A MULTIVALUED APPROACH FOR UNCERTAINTY MANAGEMENT IN PATTERN RECOGNITION PROBLEMS USING FUZZY SETS

DEBA PRASAD MANDAL

Electronics and Communication Sciences Unit

Indian Statistical Institute

Calcutta

A thesis submitted to the *Indian Statistical Institute*

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

1992

(Corrected in 1993)

To the memory of my *Father*

ACKNOWLEDGMENTS

My greatest debt of gratitude is to *Professor Sankar K. Pal*, my supervisor, for his guidence, encouragement and affection which made it less formidable for me to complete this thesis. I consider it a privilege to have come in contact with a person of rare human qualities like his.

I consider this thesis as a kind of joint venture between myself, *Dr. C. A. Murthy* and *Professor Sankar K. Pal.* Thanks are due to them for their kind permission to include the joint research work in this thesis. I owe a lot to Dr. C. A. Murthy who earnestly inspired me and helped me during the course of this work (specially in the absence of Professor Sankar K. Pal).

I express my indebtness to *Professor D. Dutta Majumder* for allowing me to use the computing facilities at the National Centre for Knowledge Based Computing, I.S.I., Calcutta and for his help and guidance during the absence of Professor Sankar K. Pal. I also record my gratitude to him for his constructive criticism during the preparation of this manuscript.

I express my deep sense of gratitude to Dr. N. R. Pal and Dr. M. K. Kundu for the help and encouragement I received from them. I owe a bundle of thanks to the members of Electronics and Communication Sciences Unit, I.S.I., Calcutta. I owe a special debt to my friends Shri A. Ghosh and Shri D. Bhandari for their assistance during the course of this work.

I wish to thank all of my friends at I.S.I., particularly Shri A. C. Naolekar, Shri G. Mukherjee, Shri S. Das and Shri B. Uma Shankar who have helped me in many ways. I also acknowledge Shri S. Chakraborty for drawing some of the diagrams and Mr. S. Bhattacharya for careful xeroxing.

I take this opportunity to express my deep sense of gratitude to my teachers at Jawaharlal Nehru University, New Delhi where I had the first exposure to Computer Sciences.

I would be failing in my duty if I do not acknowledge the help, support and inspiration that I received from all of my family members, who always kept me away from all non-academic problems of the family.

Finally, I express my sincere thanks to the authorities of I.S.I. for the facilities extended to carry out my research work.

Calcutta
December, 1992

Deba Prasad Mandal

CONTENTS

LIST OF FIGURES

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

LIST OF TABLES

Chapter 4

Chapter 5

Chapter 1

INTRODUCTION AND SCOPE OF THE THESIS

Contents

1.1 Introduction

Real life problems are rarely free from uncertainty which usually emerges from the deficiencies of information available from a situation. The deficiencies may result from incomplete, imprecise, not fully reliable, vague or contradictory information depending on the problem. Management of uncertainty in a decision making system has been an important research problem for many years.

Until the inception of the concept of fuzzy set theory in 1965 [1], the theory of probability and statistics was the primary mathematical tool for modeling uncertainty in a system/situation. Fuzzy set theory has shown enormous promise in handling uncertainties to a reasonable extent in various applications particularly in decision making models under different kinds of risks, subjective judgment, vagueness and ambiguity. This theory provides an approximate, yet effective and more flexible means of describing the behavior of systems which are too complex or too ill-defined to admit precise mathematical analysis by classical methods and tools. Since this theory is a generalization of the classical set theory, it has greater flexibility to capture various aspects of incompleteness or imperfection in information about a situation.

Pattern recognition and machine learning form a major area of research and development activity that encompasses the processing of pictorial and other non-numerical information obtained from interaction between science, technology and society. The second motivation for this spurt of activity in this field is the need for the people to communicate with the computing machines in their natural mode of communication. The third and most important motivation is that the scientists are also concerned with the idea of designing and making automata that can carry out certain tasks as we human beings do. The most salient outcome of these is the concept of fifth generation computing systems.

Machine recognition of patterns can be viewed as a two-fold task, con-

sisting of learning the invariant properties of a set of samples characterizing a class, and of deciding that a new sample is a possible member of the class by noting that it has properties common to those of the set of samples. The tasks required for developing and implementing the decision rule can be described as a transformation from the measurement space M to the feature space F and finally to the decision space D, i.e.,

$$\mathrm{M} \rightarrow \mathrm{F} \rightarrow \mathrm{D}.$$

In a pattern recognition system, the uncertainty can arise at any phase of the aforesaid tasks resulting from the incomplete or imprecise or ambiguous input information, the ill-defined and/or overlapping boundaries among the classes or regions, and the indefiniteness in defining/extracting features and relations among them. Any decision taken at a particular level will have an impact on all higher level activities. It is therefore required for a recognition system to have sufficient provision for representing these uncertainties involved at every stage, so that the ultimate output (results) of the system can be obtained with least uncertainty (and not be affected or biased much by preceding decisions).

The present thesis provides some results of investigation, both theoretical and experimental, demonstrating the effectiveness of fuzzy set theory in recognizing and handling uncertainties in certain tasks of pattern recognition. Section 1.2 deals with the description of some basic concepts, features, operations and techniques of pattern recognition. The uncertainties involved in various stages of a pattern recognition system are mentioned in section 1.3. Fuzzy models for pattern recognition and uncertainty analysis are furnished in section 1.4. These include the definition of fuzzy sets, fuzzy logic, uncertainty in membership function, measures of fuzziness, and a review of the fuzzy set theoretic approaches for pattern recognition. Section 1.5 deals with the scope of the thesis. ♠

Fig. 1.1 : Pattern recognition system [12].

1.2 Pattern Recognition

In a general setting the process of pattern recognition is visualized as a sequence of few steps, namely (i) data acquisition, (ii) feature selection, and (iii) classification [Fig. 1.1]. At the first step depending on environment within which the objects are to be classified data are gathered via a set of sensors. Afterwards, a feature space is constituted in order to reduce the space dimensionality. However, in a broader perspective this stage significantly influences the entire recognition process. Finally, at the third stage of the scheme, the classifier is constructed, namely a transformation relationship is established between features and classes. The transformation could be, for instance, a Bayesian rule of computing a posterior class probability, linear or nonlinear classifier, nearest neighbor rule, nearest prototype (mean) classifier, *etc.* [2-19].

Pattern recognition, by its nature, admits many approaches, sometimes complementary, sometimes competing, to the approximate solution of a given problem. The design of a pattern recognition system is a highly interactive process. The complex nature of this design process is summarized in Fig. 1.2.

The approach of pattern recognition can be decision theoretic or syntactic. In decision theoretic approach, once a pattern is transformed, through feature selection, to a vector in the feature space, its characteristics are expressed only by a set of numerical values. Classification can be done using deterministic or probabilistic technique. On the other hand, when a pattern

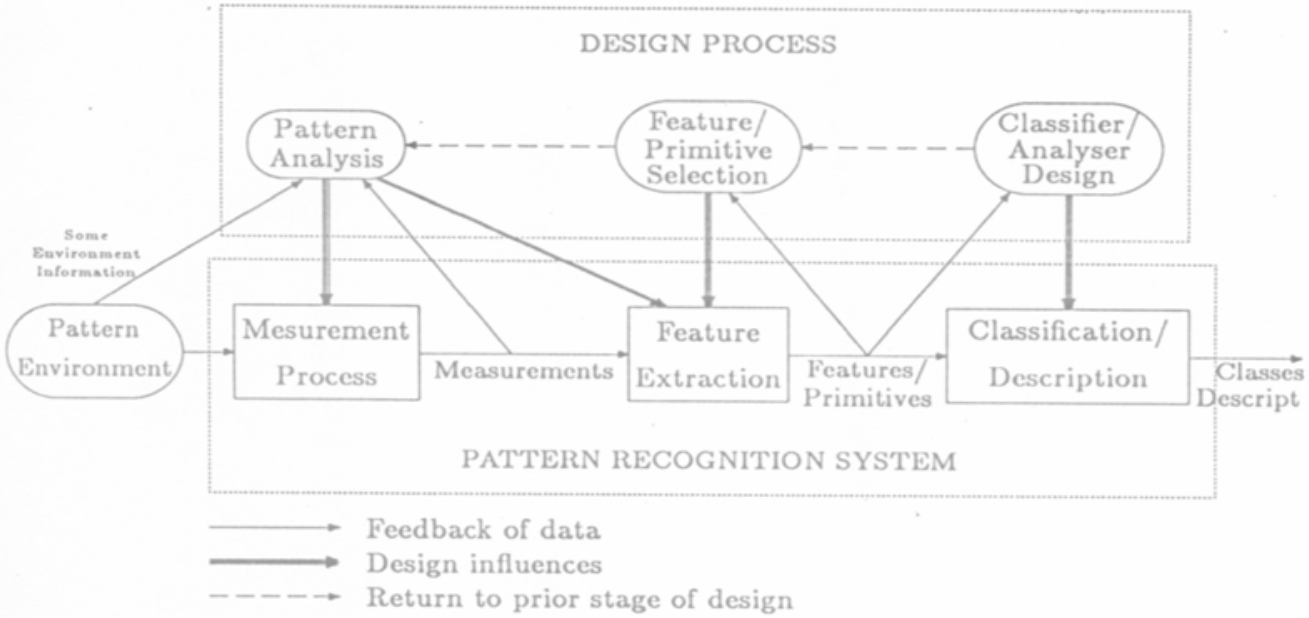


Fig. 1.2 : Development of pattern recognition systems [12].

is rich in structural information (*e.g.*, picture recognition, character recognition, scene analysis) i.e., the structural information plays an important role in describing and recognizing the patterns, the approaches used are referred to as *syntactic approaches* which deal with representation of structures via sentences, grammars, and automata. In the syntactic method, the ability of selecting and classifying the simple pattern primitives and their relationships represented by the composition operations is the vital criterion of making a system effective. Since the techniques of composition of primitives into patterns are usually governed by the formal language theory, the approach is often referred to as *linguistic approach*. A nice introduction to a variety of approaches based on this idea is provided by Fu in [11]. It may be mentioned here that the theory of syntactic pattern recognition is far more well developed than are fielded applications.

Fuzzy set theory has proved itself to be of significant importance in pattern recognition problems both using decision theoretic and syntactic approaches. On the one hand, fuzzy algorithms for different decision theoretic

recognition and image processing problems are being developed, and on the other hand, fuzzy language, fuzzy grammars and fuzzy automata theories have been developed and applied.

Since we are interested here in decision theoretic approaches, the different features and operations related to this approach are described in the following sections.

1.2.1 Data acquisition and preprocessing

Techniques of pattern recognition are applicable to data drawn from virtually any physical processes. The data may be qualitative, quantitative, or both; they may be numerical, pictorial, texture, linguistic, or any combination thereof. In general, two data structures are used in pattern recognition systems: *object data vectors* (feature vectors, pattern vectors) and *pairwise numerical data* (similarities, proximities). Object data, that is, sets of numerical vectors of Q features, are represented in the sequel as $\mathcal{Y} = \{Y_1, Y_2, \ldots, Y_t\}$, a set of t feature vectors in the Q-dimensional *measurement space* Ω_Y. The jth $(j = 1, 2, \ldots, t)$ object observed in the process has vector Y_j as its numerical representation; y_{jk} is the kth $(k = 1, 2, \ldots, Q)$ characteristic (or feature) associated with object j.

Relational data is a set of t^2 numerical relationships, say $\{r_{jj'}\}$, between pairs of objects. That is, $r_{jj'}$ represents the extent to which objects j and j' are related in the sense of some binary relation ρ. If the objects that are pairwise related by ρ are called $O = \{o_1, o_2, \ldots, o_t\}$, then ρ: $O \times O \rightarrow \mathbb{R}$. Relational data are to be found in many applications and systems, perhaps hiding in different semantic guises.

Sometimes, the data acquisition phase involves some preprocessing tasks such as noise reduction, filtering, encoding and enhancement for extracting pattern vectors. For example, if the input pattern is an image, these preprocessing operations play a crucial role for extracting salient features for its recognition. This is further explained in section 1.2.6. ♣

1.2.2 Feature selection

Feature selection is one of the major tasks in any automatic pattern recognition system. The main objective of feature selection is to retain the optimum salient characteristics necessary for the recognition process and to reduce the dimensionality of the *measurement space* Ω_Y so that the effectiveness and easily computable algorithms for classification can be devised. The problem of feature selection has two aspects. The formulation of a suitable criterion to evaluate the goodness of a feature and the selection of the optimal subset from the available features. The major mathematical measures so far devised for the estimation of feature quality are mostly statistical in nature and fall under two broad categories:

(a) Feature selection in the measurement space and

(b) Feature selection in the transformed space.

The main objective of the first category is to reduce the dimensionality of the feature set by discarding redundant information carrying features. The information content of any feature is measured by some chosen criterion and the features are then ordered according to the criterion value to represent the relative effectiveness of the features in a feature set. In contrast, feature selection techniques of the second category utilize all the information of the pattern vector and map higher dimensional pattern vector to a lower dimensional one. This is sometimes referred to as *feature extraction.*

Feature selection or extraction is therefore a process of selecting a map of the form $X = f(Y)$ by which a sample $Y(y_1, y_2, \ldots, y_Q)$ in a Q-dimensional *measurement space* Ω_Y $(\mathbb{R}^Q)$ is transformed into a point $X(x_1, x_2, \ldots, x_N)$ in an N-dimensional $(N < Q)$ *feature space* Ω_X $(\mathbb{R}^N)$.

The pioneering research on feature selection mostly deals with the statistical tools. Later, the thrust of such research was shifted to the development of various techniques based on both statistical and fuzzy approaches. Many of such approaches can be found, for example, in references [7,12,20-23]. ♣

1.2.3 Learning

In real life, the complete description of the classes is not known. We have instead, a finite and usually smaller number of samples which often provides partial information for optimal design of feature extractor or classifier. Under such circumstances, it is assumed that these samples are representative of the classes (usually consisting of an infinite number of samples). Such a set of typical patterns is called a *training set.* On the basis of the information gathered from the samples in the training set, the pattern recognition systems are designed i.e., we decide the values of the parameters of various pattern recognition models.

Learning is thus a task of constructing the regions or templates in the N-dimensional space in which labeled samples of the classes are contained. The two stages of pattern recognition, namely deriving the decision rule (learning), and using it to recognize a pattern, can be performed in two ways - (a) learning before recognition and (b) learning and recognition concurrently. In the first case, all the labeled samples are collected and based on those samples, the best decision rule is derived. This fixed decision rule is then applied without change to classify unlabeled patterns. The decision rule in the second case is adaptive (decision-directed) and is updated according to the output decision. If the learned information gradually approaches the true information, then the decisions based on the learned information will eventually approach the optimal decision as if all the information is known. Therefore, during the system's operation, the performance of the system is gradually improved. Learning process can be termed as *supervised* or *non-supervised* depending on whether the correct classification of the input patterns observed is known or not. For details on learning algorithms, readers may consult [24-31]. ♣

1.2.4 Clustering

Given any finite data set $\mathcal{X} \subset \Omega_X$ $(\mathbb{R}^N)$ of objects, the problem of clustering in $\mathcal{X}$ is to assign object labels that identify natural subgroups in the set.

Because the data are unlabeled, this problem is often called *unsupervised learning*, the word *learning* here referring to learning the correct labels for desirable subgroups. The objective is to partition $\mathcal{X}$ into a certain number (c) of natural and homogeneous subsets, where the elements of each set are as similar as possible to each other and at the same time, as different from those of the other sets as possible. Because the technique of clustering is unsupervised, clustering algorithms attempt to partition $\mathcal{X}$ based on certain assumptions and/or criteria. The output partition, which includes both the number of classes (c) and the class membership structures, is dependent on the criteria that are used to control the clustering algorithm.

Many algorithms have been developed to obtain clusters from a given data set. Among those, the c-means algorithms and their generalization, the ISODATA clustering methods, are probably the most widely used. In the case of c-means algorithms, c is assumed to be known, whereas in the case of ISODATA algorithms, c is unknown. The performance of both models is influenced by the choice of c, the initial cluster centers, the order in which the samples are taken as input, the choice of distance measure and the geometric properties of the data. In practice, the performance of any cluster-seeking algorithm depends in no small way on the cleverness of its designers and users. Labeled data are useful for testing the validity of the clustering methodologies.

The research in cluster analysis was started long back as early as in 1940s [32]. Since then scientists have proposed many approaches [33-39]. Clustering is one of the areas in which fuzzy set theory has been extensively applied. The development of fuzzy clustering algorithms is furnished in section 1.4.5. ♣

1.2.5 Classification

The problem of classification is to assign every data point in the entire feature space to one of the possible (M) pattern classes. On the contrary, the clustering algorithms label the given data set $\mathcal{X} \subset \Omega_X$ ($\mathbb{R}^N$) into a number (c) of classes (i.e., subgroups). Classifiers are usually, but not always, designed

with labeled data, in which case these problems are sometimes referred to as *supervised classification* (where the parameters of a classifier function D are learned). Many clustering algorithms are used as precursors to the design of a classifier when the only data available are unlabeled data. In such cases, the problems are sometimes referred to as *unsupervised classification.* In either case, the partitioning decision functions may be computationally explicit (*e.g.*, discriminant functions, nearest prototype rules) or implicit (*e.g.*, multilayered perceptron, k nearest neighbor rules). Various approaches of classification are described below.

A. Deterministic technique

Let $C_1, C_2, \ldots, C_j, \ldots, C_M$ be the M possible pattern classes in an N-dimensional feature space. Let $X = [x_1, x_2, \ldots, x_i, \ldots, x_N]'$ be an unknown pattern vector. In the deterministic classification approach, it is assumed that there exists only one unambiguous pattern class corresponding to each of the unknown pattern vectors.

If the pattern X is a member of class C_j, the *Discriminant (decision) function* $D_j(X)$ associated with the class C_j ($j = 1, 2, \ldots, M$) must then possess the largest value. That is, a classificatory decision would be as follows:

$$\text{Decide } X \in C_j, \qquad \text{if } D_j(X) > D_k(X) \tag{1.1}$$

in which j and k have any integer value $1, 2, \ldots, M$ and $j \neq k$. Ties are resolved arbitrarily. The decision boundary (in the feature space) between regions associated with the classes C_j and C_k respectively would be governed by the expression

$$D_j(X) - D_k(X) = 0 \tag{1.2}$$

with $j \neq k$ and $j, k = 1, 2, \ldots, M$. Many different forms satisfying Eq. (1.2) can be selected for $D_j(X)$. The functions that are often used are single prototype linear discriminant function, piecewise linear discriminant function, quadratic discriminant function, polynomial discriminant function [2-19]. •

B. Probabilistic technique

In the previous method, the features are assumed to be deterministic. But in most of the practical problems, large variations in the feature measurements and noise effect involved in experiment cannot be avoided. To demonstrate such systems, the feature values x_1, x_2, ..., x_N are considered as random values in the probabilistic approach. Bayes classifier [40], being the most established one, is described below.

Bayes classifier [40] : Let $P(C_j)$ denote the a priori probability and $p(X/C_j)$ denote the class-conditional density corresponding to the class C_j ($j = 1, 2, \ldots, M$). If the classifier decides X be from the class C_j when it actually comes from C_k, it incurs a loss equal to L_{kj}. The expected loss (conditional average loss or risk) incurred in assigning an observation X to the class C_j is given by

$$r_j(X) = \sum_{k=1}^{M} L_{kj}\ p(C_k/X) \tag{1.3}$$

where $p(C_k/X)$ represents the probability that X is from C_k. Using Bayes formula,

$$r_j(X) = \frac{1}{p(X)} \sum_{k=1}^{M} L_{kj}\ p(X/C_k) P(C_k) \tag{1.4}$$

$$\text{where} \quad p(X) = \sum_{k=1}^{M} p(X/C_k) P(C_k).$$

The pattern X is assigned to the class with the smallest expected loss. The classifier which minimizes the total expected loss is called the *Bayes Classifier*.

Let us assume that the loss (L_{kj}'s) is nil for correct decisions, and it is same for all erroneous decisions. In such cases, the expected loss [Eq. (1.4)] becomes

$$r_j(X) = 1 - \frac{p(X/C_j)P(C_j)}{p(X)}. \tag{1.5}$$

Since $p(X)$ is independent of class, the Bayes decision rule is nothing more than the implementation of the decision functions

$$D_j(X) = p(X/C_j)P(C_j), \qquad j = 1, 2, \ldots, M \tag{1.6}$$

where a pattern X is assigned to the class C_j if for that pattern $D_j(X) > D_k(X)$ for all $j \neq k$.

If one assumes $p(X/C_j)$ to be a Gaussian density with mean vector μ_j and covariance matrix Σ_j, i.e.,

$$p(X/C_j) = \frac{1}{(2\pi)^{N/2} \mid \Sigma_j \mid^{1/2}} exp\left[-\frac{1}{2}(X-\mu_j)'\Sigma_j^{-1}(X-\mu_j)\right] \qquad (1.7)$$

$$j = 1, 2, \ldots, M$$

then $D_j(X)$ becomes

$$D_j(X) = lnP(C_j) - \frac{1}{2} ln \mid \Sigma_j \mid -\frac{1}{2}(X-\mu_j)'\Sigma_j^{-1}(X-\mu_j) \qquad (1.8)$$

$$j = 1, 2, \ldots, M.$$

The decision functions in Eq. (1.8) are hyperquadrics. This is apparent, there being no terms higher than the second degree in the components of X. Thus a Bayes classifier for normally distributed patterns in reality defines second order decision surfaces between each pair of pattern classes. However, if the pattern classes are truly characterized by normal densities, no other surfaces yield better results on an average. The performance of this classifier has been compared with that of a multivalued classifier in chapters 4 and 5. •

A review on various approaches demonstrating the ways of incorporating the concept of fuzzy sets at different stages for designing such classifiers has been provided in section 1.4.6. •

Another important development during the early days of pattern recognition was the perceptron [41,42]. It may be defined as a network of elementary processors arranged in a manner reminiscent of biological neural nets which will be able to learn how to recognize and classify patterns in an autonomous manner. In such a system the processors are simple linear elements in one layer. This classical (single layer) perceptron, given two classes of patterns, attempts to find a linear decision boundary separating the two classes. If the two sets of patterns are linearly separable, the perceptron algorithm is guaranteed to find a separating hyper-plane in a finite number of steps. However,

if the pattern space is not linearly separable, the perceptron fails and it is not known when to terminate the algorithm in order to get a reasonably good decision boundary. Thus, a single layer perceptron is inadequate for situations with multiple classes and non-linear separating boundaries. This motivated the development of a multi-layer network (layers between input and output layers being called hidden layers) with non-linear learning algorithms known as the multi-layer perceptron (MLP) [41,42].

Applications of this model can be found in [43-51]. Various ways of integrating fuzzy pattern recognition and different neural networks have been described in section 1.4.8. ♣

1.2.6 Image processing and recognition

We have mentioned in section 1.2.1 that when the input to a pattern recognition system is a gray tone image, the measurement space usually involves processing tasks such as enhancement, filtering, noise reduction, segmentation, contour extraction, and skeleton extraction in order to derive salient features from the image pattern. This is what is generally known as *image processing*. These processed information are then used to compute various properties (*e.g.*, area, perimeter, centroid, *etc.*) and primitives (*e.g.*, line, corner, curve, *etc.*) of and relationships among the regions for developing decision rules/grammars. The ultimate aim is to enable the system to understand, recognize and interpret the input image pattern. Such a complete image recognition/interpretation system is called a *vision system* which may be viewed as consisting of three levels, namely low level, mid level and high level.

Basic principles and operations of image processing and computer vision are available in [52-55]. Relevance of fuzzy set theory in image analysis and recognition has been described in section 1.4.7. ♣

1.2.7 Pattern recognition applications

So far various aspects of a pattern recognition systems are discussed in brief. As mentioned in section 1.1, pattern recognition research is mostly driven by the need to process data and information obtained from the interactions between science, technology, and society in general. Researchers in this area are concerned with the idea of designing and making intelligent machines that can carry out certain tasks with skills comparable to human performance. Application areas for pattern recognition include, but are not limited to

(a) *man-machine communication* : automatic speech recognition, script recognition, speech understanding, image understanding, natural language processing;

(b) *medicine* : medical diagnosis, image analysis, disease classification;

(c) *vehicular* : automobile, airplane, train, boat controllers;

(d) *defense* : automatic target recognition, guidance, control;

(e) *police and detective* : crime and criminal detection from speech, handwriting, fingerprint, photograph;

(f) *natural resource study and estimation* : agriculture, forestry, geology, environment;

(g) *industry* : CAD, CAM, product testing and assembly, inspection and quality control;

(h) *remote sensing* : detection of man-made objects and estimation of natural resources;

(i) *domestic systems* : appliances. ♠

1.3 Uncertainties in Pattern Recognition

Uncertainties always exist either explicitly or implicitly in each and every phase of a pattern recognition system. Some of them which one often encounters while designing a pattern recognition system are explained in this

section. Let us consider, first of all, the case of a decision theoretic approach to pattern classification. With the conventional probabilistic and deterministic classifiers [2-8], the features characterizing the input patterns are considered to be quantitative (numerals) in nature. The pattern vectors having imprecise or incomplete specification are usually ignored or discarded from the design and test sets. The impreciseness (or ambiguity) may arise from various reasons. For example, instrumental error or noise corruption in the experiment may lead to partial or partially reliable information available on a feature measurement. Again, in some cases the expense incurred in extracting a very precise exact value of a feature may be high, or it may be difficult to decide on the most salient features to be extracted. For these reasons, it may become convenient to use the linguistic variables and hedges (*e.g.*, small, medium, high, very, more or less *etc.*) in order to describe the feature information. In such cases, it is not appropriate to give exact representation to uncertain feature data. Rather, it is reasonable to represent uncertain feature information by fuzzy subsets.

Again, the uncertainty in classification or clustering of patterns may arise from the overlapping nature of the various classes. This overlapping may result from fuzziness or randomness. In the conventional classification technique, it is usually assumed that a pattern belongs to only one class, which is not necessarily realistic physically, and certainly not mathematically. A pattern can and should be allowed to have degrees of membership in more than one class. It is therefore necessary to convey this information while classifying a pattern or clustering a data set.

Similarly, consider the problem of determining the boundary or shape of a class from its sampled points (i.e., training samples). There are various approaches [56-61] described in the literature which attempt to estimate an exact shape for the area in question by determining a boundary that contains (i.e., passes through) some or all of the sample points. This is not necessarily true in practice. It may be necessary to extend the boundaries to some extent to represent the possible uncovered portions by the sampled points. The extended portions should have lower possibility to be in the class than the

portions explicitly highlighted by the sample points. The size of the extended regions should also decrease with an increase in the number of sample points. This leads one to define a multivalued or fuzzy (with continuum grade of belonging) shape and boundary of a pattern class.

Let us now consider the problem of processing and recognizing a gray tone image pattern. A gray tone image possesses some ambiguity within the pixels due to the possible multivalued levels of brightness. This pattern indeterminacy is due to inherent vagueness rather than randomness. In a conventional vision system, each operation (as mentioned in section 1.2.6) involves crisp decision (i.e., yes or no, black or white, 0 or 1) to make regions, features, primitives, relations, and interpretations crisp. Since the regions in an image are not always crisply defined, uncertainty can arise at every phase of the recognition tasks. Therefore, a recognition system should have sufficient provision for representing the uncertainties involved at every stage i.e., in defining image regions, its features and relations among them, and in their matching, so that it retains as much as possible the information content of the original input image for making the final output decision.

From the aforesaid discussion, it becomes therefore convenient, natural and appropriate to avoid committing ourselves to a specific (hard) decision by allowing the segments or skeletons or contours to be fuzzy subsets of the image; the subsets being characterized by the possibility (degree) of a pixel belonging to them. Similarly, for describing and interpreting ill-defined structural information in a pattern, it is natural to define primitives (line, corner, curve *etc.*) and relations among them using labels of fuzzy sets. For example, primitives which do not lend themselves to precise definition may be defined in terms of arcs with varying grades of membership from 0 to 1 representing its belonging to more than one class. The production rules of a grammar may similarly be fuzzified to account for the fuzziness in physical relation among the primitives; thereby increasing the generative power of a grammar for syntactic recognition of a pattern.

From the aforementioned examples, we see that the concept of fuzzy sets can be used at the feature level in representing input data as an array of membership values denoting the degree of possession of certain properties, in representing linguistically phrased input features for their processing, in weakening the strong commitments for extracting ill-defined image regions, properties, primitives, and relations among them, and at the classification level, for representing class membership of objects, and for providing an estimate (or a representation) of missing information in terms of membership values. Therefore, fuzzy set theory can be incorporated in the handling of uncertainties (arising from deficiencies in the available information caused by, among others, incomplete, imprecise, ill-defined, not fully reliable, vague, and contradictory data and information) in various stages of a pattern recognition system.

