iy)=a2
                    adk(ix,ibf)=k2
                    N(k2)=N(k2)+1
                    do 121 ih=1,nh
                    S(ih,k2)=S(ih,k2)+X(ih)
                    V(ih,k2)=V(ih,k2)+iabs(Y(ih,k2)-X(ih))
                    Y(ih,k2)=X(ih)
121                 continue
        goto 201
*************************
*       Case #3         *
*************************
103     a3=ADRS(ix2,iy1)
        k3=adk(ix2,ibf1)
        if(k3.eq.k2) dis3=dis2
        if(k3.eq.k2) goto 105
        dis3=DIS(nxy,V(1,k3),S(1,k3),N(k3),X,gHV,nh)
        if(dis3.gt.nh) goto 105
                    ADRS(ix,iy)=a3
                    adk(ix,ibf)=k3
                    N(k3)=N(k3)+1
                    do 131 ih=1,nh
                    S(ih,k3)=S(ih,k3)+X(ih)
                    V(ih,k3)=V(ih,k3)+iabs(Y(ih,k3)-X(ih))
                    Y(ih,k3)=X(ih)
131                 continue
        goto 201
*************************
*       Case #5         *
*************************
105     if(ia.gt.0)then
        a5=bufa(ia)
        ia=ia-1
        else
        a=a+1
        a5=a
        endif
1051    if(bufk(ik).eq.0)then
                    k5=ik
                    bufk(ik)=ik
                    ik=1+mod(ik,nx)
                    flagk=1
        else
                    ik=1+mod(ik,nx)
                    flagk=0
        endif
        if(flagk.eq.0) goto 1051
                    ADRS(ix,iy)=a5
                    adk(ix,ibf)=k5
                    N(k5)=1
                    do 151 ih=1,nh
                    S(ih,k5)=X(ih)
                    V(ih,k5)=0
                    Y(ih,k5)=X(ih)
151                 continue
        dis5=min0(dis2,dis3)
        if(dis5.le.nhmx) then
        if(dis2.eq.dis5) then
                                dmin(ix,ibf)=2
        else
                                dmin(ix,ibf)=3
        endif
        else
                                dmin(ix,ibf)=5
        endif
*****************************************************
*       Labelling non-Edge Pixels of the Scene      *
*****************************************************
201     do 200 ix=2,nx-1
        ix1=ix-1
        ix2=ix+1
        do 210 ih=1,nh
        X(ih)=W(ih,ix,i)
210     continue
*************************
*       Case #1         *
*************************
        a1=ADRS(ix1,iy)
        k1=adk(ix1,ibf)
        dis1=DIS(nxy,V(1,k1),S(1,k1),N(k1),X,gH,nh)
        a2=ADRS(ix,iy1)
        k2=adk(ix,ibf1)
        if(k1.eq.k2) then
        dis2=dis1
        else
        dis2=DIS(nxy,V(1,k2),S(1,k2),N(k2),X,gV,nh)
        endif
        dis5=min0(dis1,dis2)
        if(dis5.gt.nh) goto 203
        if(dis5.eq.dis1) then
                    ADRS(ix,iy)=a1
                    adk(ix,ibf)=k1
                    if(N(k1).gt.nmax) goto 32
```

```
                    N(k1)=N(k1)+1
                    do 211 ih=1,nh
                    S(ih,k1)=S(ih,k1)+X(ih)
                    V(ih,k1)=V(ih,k1)+iabs(Y(ih,k1)-X(ih))
                    Y(ih,k1)=X(ih)
211                 continue
32          a3=ADRS(ix2,iy1)
            k3=adk(ix2,ibf1)
            if(k1.ne.k3) then
            dis3=DIS(nxy,V(1,k3),S(1,k3),N(k3),X,gHV,nh)
            if(dis3.le.nh) then
            armn=min0(a1,a3)
            if(armn.eq.a1) then
                    krmn=k1
                    kdlt=k3
                    adlt=a3
            else
                    krmn=k3
                    kdlt=k1
                    adlt=a1
            endif
                    do 2131 lx=1,nx
                    if(adk(lx,ibf2).eq.kdlt) adk(lx,ibf2)=krmn
                    if(adk(lx,ibf1).eq.kdlt) adk(lx,ibf1)=krmn
2131                continue
                    do 2132 lx=1,lx
                    if(adk(lx,ibf).eq.kdlt) adk(lx,ibf)=krmn
2132                continue
                    do 213 ly=1,nrp
                    do 213 lx=1,nx
                    if(ADRS(lx,ly).eq.adlt) ADRS(lx,ly)=armn
213                 continue
                    N(krmn)=N(krmn)+N(kdlt)
                    do 214 ih=1,nh
                    S(ih,krmn)=S(ih,krmn)+S(ih,kdlt)
                    V(ih,krmn)=V(ih,krmn)+V(ih,kdlt)