The present thesis confines itself in providing various new methodologies for handling uncertainties in a pattern recognition system based on fuzzy set theory. Their applications to speech data and remote sensing imagery have also been demonstrated. Before explaining the scope of the thesis in detail, we shall present a review on fuzzy set theoretic approaches for pattern recognition preceded by a brief discussion on various key features of fuzzy set theory. ♠

1.4 Fuzzy Models for Pattern Recognition

Fuzzy sets were introduced in 1965 by Zadeh [1] as a new way to represent vagueness in everyday life. They are a generalization of conventional set theory, one of the basic structures underlying computational mathematics and models. Computational pattern recognition has played a central role in the development of fuzzy models because fuzzy interpretations of data structures are a very natural and intuitively plausible way to formulate and solve various problems. Fuzzy control theory has also provided a wide variety of real, fielded system applications of fuzzy technology where the techniques of pattern recognition and image processing interact with and support many

control systems.

1.4.1 Fuzzy sets

A fuzzy set $\mathcal{A}$ in a space of points $\mathcal{X} = \{x\}$ is a class of events with a continuum of grades of membership and is characterized by a membership function $\mu_{\mathcal{A}}(x)$ which associates with each element in $\mathcal{X}$ a real number in the interval [0, 1] with the value of $\mu_{\mathcal{A}}(x)$ at x representing the grade of membership of x in $\mathcal{A}$. Formally, a fuzzy set $\mathcal{A}$ with its finite number of supports $x_1, x_2, \ldots, x_t$ is defined as a collection of ordered pairs

$$\mathcal{A} = \{(\mu_{\mathcal{A}}(x_i), x_i),\ i = 1, 2, \ldots, t\} \tag{1.9}$$

where the support of $\mathcal{A}$ is an ordinary subset of $\mathcal{X}$ and is defined as

$$S(\mathcal{A}) = \{x \mid x \in \mathcal{X} \ \text{ and } \ \mu_{\mathcal{A}}(x) > 0\}. \tag{1.10}$$

Here μ_i, the grade of membership of x_i in $\mathcal{A}$, denotes the degree to which an event x_i may be a member of $\mathcal{A}$ or belong to $\mathcal{A}$. $\mu_i = 1$ indicates strictly the containment of the event x_i in $\mathcal{A}$. If, on the other hand, x_i does not belong to $\mathcal{A}$ then $\mu_i = 0$. Basic operations related to fuzzy sets are available in [15-19,62-69]. •

1.4.2 Fuzzy logic

Logic, according to Webster's dictionary is the science of the normative formal principles of reasoning. In this sense, fuzzy logic (based on the theory of fuzzy sets) is concerned with the formal principles of approximate reasoning, with precise (classical) reasoning viewed as a limiting case. Unlike classical logic, it aims at modeling the imprecise (or inexact) modes of reasoning and thought processes (with linguistic variables) that play an essential role in the remarkable human ability to make rational decisions in an environment of uncertainty and imprecision. This ability depends, in turn, on our ability to infer an approximate answer to a question based on a store of knowledge

that is inexact, incomplete, or not totally reliable. In fuzzy logic everything, including truth, is a matter of degree [70,71].

The reason why classical logical systems can not cope with the common sense rules is as follows. First, they do not provide a system for representing the meaning of propositions expressed in a natural language when the meaning is imprecise; and second, in those cases in which the meaning can be represented symbolically in a meaning representation language (*e.g.*, a semantic network or a conceptual-dependency graph), there is no mechanism for inference.

Fuzzy logic addresses these problems in the following ways. First, the meaning of a lexically imprecise proposition is represented as an elastic constraint on a variable; and second, the answer to a query is deduced through a propagation of elastic constraints [71].

Zadeh has developed a theory of approximate reasoning [72] based on fuzzy set theory. By approximate reasoning, we mean a type of reasoning which is neither very exact nor very inexact. This theory aims at modeling the human reasoning and thinking process with linguistic variables [73-76] in order to handle both soft and hard data as well as various types of uncertainties. Many aspects of the underlying concept have been incorporated in designing decision-making systems [77-81]. Further discussion on approximate reasoning has been made in section 3.2.2 of chapter 3. •

1.4.3 Membership function and its uncertainty

Because fuzzy sets are a generalization of the classical set theory, the embedding of conventional models into a larger setting endows fuzzy models with greater flexibility to capture various aspects of incompleteness or imperfection (i.e., deficiencies) in whatever information and data are available about a real process. Another way to say this is to imagine that membership functions possess *elasticity*; thus, the higher the value of membership of an object to a class, the less the imprecisely defined concepts of the fuzzy set

must be stretched to accommodate the object. Hard membership functions, of course, are *inelastic*.

Assignment of membership function of a fuzzy subset is subjective in *nature*, and reflects the context in which the problem is viewed. It can not be assigned arbitrarily. In many cases, it is convenient to express the membership function of a fuzzy subset of the real line in terms of standard S and π *functions* [18,68].

Note that fuzzy membership function and probability density function are conceptually different. Probabilities convey information about relative frequencies of objects while fuzzy membership represent similarities of objects to imprecisely defined properties. There are many amusing articles about the relationship between fuzzy sets and probability in the literature [82-88].

Since the grade of membership is both subjective and dependent on context, some difficulty of adjudging the membership value still remains. In other words, the problem is how to assess the membership of an element to a set. This is an issue where opinions vary, giving rise to uncertainties. Two operations, namely *Bound Functions* [89] and *Spectral Fuzzy Sets* [90] have recently been defined to analyze the flexibility and uncertainty in membership function evaluation for pattern recognition problems. Besides these, there are other well defined concepts of Type 2, Type 3, ..., Type n fuzzy sets, ultrafuzzy sets, interval-valued fuzzy sets, L-valued fuzzy sets and probability sets which also consider the difficulty in settling a definite degree of ambiguity in a set [65,85,91-93].

Bound functions [89] were derived based on the properties of fuzzy correlation [94] between membership functions in order to restrict the variation in a membership function. The types of membership functions which should preferably be avoided in representing fuzzy sets are categorized with the help of these bound functions. The significance of the bound functions in selecting an S-type membership function for image segmentation problem has been reported in detail in [95].

must be stretched to accommodate the object. Hard membership functions, of course, are *inelastic*.

Assignment of membership function of a fuzzy subset is subjective in nature, and reflects the context in which the problem is viewed. It can not be assigned arbitrarily. In many cases, it is convenient to express the membership function of a fuzzy subset of the real line in terms of standard S and π functions [18,68].

Note that fuzzy membership function and probability density function are conceptually different. Probabilities convey information about relative frequencies of objects while fuzzy membership represent similarities of objects to imprecisely defined properties. There are many amusing articles about the relationship between fuzzy sets and probability in the literature [82-88].

Since the grade of membership is both subjective and dependent on context, some difficulty of adjudging the membership value still remains. In other words, the problem is how to assess the membership of an element to a set. This is an issue where opinions vary, giving rise to uncertainties. Two operations, namely *Bound Functions* [89] and *Spectral Fuzzy Sets* [90] have recently been defined to analyze the flexibility and uncertainty in membership function evaluation for pattern recognition problems. Besides these, there are other well defined concepts of Type 2, Type 3, ..., Type n fuzzy sets, ultrafuzzy sets, interval-valued fuzzy sets, L-valued fuzzy sets and probability sets which also consider the difficulty in settling a definite degree of ambiguity in a set [65,85,91-93].

Bound functions [89] were derived based on the properties of fuzzy correlation [94] between membership functions in order to restrict the variation in a membership function. The types of membership functions which should preferably be avoided in representing fuzzy sets are categorized with the help of these bound functions. The significance of the bound functions in selecting an S-type membership function for image segmentation problem has been reported in detail in [95].

The concept of spectral fuzzy sets [90] is used where, instead of a single unique membership function, a set of functions reflecting various opinions on membership elements is available so that each membership grade is attached to one of these functions. By giving due respect to all the opinions available for further processing, it reduces the difficulty (ambiguity) in selecting a single function. The various properties and its application in providing soft decision in image analysis problem are reported in [90].

In this context, the work of Kandel and Byatt [96] needs to be mentioned which shows that the concept of fuzzy expected value (FEV) can be used for dealing with missing, imprecise or unreliable information in pattern recognition problems. The idea is that, once a membership function is known, there should be some way of estimating an average value of the membership function that is representative of that possibility distribution [97-100]. Thus, if the possibility distribution is known for a feature, but the value of the membership function is missing for a specific pattern, we can use the estimated FEV in its stead. When the data do not have complete information about the distribution of the population and their grades of membership, the concept of FEV is not applicable. In order to tackle this kind of problem, the concept of fuzzy expected interval (FEI) may be effective. Details of FEV and FEI measures can be found in [17,96,101,102]. •

1.4.4 Measures of fuzziness

The term *uncertainty* has a broad semantic context. It has various meaning, namely *vague*, *questionable*, *ambiguous*, *doubtful*, *varying*, *not reliable* etc. Uncertainty that arises due to randomness in a system is called *probabilistic uncertainty*. The uncertainty that arises due to limitation of the evidence gathering and interpretating system i.e., due to a difficulty in specifying the exact solution is called *nonspecificity*. There is another type of uncertainty known as *fuzzy uncertainty* that arises due to the presence of fuzziness in the system.

Membership values determine how much fuzziness a fuzzy set contains. Quantification of the amount of imprecision captured depends on the extend to which the supporting objects (as individuals or in a group) do or do not possess the concept or property represented by the fuzzy set. The higher the extend, the lower is the fuzziness of the set, and conversely. Several measures of fuzziness are found in literature, *e.g., indices of fuzziness, crispness, entropy, certitude, ambiguity etc.* [62,83,87,103-108]. Some of these are based on notions such as the distance between a fuzzy set and its nearest (furthest) crisp set, distance between a fuzzy set and its complement, or Shannon's classical entropy function. Usually, these measures have properties such as being minimum when $\mu(x) = 0$ or 1 for all x and maximum when $\mu(x) = 0.5$ for all x. Some applications of several of these measures are reported in [18].

All the aforementioned measures provide a global measure of uncertainty (i.e., the average amount of ambiguity related to all individual supporting elements of a fuzzy set). There are some measures to assess the average amount of imprecision associated with a group (not the individual elements). *rth order entropy* H^r [107] of a fuzzy set is one of them and it provides a measure of the average uncertainty (ambiguity) in making a decision on any subset with r elements as regards its possession of the imprecise property. The significance of this measure for pattern recognition and image processing problems is described in [107].

Another information measure of a fuzzy set called *hybrid entropy* H_{hy} represents the amount of difficulty in deciding whether an element belongs to a fuzzy set or not by making a prevision on its probability of occurrence. This measure takes into account not only the fuzziness of a set, but also the underlying probability structures of the supporting elements. The significance of hybrid entropy in image enhancement and noise reduction problems is described in [107].

It is to be noted that fuzzy uncertainty differs from probability and nonspecity. Fuzzy uncertainty deals with the situations where boundaries of the sets under consideration are not sharply defined - partial occurrence of an

event. On the other hand, for probabilistic and nonspecific uncertainties there is no ambiguity about the set boundaries, but rather, about the belongingness of the elements or events to crisp sets. Interested readers may consult the articles [65,100,109-117] for further references on various uncertainty measures. •

It may be mentioned here that rough set theory [118,119] has gained a significant popularity recently in modeling uncertainty. Although the fuzzy set theory and rough set theory both offer approaches to deal with uncertainty, these two concepts are completely independent. However there is a connection between rough set theory and Dempster-Shafer theory, though they have been developed separately. Dempster-Shafer theory uses the belief function as a main tool, whereas the rough set theory makes use of the family of all sets with common lower and upper approximations [119,120]. ♣

It may be mentioned that from the very beginning of the development of fuzzy set theory, its application to pattern recognition played a very significant role. It is two fold: (i) a methodological one - this leads to treatment of fuzzy sets as a well-suited theory within which one can establish a plausible tool for modeling and mimicking cognitive process of the human being, especially those concerning recognition aspects; (ii) secondly, fuzzy sets offer a lot of novel algorithms which are useful for designing of feature analysis and classification procedures.

Recently, Bezdek and Pal [19] have provided an excellent review on various approaches which helped in the evolution of fuzzy pattern recognition. Readers may also consult in this context the review article of Pedrycz [13]. In the following sections, we present the state of art of methodology and methods of fuzzy sets for various tasks of pattern recognition. We have already explained through examples the relevance of fuzzy set theory in handling uncertainty at different stages of a pattern recognition system in section 1.3.

1.4.5 Fuzzy clustering

The objective of clustering is to partition any finite data set $\mathcal{X}$ of objects into a certain number (c) of natural and homogeneous subsets, where the elements of each set are as similar as possible to others and at the same time, as different from those of the other sets as possible. Let $1 < c < t$ and $\mathcal{X} = \{X_1, X_2, \ldots, X_t\}$ denote a set of (t) unlabeled feature vectors in $\mathbb{R}^N$. Given $\mathcal{X}$, we say that (c) fuzzy subsets $\{\mu_j\colon\ \mathcal{X} \rightarrow [0.1]\}$ are a fuzzy c-partition of $\mathcal{X}$ in case the (ct) values $\{\mu_{jk} = \mu_j(X_k),\ 1 \leq k \leq t,\ 1 \leq j \leq c\}$ satisfy three conditions [19]:

$$\begin{aligned} 0 \ \leq \quad & \mu_{jk} \quad \leq \ 1 \quad \forall j,k \\ & \sum_j \mu_{jk} \ = \ 1 \quad \forall k \\ 0 \ < \ & \sum_k \mu_{jk} \ < \ t \quad \forall j. \end{aligned} \tag{1.11}$$

Each set of (ct) values satisfying the above conditions can be arranged as a matrix $U = [\mu_{jk}]^{c \times t}$. The set of all such matrices is the nondegenerate fuzzy c-partitions of $\mathcal{X}$:

$$I_{fct} \ = \ \left\{U \text{ in } \mathbb{R}^{ct} \ \mid \ \mu_{jk} \text{ satisfies Eq. (1.11) } \forall j,k\right\}. \tag{1.12}$$

In case all the μ_{jk} are either 0 or 1, we have the subset of hard (or crisp) c-partitions of $\mathcal{X}$:

$$I_{ct} \ = \ \{U \text{ in } I_{fct} \ \mid \ \mu_{jk} \ = \ 0 \text{ or } 1 \ \ \forall j,k\}. \tag{1.13}$$

The reason these matrices are called partitions follows from the interpretation of μ_{jk} as the membership of X_k in the jth partitioning subset (cluster) of $\mathcal{X}$. I_{fct} is more realistic as a physical model than I_{ct}, for it is common experience that the boundaries between many classes of real objects are in fact ill-defined and/or overlapping. So I_{fct} provides a much richer means of representing and manipulating data that have such structures than does I_{ct}. An optimal partition of $\mathcal{X}$ in I_{fct} is the one which groups together object data vectors that share some well-defined mathematical similarity. We mention that c is assumed to be known; otherwise, its value becomes a part of the clustering problem. This facet of the problem is often called the *cluster validity* question.

Interested readers should consult [15] for more details about the algebraic and geometric nature of the partition spaces.

A pioneering application of the theory of fuzzy sets to clustering analysis was made in 1969 by Ruspini [33]. He first suggested the structure (I_{fct}) of fuzzy c-partition spaces. He further defined and analyzed the first fuzzy objective function algorithm for generating fuzzy c-partions of a finite set of unlabeled data. Almost all of the fuzzy clustering models that are currently used are based on Ruspini's idea.

The approach proposed by Gitman and Levine in [34] demonstrates an way of decomposing *mixtures* or data with multimodality, using fuzzy sets. They have suggested augmentation of the input data with numbers that represented linguistic characteristics of the inputs, to decompose the data with this added information. The research work by Dunn and Bezdek [36,37,121] on the fuzzy ISODATA (or fuzzy c-means) algorithms may be considered as a landmark in the theory of cluster analysis. Their contribution is extremely important and has had lasting and far-reaching implications in the area of fuzzy clustering.

Roubens [122] made an important contribution in the area of relational clustering based on optimizing an objective function of the unknown fuzzy partition and the data at hand. Some of the early indices of fuzziness that were being studied in the context of cluster validity were also discussed in [122]. Gustafson and Kessel [123] incorporated an important concept of localized shape matching via norms that adopted to the shapes of single clusters. Windham [124] extended the ideas of geometric matching of algorithms to data somewhat beyond the confines of fuzzy models.

Backer [38] introduced a clustering model which tries to achieve an optimal decomposition of a group of objects into a collection of induced fuzzy sets by means of a *point-to-subset affinity* concept on the basis of the structural properties among objects in the representation space. He suggested a number of affinity mechanisms based on the distance concept, the neighborhood concept, and the probabilistic concept. An iterative optimization-partitioning

procedure has been adopted to obtain decomposition of the sample points to the final clusters. The iterative procedure is guided by some criterion functions based on measures of fuzziness, inter fuzzy set distance, and measure of fuzzy similarity. Recently Pal and Mitra [125] formulated a fuzzy dynamic clustering algorithm based on the concept of induced fuzzy sets when the number of classes is unknown.

There have been many studies [126,127] on the idea of synthesis between fuzzy clustering models and statistical decision theory. One of the major issues in the clustering approaches is the verification of their convergence property. The research works [127-129] address issues about convergence of the fuzzy c-means algorithms. There is another significant research area in clustering which concerns the problem of clustering validity. This issue has been addressed in the articles [130-134]. Additional results on fuzzy clustering are available in [135,136]. •

1.4.6 Fuzzy classification (classifier design)

A more ambitious, difficult, and potentially useful computational problem than clustering, *classifier design* refers to finding a hard or fuzzy partition of $\mathbb{R}^N$ itself. A classifier is capable, once it is defined, of labeling every data point in the entire space $\mathbb{R}^N$. To characterize fuzzy classifier, let us first define the hard and fuzzy label vector sets in $\mathbb{R}^M$ [19] :

$$\begin{aligned} J_M &= \left\{ Y \in \mathbb{R}^M \mid \sum_k y_k = 1;\ y_k \in \{0,1\}\ \forall k \right\} \\ &= \text{hard label vectors for } M \text{ classes.} \end{aligned} \tag{1.14}$$

$$\begin{aligned} J_{fM} &= \left\{ Y \in \mathbb{R}^M \mid \sum_k y_k = 1;\ y_k \in [0,1]\ \forall k \right\} \\ &= \text{fuzzy label vectors for } M \text{ classes.} \end{aligned} \tag{1.15}$$

J_M is just the usual canonical basis of Euclidean M-space, and J_{fM} is its convex hull. As usual, $J_M \subset J_{fM}$.

With J_{fM}, a *classifier* is defined on $\mathbb{R}^N$ as any function D imaged in J_{fM}.

That is, classifiers are a special kind of vector field, which is generally denoted as

$$\mathcal{D} : I\!R^N \rightarrow J_{fM} \subset R^M.$$

Thus the value of $\mathcal{D}$ at any $X \in I\!R^N$ is $Y = \mathcal{D}(X)$, the label vector for X in J_{fM}.

Fuzzy classifiers design almost always means arriving at a hard classifier (such as in Eq. (1.1)), but uses the idea of fuzziness somewhere upstream. The approaches [22,39,137-146] show that the incorporation of fuzzy ideas in the model leading to a hard classifier design do indeed (sometimes) yield better hard classifiers than those that result from looking for a hard design to begin with. This can again be attributed to the idea of embedding: we find a better solution to a crisp problem by looking in a large space at first, which has different (usually less) constraints and therefore allows the algorithm more freedom to avoid errors forced by commission to hard answers in intermediate stages.

The research on the application of fuzzy set theory to supervised pattern recognition was started in 1966 in the seminal note of Bellman, Kalaba and Zadeh [137] where the two basic operations, namely *abstraction* and *generalization* were proposed. Abstraction in fuzzy set theory means estimation of a membership function μ of a fuzzy class from the training samples. Having obtained the estimate, generalization is performed when this estimate is used to compute the values of μ for unknown objects not contained in the training set. Consideration of linguistic features and fuzzy relations in representing a class has also been suggested by Zadeh [138]. The work by Pal and Dutta Majumder [139] outlines an early application of fuzzy sets for decision theoretic classification, where a pattern is considered as an array of linguistically pharased features denoting certain properties and where each of these features is a fuzzy set. The variation of the recognition score (for speech data) with the change of fuzziness in the linguistically phrased feature values has subsequently been reported in [140]. These classifiers have also been used for designing a self-supervised recognition system [26,141,142]. Nath *et al.* [143,144] proposed a classification model applicable in the soft sciences (*e.g.*,

medical diagnostics) where enough apriori knowledge about the classifier is available from expert in linguistic form. This algorithm involves the theory of approximate reasoning [72,77,78] and fuzzy relations between fuzzy statements.

A fuzzy version of the well known *k-nearest neighbor (k-NN)* classifier has been provided by Keller *et al.* [39]. In the conventional approach [2-8], each of the labeled samples is given equal importance in deciding the class membership of an unknown pattern; this frequently causes problems in places where the labeled samples overlap. This is tackled in [39] by providing fuzzy label vectors for the samples as an indication of their class representativeness and subsequently leads to a fuzzy classification rule. The fuzzy version of the k-nn rules seems to offer better performance (lower error rates) than crisp rules [39,147].

An adaptive system can be viewed as a learning machine in which system's decisions gradually approach optimal decisions by acquiring the necessary information from observed patterns. The approaches considered so far use a specified set of labeled data for training of a classifier before it is used for classifying unknown patterns. Devi and Sharma [28] proposed an adaptive algorithm using a fuzzy approximation to the gradient descent technique for training a classifier sequentially. They suggested a method for eliminating or discarding doubtful or unreliable samples from the training procedure.

Although, the task of feature selection plays an important role in designing a pattern recognition system, the research in this area using fuzzy set theory has not been significant. Bezdek and Castelaz [22] showed an application of fuzzy c-means clustering algorithm to select an optimum feature subset from the available features so that there is no appreciable loss of classifier performance with the reduced set of features. Pal and Chakraborty explained in [146] an application of fuzziness measures (the index of fuzziness, entropy, and π-ness) of a set in selecting features without going through classification. This work has then been extended to evaluate the importance of any subset of features to provide an average quantitative index of goodness.

Chang and Pavlidis [145] incorporated the concept of fuzzy decision trees in developing an efficient algorithm for making decisions in pattern recognition problems. Fuzzy tree automata are defined by Lee [148] for processing fuzzy tree representations of patterns using syntactic recognition. This work shows how membership functions for structural patterns can be defined and how fuzzy language can be used for handling imprecision in structural pattern recognition. Further works on fuzzy syntactic classifier is available in [11,25,149-153].

There have been many attempts showing the application of fuzzy set theoretic approaches to real life recognition problems. Some significant attempts can be found in literature [149,141,154,155] for recognizing speech patterns which are biological in origin and the patterns manifest a considerable amount of fuzziness (vagueness). Pathak and Pal [152,156] demonstrated an application of fuzzy and fractionally fuzzy grammars [149,150] in syntactic recognition of ages of different bones from x-ray image patterns. They have shown that incorporation of the concept of fuzziness in defining sharp, fair, and gentle curves and the production rules used enable one to work with a smaller number of primitives and to use the same set of rules and nonterminals at each stage. Therefore, one needs to parse an input string with only one grammar at each stage, unlike the case in the nonfuzzy approach, where one may have to parse each string by more than one grammar, in general, at each stage. However, this has to be balanced against the fact that the fuzzy grammars are not as simple as the corresponding nonfuzzy grammars. Furthermore, these grammars need not be unambiguous, whereas nonambiguity is an absolutely necessary requirement for the nonfuzzy approach.

Automatic recognition of handwritten characters is another area where ambiguity occurs because of imprecision in writing rather than from randomness, and the fuzzy set theory has been used quite extensively both in feature extraction and in classification. The research reported in [157-161] is representative of the development in the area. Some other significant applications can be found in [14,15,64,68,162-165].

Recently, rule based systems have gained popularity in pattern recognition activities. By modeling the rules and facts in terms of fuzzy sets, it is possible to make interfaces using the concept of approximate reasoning. Such a system has been designed recently for automatic target recognition using 40 rules [166]. A knowledge based approach using Dempster-Shafer theory of evidence [109] has also been formulated [167] for managing uncertainty in object recognition problem when features fail to be homogeneous. Meaningful pay-offs are defined in this context. The problem is tackled by considering masses with fuzzy focal elements. An evidential approach to problem solving was also developed when a large number of knowledge systems (which might give contradictory or inconsistent information) is available [168]. The definitions of credibility and plausibility of Dempster-Shafer theory of evidence when evidences and propositions are both fuzzy in nature are also available [169,170]. •

1.4.7 Fuzzy image processing and recognition

While the application of fuzzy sets in cluster analysis and classifier design (as described earlier) was in the process of development, an important and related effort in fuzzy image processing and analysis was evolving more or less in parallel. This evolution was based on the realization that many of the basic concepts in image analysis (*e.g.*, edge, corner, line, region, relation *etc.*) do not lend themselves well to precise definition.

As mentioned in section 1.3, a gray tone image possesses ambiguity within each pixel because of the possible multivalued levels of brightness. If the gray levels are scaled to lie in the range $[0, 1]$, we can regard the gray level of a pixel as its degree of membership in the set of high valued "bright" pixels. Thus a gray image can be viewed as a fuzzy set. In section 1.3, we explained the relevance of fuzzy set theory in weakening hard decisions for ill-defined image regions for reducing uncertainties in a recognition system. Some of the significant research done in fuzzy image processing and recognition are described below in short.

Fuzzy geometry of a gray image is an area where an extensive work has been carried out. Rosenfeld explained in [171,172] various concepts of fuzzy geometry including the topological concepts of connectedness, adjacency and surroundness, area, perimeter, compactness, height, width, extent, diameter, *etc.* Other geometric properties (*e.g.,* length, breadth, index of area coverage, adjacency, major axis, minor axis, center of gravity, and density) have recently been defined in [173]. The fuzzy versions of various imaging operations, such as shrinking and expanding, thinning, splitting and merging, enhancement and edge detection have been developed [18,106,174-183]. The problem of evaluation of image quality has also been dealt with fuzzy set theory [184-187].

The problem of image segmentation (which is a clustering problem) plays a key role in many areas of recognition, analysis, and description. Prewitt first suggested [188] that the results of segmentation should be fuzzy subsets rather than ordinary subsets. Research in this area can be found in [90,95,181,189-195]. There have been several attempts to extract fuzzy primitives (or features) from fuzzy edges and segmented outputs of the image regions for shape analysis, matching, and recognition. Readers may consult the articles [18,136,148,152,163,196-201]. •

1.4.8 Use of artificial neural networks

Artificial neural networks (ANN) are signal processing systems that try to emulate the human brain, i.e., the behavior of biological nervous systems, by providing a mathematical model of combination of numerous neurons connected in a network. These models are reputed to have the following characteristics: adaptivity (the ability to adjust when given new information), speed (via massive parallelism), fault tolerance (to missing, confusing and/or noisy data) and optimality (as regards error rates in performance). Though there are some common features between different proposed artificial neural network models, they differ in final details and in the underlying philosophy. They are mainly categorized according to the learning strategies involved in

them. For example, Hopfield's model [202,203] employs associative types of learning, multi-layer perceptron [41,42] employs hetero-associative learning, and Kohonen model of self-organizing feature map [204], adaptive resonance theory of Carpenter and Grossberg [205,206] and neo-cognitron of Fukushima [207] try to model the function of regularity detectors. Applications of these networks to pattern recognition problem are discussed in [41,42,47-49,206].

A large number of researchers are now trying to integrate the concepts of neural networks and fuzzy sets in order to exploit the merits of two technologies for developing pattern recognition systems [43,44,46]. The fusion is being mainly tried out in the following two ways [19,208]. First, the concept of fuzziness can be incorporated into an artificial neural network framework, i.e., to build *fuzzy neural networks.* For example, the target output of the neural network during training can be fuzzy label vectors. In this case the network itself is functioning as a *fuzzy classifier* [209,210]. The integration/transfer function at each node can be altered by fuzzy operators so that they perform some sort of fuzzy aggregation (i.e., fuzzy union, intersection etc.) [135]. Input representation can also be considered to be fuzzy. Attempts are also made in [211] to use fuzziness measures of a fuzzy set for modeling error of neural networks. In [212] an approach is provided to design optimal network architecture by optimization of fuzziness of a set. Moreover, there are suggestions in the literature about making individual neurons fuzzy [213,214].

The second fusion methodology is to use neural networks for a variety of computational tasks within the framework of a pre-existing fuzzy model. In this category, for example, attempts are made to have membership function representation by neural networks [215] and implementation of fuzzy logic operations with neural networks [216,217]. For further references on this issue one can consult [43,51].