                    Y(ih,krmn)=X(ih)
214                 continue
            ik=kdlt
            bufk(ik)=0
            ia=ia+1
            bufa(ia)=adlt
            endif
            endif
*****************************
*           Case #2         *
*****************************
            else
                    ADRS(ix,iy)=a2
                    adk(ix,ibf)=k2
                    if(N(k2).lt.nmax) then
                    N(k2)=N(k2)+1
                    do 221 ih=1,nh
                    S(ih,k2)=S(ih,k2)+X(ih)
                    V(ih,k2)=V(ih,k2)+iabs(Y(ih,k2)-X(ih))
                    Y(ih,k2)=X(ih)
221                 continue
                    endif
            endif
            goto 206
*****************************
*           Case #3         *
*****************************
203         a3=ADRS(ix2,iy1)
            k3=adk(ix2,ibf1)
            if(k3.eq.k2) dis3=dis2
            if(k3.eq.k2) goto 204
            if(k3.eq.k1) dis3=dis1
            if(k3.eq.k1) goto 204
            dis3=DIS(nxy,V(1,k3),S(1,k3),N(k3),X,gHV,nh)
            if(dis3.gt.nh) goto 204
                    ADRS(ix,iy)=a3
                    adk(ix,ibf)=k3
                    if(N(k3).gt.nmax) goto 33
                    N(k3)=N(k3)+1
                    do 231 ih=1,nh
                    S(ih,k3)=S(ih,k3)+X(ih)
                    V(ih,k3)=V(ih,k3)+iabs(Y(ih,k3)-X(ih))
                    Y(ih,k3)=X(ih)
231                 continue
33          a4=ADRS(lx1,iy1)
            k4=adk(lx1,ibf1)
            if(k3.eq.k4) goto 206
            dis4=DIS(nxy,V(1,k4),S(1,k4),N(k4),X,gHV,nh)
            if(dis4.gt.nh) goto 206
            armn=min0(a3,a4)
            if(armn.eq.a4) then
                    krmn=k4
                    kdlt=k3
                    adlt=a3
            else
                    krmn=k3
                    kdlt=k4
                    adlt=a4
            endif
                    do 2331 lx=1,nx
                    if(adk(lx,lbf2).eq.kdlt) adk(lx,lbf2)=krmn
                    if(adk(lx,lbf1).eq.kdlt) adk(lx,lbf1)=krmn
2331                continue
                    do 2332 lx=1,ix
                    if(adk(lx,ibf).eq.kdlt) adk(lx,lbf)=krmn
2332                continue
                    do 233 ly=1,nrp
                    do 233 lx=1,nx
                    if(ADRS(lx,ly).eq.adlt) ADRS(lx,ly)=armn
233                 continue
                    N(krmn)=N(krmn)+N(kdlt)
                    do 234 lh=1,nh
                    S(ih,krmn)=S(ih,krmn)+S(ih,kdlt)
                    V(ih,krmn)=V(ih,krmn)+V(ih,kdlt)
                    Y(ih,krmn)=X(ih)
234                 continue
            ik=kdlt
            bufk(ik)=0
            ia=ia+1
            bufa(ia)=adlt
            goto 206
*****************************
*           Case #4         *
*****************************
204         a4=ADRS(ix1,iy1)
            k4=adk(lx1,ibf1)
            if(k4.eq.k3) dis4=dis3
            if(k4.eq.k3) goto 205
            if(k4.eq.k2) dis4=dis2
            if(k4.eq.k2) goto 205
            if(k4.eq.k1) dis4=dis1
            if(k4.eq.k1) goto 205
            dis4=DIS(nxy,V(1,k4),S(1,k4),N(k4),X,gHV,nh)
            if(dis4.gt.nh) goto 205
                    ADRS(lx,ly)=a4
                    adk(ix,ibf)=k4
                    if(N(k4).gt.nmax) goto 206
                    N(k4)=N(k4)+1
                    do 241 ih=1,nh
                    S(ih,k4)=S(ih,k4)+X(ih)
                    V(ih,k4)=V(ih,k4)+iabs(Y(ih,k4)-X(ih))
                    Y(ih,k4)=X(ih)
241                 continue
            goto 206
*****************************
*           Case #5         *
*****************************
205         if(ia.gt.0) then
            a5=bufa(ia)
            ia=ia-1
            else
            a=a+1
            a5=a
            endif
2051        if(bufk(ik).eq.0)then
                    k5=ik
                    bufk(ik)=ik
                    ik=1+mod(ik,nx)
                    flagk=1
            else
                    ik=1+mod(ik,nx)
                    flagk=0
            endif
            if(flagk.eq.0) goto 2051