1.5 Scope of the Thesis

The aim of the thesis is to present some results of investigation, both theoretical and experimental, that demonstrate the effectiveness of the theory of fuzzy sets in formulating some methodologies for handling uncertainties (which may arise from imprecise, incomplete and linguistic input information, and overlapping/ill-defined class boundaries) in certain tasks of pattern recognition. Initially, a procedure for providing multivalued shape of a pattern class from its sampled points has been described. Then the design aspects of two multivalued recognition systems are presented along with the theoretical analysis of their performance. The effectiveness of the proposed systems has been demonstrated on some artificially generated pattern sets as well as some real life problems such as speech recognition and remotely sensed image analysis for identifying ill-defined object regions. The results of investigations are summarized below on the basis of chapter headings.

1.5.1 Determining multivalued shape of a pattern class [218-220]

An important problem in pattern recognition is determining the shape of a pattern class from a set of training samples (i.e., sampled points). A method has been developed in chapter 2 in which the concept of fuzzy sets has been used in performing this task. In estimating the shape, the portions not covered by the sampled points are assigned some fuzzy membership values denoting the degrees of their belongingness to the actual class. Therefore, unlike the conventional approaches [56-61] the proposed method does not attempt to provide crisp boundary from incomplete sample set.

The procedure has two phases. Phase I deals with the decomposition of the sample set into some groups of nearly parallelepiped (rectangular in $I\!R^2$) shape depending on the geometric complexity of the sample set. Phase II determines each of the subclasses corresponding to the sample groups separately, aggregates them and obtains the multivalued shape/boundary of the

pattern class. The theory is initially described for the pattern classes in $I\!R^2$, and then its extension to $I\!R^N$ is elaborated.

The effectiveness of the methodology has been demonstrated on some artificially generated data sets (in $I\!R^2$ and $I\!R^3$) and also on the real life speech data. The convergence of the estimated shape to the original one has been demonstrated experimentally. The convergence property is also verified analytically using Hausdorff metric [221] and a newly defined metric.

The complexity of the proposed algorithm increases exponentially with the dimension (N) of the feature space. It becomes computationally expensive for its implementation in higher dimensional feature spaces ($N > 3$), although we could formulate the algorithm for any dimension.

1.5.1 Multivalued recognition system using linguistic property based decomposition [222,223]

A multivalued recognition system based on fuzzy sets and approximate reasoning has been described in chapter 3 which is capable of handling various imprecise input patterns and of providing four state linguistic (natural) decisions. The input is considered to be in *linguistic*, *quantitative*, *set* or in *mixed form*. For the purpose of processing, an input has been viewed as consisting of various combinations of the three primary linguistic properties *small*, *medium* and *high* possessed by its different features to some degree. The various uncertainties (ambiguities) in the input statement is managed by providing/modifying membership values heuristically to a great extent. Unlike the conventional fuzzy set theoretic approaches, the sets *small* and *high* have been represented here by π functions.

The weight matrices corresponding to various properties and classes have been taken into account in the Zadeh's composition rule of inference [72] in order to make the analysis more effective. The four state linguistic output decision is associated with a confidence factor denoting the degree of certainty of the decision and provides a low rate of misclassification as compared to the conventional two-state systems. The effectiveness of the algorithm has been demonstrated on speech recognition problem.

1.5.3 Multivalued recognition system using geometric structure based decomposition [224-226]

The multivalued recognition system (as described in chapter 3) considers three linguistically phrased property sets (namely *small*, *medium* and *high*) for the input patterns so that each feature information can be viewed to have these properties to some degree. As a result, the entire feature space is decomposed into 3^N (N being the number of features) overlapping *space subdomains* irrespective of the structure and the relative position of pattern classes.

In chapter 4 we have described another recognition system for providing improved performance where the entire feature space is decomposed into some overlapping *space subdomains* depending on the geometric structure (described in chapter 2) and the relative position of the pattern classes found in the training samples. The system is trained with deterministic data, but in the recognizing (testing) phase it can handle input data in either of the four forms (i.e., *linguistic*, *quantitative*, *set* and *mixed forms*). Moreover, there is no concept of weight matrix as used in chapter 3.

A relational matrix corresponding to the *space subdomains* and the pattern classes has been considered in the compositional rule of inference [72] in order to recognize the samples. The system provides linguistic output decisions in four states, namely *single choice*, *first-second choice*, *combined choice* and *null choice*. Each decision is associated with a confidence factor denoting its degree of certainty. The effectiveness of the system has been demonstrated on some artificially generated patterns and also on the real life speech data.

1.5.4 Theoretical analysis of multivalued recognition system [226,227]

The multivalued recognition system (as described in chapter 4) has the ability to discriminate the non-overlapping, and overlapping and no-class (i.e.,

ambiguous) regions for providing output decisions in four states. The *single choices* correspond to the non-overlapping regions, whereas the overlapping regions are reflected by the *first-second* and *combined choices*. The *null choices* reflect the portions outside the pattern classes and/or the portions of the pattern classes uncovered by the training samples. A theoretical analysis of these characteristics and the performance of the recognition system has been provided in chapter 5.

The regions corresponding to the four output forms are initially determined analytically, and the estimates of the overlapping, non-overlapping and no-class regions in the feature space are then obtained. Various situations in one and two dimensional feature spaces with two class problems have been considered. It has been shown theoretically that with the increase in the size of the training samples, the estimates of the overlapping, non-overlapping and no-class regions tend to their actual sizes. All analytical findings have been substantiated with experimental results. As a comparative study of the recognition system with a conventional one, the Bayes classifier is implemented on the same data sets. Bayes decision boundaries are always found to lie within the *combined choice* region of the proposed system. The present investigation, in turn, establishes analytically the justification of considering output decisions in four states for managing uncertainties arising from ambiguous regions.

1.5.5 Analysis of satellite imagery for identifying ill-defined object regions [226]

Chapter 6 describes another real life application of the multivalued recognition system (chapter 4) by analyzing Indian Remote Sensing (IRS) satellite imagery for detecting various ill-defined object regions, namely *roads*, *bridges*, *islands*, *sandbeds*, *city area*, *township/industrial areas*. The multivalued recognition system has initially been applied to classify (based on the spectral knowledge of the image) the image pixels into six classes corresponding to six land cover types, namely *pond water*, *turbid water*, *concrete structure*, *habi-*

tation, *vegetation* and *open space*. The *green* and *infrared* band information are used as features for the classification. The clustered images are processed further for detecting various object regions. In order to detect certain targets, some spatial knowledge about them and their inter-relationships have been incorporated in the algorithms using some heuristic rules.

The concept of multiple choices of the recognition system is found to be very effective for detecting the *road* patterns from concrete structure pixels. Because of the low pixel resolution ($36.25m \times 36.25m$ for IRS imagery) of the remotely sensed images, many portions of the roads may not be classified as concrete structures. In order to identify them, a traversal algorithm through the detected *road* pixels has been proposed here. Some of the movements in the traversal algorithm are governed by only the *second* and *combined choices* of the multivalued recognition system. A thinning algorithm [228] is applied in order to facilitate the traversal process. Some morphological operations [53,55] are used on the clustered image for locating the *city* and *township/industrial areas*. The results are found to agree well with the ground truths.

1.5.6 Conclusions and scope of further research

The concluding remarks and the scope of further research are presented in chapter 7.

Chapter 2

DETERMINING MULTIVALUED SHAPE OF A PATTERN CLASS

Contents

2.1 Introduction

The present chapter deals with the problem of determining the pattern class and its multivalued shape/boundary from sampled points (training samples). Once these are computed, some salient features of the class can then be extracted which are useful in making decisions about a course of action (*e.g.*, identification, classification and pattern description) to be taken later. This will also reduce the storage requirement of the complete pattern-class.

It may be noted that in most of the real life pattern recognition problems, the complete description of a pattern class is not known. Instead, a few sampled points are usually available which are assumed to represent the class. Hence determining the pattern class and its shape from sampled points (or a set of training samples) is extremely useful.

There are various approaches described in the literature for determining the shape of a pattern class from sampled points [56-61]. These methods are mostly heuristic in nature and they provide an exact boundary or shape of the pattern class. One of the inherent observations about these algorithms is that the boundary of the class is restricted by the sampled points. This need not be true because the resulting boundary leaves certain regions not confined in it, although it should be. So, it is necessary to extend the boundaries to some extent to handle the possible uncovered portions by the sampled points. The extended portions should have the following two properties:

(i) As the number of sampled points increases, the extended portions should decrease.

(ii) The extended portions should have less possibility to be in the pattern class than the portions explicitly highlighted by the sampled points.

The second property leads to define a multivalued or fuzzy (with continuum grade of belongingness) boundary of a pattern class. The basic concept of one of the existing methods [56-61] is described below in short for illustration.

Edelsbrunner *et al.* [61] introduced the notion of the *α-shape* of a finite set of points, for arbitrary real α. This notion is a generalization of the *convex hull*. Given a set S of t points in a plane, the *convex hull* of S may be defined as the intersection of all closed half planes that contain all points of S. The *α-hull* of S is the complement of the union of all open discs of radius not less than $1/\alpha$ (for arbitrary negative real α) which contains no point of S. The shape of the set is determined using *α-hulls*. If the selection of α is proper then the shape of the set obtained by the method resembles the intuitive shape of the planar set. These α-hulls have been found to be extremely useful in computational geometry [229]. The selection of α is very important for the method, but the authors did not provide any criterion for the selection of α. Also this method does not provide the multivalued shape of a pattern class.

We have formulated here a methodology for providing multivalued (fuzzy) shape of a pattern class from a set of sampled points. The procedure can be viewed in two phases. Phase I is concerned with the decomposition of sample set into groups of nearly N-dimensional parallelepiped (henceforth parallelepiped) shape. The decomposition is based on the boundary variations (geometric complexity) of the pattern class found in the sampled points. A window approach has been incorporated here to find the boundary variations of the sample groups or sets. Phase II determines each of the subclasses corresponding to the groups separately, puts them together, and finds the multivalued shape of a pattern class. The effectiveness of the procedure has been demonstrated on some artificially generated data sets in $\mathbb{R}^2$ and $\mathbb{R}^3$ feature spaces. The practical applicability of the method has been demonstrated on a real life speech data set. The convergence of the estimated classes to the original ones has been verified using Hausdorff metric and a newly defined metric.

In section 2.2, some basic concepts along with the block diagram of the proposed method are discussed. Section 2.3 deals with the procedure for decomposing a training sample set. Section 2.4 provides an approach to determine the multivalued boundary of a pattern class. Experimental results

are provided in section 2.5. It is to be mentioned here that the sections 2.3, 2.4 and 2.5 concern with two dimensional feature space only. The extension of the procedure to higher dimension ($\mathbb{R}^N$) and its implementation to the pattern classes in $\mathbb{R}^3$ are provided in section 2.6. Section 2.7 deals with the discussion of the convergence property of the proposed algorithm. ♠

2.2 Some Basic Concepts and Block Diagram

Some of the basic concepts which are useful in developing the proposed method are initially stated here. The block diagram of the procedure is then provided.

2.2.1 Some basic concepts

A. Pattern Class

In most of the real life problems, pattern classes are bounded. Thus the pattern classes considered here are all bounded. A formal definition of pattern class in $\mathbb{R}^N$ is given below using topological and measure theoretic concepts.

Definition 2.1 : *A set $\mathcal{A} \subseteq \mathbb{R}^N$ is said to be a pattern class [61] if*

(*i*) *$\mathcal{A}$ is path connected compact,*

(*ii*) *$cl(Int(\mathcal{A})) = \mathcal{A}$, [cl means closure, Int means interior]*

(*iii*) *$Int(\mathcal{A})$ is path connected and*

(*iv*) *$\lambda(\delta\mathcal{A}) = 0$ where $\delta\mathcal{A} = \mathcal{A} \cap cl(\mathcal{A}^c)$ and λ is the Lebesgue measure on $\mathbb{R}^N$.*

The relevance of the properties (*i*), (*ii*), (*iii*) and (*iv*) of Definition 2.1 is provided in [61]. Let $\mathbb{B} = \{\mathcal{A} : \mathcal{A}$ satisfies Definition 2.1$\}$. $\mathbb{B}$ is the collection of all classes in $\mathbb{R}^N$. Any $\mathcal{A} \in \mathbb{B}$ is referred to as the pattern class. •

B. Accuracy factor

It has been argued in the previous section that the boundary of a pattern class obtained from its sampled points should be extended to some extent to highlight the possible uncovered portions of the class by the sampled points. An *accuracy factor* (δ_t) based on the number of sampled points (t) is considered here for the said extension to manage the uncertainty. δ_t satisfies [230]

$$\frac{1}{t^{1/N}} < \delta_t < \frac{1}{t^{1/(N+1)}} \qquad (2.1)$$

so that as t increases, $\delta_t \to 0$ and $t\delta_t^N \to \infty$. Since δ_t decreases with the increase of t, the accuracy of the obtained boundary also increases with the increase of t. The selection of an *accuracy factor* is guided by the inequality (2.1) and its justification is analyzed below. An existing definition and a theorem have been defined in this regard.

Definition 2.2 [221] : *Let X_1, X_2, ..., X_t be independent and identically distributed (i.i.d.) random vectors from a distribution P_α which supports a set α. Let α_t^* be an estimated set on the basis of X_1, X_2, ..., X_t. Then α_t^* is said to be consistent estimate of α if*

$$E_\alpha [\mu(\alpha_t^* \triangle \alpha)] \to 0 \quad as\ t \to \infty$$

where E represents expectation, μ is the σ-finite and $\triangle$ represents symmetric difference. •

Theorem 2.1 [221] : *Let $\varepsilon_t \to 0$ and $t\varepsilon_t^2 \to \infty$ and P_α be the uniform distribution over α where $\alpha \in \mathbb{B}$ and $\alpha \subseteq \mathbb{R}^2$. Let $\alpha_t^* = \cup_{i=1}^t \{X :| X_i - X | \leq \varepsilon_t \}$. Then $E_\alpha [\lambda(\alpha_t^* \triangle \alpha)] \to 0$ as $t \to \infty$ where λ is Lebesgue measure in $\mathbb{R}^2$.* •

Corollary 2.1.1 : *Theorem 2.1 holds for any N (≥ 2) dimensional space. Let $\varepsilon_t \to 0$ and $t\varepsilon_t^N \to \infty$ and P_α be the multivariate (with N variates) uniform distribution over $\alpha \in \mathbb{B}$. Let $X_i = (x_{i_1}, x_{i_2}, \ldots, x_{i_N})'$. Let $\alpha_t^* = \cup_{i=1}^t \{ [x_{i_1} - \varepsilon_t , x_{i_1} + \varepsilon_t] \times [x_{i_2} - \varepsilon_t , x_{i_2} + \varepsilon_t] \times \ldots \times [x_{i_N} - \varepsilon_t , x_{i_N} + \varepsilon_t] \}$ where $\varepsilon_t \to 0$ and $t\varepsilon_t^N \to \infty$. Then α_t^* is a consistent estimator of α.* •

B. Accuracy factor

It has been argued in the previous section that the boundary of a pattern class obtained from its sampled points should be extended to some extent to highlight the possible uncovered portions of the class by the sampled points. An *accuracy factor* (δ_t) based on the number of sampled points (t) is considered here for the said extension to manage the uncertainty. δ_t satisfies [230]

$$\frac{1}{t^{1/N}} < \delta_t < \frac{1}{t^{1/(N+1)}} \qquad (2.1)$$

so that as t increases, $\delta_t \to 0$ and $t\delta_t^N \to \infty$. Since δ_t decreases with the increase of t, the accuracy of the obtained boundary also increases with the increase of t. The selection of an *accuracy factor* is guided by the inequality (2.1) and its justification is analyzed below. An existing definition and a theorem have been defined in this regard.

Definition 2.2 [221] : *Let X_1, X_2, ..., X_t be independent and identically distributed (i.i.d.) random vectors from a distribution P_α which supports a set α. Let α_t^* be an estimated set on the basis of X_1, X_2, ..., X_t. Then α_t^* is said to be consistent estimate of α if*

$$E_\alpha [\mu(\alpha_t^* \triangle \alpha)] \to 0 \quad as\ t \to \infty$$

where E represents expectation, μ is the σ-finite and $\triangle$ represents symmetric difference. •

Theorem 2.1 [221] : *Let $\varepsilon_t \to 0$ and $t\varepsilon_t^2 \to \infty$ and P_α be the uniform distribution over α where $\alpha \in \mathbb{B}$ and $\alpha \subseteq \mathbb{R}^2$. Let $\alpha_t^* = \cup_{i=1}^t \{X :\mid X_i - X \mid \leq \varepsilon_t\}$. Then $E_\alpha [\lambda(\alpha_t^* \triangle \alpha)] \to 0$ as $t \to \infty$ where λ is Lebesgue measure in $\mathbb{R}^2$.* •

Corollary 2.1.1 : *Theorem 2.1 holds for any N (≥ 2) dimensional space. Let $\varepsilon_t \to 0$ and $t\varepsilon_t^N \to \infty$ and P_α be the multivariate (with N variates) uniform distribution over $\alpha \in \mathbb{B}$. Let $X_i = (x_{i_1}, x_{i_2}, \ldots, x_{i_N})'$. Let $\alpha_t^* = \cup_{i=1}^t \{ [x_{i_1} - \varepsilon_t , x_{i_1} + \varepsilon_t] \times [x_{i_2} - \varepsilon_t , x_{i_2} + \varepsilon_t] \times \ldots \times [x_{i_N} - \varepsilon_t , x_{i_N} + \varepsilon_t] \}$ where $\varepsilon_t \to 0$ and $t\varepsilon_t^N \to \infty$. Then α_t^* is a consistent estimator of α.* •

Corollary 2.1.2 : *Let* $\alpha_t^* = \cup_{i=1}^{t} \{ [x_{i_1} - \varepsilon_{1_t} , x_{i_1} + \varepsilon_{1_t}] \times [x_{i_2} - \varepsilon_{2_t} , x_{i_2} + \varepsilon_{2_t}] \times \ldots \times [x_{i_N} - \varepsilon_{N_t} , x_{i_N} + \varepsilon_{N_t}] \}$ *where* $\varepsilon_{j_t} \to 0$ *and* $t\varepsilon_{j_t}^N \to \infty$ *for* $j = 1, 2, \ldots, N$. *Then also* α_t^* *is a consistent estimator to* α. •

Corollary 2.1.3 : *Let* $a_{j_t} \to a_j$ *where* $a_j > 0$ *for* $j = 1, 2, \ldots, N$. *Let* $\varepsilon_{j_t} = a_{j_t}\delta_t$ *where* $\delta_t \to 0$ *and* $t\delta_t^N \to \infty$. *Then* α_t^*, *as defined in corollary 2.1.2, is a consistent estimator to* α. •

Notes :

(i) The above theorem and corollaries basically take the union of certain neighborhoods for every point as an estimate of the original set α.

(ii) In Theorem 2.1, the distributions were assumed to be (multivariate) uniform. But it can be shown for any continuous distribution on a compact path connected α, similar result holds [61]. Observe that a pattern class can always be assumed to be a path connected and compact set. Hence corollary 2.1.3 can be used for any continuous distribution.

(iii) Corollary 2.1.3 has been used in the present paper where a_{j_t}'s are taken to be the range of the individual (jth) feature values in the set of sampled points and δ_t satisfies the inequality (2.1). •

For the pattern classes in $I\!R^2$ (i.e., $N = 2$), δ_t satisfies [218]

$$\frac{1}{t^{0.49}} \leq \delta_t \leq \frac{1}{t^{0.34}} \tag{2.2}$$

Fig. 2.1 shows the allowed range of δ_t [inequality (2.2)] for different values of t for the pattern classes in $I\!R^2$. •

C. Coverage factors

Each individual sampled point represents a covered area of the pattern class in the feature space. The proposed method intends to find the multivalued shape of a class from its sampled points or training samples. Hence, in order to find the possible uncovered portions of the class by the sampled points, the boundaries are extended to some extent. The extended portions should have

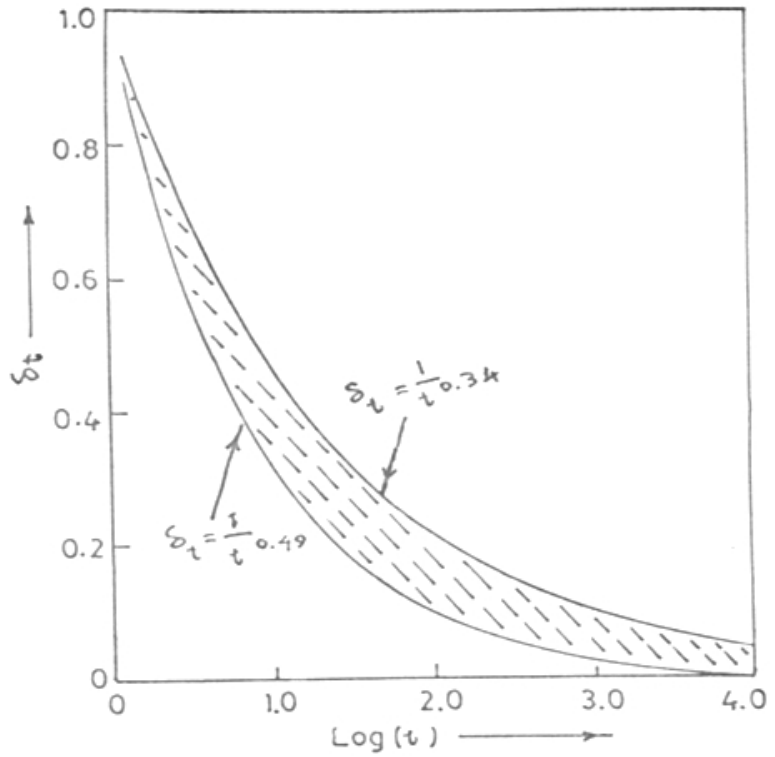


Figure 2.1 : Allowed range of δ_t for different sample sizes (t) in $\mathbb{R}^2$.

less possibility to be in the pattern class than the portions explicitly highlighted by the sampled points. To decide the extent of the covered/extended portions, the factors, named as *coverage factors*, are defined below across different feature axes.

Let $X_1, X_2, \ldots, X_t$ be the training samples where $X_i = (x_{i_1}, x_{i_2}, \ldots, x_{i_N})'$ and x_{i_j} denotes the jth feature value of the ith sample. Let MAX_j and MIN_j denote the maximum and minimum sample feature values respectively in the sample set corresponding to the jth $(j = 1, 2, \ldots, N)$ feature i.e.,

$$MAX_j = \max_{i=1,2,\ldots,t} \{x_{i_j}\} \quad \text{and} \quad MIN_j = \min_{i=1,2,\ldots,t} \{x_{i_j}\}$$

The *coverage factor* for the set of sampled points corresponding to the jth feature, denoted by ε_j $(j = 1, 2, \ldots, N)$, is defined as

$$\varepsilon_j = (MAX_j - MIN_j) \times \delta_t \tag{2.3}$$

where δ_t is the accuracy factor. When the number (t) of sampled points increases, the accuracy factor (δ_t) decreases and correspondingly the values

of the coverage factors (ϵ_j's) also decrease. Thus, the extended portions also decrease and correspondingly the accuracy of the boundary increases. ♣

D. Hole in a pattern class

A path connected and compact set is referred to here as a pattern class. If it happens that within the range of the pattern class or set, some portions do not belong to the class, then the portions are referred to as the holes. The intuitive idea behind holes of a pattern class can be put mathematically by the following definition.

Definition 2.3 : *A pattern class A is said to have k holes [231] if*

$$A^c = B \cup C_1 \cup C_2 \cup \ldots \cup C_k \qquad \textit{such that}$$

(*i*) B *and* C_i *are path connected sets for* $i = 1, 2, ..., k$,

(*ii*) B *is unbounded and* $C_1, C_2, \ldots, C_k$ *are bounded,*

(*iii*) $B \cup C_{i_1} \cup C_{i_2} \cup \ldots \cup C_{i_r}$ *is a disconnected set for* $1 \leq i_1 \leq i_2 \leq \ldots \leq i_r \leq k$, *where* $1 \leq r \leq k$ *and*

(*iv*) $C_{i_1} \cup C_{i_2} \cup \ldots \cup C_{i_r}$ *is a disconnected set* $1 \leq i_1 < i_2 < \ldots < i_r \leq k$, *where* $2 \leq r \leq k$.

Then $C_1, C_2, \ldots, C_k$, in the Definition 2.3, are said to be the holes of A. The hole detection procedure is described in section 2.3.1. ♣

2.2.2 Block diagram

The block diagram of the proposed procedure for determining the multi-valued class boundary from a set of sampled points is shown in Fig. 2.2. It consists of two parts, namely the *decomposition* and the *fuzzy processor*. The decomposition section deals with the decomposition of the sample set into some groups of nearly parallelepiped shape. The fuzzy processor determines

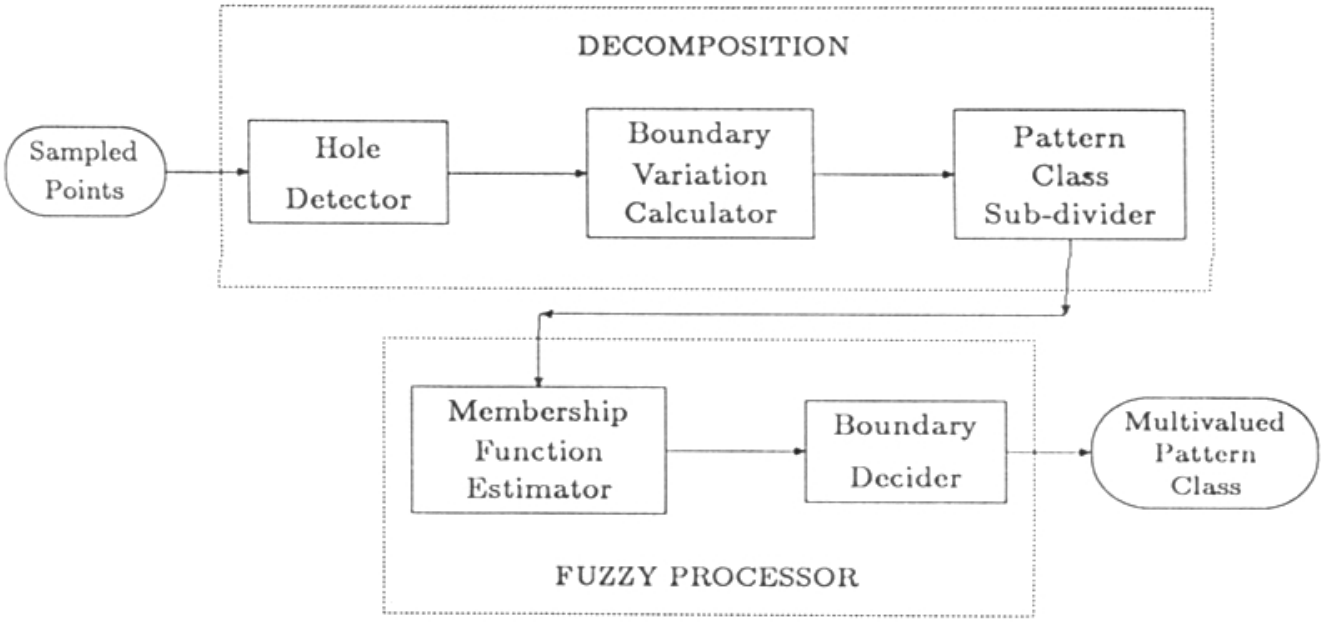


Figure 2.2 : Block diagram.

each of the subclasses corresponding to the sample groups separately and all these subclasses are combined to compute the multivalued shape of the pattern class.

The decomposition section consists of three blocks as shown in Fig. 2.2. The *hole detector* block decomposes the training sample set with holes into groups to find the hole information. The *boundary variation calculator* block finds the values of the boundary variation factors corresponding to all possible boundary directions. These boundary variation values are analyzed in the *pattern class sub-divider* block to decompose (if necessary) the sample set into groups.

The theory of fuzzy sets has been used in the fuzzy processor section to extend the boundary of the sample set and also to relate every point in the whole feature space to its possibility to be in the pattern class. The *membership function estimator* block decides about the compatibility/membership functions to represent each of the subclasses corresponding to the sample

groups. The *boundary decider* block determines each of the subclasses separately, puts them together and finds the multivalued shape of a pattern class. ♣

One of the major underlying features of the procedure is that any pattern class can be represented as a collection of nearly parallelepiped shaped subclasses. The concept of decomposition of the sample set is found very useful in developing a multivalued recognition system (described chapter 4). Moreover, all the parameters like *accuracy factor*, *coverage factors* etc. required in the algorithm are determined automatically from the sampled points.

The following two sections (i.e., sections 2.3 and 2.4) describe in detail the operations of different blocks of Fig. 2.2 for pattern classes in $I\!R^2$. ♠

2.3 Decomposition

The decomposition section consists of three blocks, namely *hole detector*, *boundary variation calculator* and *pattern class sub-divider*. It takes the training samples of a pattern class as an input and decomposes the training sample set into groups if the sample set is found to be not nearly rectangular in shape. To obtain this decomposition, a procedure based on some overlapping windows is adopted here. Before describing the operations of various blocks of decomposition, the approach to generate the windows is furnished below.

Generation of windows :

The two features under consideration are referred to as the first (F_1) and second (F_2) feature axes respectively. In the proposed approach, one of the axes is considered as the base axis and then the other axis is considered as the height axis. Consequently the coverage factors corresponding to the base and height features are referred to as the *base coverage factor* (ε_b) and *height threshold factor* (ε_h) respectively.

Now the sampled points are first of all arranged in ascending order according to the base feature values. The first window starts with the first

sample of the ordered sample set and it includes all those samples one after another in ascending order until its base coverage length exceeds ε_b. Assume that the generated first window ends with the kth sample of the ordered sample set. Then the second window will end with the $(k+1)$th sample and to find the starting point of this window, it proceeds backward from kth sample until its base coverage length exceeds ε_b. Similarly other windows are constructed by including one new sample at the end and excluding some samples from the beginning of the previous window such that the base coverage length would atleast be ε_b. The last window ends with the last ordered sample i.e., sample with the highest base value. Thus, some overlapping windows of the sample points are generated utilizing the sample base values and the base coverage factor (ε_b).

The maximum and the minimum height values are found from each of the windows and these are taken to be the upper (u) and lower (l) boundary values respectively of that window. The combination of the upper boundary values highlights the upper boundary of the training sample set and the combination of the lower boundary values provides the lower boundary of the training sample set.

The way in which the boundary of a training sample set is obtained using the aforementioned procedure for generating windows, is explained below considering a typical pattern class [Fig. 2.3(a)]. A hypothetical training sample set is shown in Fig. 2.3(a), where the locations of samples in the feature space are shown by cross $(\times)$ marks. Initially, F_1 is considered as the base and few windows are generated. In such windows, F_2 is considered as the height feature. A typical window is shown by dotted lines in Fig. 2.3(b) where the samples in the window are shown by tick $(\surd)$ marks. The boundary values corresponding to the lower and upper boundary directions (referred to as 2_l and 2_u respectively) are also marked for the window. Based on these boundary values, the rough boundaries in the coded directions 2_l and 2_u are drawn in Fig. 2.3(c). To find the boundaries in the lower and upper directions of F_1 (i.e., directions 1_l and 1_u), some windows are first of all generated considering F_2 as the base feature. A typical window with its sample points and boundary

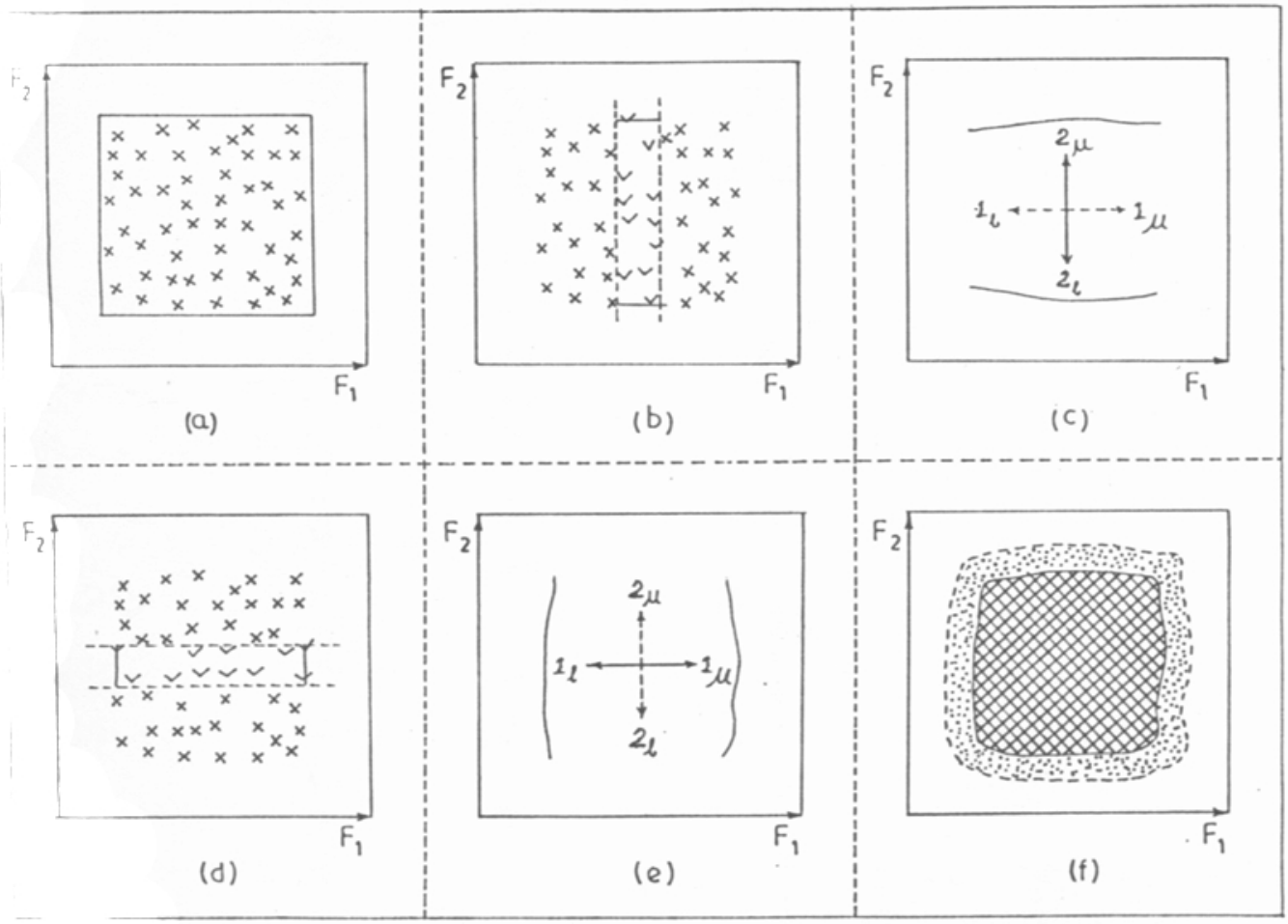


Figure 2.3 : Illustrating the concept of boundary in $\mathbb{R}^2$.

values is shown in Fig. 2.3(d). Fig. 2.3(e) shows the rough boundaries in the coded directions 1_l and 1_u.

Combining the boundaries in figures 2.3(c) and 2.3(e), the complete boundary of a pattern class is obtained [Fig. 2.3(f)]. To incorporate the possible uncovered portions of the pattern class by the training set [Fig. 2.3(a)], these boundaries are extended to some extent (depending on the coverage factors). To visualize the said extension, the extended boundary of the pattern class is conceptually drawn in Fig. 2.3(f). The extended portion should have lower possibility to be in the class than the portions explicitly highlighted by the sample points. The extended regions decrease with the increase of the sample size (t). •

2.3.1 Hole detector

The concepts about holes of a pattern class are provided in section 2.2.1*D*. The adopted procedure to find holes from a set of sampled points is discussed below.

Procedure : The procedure considers F_1 axis as the base feature and the F_2 axis as the height. Correspondingly ε_1 and ε_2 are considered as the *base coverage factor* (ε_b) and *height threshold factor* (ε_h) respectively. Windows are now generated using the aforementioned approach. The sample points in each window are then arranged in ascending order according to the height (F_2) sample values. If the difference of height values of any two consecutive samples within a window exceeds ε_h, then a hole is assumed to be present between the said sample pair. Let h' and h'' be the height feature values corresponding to two such sample points. To illustrate this finding, a pattern class with a hole is shown in Fig. 2.4 in which P_1 and P_2 are the sample points which satisfy the above condition and the corresponding window is shown using dotted lines.

To detect the hole, the sample set is decomposed into two groups according to whether the height values (i.e., F_2 values) are less than $(h' + h'')/2$ or not. The decomposition leads to finding two groups where none of them possess a hole. In Fig. 2.4, the line with dashes indicates the split.

The aforementioned routine is repeated until every sample group is found to be not containing any hole. When the subclasses corresponding to the sample groups are combined in the boundary decider block [section 2.4], the holes will be excluded from the final shape of the pattern class. It is to be realized that this procedure detects not only the holes elongated across F_2 feature axis but also the holes elongated across F_1 feature axis. Note that the proposed procedure can not detect the holes with sizes less than ε_1 and/or ε_2 corresponding to the feature axes F_1 and F_2 respectively. Hence the detectable minimum hole size depends on coverage factors ε_1 and ε_2, which in turn depends on the accuracy factor (δ_t) and finally depends on the sample

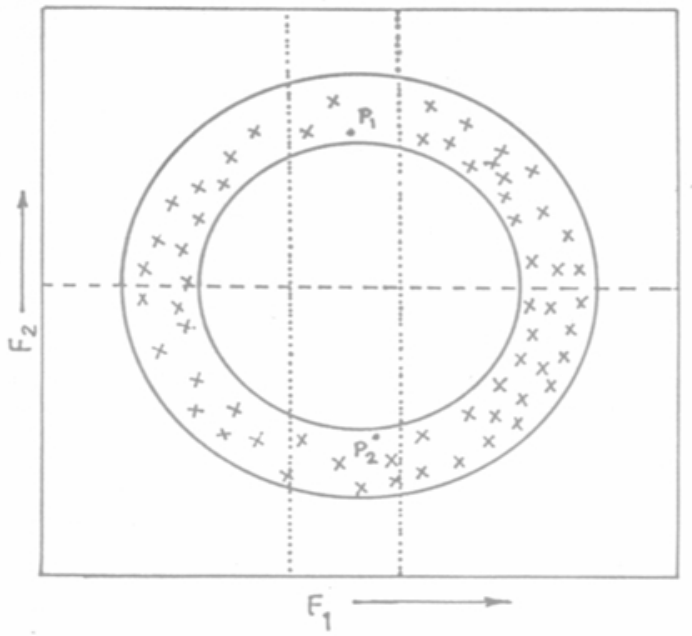


Figure 2.4 : A typical pattern class with a hole.

size (t).

It is also to be mentioned here that the procedure decomposes a training sample set not only for a hole, but also for some particular type of concave boundaries. For example, sample sets having shapes like '$\subset$', '$\pm$' or '$\angle$' may also be decomposed although these do not possess any hole. But this does not create any problem in the overall shape determining procedure, since such sample sets anyway had to be decomposed in the pattern class sub-divider block. ♣

2.3.2 Boundary variation calculator

This block tries to detect the geometric structure of the pattern class from a set of sampled points. The boundaries in four perpendicular directions [Fig. 2.3(b)] are considered here to find the boundary variation values. Our approach to generate the windows from a sample set and consequently to find

the boundary values are described earlier. It may be recalled that the maximum and the minimum height values of each window are considered as the boundary values in the upper and lower boundary directions respectively corresponding to the height feature axis. The approach to calculate the boundary variations is discussed below.

Calculation of boundary variations : To describe the approach, let us consider the boundary variation values in a particular direction, say d ($d \in \{1_l, 1_u, 2_l, 2_u\}$. The procedure to obtain the boundary values has already been discussed previously. Henceforth, it is assumed that there are w windows and their boundary values are H_i $(i = 1, 2, \ldots, w)$. A *boundary variation factor*, denoted by V_d in direction d is defined as

$$V_d = \left[\sum_{i=2}^{w} (H_i - H_{i-1})^2\right] / \varepsilon_h^2 \qquad (2.4)$$

where ε_h is the *height threshold factor* for the considered direction d. Here the division factor ε_h^2 is used to make the variation factor V_d unitless.

Now let MAX_H and MIN_H be the maximum and minimum of the boundary values H_i's $(i = 1, 2, \ldots, w)$ respectively. If the difference of MAX_H and MIN_H does not exceed ε_h (i.e., if $(MAX_H - MIN_H) \leq \varepsilon_h$) then the sample set is assumed to be nearly rectangular in shape. In such a case, the *variation factor* V_d is assumed to be zero i.e., make $V_d = 0$. Otherwise the sample set is considered as decomposable in the direction d.

Initially, F_1 axis is considered as the base to generate the windows and F_2 axis is considered as the height to find the boundary variation factors V_{2_l} and V_{2_u} for the coded directions 2_l and 2_u respectively. Similarly, by reversing the roles of F_1 and F_2 axes above, the boundary variation factors V_{1_l} and V_{1_u} for the coded directions 1_l and 1_u are calculated. Thus, the boundary variation factors for the considered four boundary directions are obtained in this block.

2.3.3 Pattern class sub-divider

This block analyzes the *boundary variation factors* to determine whether the training sample set is to be decomposed or not. If the sample set is to be decomposed then this block decomposes it into few groups. Before actually dividing the sample set, the decision about the direction of decomposition is to be made. To decide this, it finds the direction in which the value of the variation factor is maximum. That is, the direction $D \in \{1_l, 1_u, 2_l, 2_u\}$ is found where

$$V_D = \max \{V_{1_l}, V_{1_u}, V_{2_l}, V_{2_u}\}.$$

If $V_D = 0$, then the sample set is assumed to be nearly rectangular in shape and it is not further decomposable.

Otherwise i.e., if $V_D > 0$, then it is assumed that the sample set is not nearly rectangular in shape and it is to be decomposed into groups. Now from the direction of decomposition (i.e., D) the windows with their base and boundary values, and the corresponding *height threshold factor* ε_h are recalled. The samples are then arranged in ascending order according to the base values.

For making a cluster of windows, the maximum boundary value is found. The starting window for the cluster is taken as that window which have the maximum boundary value. The position of the starting window is noted. The following windows from the starting windows are assigned one after another in the cluster until the differences between the boundary values of the current window and the starting window exceed the *height threshold factor* (ε_h). Similarly, the preceding windows are also put in the window cluster. The samples lying in the window cluster are assigned to the first sample group.

The aforesaid routine is repeated on the remaining windows until all the windows are exhausted. This leads to the formation of window clusters. Every window cluster results in a group of sample points. Thus, the given training sample set is decomposed into a few group of sample points.

The decomposition procedure is applied on the sample groups repeatedly

until all the groups are found to be nearly rectangular in shape. It is to be observed that the larger the number of groups, the greater will be the accuracy of the shape of the multivalued pattern class obtained. ♠

2.4 Fuzzy Processor

The previous section dealt with the decomposition of a training set into few sample groups of nearly rectangular shape. The present section is devoted to obtaining the subclasses corresponding to these groups and combining them to get the estimated multivalued shape of the pattern class. This section consists of two blocks, namely the *membership function estimator* and the *boundary decider*.

The concept of membership functions in the light of fuzzy set theory is brought here to represent each of the subclasses corresponding to the groups separately. Membership functions have also been used to relate every point in the entire feature space to its possibility to be in the pattern class. The membership function estimator block finds the appropriate membership functions to represent each of the subclasses corresponding to the sample groups. The boundary decider block determines each of the subclasses separately, puts them together and obtains the multivalued shape of the pattern class.

2.4.1 Membership function estimator

For any feature point, the possibility of being a member of a class is maximum if it lies in the centre of the class. As its distances from the the centre increases, the membership value decreases and ultimately go to zero. Any function having this property may be considered as the representative membership function for the (sub) pattern class corresponding to a sample group. As the π function is well established to dictate this property [18,67], it is considered here as the representative membership function.

Thus, the subclasses corresponding to the sample groups are character-

ized by different π functions across different axes of the form $\pi(x;\ \alpha_{k_j},\ \beta_{l_{k_j}},\ \beta_{u_{k_j}},\ \gamma_{l_{k_j}},\ \gamma_{u_{k_j}})$ where k indicates the group number ($k = 1, 2, \ldots, \eta$, η denotes the number of groups); j indicates the axis number ($j = 1, 2$); α_{k_j} is the peak value where the membership value is 1.0; $\beta_{l_{k_j}}$ and $\beta_{u_{k_j}}$ are the lower and upper most ambiguous points where the membership values are 0.5, and $\gamma_{l_{k_j}}$ and $\gamma_{u_{k_j}}$ are the lower and upper end points beyond which the membership values are zero. The functional form of such a π function [218] is stated below:

$$\pi(x; \alpha, \beta_l, \beta_u, \gamma_l, \gamma_u) = \begin{cases} S\,(x; \gamma_l, \beta_l, \alpha) & \text{if } x \leq \alpha \\ 1 - S\,(x; \alpha, \beta_u, \gamma_u) & \text{if } x > \alpha \end{cases} \tag{2.5}$$

where

$$S(x; a, b, c) = \begin{cases} 0 & \text{if } x \leq a \\ \frac{1}{2}\left(\frac{x-a}{b-a}\right)^2 & \text{if } a < x \leq b \\ 1 - \frac{1}{2}\left(\frac{x-c}{b-c}\right)^2 & \text{if } b < x \leq c \\ 1 & \text{if } x > c. \end{cases} \tag{2.6}$$

Note that unlike the standard S function [18,67], here $b \neq \frac{a+c}{2}$.

The structure of such a π function is shown in Fig. 2.5. Given a particular group of samples and a particular feature, the domain of the π function is $[\gamma_l, \gamma_u]$. It is assumed that the extended portions are $[\gamma_l, \beta_l]$ and $[\beta_u, \gamma_u]$, and the highlighted portion by the training sample set is $[\beta_l, \beta_u]$. These extended portions ultimately provide the multivalued shape of the pattern class.

Determination of membership functions : To determine the membership functions (which are taken as π functions), the parameters of them corresponding to various sample groups are to be evaluated. Here each of the sample groups is considered separately. Let MAX_{k_j} and MIN_{k_j} be the maximum and minimum of the training sample set respectively corresponding to jth ($j = 1, 2$) feature and kth ($k = 1, 2, \ldots, \eta$) sample group. Then the parameters of the π function corresponding to jth feature and kth subclass (i.e., kth sample group) are assigned as follows:

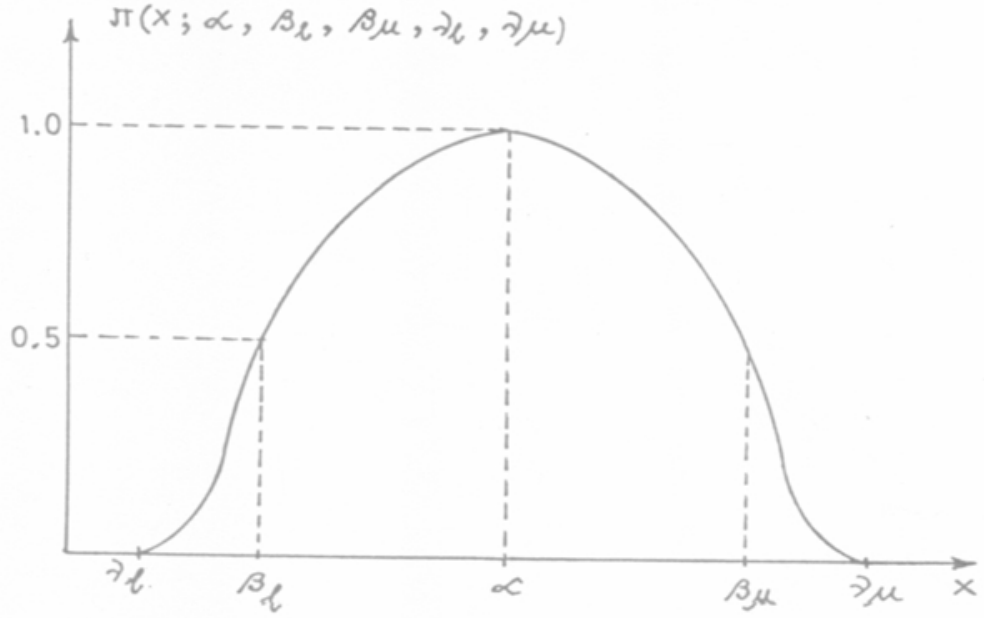


Figure 2.5 : Pie (π) function.

$$
\begin{aligned}
\alpha_{k_j} &= \frac{MAX_{k_j}+MIN_{k_j}}{2} \quad ; \\
\beta_{l_{k_j}} &= MIN_{k_j} \quad ; \quad \beta_{u_{k_j}} = MAX_{k_j} \quad ; \\
\gamma_{l_{k_j}} &= MIN_{k_j} - \varepsilon_j \quad ; \quad \gamma_{u_{k_j}} = MAX_{k_j} + \varepsilon_j \quad ; \\
& \qquad\qquad j = 1, 2 \; ; \quad k = 1, 2, \ldots, \eta \quad ;
\end{aligned}
\tag{2.7}
$$

where ε_j is the coverage factor for the jth feature [Eq. (2.3)]. ♣

2.4.2 Boundary decider

In the previous sub-section, membership functions corresponding to all the sample groups along each feature axis are determined. Using these functions, each of the subclasses corresponding to the sample groups is estimated and those are finally combined to obtain the estimated (multivalued) shape of the pattern class. All planar points in the feature space are labeled with their degree of possibilities to be in the class. To show the shape of a pattern class in the plane, the whole feature range is divided into small rectangles and these

are referred to as the Feature Space Cells (henceforth FSCs). The size of all the FSCs are same and these are made as small as possible such that each FSC can be distinguished in the feature space. Thus all these FSCs are labeled in terms of their possibility values to be in the pattern class. The FSCs with zero possibility value are taken to be outside the class. The method to obtain these possibility values is described below.

Procedure : Let $(x_1, x_2)'$ be a typical feature values of such a FSC. Suppose μ_{k_j} denote the membership value of the FSC corresponding to kth $(k = 1, 2, \ldots, \eta)$ subclass (i.e., kth sample group) and jth $(j = 1, 2)$ feature. μ_{k_j} is calculated from the corresponding π function i.e.,

$$\mu_{k_j} = \pi\left(x_j; \alpha_{k_j}, \beta_{l_{k_j}}, \beta_{u_{k_j}}, \gamma_{l_{k_j}}, \gamma_{u_{k_j}}\right) \tag{2.8}$$

The combined membership of the FSC corresponding to kth $(k = 1, 2, \ldots, \eta)$ subclass, say μ_k, is defined as the geometric mean of μ_{k_1} and μ_{k_2} i.e.,

$$\mu_k = (\mu_{k_1} \times \mu_{k_2})^{1/2}. \tag{2.9}$$

Now, the possibility, say θ, of the FSC to be in the estimated pattern class is defined as the maximum of the membership values of the subclasses. That is,

$$\theta = \max_{k=1,2,\ldots,\eta} \{ \mu_k \} \tag{2.10}$$

Let τ be the number of subclasses for which the combined membership values (μ_k's) of the said FSC are positive. To incorporate the effect of the neighboring subclasses with positive membership values in the estimated pattern class, the value of θ is increased to $\theta^{1/\tau}$ for $\tau > 1$. That is, when the possibility values of the FSC (μ_k's) are positive for two or more subclasses, then it indicates that the said FSC has the possibility to lie in those subclasses, which in turn increases the possibility of the FSC to be in the finally obtained pattern class. •

A method has been described earlier to find the possibility value (θ) of a FSC to be in the pattern class. Note that $0 \leq \theta \leq 1$. If the value of θ is

zero, then the FSC is considered to lie outside the pattern class. Otherwise the FSC belongs to the pattern class with the possibility θ.

To obtain the complete shape of the pattern class, the aforesaid routine is repeated for every FSC in the feature domain. Thus, all the FSCs are labeled with their possibility values to be in the pattern class, and as a result, the multivalued shape of the pattern class is obtained. ♣

It may be mentioned here that the proposed shape determining procedure is intuitive to a great extent. In the next section, the implementation and the usefulness of the procedure for $\mathbb{R}^2$ are discussed. ♠

2.5 Implementation and results

2.5.1 Artificially generated data

To show the effectiveness of the proposed system, different possible pattern classes were generated artificially and the procedure was implemented on them. The obtained boundaries were found to be quite satisfactory in all the cases. Figures 2.6(a)-(d) show four typical pattern classes in $\mathbb{R}^2$. Note that the pattern class in Fig. 2.6(c) is originally rectangular in shape, while others are not rectangular. The class shown in Fig. 2.6(d) has a hole whereas others do not have any hole. Training samples of size 50 are chosen randomly from each of the four classes. The multivalued shape of these classes are shown in figures 2.7(a)-(d) with 0.15 as the accuracy factor (δ_t).

Initially in all the cases, the feature spaces are decomposed into small rectangles (which are referred to as Feature Space Cells or FSCs) and the FSCs are labeled in terms of their possibility values (θ) to be in the estimated pattern class. Recall that $0 \leq \theta \leq 1$. The multivalued shapes are shown by various gray values in figures 2.7(a)-(d). Here the FSCs with gray value zero (absolute white) denote that the FSCs are completely outside the pattern class. To obtain the *crisp* version of an estimated multivalued pattern class,

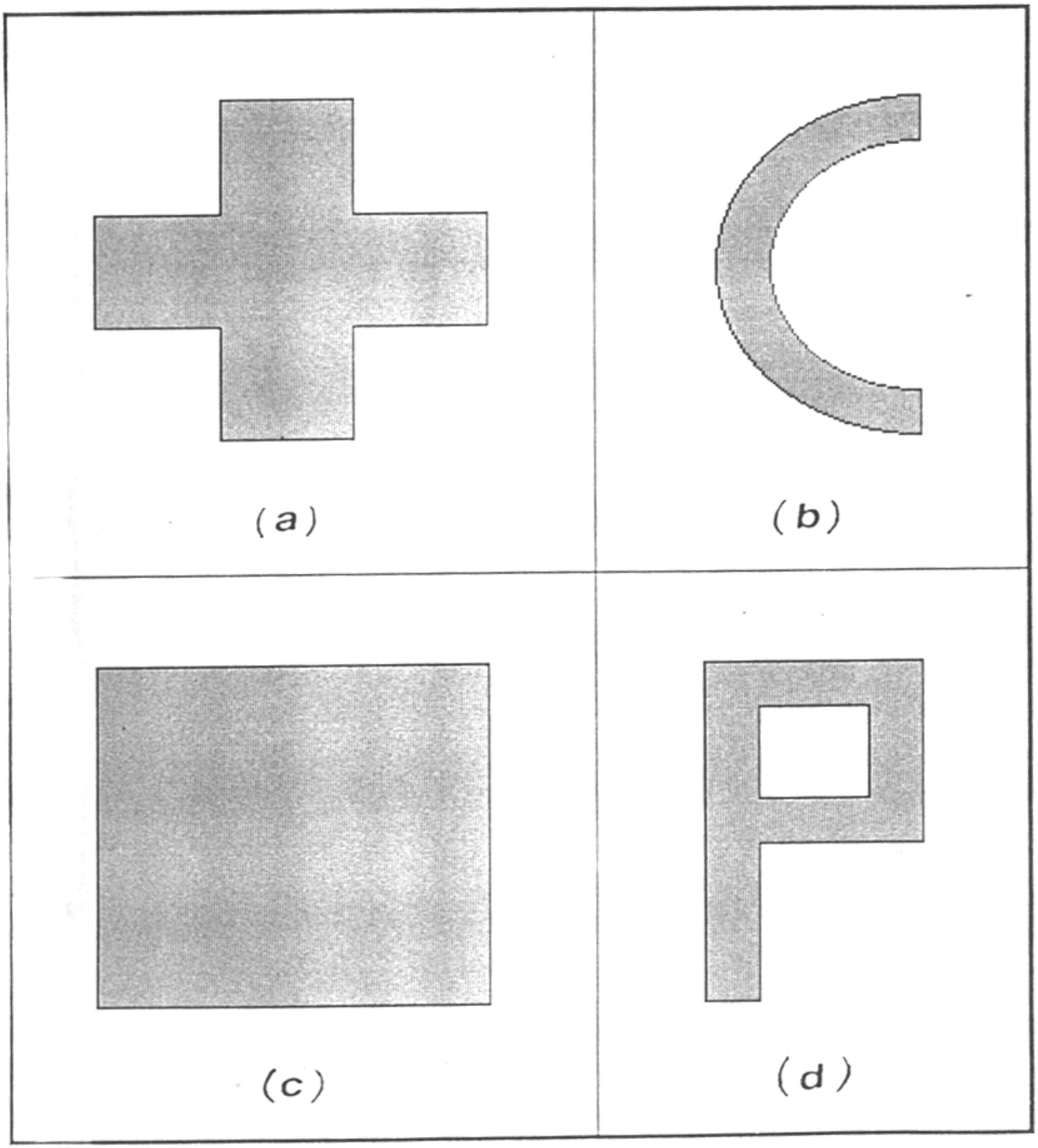


Figure 2.6 : (a)-(d) Four typical pattern classes.