                    ADRS(lx,ly)=a5
                    adk(lx,ibf)=k5
                    N(k5)=1
                    do 251 ih=1,nh
                    S(ih,k5)=X(ih)
                    V(ih,k5)=0
                    Y(ih,k5)=X(ih)
251                 continue
            dis5=min0(dis1,dis2,dis3,dis4)
            if(dis5.le.nhmx) then
            if(dis1.eq.dis5) then
                                dmin(ix,lbf)=1
            elseif(dis2.eq.dis5) then
                                dmin(ix,lbf)=2
            elseif(dis3.eq.dis5) then
                                dmin(ix,lbf)=3
            else
                                dmin(ix,lbf)=4
            endif
            else
```

```
                                    dmin(ix,ibf)=5
            endif
*****************************
*           Case #6         *
*****************************
206         if(ip.eq.2) goto 200
            adlt=ADRS(ix1,iy1)
            kdlt=adk(ix1,ibf1)
            if(N(kdlt).eq.1) then
            iflag=0
            if(dmin(ix1,ibf1).eq.1) then
            if(ix1-1.eq.0) print *, 'ERROR 1',ix1,ip-1
                                    armn=ADRS(ix1-1,iy1)
                                    krmn=adk(ix1-1,ibf1)
            elseif(dmin(ix1,ibf1).eq.2) then
                                    armn=ADRS(ix1,iy2)
                                    krmn=adk(ix1,ibf2)
            elseif(dmin(ix1,ibf1).eq.3) then
                                    armn=ADRS(ix,iy2)
                                    krmn=adk(ix,ibf2)
            elseif(dmin(ix1,ibf1).eq.4) then
            if(ix1-1.eq.0) print *, 'ERROR 4',ix1,ip-1
                                    armn=ADRS(ix1-1,iy2)
                                    krmn=adk(ix1-1,ibf2)
            elseif(dmin(ix1,ibf1).eq.5) then
                                    iflag=1
            else
            print *,'iy',ip-1,'ix',ix1,'dmin',dmin(ix1,ibf1),'N',N(kdlt)
            endif
                        if(iflag.eq.0) then
                                    if(N(krmn).eq.1) then
                                    N(krmn)=2
                                    do 261 ih=1,nh
                                    S(ih,krmn)=2*S(ih,krmn)
261                                 continue
                                    endif
                                    adk(ix1,ibf1)=krmn
                                    ADRS(ix1,iy1)=armn
            lk=kdlt
            bufk(lk)=0
                        ia=ia+1
                        bufa(ia)=adlt
                        endif
            endif
200         continue
*********************************************
*           Labeling the last Column of Scene *
*********************************************
            ix=nx
            ix1=ix-1
            do 310 ih=1,nh
            X(ih)=W(ih,ix,i)
310         continue
*****************************
*           Case #1         *
*****************************
            a1=ADRS(ix1,iy)
            k1=adk(ix1,ibf)
            dis1=DIS(nxy,V(1,k1),S(1,k1),N(k1),X,gH,nh)
            if(dis1.gt.nh) goto 302
                        ADRS(ix,iy)=a1
                        adk(ix,ibf)=k1
                        N(k1)=N(k1)+1
                        do 311 ih=1,nh
                        S(ih,k1)=S(ih,k1)+X(ih)
                        V(ih,k1)=V(ih,k1)+iabs(Y(ih,k1)-X(ih))
                        Y(ih,k1)=X(ih)
311                     continue
            goto 306
*****************************
*           Case #2         *
*****************************
302         a2=ADRS(ix,iy1)
            k2=adk(ix,ibf1)
            if(k2.eq.k1) dis2=dis1
            if(k2.eq.k1) goto 304
            dis2=DIS(nxy,V(1,k2),S(1,k2),N(k2),X,gV,nh)
            if(dis2.gt.nh) goto 304
                        ADRS(ix,iy)=a2
                        adk(ix,ibf)=k2
                        N(k2)=N(k2)+1
                        do 321 ih=1,nh
                        S(ih,k2)=S(ih,k2)+X(ih)
                        V(ih,k2)=V(ih,k2)+iabs(Y(ih,k2)-X(ih))
                        Y(ih,k2)=X(ih)
321                     continue
            goto 306