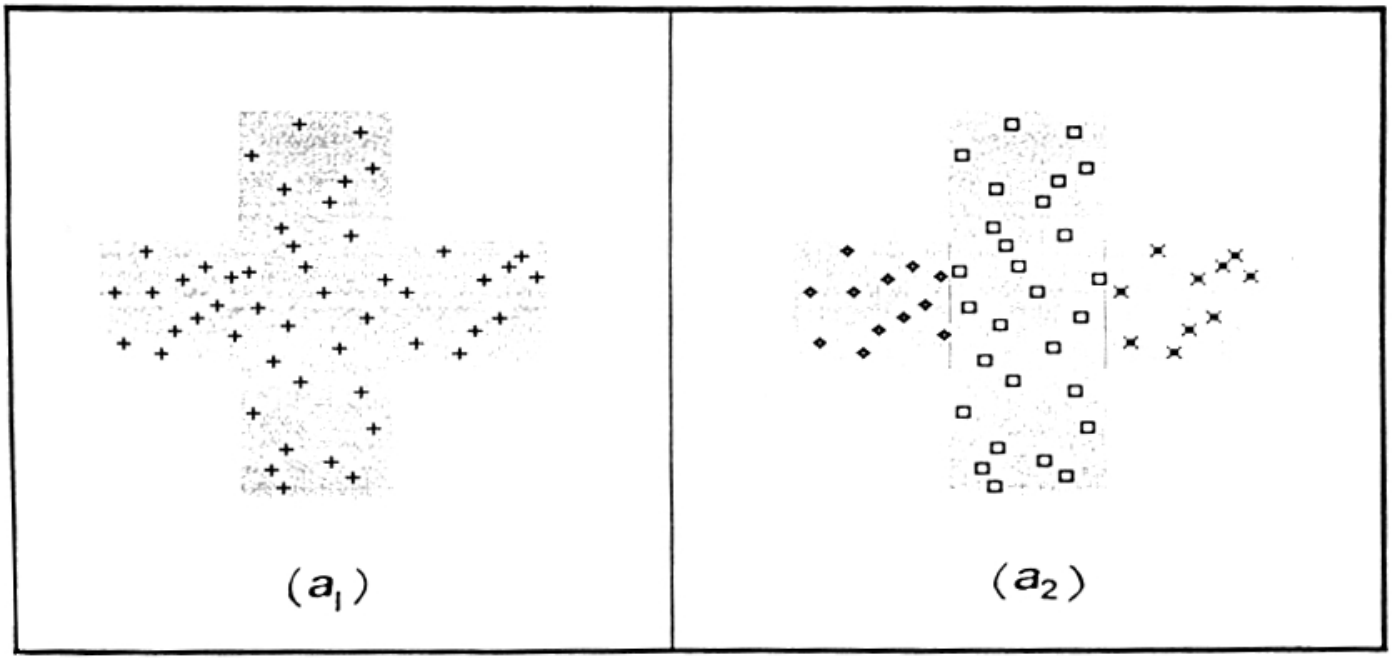


Figure 2.8 : (a_1) A set of training samples from the class in Fig. 2.6(a). (a_2) Three decomposed groups of the training set in (a_1).

the FSCs with possibility possibility values (θ) less than 0.5 may be considered to be outside the class.

Observe that the algorithm has not decomposed the training sample set of the class in Fig. 2.6(c) because it is nearly rectangular in shape. But the sample set of the classes in figures 2.6(a) and 2.6(b) are decomposed into three and fourteen sample groups respectively. Note that the class shown in Fig. 2.6(d) has a hole. The algorithm initially decomposed the training sample set into two sample groups to detect the hole and finally the sample set has been decomposed into seven sample groups. The proposed technique is explained below for the pattern class shown in Fig. 2.6(a).

Fig. 2.8(a_1) shows a training sample set corresponding to the class shown in Fig. 2.6(a). The algorithm first searches for the holes, but it could not find them. It then considered the boundaries in four perpendicular directions and found that the sample set can be decomposed in all the four directions. The boundary variation is found maximum in the coded direction 1_l. Accordingly,

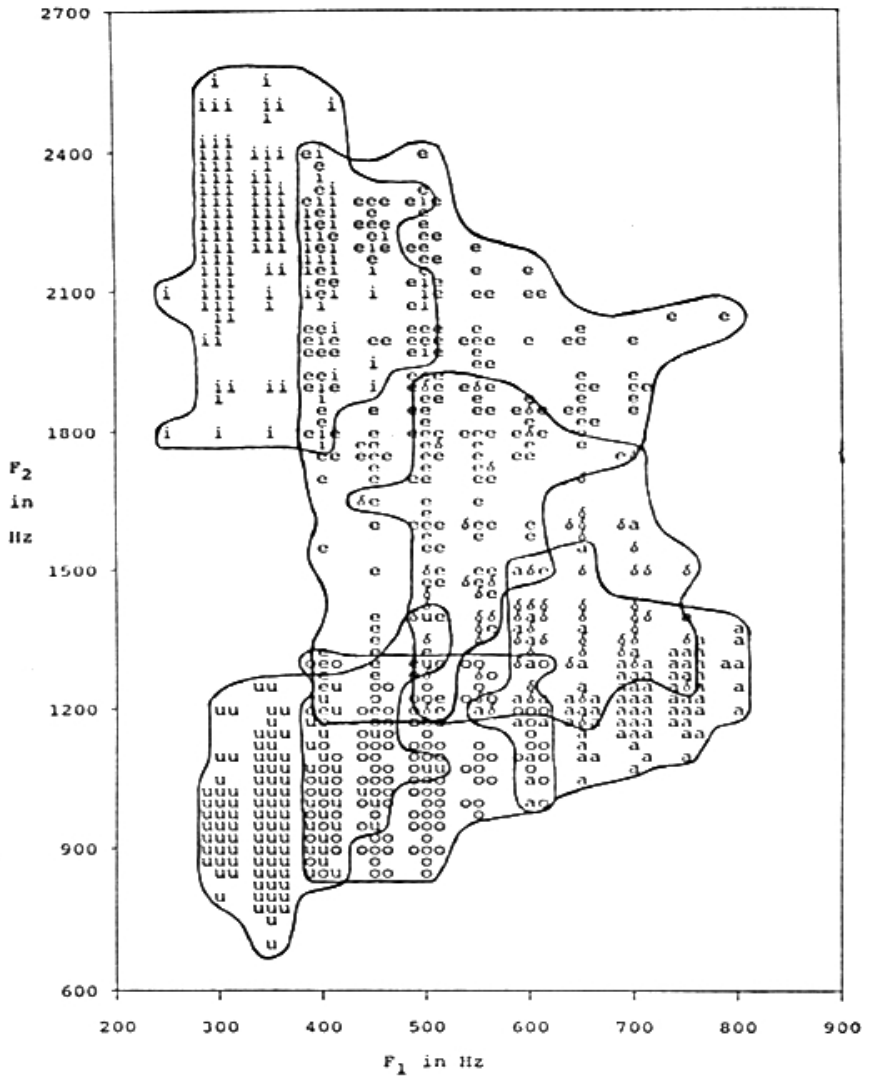


Figure 2.9 : Telugu vowel data set.

the training sample set is decomposed into three sample groups with 12, 28 and 10 samples respectively [Fig. 2.8(a_2)]. Here the sample points of different groups are shown by three different characters. Each of these sample groups are then analyzed and it could not find enough boundary variation in any of the four directions for which these groups can be decomposed again. As a result, it obtained three groups [Fig. 2.8(a_2)] of nearly rectangular shape. The three subclasses are determined based on the samples of these three groups. Finally these subclasses are combined to obtain the multivalued shape [Fig. 2.7(a)] of the pattern class in Fig. 2.6(a).

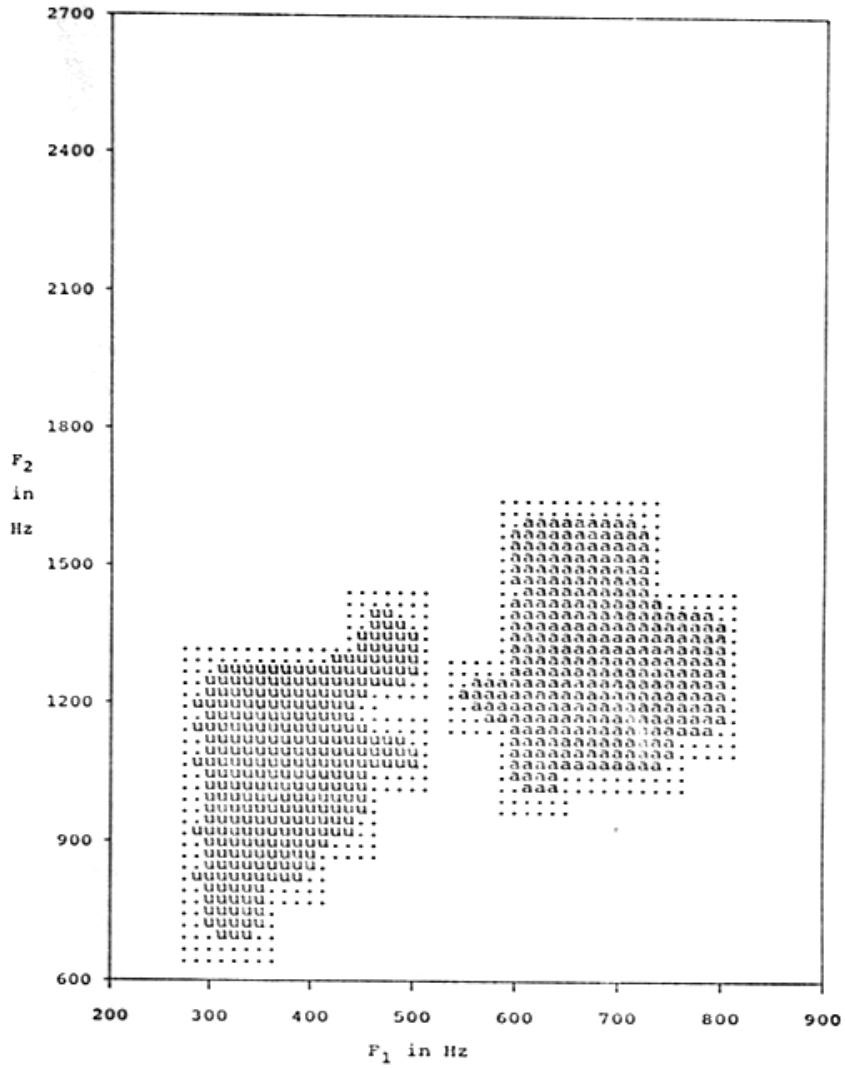


Figure 2.10(a) : Estimated classes corresponding to the vowels /a/ & /u/.

2.5.2 Speech data

To examine the practical applicability, the algorithm has been implemented on a set of Indian Telugu Vowel sounds in consonant-vowel-consonant context uttered by three speakers in the age group 30 to 35 years. Fig. 2.9 shows the typical feature space of six vowel classes /δ/, /a/, /i/, /u/, /e/ and /o/ with 72, 89, 172, 151, 207 and 180 samples respectively corresponding to the features F_1 and F_2. Here F_1 and F_2 denote the first and second formant frequencies which were obtained through spectrum analysis of the speech data. Details of feature extraction procedure can be found in [18,139]. The classes are seen to be overlapping and their boundaries are ill-defined (fuzzy).

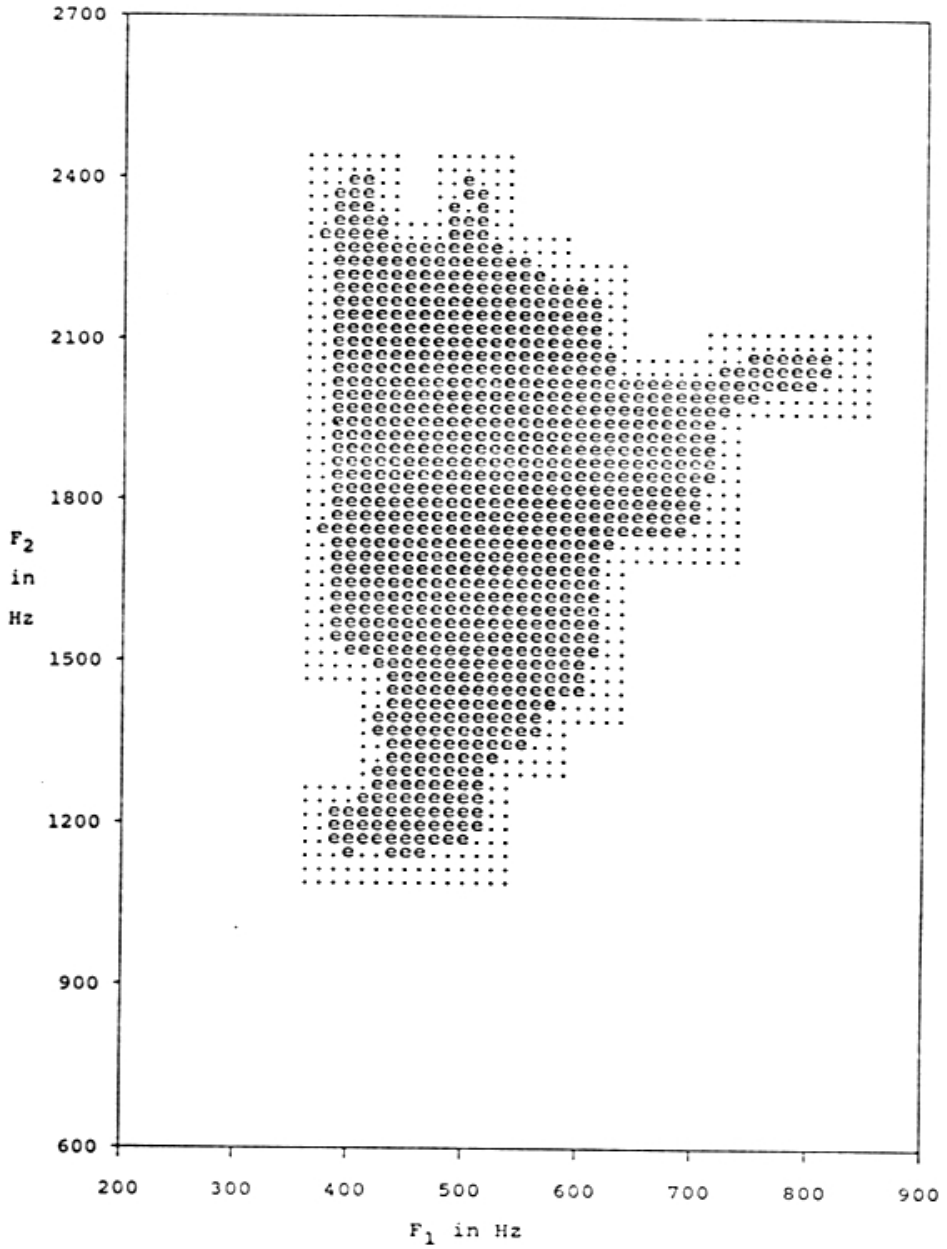


Figure 2.10(b) : Estimated class corresponding to the vowel /e/.

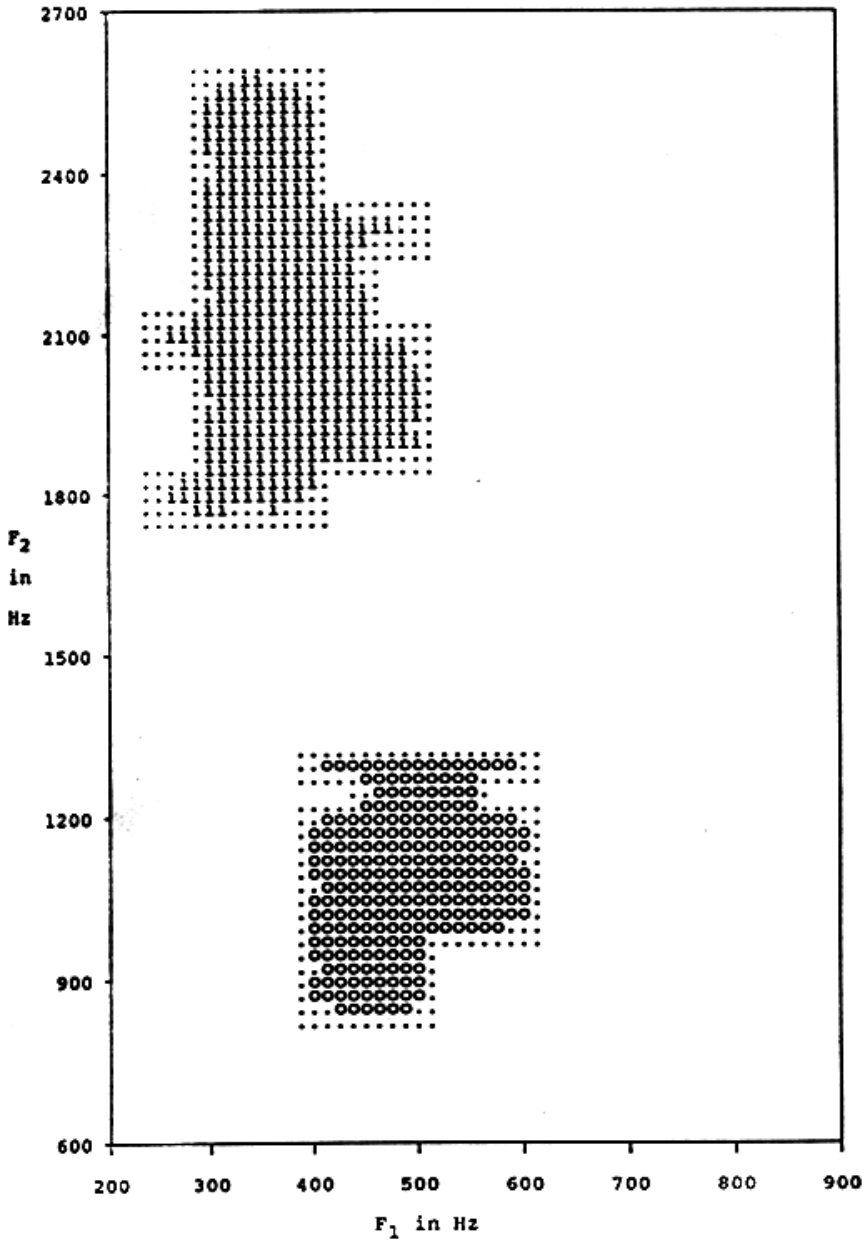


Figure 2.10(c) : Estimated classes corresponding to the vowels /i/ & /o/.

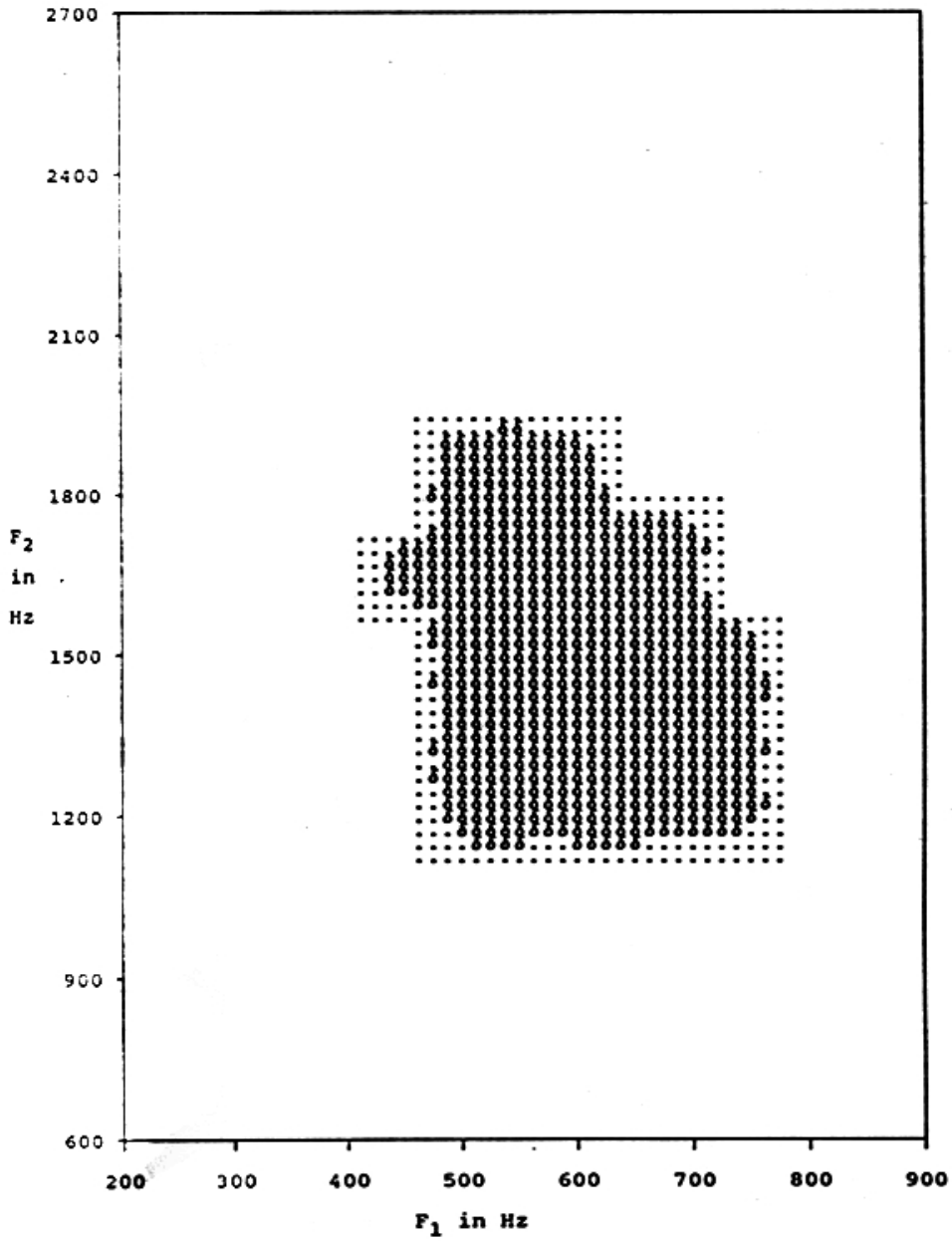


Figure 2.10(d) : **Estimated class corresponding to the vowel /δ/.**

The proposed algorithm has been applied on each of the six vowel classes separately where the total available data are assumed as the sampled points. That is, for the classes /δ/, /a/, /i/, /u/, /e/ and /o/, the number of sampled points are 72, 89, 172, 151, 207 and 180 respectively and correspondingly the accuracy factors are considered as 0.12, 0.12, 0.10, 0.10, 0.08 and 0.10 respectively. Fig. 2.10(a) shows the obtained multivalued shapes corresponding to the vowel classes /a/ and /u/; Fig. 2.10(b) shows the shape corresponding to the class /e/; Fig. 2.10(c) shows the estimated shapes corresponding to the classes /i/ and /o/, and Fig. 2.10(d) shows the shape corresponding to the class /δ/. In figures 2.10(a)-(d), the FSCs with possibility values (θ) ≥ 0.5 are represented by the corresponding vowel characters; the FSCs with θ satisfying $0 < \theta < 0.5$ are represented by dots and the FSCs with $\theta = 0$ are shown as blanks. Observe that the estimated vowel classes are good representations of their original classes. ♠

2.6 Extension to Higher Dimension

So far, we have dealt with the description and the implementation of the shape determining procedure for the pattern classes in $I\!R^2$. The extension concept of the procedure to the pattern classes in $I\!R^N$ ($N > 2$) is furnished here. For the case of $I\!R^2$ feature space, we first decomposed the training set into some sample groups of nearly rectangular shapes and finally the subclasses corresponding to the sample groups are combined to obtain the multivalued shape of the pattern class (in $I\!R^2$). The same spirit is adopted for $I\!R^N$ feature space i.e., we first decompose (based on a window approach) the training sample set into some groups of nearly N-dimensional parallelepiped shapes and these are combined to find the final shape of the pattern class (in $I\!R^N$). The two parts of the shape determining procedure i.e., Decomposition and Fuzzy processor are first of all described. The generalized decomposition concept is also elaborated for a pattern class in $I\!R^3$. The algorithm is then demonstrated on some artificially generated pattern classes in $I\!R^3$.

2.6.1 Decomposition

A pattern class is represented here by a set of sampled points in $\mathbb{R}^N$. This block tries to detect the geometric structure of the pattern class from the sample set. The N features axes under consideration are referred to as the first (F_1), second (F_2), ..., Nth (F_N) axes respectively. Initially the window generation procedure from a sample set in R^N will be described below. Later, the proposed methodology for finding the shape will be elaborated using the window generation procedure.

Generation of windows : A window based approach is adopted here to find the boundary variations of a sample set in $\mathbb{R}^N$ and the approach is same as described for $\mathbb{R}^2$. Initially, one of the feature axes is considered as the base feature and the corresponding coverage factor is referred to as the base coverage factor (ε_b). The training samples are first of all arranged in ascending order according to the base feature values. Then, depending on the base feature values and the base coverage factor ε_b, the same procedure, as stated for $\mathbb{R}^2$, is followed here to generate windows. These windows are such that base coverage length of each window would atleast be ε_b.

Here for a particular window, the base feature values of the samples are assumed to be same and the rest (N-1) features for the samples may take any value. So the generated windows are visualized here to belong in $\mathbb{R}^{N-1}$ feature space; although the original sample set belongs to $\mathbb{R}^N$. In this sense, the proposed method of generating windows always results in reducing the dimension of the sample set by one. In this context, note that, the initial training sample set may itself be considered to belong to an N-dimensional window.

Based on the aforesaid concepts, the multivalued shape determining procedure is extended to $\mathbb{R}^N$. The approach has been applied on a sample set in $\mathbb{R}^N$ in the following way. To start with, the feature F_{i_1} is considered as the base feature (axis) and correspondingly a few windows in $\mathbb{R}^{N-1}$ are formed. For each of these windows, another axis F_{i_2} is considered as the base fea-

ture and windows in R^{N-2} are formed. Repeating this process sequentially with F_{i_3}, F_{i_4}, ..., $F_{i_{N-1}}$ as the base features axes, some one-dimensional (1-D) windows are generated eventually. The values of the left out feature F_{i_N} are considered here as the height values. The maximum and the minimum height sample values are found for each window and these are taken to be the upper and the lower *boundary values* respectively for that window. The combination of the upper boundary values of all 1-D windows, generated by sequentially taking F_{i_1}, F_{i_2}, ..., $F_{i_{N-1}}$ as the base axes, highlights the upper boundary of the training sample set in the F_{i_N} feature direction. Similarly, the combination of the lower boundary values provides the lower boundary of the sample set across F_{i_N} feature axis.

A. Hole detector

The operation of this block is same as that of the one described in section 2.3.1. The procedure sequentially considers F_1, F_2, ..., F_{N-1} as the base features (and correspondingly ε_1, ε_2, ..., ε_{N-1} as the base coverage factors) to generate few 1-D windows with F_N as the height feature. Here the coverage factor ε_N across the axis F_N is referred to as the height threshold factor. The samples in each window are then arranged in ascending order according to the height (F_N) sample values. If the difference of height values of any two consecutive samples within a window exceeds ε_N, then a hole is assumed to be present between the said sample pair. Let h' and h'' be the height feature values corresponding to the above two such sample points. To detect the hole, the sample set is decomposed into two groups according to whether the height values (i.e., F_N values) are less than $(h' + h'')/2$ or not.

The above routine is repeated until every sample group is found to be not containing any hole. When the subclasses corresponding to the sample groups are combined, the holes are excluded from the final shape of the pattern class. •

B. Boundary variation calculator

To find the boundary variations of the sample set in $\mathbb{R}^N$, $2N$ perpendicular directions (coded as 1_l, 1_u, 2_l, 2_u, ..., N_l, N_u) corresponding to the lower and upper boundary directions along the N feature axes are considered. The N features are denoted here by $F_{i_1}, F_{i_2}, \ldots, F_{i_N}$ such that $i_k \in \{1, 2, \ldots, N\}$ for $k = 1, 2, \ldots, N$ and $i_k \neq i_{k'}$ for $k \neq k'$. The procedure to find the boundary variation values is described below for a training sample set in $\mathbb{R}^N$. Though the notation used here may seem to be cumbersome, it may be noted that the procedure is the natural extension of the 2-D case (described in section 2.3.2).