*****************************
*           Case #4         *
*****************************
304         a4=ADRS(ix1,iy1)
            k4=adk(ix1,ibf1)
            if(k4.eq.k2) dis4=dis2
            if(k4.eq.k2) goto 305
            if(k4.eq.k1) dis4=dis1
            if(k4.eq.k1) goto 305
            dis4=DIS(nxy,V(1,k4),S(1,k4),N(k4),X,gHV,nh)
            if(dis4.gt.nh) goto 305
                        ADRS(ix,iy)=a4
                        adk(ix,ibf)=k4
                        N(k4)=N(k4)+1
                        do 341 ih=1,nh
                        S(ih,k4)=S(ih,k4)+X(ih)
                        V(ih,k4)=V(ih,k4)+iabs(Y(ih,k4)-X(ih))
                        Y(ih,k4)=X(ih)
341                     continue
            goto 306
*****************************
*           Case #5         *
*****************************
305         if(ia.gt.0) then
            a5=bufa(ia)
            ia=ia-1
            else
            a=a+1
            a5=a
            endif
3051        if(bufk(ik).eq.0)then
                        k5=ik
                        bufk(lk)=ik
                        ik=1+mod(lk,nx)
                        flagk=1
            else
                        ik=1+mod(ik,nx)
                        flagk=0
            endif
            if(flagk.eq.0) goto 3051
                        ADRS(ix,iy)=a5
                        adk(ix,ibf)=k5
                        N(k5)=1
                        do 351 ih=1,nh
                        S(ih,k5)=X(ih)
                        V(ih,k5)=0
                        Y(ih,k5)=X(ih)
351                     continue
            dis5=min0(dis1,dis2,dis4)
            if(dis5.le.nhmx) then
            if(dis1.eq.dis5) then
                                    dmin(ix,ibf)=1
            elseif(dis2.eq.dis5) then
                                    dmin(ix,ibf)=2
            else
                                    dmin(ix,ibf)=4
            endif
            else
                                    dmin(ix,ibf)=5
            endif
*****************************
*           Case #6         *
*****************************
306         if(ip.eq.2) goto 2000
            do 360 i6=1,2
            ix1=nx-(2-i6)
            adlt=ADRS(ix1,iy1)
            kdlt=adk(ix1,ibf1)
            if(N(kdlt).eq.1) then
            iflag=0
            if(dmin(ix1,ibf1).eq.1) then
                                    armn=ADRS(ix1-1,iy1)
                                    krmn=adk(ix1-1,ibf1)
            elseif(dmin(ix1,ibf1).eq.2) then
                                    armn=ADRS(ix1,iy2)
                                    krmn=adk(ix1,ibf2)
            elseif(dmin(ix1,ibf1).eq.3) then
            if(ix1+1.gt.nx) print 'ERORR 3',ix1,ip-1
                                    armn=ADRS(ix1+1,iy2)
                                    krmn=adk(ix1+1,ibf2)
            elseif(dmin(ix1,ibf1).eq.4) then
                                    armn=ADRS(ix1-1,iy2)
                                    krmn=adk(ix1-1,ibf2)
            elseif(dmin(ix1,ibf1).eq.5) then
                                    iflag=1
            else
            print *,'iy',ip-1,'ix',ix1,'dmin',dmin(ix1,ibf1),'N',N(kdlt)
            endif
                        if(iflag.eq.0) then
                                    if(N(krmn).eq.1) then
                                    N(krmn)=2
                                    do 361 ih=1,nh
                                    S(ih,krmn)=2*S(ih,krmn)
```

```
361                                     continue
                                        endif
                                        adk(ix1,ibf1)=krmn
                ik=kdlt
                bufk(ik)=0
                                        ADRS(ix1,iy1)=armn
                        ia=ia+1
                        bufa(ia)=adlt
                        endif
                endif
360             continue
*****************************************
*               Transmitting the Features
*****************************************
                ix=1
401             ix=0
410             ix=ix+1
                if(ADRS(ix,iy2).eq.ADRS(ix,iy1)) then