Initially assuming the feature F_{i_1} as the base (and correspondingly ε_{i_1} as the base coverage factor), some windows are generated from the total training set. Let the number of windows generated be denoted by q_{i_1}. As mentioned earlier, in each of these q_{i_1} windows, the F_{i_1} feature values of the samples are considered to be same and so the samples are visualized to belong in $\mathbb{R}^{N-1}$ feature space with $F_{i_2}, F_{i_3}, \ldots, F_{i_N}$ as the feature axes. Let a particular window, say j_1th one, be denoted by $W_{i_1}^{j_1}$ where $j_1 = 1, 2, \ldots, q_{i_1}$. Now assuming the feature F_{i_2} as the base (and correspondingly ε_{i_2} as the base coverage factor), the sample set of the window $W_{i_1}^{j_1}$ ($\in \mathbb{R}^{N-1}$) results in $q_{i_1(i_2)}^{j_1}$ windows in (N-2) dimensional space with $F_{i_3}, F_{i_4}, \ldots, F_{i_N}$ as the feature axes. Similarly, considering sequentially the features $F_{i_3}, F_{i_4}, \ldots, F_{i_{N-2}}$ as the bases (and correspondingly $\varepsilon_{i_3}, \varepsilon_{i_4}, \ldots, \varepsilon_{i_{N-2}}$ as the respective base coverage factors), some 2-D windows will be generated and the number of such windows is denoted by $q_{i_1(i_2(\ldots(i_{N-3}(i_{N-2}))\ldots))}^{j_1(j_2(\ldots(j_{N-3})\ldots))}$ (where $j_k = 1, 2, \ldots, q_{i_1(i_2(\ldots(i_{k-1}(i_k))\ldots))}^{j_1(j_2(\ldots(j_{k-1})\ldots))}$, for $k = 1, 2, \ldots, N$-3). These windows are visualized here to belong to $\mathbb{R}^2$ feature space with $F_{i_{N-1}}$ and F_{i_N} as the feature axes. Thus the original sample set (belonging to $\mathbb{R}^N$) after the above mentioned operations, gives rise to a collection of 2-D windows.

The feature $F_{i_{N-1}}$ is now assumed to be the base (and correspondingly $\varepsilon_{i_{N-1}}$ is assumed as the base coverage factor) so that the samples of the j_{N-2}th window $W_{i_1(i_2(\ldots(i_{N-2})\ldots))}^{j_1(j_2(\ldots(j_{N-2})\ldots))}$ results in $q_{i_1(i_2(\ldots(i_{N-2}(i_{N-1}))\ldots))}^{j_1(j_2(\ldots(j_{N-2})\ldots))}$ 1-D windows with F_{i_N}

as the feature axis. For the samples belonging to these 1-D windows, the F_{i_N} feature values are now considered as the height values. The maximum and the minimum height sample values are found in each of the 1-D windows and these are taken as the upper and the lower boundary values respectively for the respective windows.

Let $H^{i_N : d : k}_{i_1(i_2(\ldots(i_{N-2})\ldots))}$ denotes the boundary value of the kth ($k = 1, 2, \ldots, q^{j_1(j_2(\ldots(j_{N-2})\ldots))}_{i_1(i_2(\ldots(i_{N-2}(i_{N-1}))\ldots))}$) 1-D window (generated by taking $F_{i_{N-1}}$ as the base) for the feature F_{i_N} in the direction d ($d \in \{l, u\}$ where l and u stand for lower and upper boundary directions respectively). For each of the $q^{j_1(j_2(\ldots(j_{N-3})\ldots))}_{i_1(i_2(\ldots(i_{N-3}(i_{N-2}))\ldots))}$ 2-D windows, the upper and the lower *boundary variation factors*, denoted by $V^{i_N : d : j_{N-2}}_{i_1(i_2(\ldots(i_{N-1})\ldots))}$, are defined based on their respective 1-D windows as follows

$$V^{i_N : d : j_{N-2}}_{i_1(i_2(\ldots(i_{N-1})\ldots))} = \left[\sum_{k=2}^{q^{j_1(j_2(\ldots(j_{N-2})\ldots))}_{i_1(i_2(\ldots(i_{N-2}(i_{N-1}))\ldots))}} \left(H^{i_N : d : k}_{i_1(i_2(\ldots(i_{N-2})\ldots))} - H^{i_N : d : k-1}_{i_1(i_2(\ldots(i_{N-2})\ldots))} \right)^2 \right] / \varepsilon^2_{i_N} \tag{2.11}$$

$$j_{N-2} = 1, 2, \ldots, q^{j_1(j_2(\ldots(j_{N-3})\ldots))}_{i_1(i_2(\ldots(i_{N-3}(i_{N-2}))\ldots))}$$
$$d \in \{l, u\}$$

where ε_{i_N} is the *coverage factor* for the feature F_{i_N}. Here the division factor $\varepsilon^2_{i_N}$ is used to make the variation factor unitless. Note that Eq. (2.11) is the generalized version of Eq. (2.4), which was stated for $\mathbb{R}^2$.

Now, let MAX_H and MIN_H be the maximum and minimum of the boundary values respectively in a particular direction d and corresponding to a particular two dimensional window $W^{j_1(j_2(\ldots(j_{N-2})\ldots))}_{i_1(i_2(\ldots(i_{N-2})\ldots))}$. If the difference of MAX_H and MIN_H does not exceed ε_{i_N} (i.e., if $(MAX_H - MIN_H) \leq \varepsilon_{i_N}$), then the *boundary variation* for F_{i_N} in the direction d corresponding to the said 2-D window is considered to be insignificant. In such a case, the *variation factor* is assumed to be zero i.e., make $V^{i_N : d : j_{N-2}}_{i_1(i_2(\ldots(i_{N-1})\ldots))} = 0$. Otherwise the sample set is considered to be decomposable in the direction d for F_{i_N}.

Note that the order in which the features have been considered for the above generation of windows in $\mathbb{R}^2$ is $i_1, i_2, \ldots, i_{N-2}$. It may be observed

that for every such sequence, a set of aforementioned 2-D windows can be generated. Thus all possible such sequences of features are considered and correspondingly boundary variation factors for the lower and upper boundaries across the left over two features are calculated for each of the 2-D windows. All these boundary variation factors are analyzed in the pattern class sub-divider block to decide whether sample group is to be decomposed or not.

Pattern classes in $\mathbb{R}^3$:

As the notations used to describe the generalized approach seem to be cumbersome, it is now explained for the pattern class in $\mathbb{R}^3$. For a better understanding, a cubic shaped pattern class, as shown in Fig. 2.11(a) is considered. Initially, F_1 is assumed to be the base feature to generate q_1 2-D windows. A typical (say, j_1th) window $W_1^{j_1}$ ($j_1 = 1, 2, \ldots, q_1$) is distinctly marked in Fig. 2.11(a). The 2-D view of this window with some typical training samples is shown in Fig. 2.11(b) where the F_1 feature is taken to be same (as the mid value of the ranges of F_1 in the window). Then considering F_2 as the base, the sample set of the window $W_1^{j_1}$ results in $q_{1(2)}^{j_1}$ windows in F_3 feature space. Now find the boundary values of each of $q_{1(2)}^{j_1}$ 1-D windows, which are denoted by $H_{1(2)}^{3\,:\,d\,:\,k}$ ($k = 1, 2, \ldots, q_{1(2)}^{j_1}$ and $d \in \{l, u\}$). Based on these boundary values, the boundary variation factors of F_3 in direction d corresponding to their 2-D windows are calculated using Eq. (2.11) as follows

$$V_{1(2)}^{3\,:\,d\,:\,j_1} = \left[\sum_{k=1}^{q_{1(2)}^{j_1}} \left(H_{1(2)}^{3\,:\,d\,:\,k} - H_{1(2)}^{3\,:\,d\,:\,k-1} \right)^2 \right] / \varepsilon_3^2$$

$$j_1 = 1, 2, \ldots, q_1$$
$$d \in \{l, u\}.$$

Similarly, by taking F_3 as the base feature, every sample window $W_1^{j_1}$ (generated by assuming F_1 as the base on the original sample set), gives rise to $q_{1(3)}^{j_1}$ 1-D windows in F_2 feature space. Finally the boundary variation factors of F_2 in the lower and upper boundary directions corresponding to their 2-D windows are obtained. So for all the above q_1 2-D windows, the boundary variation factors $V_{1(2)}^{3\,:\,l\,:\,j_1}$, $V_{1(2)}^{3\,:\,u\,:\,j_1}$, $V_{1(3)}^{2\,:\,l\,:\,j_1}$, $V_{1(3)}^{2\,:\,u\,:\,j_1}$ ($j_1 = 1, 2, \ldots, q_1$) are calculated.

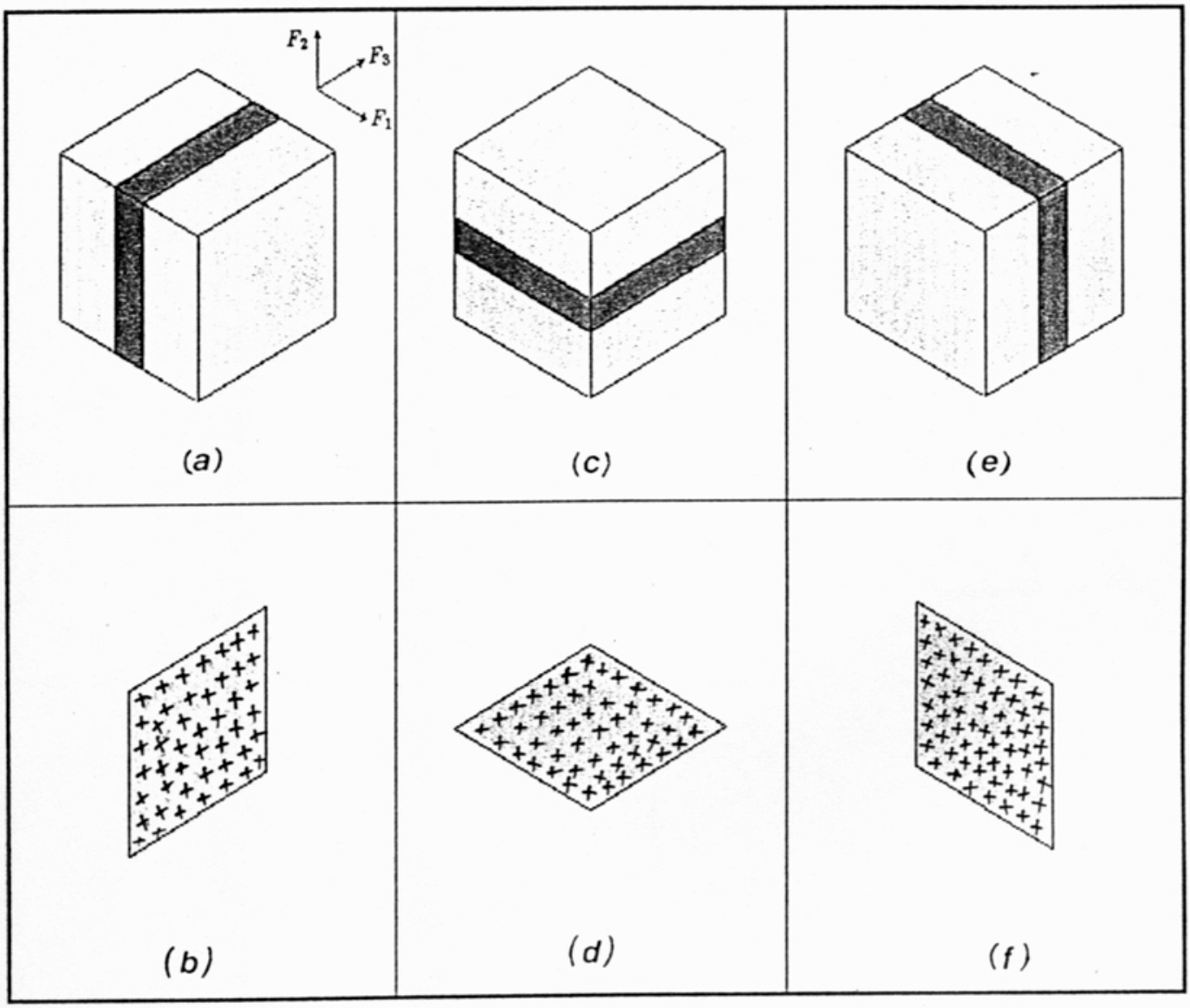


Figure 2.11 : Concept of windows for a pattern class in $\mathbb{R}^3$.

If F_2 is initially taken to be the base, the original sample set results in q_2 2-D windows with F_1 and F_3 as the feature axes. A typical of such windows is marked in Fig.2.11(c). The 2-D view of the marked window is provided in Fig. 2.11(d). Corresponding to each of the q_2 2-D windows, the boundary variation factors $V_{2(1)}^{3:l:j_2}$, $V_{2(1)}^{3:u:j_2}$, $V_{2(3)}^{1:l:j_2}$, $V_{2(3)}^{1:u:j_2}$ $(j_2 = 1, 2, \ldots, q_2)$ are obtained.

Similarly, if F_3 is first considered to be the base, the initial sample set results in q_3 2-D windows with F_1 and F_2 as the feature axes. A typical windows is marked in Fig.2.11(e) and its 2-D view is provided in Fig. 2.11(f). Corresponding to each of the q_3 2-D windows, the boundary variation factors $V_{3(1)}^{2:l:j_3}$, $V_{3(1)}^{2:u:j_3}$, $V_{3(2)}^{1:l:j_3}$, $V_{3(2)}^{1:u:j_3}$ $(j_3 = 1, 2, \ldots, q_3)$ are determined.

Thus there are in total $4(q_1 + q_2 + q_3)$ boundary variation factors corresponding to a sample set in $I\!R^3$. All these variation factors are analyzed in the next block i.e., pattern class sub-divider block. •

C. Pattern class sub-divider

This block analyzes the boundary variation factors to determine whether the training sample set is to be decomposed or not. To decide this, it finds the maximum of all the variation factors. If this value is zero, then the sample set is assumed to be nearly parallelepiped in shape and it is not further decomposable. Otherwise it is assumed that the sample set is not nearly parallelepiped in shape and it is to be decomposed into few groups. Corresponding to the 2-D window and the direction of the maximum variation factor, the 1-D windows with their base and boundary values, and the coverage factor corresponding to the height feature (referred to as height threshold factor ϵ_h) are marked. Based on these values, the decomposition is made. The samples in the particular 2-D window are then arranged in ascending order according to the base values.

For making a cluster of 1-D windows, the maximum boundary value is

found. The starting window for the cluster is taken as the 1-D window having the maximum boundary value. The position of the starting window is noted. The following 1-D windows from the starting windows are assigned one after another in the cluster until the differences between the boundary values of the current 1-D window and the starting window exceeds the height threshold factor (ϵ_h). Similarly the preceding 1-D windows are also put in the window cluster. The maximum (say, MAX_b) and the minimum (say, MIN_b) base values of the samples in the window cluster are found. Now from all the samples in the considered sample set/group, the samples with the base lying between MIN_b and MAX_b are assigned to the first sample group.

The aforesaid routine is repeated on the remaining 1-D windows until all the marked 1-D windows are exhausted. This leads to the formation of window clusters. Every window cluster results in a group of sample points. Thus, the given training sample set is decomposed into a few group of sample points.

The decomposition procedure is applied on each of the sample groups repeatedly until all the groups are found to be nearly parallelepiped in shape. ♣

2.6.2 Fuzzy processor

A training sample set is decomposed in the previous section into few groups of nearly parallelepiped shape. Here the subclasses corresponding to these groups are determined separately and finally these are combined to obtain the multivalued shape of the pattern class.

A. Membership function estimator

The subclasses corresponding to the groups are characterized by different π functions across different axes of the form $\pi(x; \alpha_{k_j}, \beta_{l_{k_j}}, \beta_{u_{k_j}}, \gamma_{l_{k_j}}, \gamma_{u_{k_j}})$ [Eq. (2.5) and Fig. 2.5] where k indicates the group number ($k = 1, 2, \ldots, \eta$, η denotes the number of groups); j indicates the axis number ($j = 1, 2, \ldots, N$).

The meaning of the parameter of the π functions are same as mentioned in section 2.4.1 in relation to the pattern classes in $I\!R^2$. Using Eq. (2.7), the parameters are determined for each subclasses corresponding to sample groups and each feature axes. •

B. Boundary decider

To show the shape of a pattern class in the feature space, the entire feature range is divided into small units of parallelepiped shape and these small units are referred to as the Feature Space Cell or FSC. The size of all the FSCs are same and these are made as small as possible such that each FSC can be distinguished in the feature space. All these FSCs are labeled in terms of their possibility values (θ) to be in the pattern class.

Let $(x_1, x_2, \ldots, x_N)'$ be a typical feature value of such a FSC. The membership value (μ_{k_j}) of the FSC corresponding to kth ($k = 1, 2, \ldots, \eta$) subclass (i.e., kth sample group) and jth ($j = 1, 2, \ldots, N$) feature is calculated from the corresponding π function using Eq. (2.8). The combined membership (μ_k) of the FSC corresponding to kth ($k = 1, 2, \ldots, \eta$) subclass is defined as the geometric mean of μ_{k_j}'s i.e.,

$$\mu_k = (\mu_{k_1} \times \mu_{k_2} \times \ldots \times \mu_{k_N})^{1/N}.$$

The possibility value (θ) of the FSC is then determined using Eq. (2.10). Depending on the membership values of the FSC in the neighboring subclasses, the value of θ is similarly modified as done for $I\!R^2$ feature space. Thus, by labeling all the FSCs of the feature domain with their possibility values, the multivalued shape of a pattern class in $I\!R^N$ is obtained. The implementation of the extension concepts is provided below. ♣

2.6.3 Implementation and results

The effectiveness of the proposed procedure has already been demonstrated in section 2.5 with some artificially generated data sets and also with the real

life speech data which were in $\mathbb{R}^2$. The behaviour of the generalized procedure is demonstrated here with three artificially generated pattern classes in $\mathbb{R}^3$. The classes are shown in figures 2.12(a), 2.13(a) and 2.14(a). For all 3-D classes, the isometric views are provided. Note that Fig. 2.14(a) shows only the lower bisecting (with respect to F_2 feature) portion of a pattern class which has the same external view with the class in Fig. 2.12(a). Actually this class is having a hole and externally the hole can not be shown. For demonstrating the hole detecting capabilities of the proposed algorithm, only the lower bisecting portion of a pattern class is displayed in Fig. 2.14(a).

Training samples of size 150 are chosen randomly from each of the three classes and correspondingly the accuracy factor (δ_t) is assumed to be 0.20. Note that the extracted classes are multivalued. Hence, in order to demonstrate the concept of the multivalued shapes, three levels of estimated classes based on the possibility values (θ), namely $\theta \geq 0.5$, $\theta \geq 0.25$, $\theta > 0$ are only shown. The [illegible] are all parallelepiped (with 3 sides) in shape and their sizes are made as small as possible so that each of them can be distinguished in the feature space. Figures 2.12(b), 2.12(c) and 2.12(d) show the estimated shapes with $\theta \geq 0.5$, $\theta \geq 0.25$, $\theta > 0$ respectively corresponding to the pattern class in Fig. 2.12(a). Figures 2.13(b), 2.13(c) and 2.13(d) show the estimated shapes with $\theta \geq 0.5$, $\theta \geq 0.25$, $\theta > 0$ respectively corresponding to the pattern class in Fig. 2.13(a). Figures 2.14(b), 2.14(c) and 2.14(d) show the lower bisected portion of the estimated shapes with $\theta \geq 0.5$, $\theta \geq 0.25$, $\theta > 0$ respectively corresponding to the lower bisected portion as shown in Fig. 2.14(a) of a pattern class.

To give a 3-D feeling of the training sets, the patterns of the class in Fig. 2.12(a) are shown in Fig. 2.15. Dotted lines corresponding to all the sample points are drawn from the $F_1 \times F_3$ feature plane (with F_2 as the minimum value in the pattern class). The actual locations of the sample points are distinctly marked. Based on this sample set, the output as shown in figures 2.12(b), 2.12(c) and 2.12(d) with $\theta \geq 0.5$, $\theta \geq 0.25$, $\theta > 0$ respectively are obtained. ♠

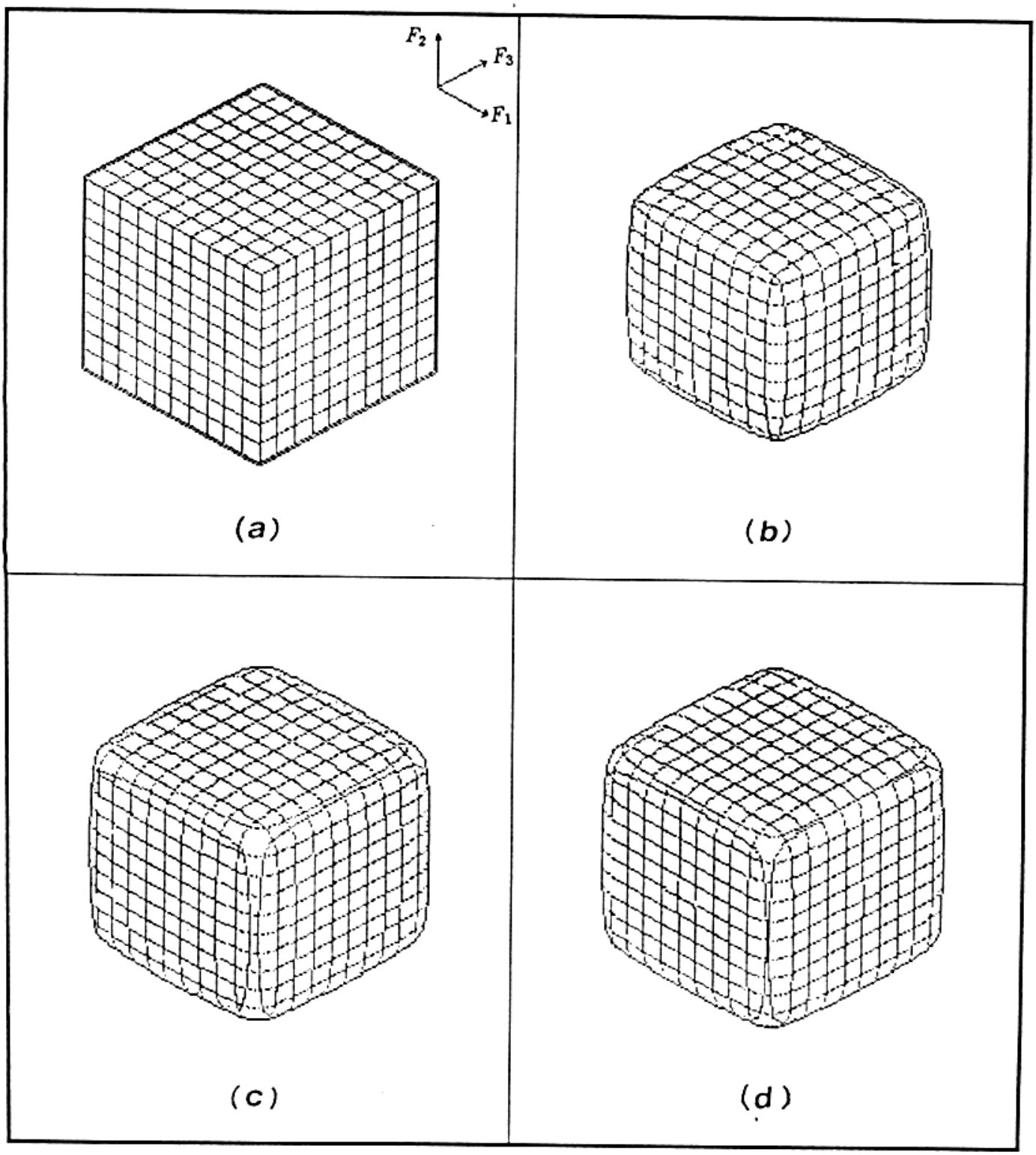


Figure 2.12 : (a) A pattern class; (b)-(d) corresponding estimated versions with $\theta \geq 0.5$, $\theta \geq 0.25$, $\theta > 0$ respectively.

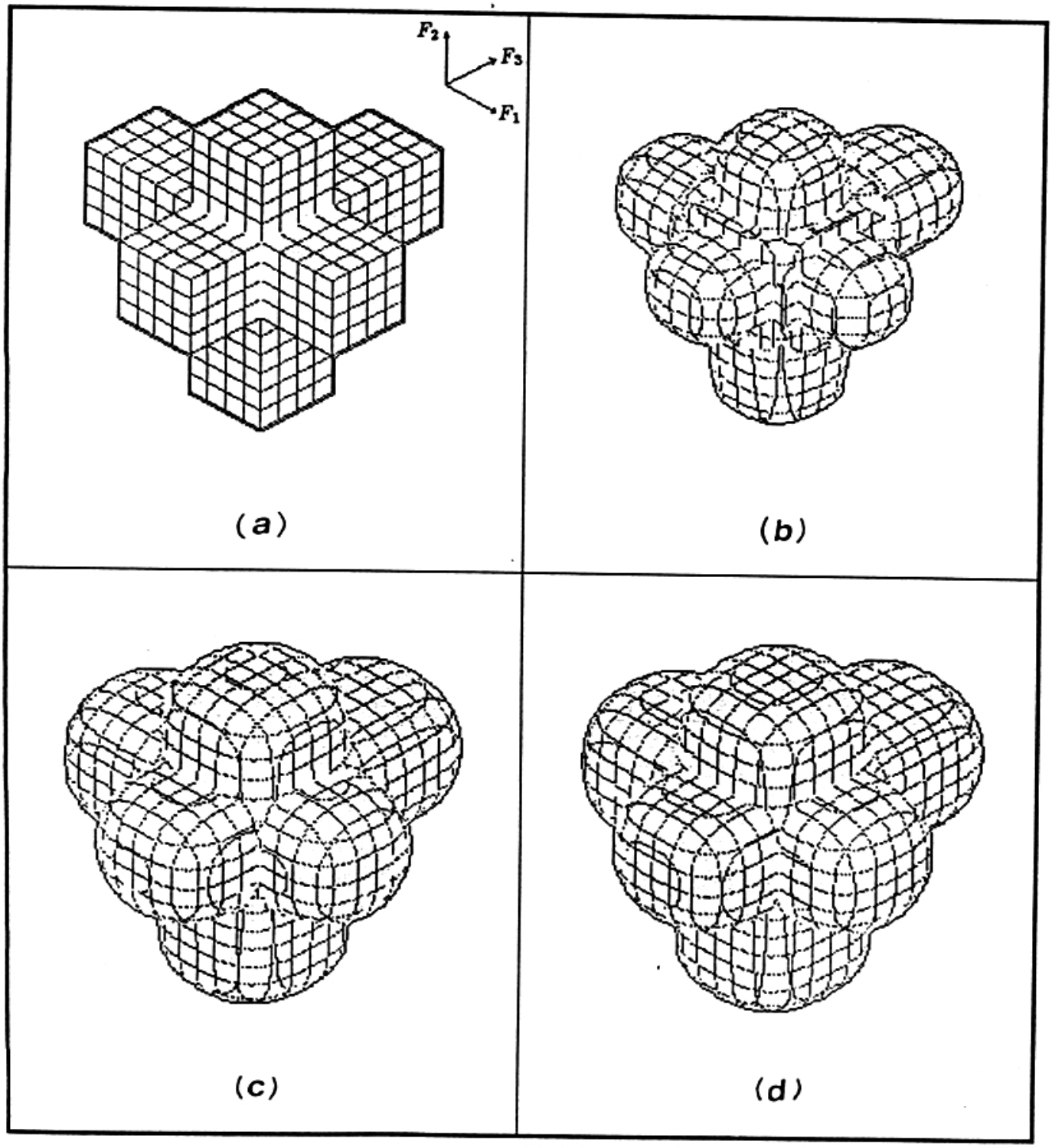


Figure 2.13 : (a) A pattern class; (b)-(d) corresponding estimated versions with $\theta \geq 0.5$, $\theta \geq 0.25$, $\theta > 0$ respectively.

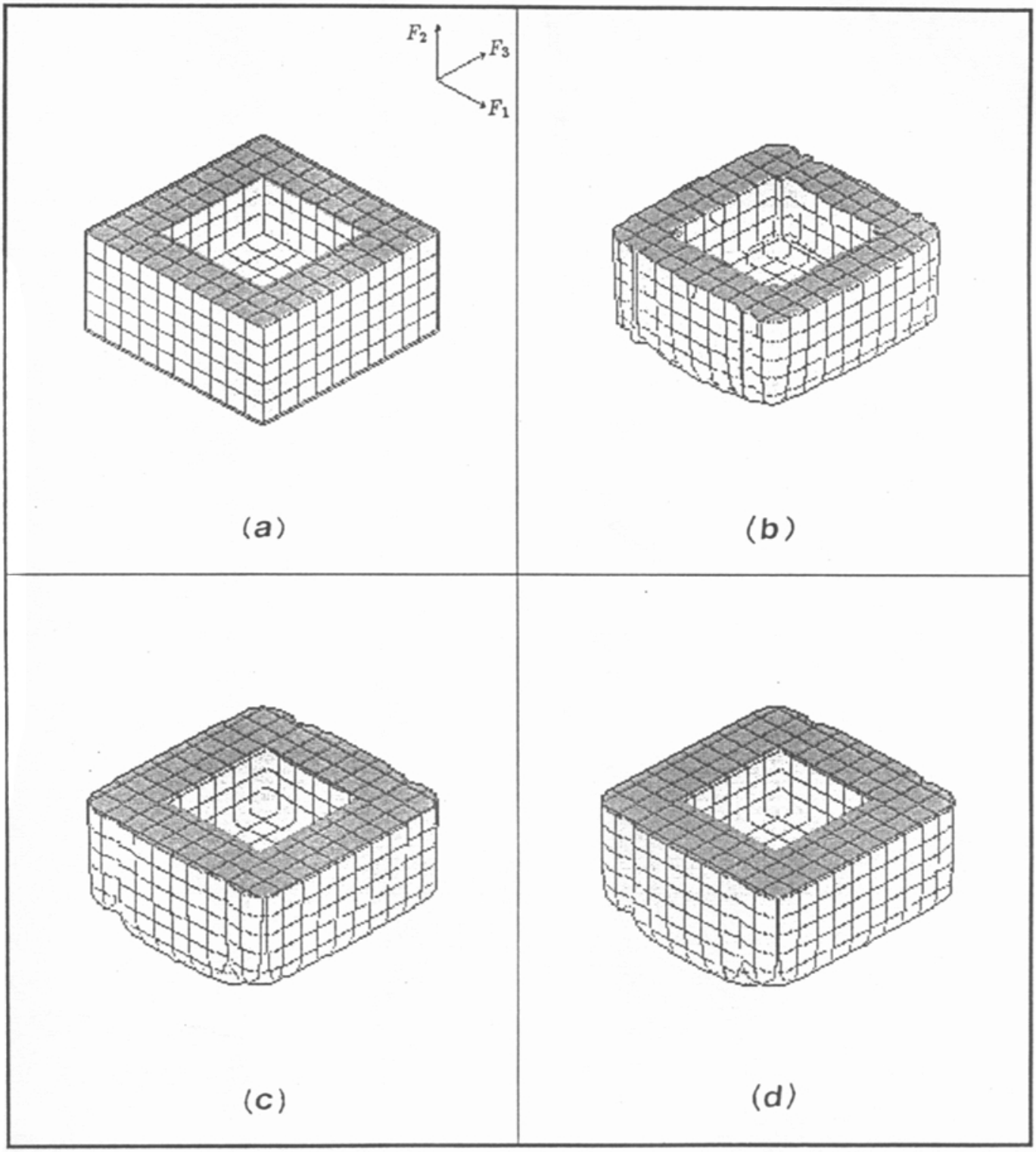


Figure 2.14 : (a) Lower segmented portion of a pattern class; (b)-(d) corresponding estimated versions with $\theta \geq 0.5$, $\theta \geq 0.25$, $\theta > 0$ respectively.

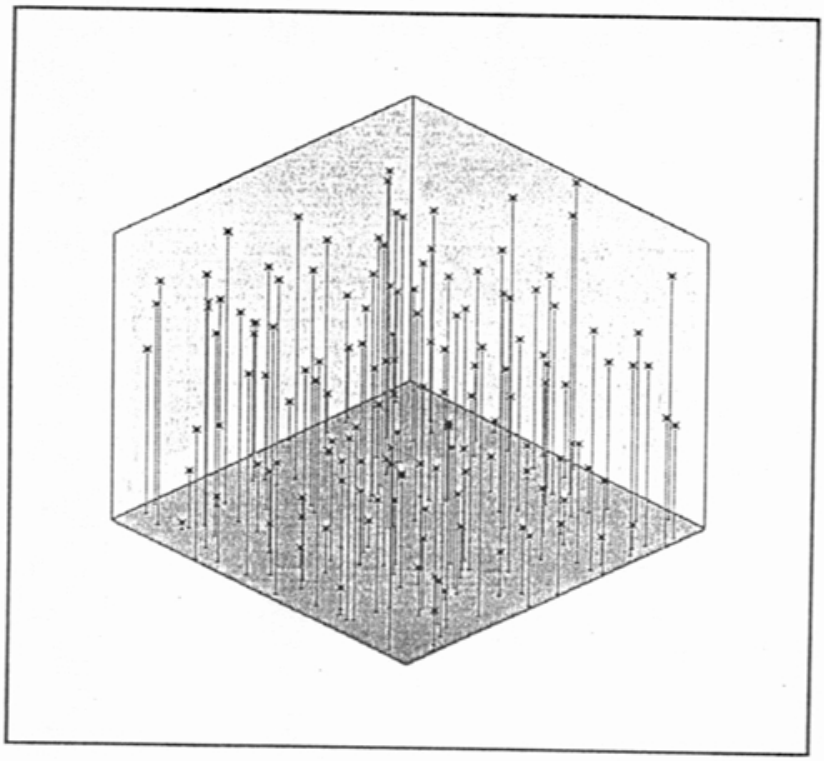


Figure 2.15 : A set of training samples from the class in Fig. 2.12(a).

2.7 Convergence with Sample Size

Convergence of the estimated pattern class to the original pattern class is shown here experimentally and analytically for any dimension $N \geq 2$. For any shape determining approach based on sampled points, the accuracy of the estimated shapes, in general, should improve with the increase of the size (number) of the sampled points. It will be shown in this section that the proposed multivalued shape determining procedure also has this property. As the sample size (t) increases, the value of the accuracy factor (δ_t) decreases and hence accordingly the accuracy of the estimated multivalued shape increases.

2.7.1 Experimental verification

Two artificially generated pattern classes have been considered in this section to demonstrate the convergence property of the proposed algorithm. The first one is a disc [Fig. 2.16(a)] with radius 2 and centre at $(3, 3)$. Five differ-

ent sets of data are chosen randomly from it with sizes 50, 100, 150, 200 and 250. The algorithm was applied assuming these five data sets as sample points with accuracy factors (δ_t) 0.16, 0.13, 0.10, 0.08 and 0.07 respectively. The corresponding estimated multivalued classes are shown in figures 2.16(b)-(f).

The convergence property has also been verified for another pattern class of spherical shape [Fig. 2.17] with radius 2 and centre at $(3,3,3)$. Four different sets of data are chosen randomly from it with sizes 150, 300, 500 and 1200 respectively and the values of δ_t are considered as 0.20, 0.16, 0.13 and 0.10 respectively. Here also three levels of extracted classes are shown corresponding to $\theta \geq 0.5$, $\theta \geq 0.25$, $\theta > 0$. Figures 2.18(a)-(d) show the estimated classes with $\theta \geq 0.5$ based on the selected sample sets of sizes 150, 300, 500 and 1200 respectively corresponding to the pattern class in Fig. 2.17. Figures 2.19(a)-(d) show the estimated classes with $\theta \geq 0.25$ based on 150, 300, 500 and 1200 training samples respectively. Figures 2.20(a)-(d) show the estimated classes with $\theta > 0$ based on 150, 300, 500 and 1200 training samples respectively.

It can be seen from these results that as the sample size (t) increases the estimated classes are gradually converging to the original pattern class. ♣

2.7.2 Criteria for goodness of fit

The aforementioned convergence property is also verified analytically. For this purpose, a distance measure (metric) between two sets may be used and the value of the measure should tend towards zero as the sample size $(t) \rightarrow \infty$. Here two metrics are used. One of them is the Hausdorff metric and the other one is a new similarity metric that has been defined here. It has been shown that the values of both the metrics tend toward zero as $t \rightarrow \infty$.

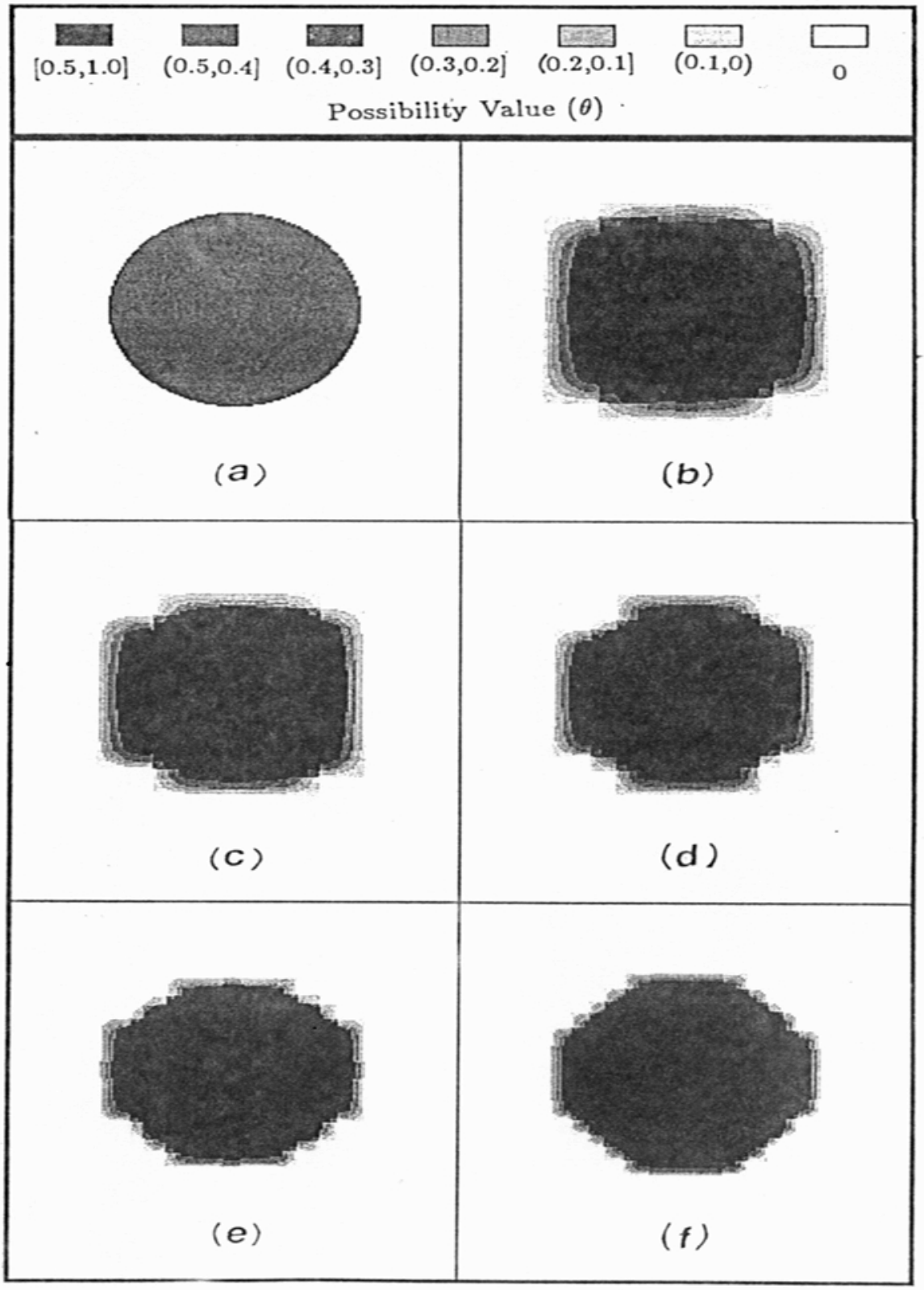


Figure 2.16 : (a) A circular class; (b)-(f) corresponding estimated versions based on 50, 100, 150, 200 and 250 samples respectively.

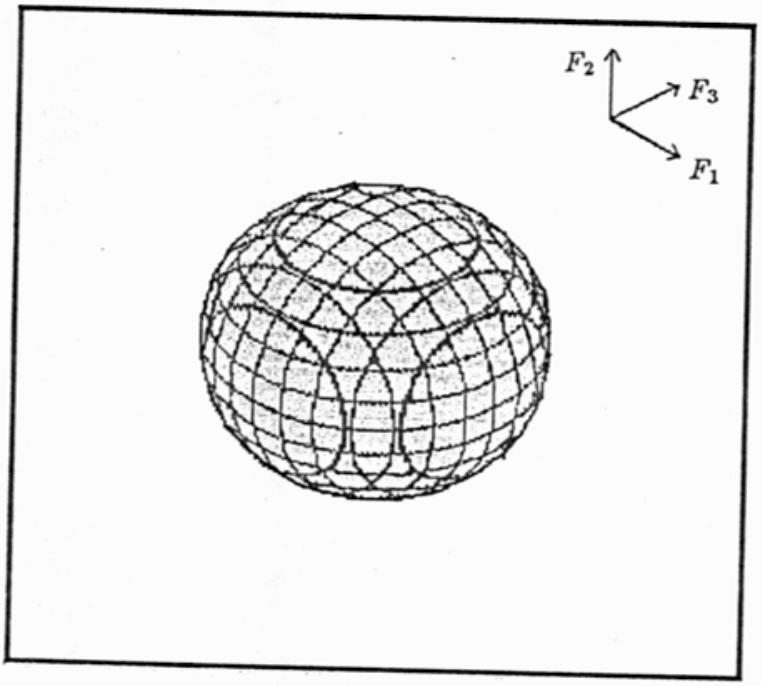


Figure 2.17 : A spherical class.

A. Hausdorff metric

Normally, to find the similarity between sets, a distance measure is often used. Hausdorff metric [221] has been used here for this purpose.

Let (X, d) be a metric space. For any compact subset of A of X, define

$$\delta(y, A) = \inf_{x \in A} d(x, y)$$

where inf means infimum. Note that $\delta(y, A)$ is finite and $\exists x_0 \in A$ such that $\delta(y, A) = d(x_0, y)$. Now the definition of the Hausdorff metric is given below.

Definition 2.4 [221] : *Let A and B be two compact subsets in $\mathbb{R}^N$. Then the distance between A and B, denoted by $Dist(A, B)$, is defined as*

$$Dist(A, B) = \max \left\{ \sup_{x \in A} \delta(x, B) \ , \ \sup_{y \in B} \delta(y, A) \right\} \qquad (2.12)$$

the sets A and B are non-empty. Here sup *and* max *stand for supremum and maximum respectively.* •

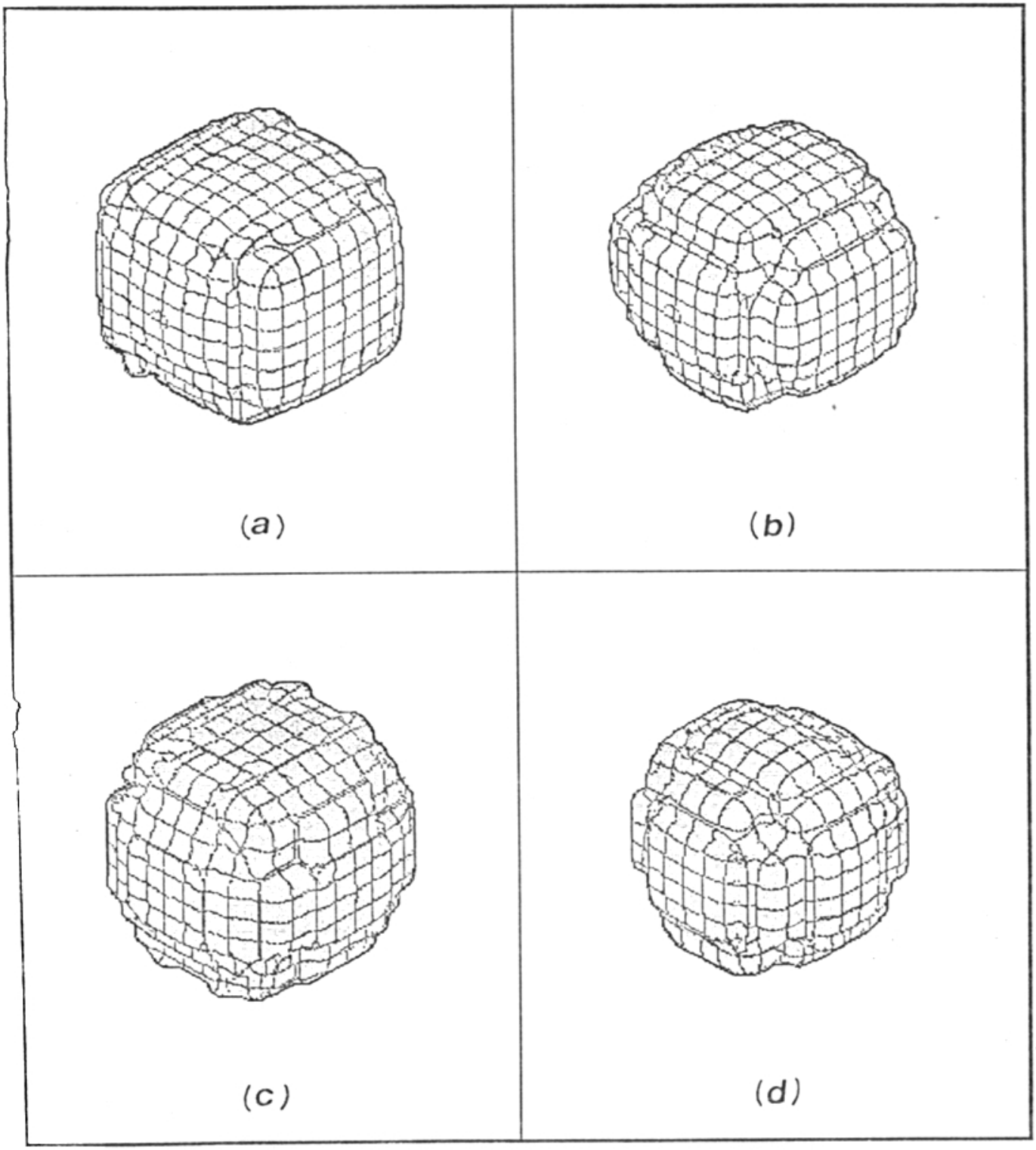


Figure 2.18 : (a)-(d) Estimated versions of the class in Fig. 2.17 with $\theta \geq 0.5$ based on 150, 300, 500 and 1200 samples respectively.

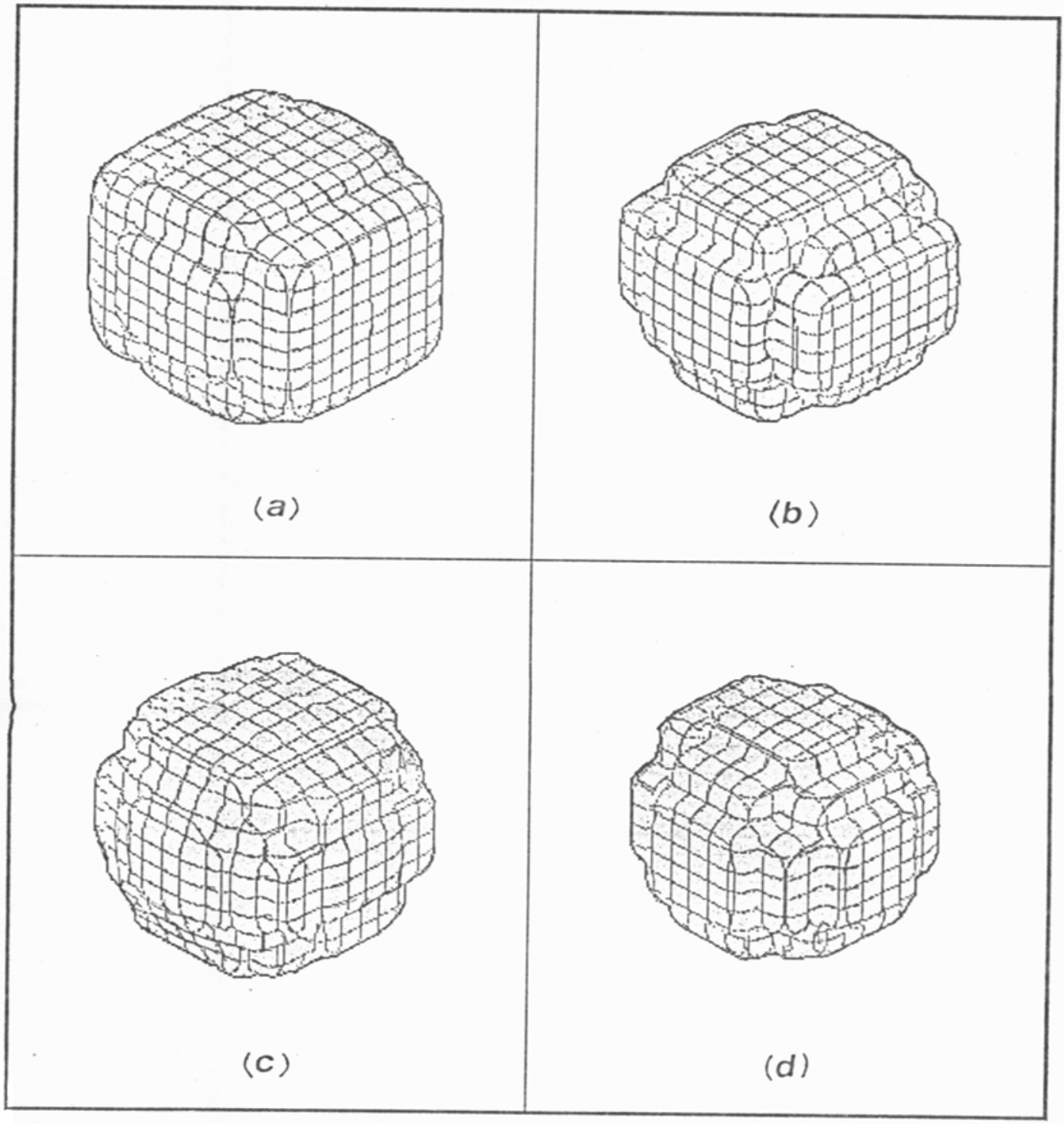


Figure 2.19 : (a)-(d) Estimated versions of the class in Fig. 2.17 with $\theta \geq 0.25$ based on 150, 300, 500 and 1200 samples respectively.

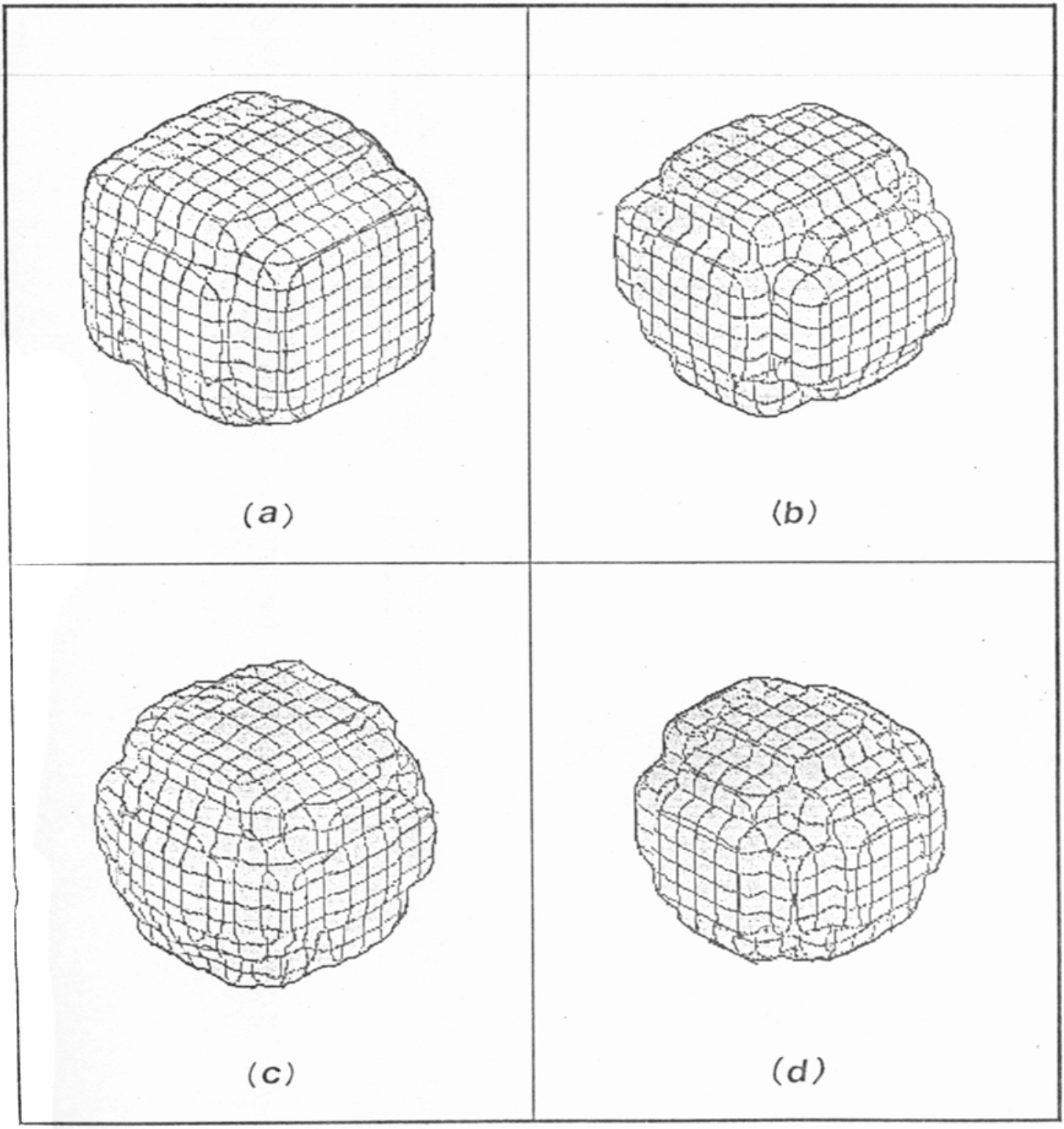


Figure **2.20 :** (a)-(d) Estimated versions of the class in Fig. 2.17 with $\theta > 0$ based on 150, 300, 500 and 1200 samples respectively.

If the original sets are assumed to be finite, the sup and inf can be replaced by max and min (minimum) respectively and then $Dist(\mathcal{A}, \mathcal{B})$ is defined as

$$Dist(\mathcal{A}, \mathcal{B}) = max \left\{ \max_{x \in \mathcal{A}} \delta(x, \mathcal{B}) \ , \ \max_{y \in \mathcal{B}} \delta(y, \mathcal{A}) \right\} \qquad (2.13)$$

where $\delta(x, \mathcal{B}) = \min_{y \in \mathcal{B}} d(x, y)$

and $\delta(y, \mathcal{A}) = \min_{x \in \mathcal{A}} d(x, y)$.

This distance measure $Dist(\mathcal{A}, \mathcal{B})$ is considered here as one of the criteria for goodness of fit, where $\mathcal{A}$ is considered as the boundary of the estimated set or class and $\mathcal{B}$ is considered as the boundary of the original class. This distance measure has initially been applied on the five estimated sets or classes [figures 2.16(b)-(f)] with the original set [Fig. 2.16(a)] in the following way.

The boundary of the disc [Fig. 2.16(a)] is approximated by 180 equally spaced points. This set of 180 points is considered here as the set $\mathcal{B}$. Note that the estimated classes are multivalued. Hence, in order to apply this measure, three levels of estimated boundary based on the possibility values (θ), namely $\theta \geq 0.5$, $\theta \geq 0.25$ and $\theta > 0$ are considered. The values of the $Dist$ measure are shown by a graph in Fig. 2.21(a).