                        ix=ix+1
                        ix=1
                        if(ix.gt.nx) ix=nx
                endif
                if(ix.lt.nx) goto 410
                if(ix.gt.nx) goto 2000
                iflag=0
                if(ix.gt.1)then
                        do 420 ix=1,ix-1
                        if(ADRS(ix,iy2).eq.ADRS(ix,iy2)) iflag=1
420                     continue
                endif
                if(iflag.eq.0) then
                a5=ADRS(ix,iy2)
                k5=adk(ix,ibf2)
                ik=k5
                bufk(ik)=0
                do 430 ih=1,nh
                        X(ih)=nint(float(S(ih,k5))/float(N(k5)))
430             continue
                write(10,*) a5
                write(10,*) X
                endif
                ix=ix+1
                goto 401
2000            iy=1+mod(iy,nrp)
                iy1=1+mod(iy1,nrp)
                iy2=1+mod(iy2,nrp)
                ibf=1+mod(ibf,3)
                ibf1=1+mod(ibf1,3)
                ibf2=1+mod(ibf2,3)
                if(ip.ge.nrp) then
                iy3=1+mod(iy3,nrp)
                        do 450 ix=1,nx
                        ij=ADRS(ix,iy3)
                        ID=32+mod(ij,95)
                        scen(ix)=char(ID)
450                     continue
                write(11,*) (ADRS(ix,iy3),ix=1,nx)
                endif
2999            continue
*****************************************
*               Processing the last line
*****************************************
                do 3100 ix=1,nx
                adlt=ADRS(ix,iy1)
                kdlt=adk(ix,ibf1)
                if(N(kdlt).eq.1) then
                iflag=0
                if(dmin(ix,ibf1).eq.1) then
                if(ix-1.eq.0) print *, 'ERROR 1',ix,ip-1
                                        armn=ADRS(ix-1,iy1)
                                        krmn=adk(ix-1,ibf1)
                elseif(dmin(ix,ibf1).eq.2) then
                                        armn=ADRS(ix,iy2)
                                        krmn=adk(ix,ibf2)
                elseif(dmin(ix,ibf1).eq.3) then
                                        armn=ADRS(ix+1,iy2)
                                        krmn=adk(ix+1,ibf2)
                elseif(dmin(ix,ibf1).eq.4) then
                if(ix-1.eq.0) print *, 'ERROR 4',ix,ip-1
                                        armn=ADRS(ix-1,iy2)
                                        krmn=adk(ix-1,ibf2)
                elseif(dmin(ix,ibf1).eq.5) then
                                        iflag=1
                else
                print *,'iy',ip-1,'ix',ix,'dmin',dmin(ix,ibf1),'N',N(kdlt)
                endif
                        if(iflag.eq.0) then
                                if(N(krmn).eq.1) then
                                N(krmn)=2
                                do 3110 ih=1,nh
                                S(ih,krmn)=2*S(ih,krmn)
3110                            continue
                                endif
                                adk(ix,ibf1)=krmn
                ik=kdlt
                bufk(ik)=0
                                ADRS(ix,iy1)=armn
                        ia=ia+1
                        bufa(ia)=adlt
                        endif
                endif
3100            continue
*****************************************
*               Transmitting the Features
*****************************************
                ix=1
3001            ix=0
3200            ix=ix+1
                if(ADRS(ix,iy2).eq.ADRS(ix,iy1)) then
                        ix=ix+1
                        ix=1
                        if(ix.gt.nx) ix=nx
                endif
                if(ix.lt.nx) goto 3200
                if(ix.gt.nx) goto 3002
                iflag=0
                if(ix.gt.1)then
                        do 3210 ix=1,ix-1
                        if(ADRS(ix,iy2).eq.ADRS(ix,iy2)) iflag=1
3210                    continue
                endif
                if(iflag.eq.0) then
                a5=ADRS(ix,iy2)
                k5=adk(ix,ibf2)
                do 3220 ih=1,nh
                        X(ih)=nint(float(S(ih,k5))/float(N(k5)))
3220            continue
                write(10,*) a5
                write(10,*) X
                endif
                ix=ix+1
                goto 3001
3002            ik=1
                bufk(ik)=ADRS(1,iy1)
                k5=adk(1,ibf1)
                do 3230 ih=1,nh
                        X(ih)=nint(float(S(ih,k5))/float(N(k5)))
3230            continue
                write(10,*) bufk(ik)
                write(10,*) X
                do 3300 ix=1,nx
                iflag=1
                        do 3310 ik=1,ik
                        if(ADRS(ix,iy1).eq.bufk(ik)) iflag=0
3310                    continue
                if(iflag.eq.1)then
                ik=ik+1
                bufk(ik)=ADRS(ix,iy1)
                k5=adk(ix,ibf1)
                do 3330 ih=1,nh
                X(ih)=nint(float(S(ih,k5))/float(N(k5)))
3330            continue
                write(10,*) bufk(ik)
                write(10,*) X
                endif
3300            continue
                do 500 iy=2,nrp
                iy3=1+mod(iy3,nrp)
                        do 510 ix=1,nx
                        ij=ADRS(ix,iy3)
                        ID=32+mod(ij,95)
                        scen(ix)=char(ID)
510                     continue
                write(11,*) (ADRS(ix,iy3),ix=1,nx)
500             continue
                stop
                end
*****************************************
*                       Functional No.1
*****************************************
                function DIS(nw,V,S,n,X,T,nh)
                integer DIS,V(nh),S(nh),X(nh),T(nh),n,nh,nw
                a=0.0
                if(n.gt.100) then
                do 20 i=1,nh
                a=a+abs((S(i)-(n*X(i)))/float(V(i)))
20              continue
                else
                do 10 i=1,nh
                a=a+abs(((S(i)/float(n))-float(X(i)))*(nw/float(T(i))))
```

```
10          continue
            endif
            DIS=nint(a)
            return
            end
*************************************
*                    Functional No.2                *
*************************************
            function DIS2(nw,V,S,n,X,T,nh)
            integer DIS,V(nh),S(nh),X(nh),T(nh),n,nh,nw
            a=0.0
            b=1.0/float(n+24)
            c=24./float(nw*(n+24))
            do 10 i=1,nh
            a=a+abs(((S(i)/float(n))-float(X(i)))/(b*V(i)+c*T(i)))
10          continue
            DIS=nint(a)
            return
            end
```

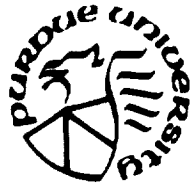
VITA

VITA

Mohammad Hassan Ghassemian Yazdi was born on [redacted], in [redacted]. He graduated from Tehran College of Telecommunication, Tehran, Iran, in May 1980, received the B.S.E.E. degree in communication engineering, and was awarded the Distinguish Student Honoree: highest G.P.A. ever obtained at the university up to the present, May 1988 (3.94/4.00). Then he was employed by ITRC (Iran Telecommunication Research Center) as a circuit designer and research assistant in the digital switching system lab. One of his works at ITRC was designing a new PBX system whose operation and quality is distinct from conventional PBX systems. After participation in the designing and development of bilingual, Farssi/Latin, computer terminals with MERC (Material and Energy Research Center), he entered the graduate school of Electrical Engineering at Purdue University in the Fall of 1983.

He received the M.S.E.E. degree from Purdue University, West Lafayette, IN, in December 1984. Since the Fall of 1985 he has been a teaching assistant and later a research assistant at the Purdue School of Electrical Engineering. He completed work toward his Ph.D. degree in electrical engineering with research interests in pattern recognition and data processing for remote sensing systems, multidimensional digital signal processing, and signal representation.

He married Miss Shokouh [redacted] in May 1980. They have three daughters: Sarah six, Maryam five, and Ehsaneh two years old.


purdue university