The distance measure $Dist$ has also been applied on the estimated sets or classes [figures 2.18(a)-(d), 2.19(a)-(d) and 2.20(a)-(d)] with the original set in Fig. 2.17. The boundary of the sphere is approximated by 5675 equally spaced points and this set of 5675 points is considered here as the set $\mathcal{B}$. Here also three levels of estimated boundary based on the possibility values (θ), namely $\theta \geq 0.5$, $\theta \geq 0.25$ and $\theta > 0$ are considered. The values of the $Dist$ measure are shown by a graph in Fig. 2.21(b). From the above results [figures 2.21(a) & (b)], the convergence property of the proposed algorithm is verified. ♣

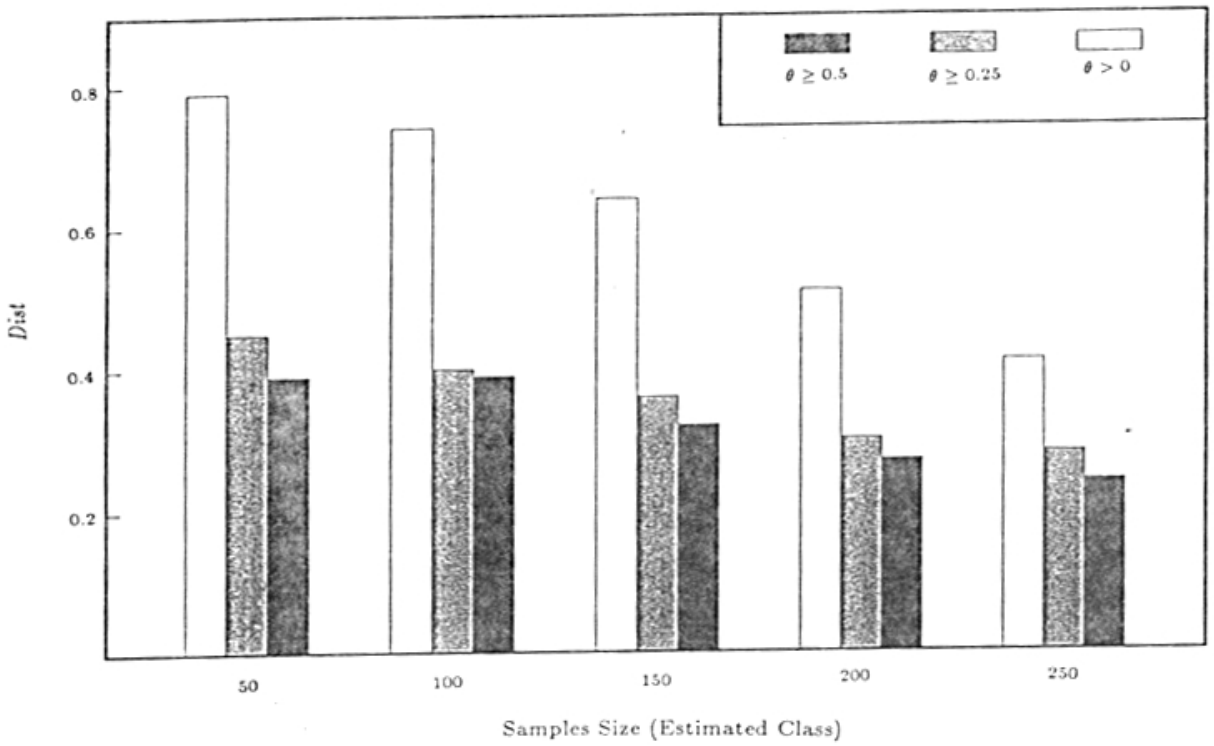


Figure 2.21(a) : Values of *Dist* measure between the estimated classes in figures 2.16(b)-(f) and the actual class in Fig. 2.16(a).

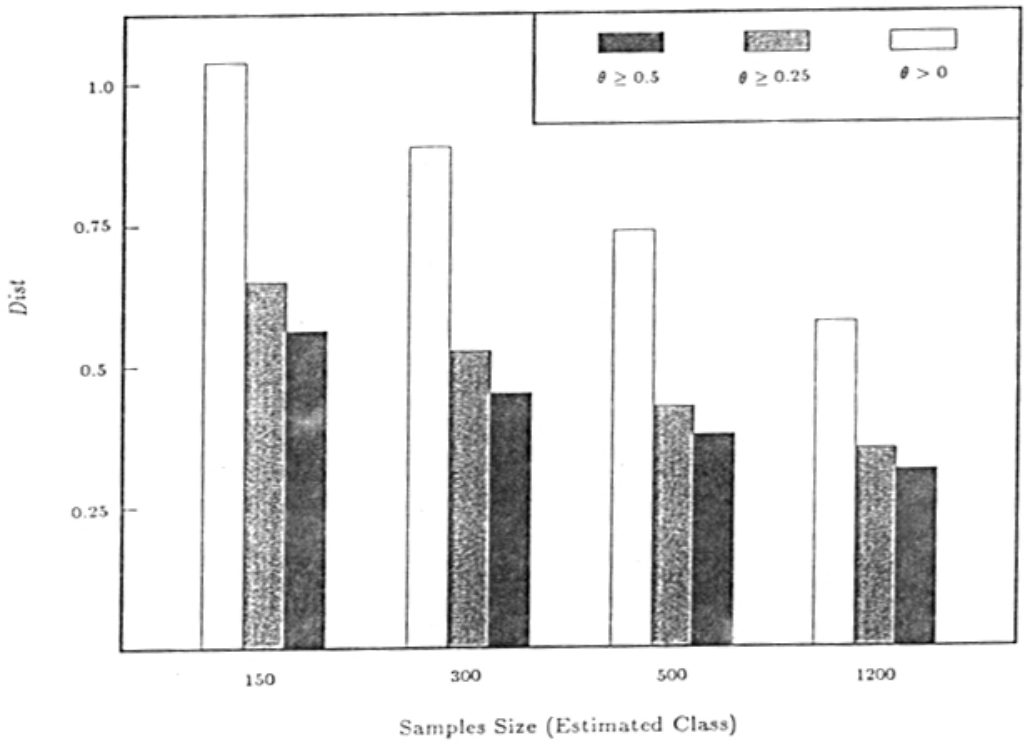


Figure 2.21(b) : Values of *Dist* measure between the actual class in Fig. 2.17 and its estimated versions.

B. A new metric

Note that the Hausdorff metric reflects the overall similarity between two closed sets. In order to incorporate the similarity of each of the elements of the sets, a new measure has been defined below.

Definition 2.5 : *Let $\mathcal{A}$ and $\mathcal{B}$ be 2 finite sets with $n_{\mathcal{A}}$ and $n_{\mathcal{B}}$ elements respectively. Then a similarity measure between $\mathcal{A}$ and $\mathcal{B}$, denoted by $Sim(\mathcal{A},\mathcal{B})$, is defined as*

$$Sim(\mathcal{A},\mathcal{B}) = \frac{1}{n_{\mathcal{A}}}\sum_{x\in\mathcal{A}}\delta(x,\mathcal{B}) + \frac{1}{n_{\mathcal{B}}}\sum_{y\in\mathcal{B}}\delta(y,\mathcal{A}) + Dist(\mathcal{A},\mathcal{B}) \tag{2.14}$$

Here the first term denotes the average similarity of the elements of $\mathcal{A}$ to $\mathcal{B}$, the second term denotes the average similarity of the elements of $\mathcal{B}$ to $\mathcal{A}$ and the last term $Dist(\mathcal{A},\mathcal{B})$ denotes the overall similarity between $\mathcal{A}$ and $\mathcal{B}$. This is a metric and its proof is provided below. •

Proposition 2.1 : *$Sim(\mathcal{A},\mathcal{B})$ is a metric.*

Proof : For $\delta(\mathcal{A},\mathcal{B})$, the following relations hold [221]

$$\begin{array}{lllll} (i) & \delta(\mathcal{A}^c,\mathcal{B}^c) & = & \delta(\mathcal{A},\mathcal{B}) \\ (ii) & \{\delta(x,\mathcal{A})=0\} & \Rightarrow & \{x\in\mathcal{A}^c\} \\ (iii) & \delta(\mathcal{A},\mathcal{C}) & \leq & \delta(\mathcal{A},\mathcal{B})+\delta(\mathcal{B},\mathcal{C})+\rho(\mathcal{B}) \end{array} \tag{2.15}$$

where $\rho(\mathcal{B})$ denotes the diameter of the set $\mathcal{B}$ and it is defined as the least upper bound of the distances of the elements of $\mathcal{B}$. •

Let $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ be three finite sets in $\mathbb{R}^N$. Clearly $Sim(\mathcal{A},\mathcal{B}) = 0$ iff $\mathcal{A}=\mathcal{B}$. Again it is obvious from the definition that $Sim(\mathcal{A},\mathcal{B}) = Sim(\mathcal{B},\mathcal{A})$. It only remains to prove the triangular inequality i.e.,

$$Sim(\mathcal{A},\mathcal{C}) \;\leq\; Sim(\mathcal{A},\mathcal{B}) \;+\; Sim(\mathcal{B},\mathcal{C}) \tag{2.16}$$

Without loss of generality, we can assume that

$$Dist(\mathcal{A},\mathcal{C}) \;=\; \max\left\{\max_{x\in\mathcal{A}}\;\delta(x,\mathcal{C})\;,\;\max_{z\in\mathcal{C}}\;\delta(z,\mathcal{A})\right\} \;=\; \max_{x\in\mathcal{A}}\;\delta(x,\mathcal{C}) \tag{2.17}$$

$$So,\;\; Sim(\mathcal{A},\mathcal{B}) = \frac{1}{n_{\mathcal{A}}}\sum_{x\in\mathcal{A}}\delta(x,\mathcal{B}) + \frac{1}{n_{\mathcal{B}}}\sum_{y\in\mathcal{B}}\delta(y,\mathcal{A}) + \max_{x\in\mathcal{A}}\;\delta(x,\mathcal{C}) \tag{2.18}$$

If $x\in\mathcal{A}$ and $y\in\mathcal{B}$, then by Eq. (2.15),

$$\delta(x,\mathcal{C}) \;\leq\; d(x,y) \;+\; \delta(\mathcal{B},\mathcal{C}) \;\leq\; d(x,y) \;+\; Dist(\mathcal{B},\mathcal{C})$$

whence

$$\begin{array}{lcccccc} \delta(x,\mathcal{C}) & \leq & \inf\limits_{y\in\mathcal{B}} d(x,y) & + & Dist(\mathcal{B},\mathcal{C}) & = & \delta(x,\mathcal{B}) + Dist(\mathcal{B},\mathcal{C}) \\ \Rightarrow & & \max\limits_{x\in\mathcal{A}} \delta(x,\mathcal{C}) & \leq & \max\limits_{x\in\mathcal{A}} \delta(x,\mathcal{B}) & + & Dist(\mathcal{B},\mathcal{C}) \\ & & & \leq & Dist(\mathcal{A},\mathcal{B}) & + & Dist(\mathcal{B},\mathcal{C}) \end{array} \tag{2.19}$$

By Eq. (2.15) and assuming $\mathcal{A} = \{x\}$,

$$\delta(x,\mathcal{C}) \leq \delta(x,\mathcal{B}) + \delta(\mathcal{B},\mathcal{C})$$

$$\Rightarrow \frac{1}{n_{\mathcal{A}}} \sum_{x\in\mathcal{A}} \delta(x,\mathcal{C}) \leq \frac{1}{n_{\mathcal{A}}} \sum_{y\in\mathcal{B}} \delta(x,\mathcal{B}) + \delta(\mathcal{B},\mathcal{C}) \tag{2.20}$$

$$Again, \quad \delta(\mathcal{B},\mathcal{C}) = \inf_{y\in\mathcal{B}} \delta(y,\mathcal{C}) \leq \frac{1}{n_{\mathcal{B}}} \sum_{y\in\mathcal{B}} \delta(y,\mathcal{C}) \tag{2.21}$$

Using Eq. (2.21) in Eq. (2.20), we get

$$\frac{1}{n_{\mathcal{A}}} \sum_{x\in\mathcal{A}} \delta(x,\mathcal{C}) \leq \frac{1}{n_{\mathcal{A}}} \sum_{x\in\mathcal{A}} \delta(x,\mathcal{B}) + \frac{1}{n_{\mathcal{B}}} \sum_{y\in\mathcal{B}} \delta(y,\mathcal{C}) \tag{2.22}$$

Similarly, one can get

$$\frac{1}{n_{\mathcal{C}}} \sum_{z\in\mathcal{C}} \delta(z,\mathcal{A}) \leq \frac{1}{n_{\mathcal{C}}} \sum_{z\in\mathcal{C}} \delta(z,\mathcal{B}) + \frac{1}{n_{\mathcal{B}}} \sum_{y\in\mathcal{B}} \delta(y,\mathcal{A}) \tag{2.23}$$

Using equations (2.22), (2.23) and (2.19) in Eq. (2.18),

$$\begin{array}{lcccccc} Sim(\mathcal{A},\mathcal{C}) & \leq & \frac{1}{n_{\mathcal{A}}} \sum\limits_{x\in\mathcal{A}} \delta(x,\mathcal{B}) & + & \frac{1}{n_{\mathcal{B}}} \sum\limits_{y\in\mathcal{B}} \delta(y,\mathcal{C}) & & [by(2.22)] \\ & + & \frac{1}{n_{\mathcal{C}}} \sum\limits_{z\in\mathcal{C}} \delta(z,\mathcal{B}) & + & \frac{1}{n_{\mathcal{B}}} \sum\limits_{y\in\mathcal{B}} \delta(y,\mathcal{A}) & & [by(2.23)] \\ & + & Dist(\mathcal{A},\mathcal{B}) & + & Dist(\mathcal{B},\mathcal{C}) & & [by(2.19)] \\ = & \frac{1}{n_{\mathcal{A}}} \sum\limits_{x\in\mathcal{A}} \delta(x,\mathcal{B}) & + & \frac{1}{n_{\mathcal{B}}} \sum\limits_{y\in\mathcal{B}} \delta(y,\mathcal{A}) & + & Dist(\mathcal{A},\mathcal{B}) & \\ + & \frac{1}{n_{\mathcal{B}}} \sum\limits_{y\in\mathcal{B}} \delta(y,\mathcal{C}) & + & \frac{1}{n_{\mathcal{C}}} \sum\limits_{z\in\mathcal{C}} \delta(z,\mathcal{B}) & + & Dist(\mathcal{B},\mathcal{C}) & \\ \Rightarrow & Sim(\mathcal{A},\mathcal{C}) & \leq & Sim(\mathcal{A},\mathcal{B}) & + & Sim(\mathcal{B},\mathcal{C}) & \end{array}$$

Hence the defined Sim is a metric. •

Since the defined metric Sim can be used to find the similarity between any two finite sets, it has been considered as another criterion for goodness of fit for the proposed shape estimation method. It has initially been applied between each of the five estimated sets (with three levels of boundaries as $\theta \geq 0.5$, $\theta \geq 0.25$ and $\theta > 0$) [figures 2.16(b)-(f)] and the original set [Fig. 2.16(a)] in the same way as the previous case. The values of the Sim measure are shown in Fig. 2.22(a).

The metric *Sim* has finally been applied between the pattern class in Fig. 2.17 and its estimated multivalued classes [figures 2.18(a)-(d), 2.19(a)-(d) and 2.20(a)-(d)]. The values of the *Sim* measure are provided in Fig. 2.22(b) where the convergence property is seen to be verified.

Hence, the convergence property of the proposed shape determining procedure is established both experimentally and analytically.

Some remarks :

- The complexity of the proposed algorithm increases exponentially with the dimension (N) of the feature space. Again, it is very difficult to use the Hausdorff metric as well as the newly defined metric *sim* to verify the goodness of the estimated shape of a pattern class for higher dimensions ($N > 3$). It becomes computationally expensive for its implementation in higher dimensional feature spaces, although we could formulate the algorithm for any dimension.

- Note that for a given support P_α over the set α, sometimes, a fewer number of sample points may give a better representation of α (i.e., Hausdorff distance very small) than a large number of points. Moreover, if one considers various distributions over α, then it is not possible to get a unique number of points t such that for every $T > t$, Hausdorff distance is less than 0.05 for every α and for every continuous probability distribution P_α over α, and the number of points being T. At most one may try to get results like "$P_\alpha\{$Hausdorff distance between the estimated set based on a training set of size t and α is less than $0.05\} > 1 - \epsilon \;\; \forall\, t > T_\alpha$ where ϵ is a small positive number and T_α is an integer depending on α". Obtaining this result for any $\alpha \subseteq I\!R^2$ and making P_α independent of α is a difficult task. The problem becomes further complicated for $\alpha \subseteq I\!R^N$, $N > 2$. So the problems of minimum sample size requirement (t) and rate of convergence of the proposed method have not been dealt with here. ♠

In this chapter, we have demonstrated an application of the fuzzy set theory in estimating multivalued (with continuum grade of belongingness) shape/boundary of a pattern class from a few sampled points by representing some portions uncovered by the sampled points in terms of membership values ($0 < \theta \leq 1$) to be in the estimated class. In the next chapter, we will be demonstrating another application of fuzzy set theory in formulating a multivalued linguistic recognition system which can accept imprecise or incomplete input and provide linguistic (soft) decisions associated with their certainty values.

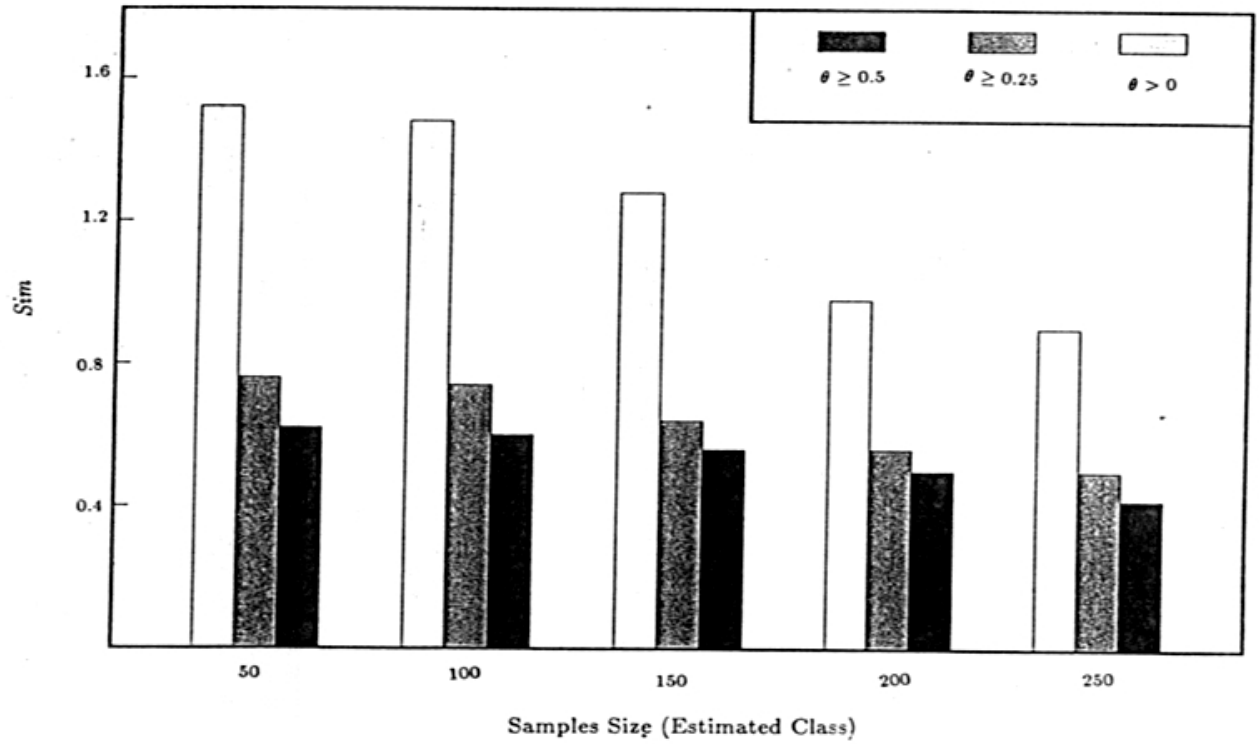


Figure 2.22(a) : Values of *Sim* measure between the estimated classes in figures 2.16(*b*)-(*f*) and the actual class in Fig. 2.16(*a*).

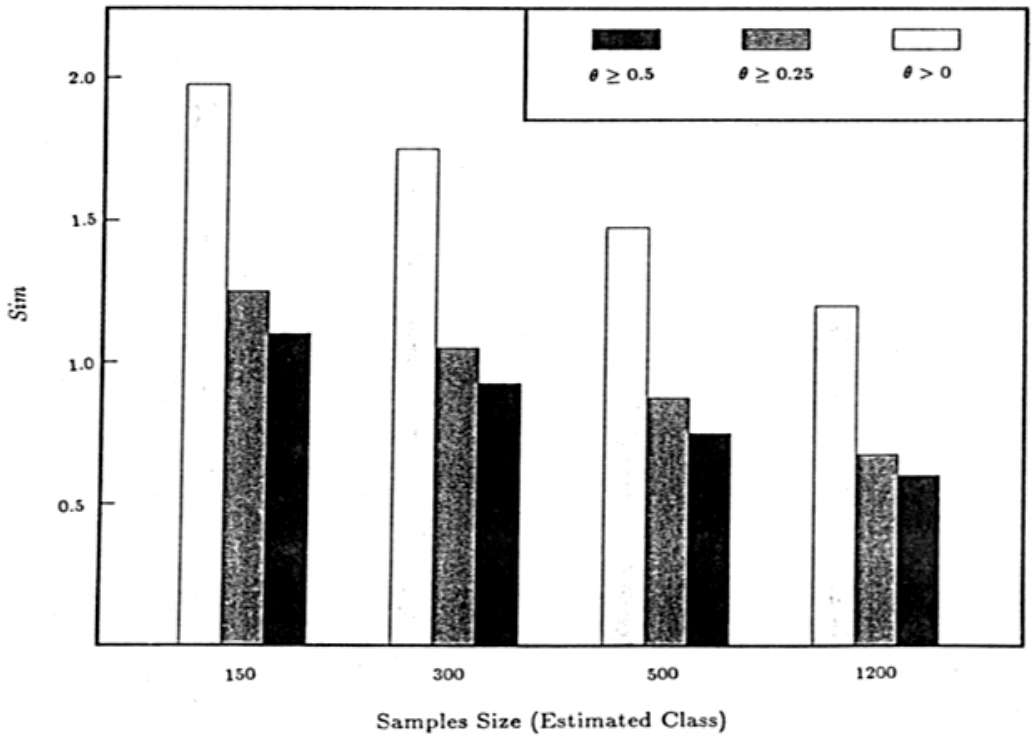


Figure 2.22(b) : Values of *Sim* measure between the actual class in Fig. 2.17 and its estimated versions.

Chapter 3

MULTIVALUED RECOGNITION SYSTEM USING LINGUISTIC PROPERTY BASED DECOMPOSITION

Contents

3.1 Introduction

With the conventional probabilistic and deterministic [2-11] classifiers, the features characterizing the input patterns are considered to be quantitative (exact numerals) in nature. The patterns having imprecise or incomplete information are usually ignored or discarded from their designing and testing processes. The impreciseness (or ambiguity) [70,232] may arise from various reasons. For example, instrumental error or noise corruption in the experiment may lead to having partial/unreliable information available on a feature measurement F *viz.*, F is about 500 (*mixed form*) or F is between 400 and 500 (*set form*). Again, in some cases the expense incurred in extracting exact value of feature may be high or, it may be difficult to decide on the actual salient features to be extracted; on the other hand, it may become convenient to use the linguistic variables and hedges e.g., *small*, *medium*, *high*, *very*, *more or less etc.* in order to describe feature information (*viz.* F is *very small*). There has recently been an attempt [143,144] to provide the design concept of a classifier which needs enough apriori knowledge from experts, in linguistic form only, regarding the classification problem (*viz.* medical diagnosis).

In the present chapter, a multivalued recognition system based on the theories of fuzzy set and approximate reasoning is described which is capable of handling all the aforesaid impreciseness in pattern without consulting any expert. The classifier can be viewed as *general* because it takes feature input in both exact and inexact forms. As the linguistic representation contains summarized information, it is difficult to convert the linguistic information into a quantitative form. On the other hand, it is easier to convert any information into linguistic form. Keeping this in mind, the algorithm considers only three primary linguistic properties namely *small*, *medium* and *high* so that any input information can be thought of possessing various combination of these properties to some degree. As a result, the entire feature space is decomposed into 3^N (N being the number of features) overlapping *space sub-*

domains representing various property combinations, thereby providing more local information of the feature space for making decisions. Based on these properties, the various membership values of imprecise input are assigned and modified. The membership/compatibility functions for *small* and *high* have been represented by π functions. Note that this is a major deviation from the standard fuzzy set theoretic approach where these are represented by (1-S) and S type functions respectively. Since all the primary feature properties are not equally important in characterizing a class, a concept of weighting coefficient has also introduced.

The recognition system uses Zadeh's compositional rule of inference [72] and gives a natural output decision associated with its certainty (or validity). Finally, the effectiveness of the proposed linguistic system has been demonstrated on speech recognition problem where the classes have ill-defined boundaries and the input feature information have the aforementioned impreciseness.

In section 3.2, an introduction of linguistic variable and approximate reasoning is provided. Section 3.3 describes the basic features and the block diagram of the multivalued recognition system. The description of different blocks is provided in sections 3.4, 3.5 and 3.6. Results on speech recognition problem are discussed in section 3.7. ♠

3.2 Linguistic Variable and Approximate Reasoning

3.2.1 Linguistic variable

By a linguistic variable, we mean a variable whose values are not numbers but words or sentences in a natural language.

Definition 3.1 [73-75] : *A linguistic variable is characterized by a quintuple* $(X, T(X), \mathcal{X}, G, M)$ *in which* X *is the name of the variable;* $T(X)$ *is the term set of* X*;* $\mathcal{X}$ *is a universe of discourse;* G *is a syntactic rule which generates the terms in* $T(X)$*; and* M *is a semantic rule which associates with each linguistic value* X *its meaning where* $M(X)$ *denotes a fuzzy subset of* $\mathcal{X}$*. For a particular* X*, the name generated by* G *is called a term.* •

Example 3.1 : Suppose X is a linguistic variable having the label *height* with $\mathcal{X} = [0, 250]$. Terms of this linguistic variable, which are fuzzy sets, could be called *tall*, *short*, *very tall*, and so on. The base variable $\mathcal{X}$ is the height in cm. of persons. $M(X)$ is the rule that assigns a meaning, that is, a fuzzy set of the term

$$M(tall) = \{x, \mu_{tall}(x)\}, \qquad x \in [0, 250],$$

where

$$\mu_{tall}(x) = \begin{cases} 0 & \text{for } x \in [0, 150] \\ [1 + \{\frac{x-150}{10}\}^{-2}]^{-1} & \text{for } x \in [150, 250]. \end{cases}$$

$T(X)$ will define the term set of the variable X. In this case

$$T(height) = \{tall,\ very\ tall,\ not\ very\ tall,\ quite\ tall,\ short,\ more\ or\ less\ short,\ \ldots\}$$

where $G(X)$ is the rule which generates the terms in the term set $T(X)$.

Definition 3.2 : *A linguistic hedge or a modifier is an operator, which modifies the meaning of a term or more generally of a fuzzy set. If* A *is a*

fuzzy set then the modifier m generates the composite term $b = m(\mathcal{A})$.

The mathematical operators which are used very frequently as modifiers are

$$\begin{aligned} &\text{Concentration :} && \mu_{CON(\mathcal{A})}(x) &&= && (\mu_{\mathcal{A}}(x))^2 \\ &\text{Dilation :} && \mu_{DIL(\mathcal{A})}(x) &&= && (\mu_{\mathcal{A}}(x))^{\frac{1}{2}} \end{aligned} \tag{3.1}$$

and contrast intensification :

$$\mu_{INT(\mathcal{A})}(x) = \begin{cases} 2[\mu_{\mathcal{A}}(x)]^2 & \text{for} \quad \mu_{\mathcal{A}}(x) \in [0, 0.5] \\ 1 - 2[1 - \mu_{\mathcal{A}}(x)]^2 & \text{otherwise.} \end{cases} \tag{3.2}$$

Generally, the following linguistic hedges are associated with the aforementioned operators. If $\mathcal{A}$ is a term (fuzzy set) then

$$\begin{aligned} &very \;\; \mathcal{A} && \equiv && CON(\mathcal{A}) \\ &More \; or \; less \;\; \mathcal{A} && \equiv && DIL(\mathcal{A}) \\ &plus \;\; \mathcal{A} && \equiv && \mathcal{A}^{1.25} \\ &slightly \;\; \mathcal{A} && \equiv && INT\,[plus \; \mathcal{A} \;\; and \;\; not(very \; \mathcal{A})] \end{aligned}$$

3.2.2 Approximate reasoning

Zadeh has developed a theory of approximate reasoning [72] based on fuzzy set theory. This theory aims at modeling the human reasoning and thinking process with linguistic variable in order to handle both soft and hard data as well as various types of uncertainty. Many aspects of the underlying concept have been incorporated in designing decision making systems [17-19,77-80,233] along with their applications.

By approximate reasoning, we mean a type of reasoning which is neither very exact nor very inexact. Consider a fuzzy proposition p of the form

$$p \equiv x \; is \; Q$$

where x is a name of an object and Q is a label of a fuzzy subset of a universe

$\mathcal{X}$. p can be expressed by the relational assignment equation [68] as

$$\mathcal{R}(\mathcal{A}(x)) \quad = \quad Q$$

where $\mathcal{A}$ is an implied attribute of x i.e., an attribute which is implied by x and Q; $\mathcal{R}$ denotes a fuzzy restriction on $\mathcal{A}(X)$ to which the value Q is assigned by the relational assignment equation.

Example 3.2 : Let

$p \quad \equiv$ This tomato is very red.

So the corresponding relational assignment equation will be

$\mathcal{R}$ (colour(This tomato)) $\equiv$ very red.

For an illustration of approximate reasoning, let us consider the proposition p as a premise. The conclusion corresponding to an implication may be as follows:

Implication : If a tomato is red then the tomato is ripe.

Conclusion : This tomato is very ripe.

Various methods [72,77,78] have been suggested for the previous type of fuzzy conditional implication. We have restricted ourselves to Zadeh's approach [72] while developing the recognition system. The definition of Zadeh's composition rule of inference is as follows.

Definition 3.3 [72] : *Let $\mathcal{A}$ and $\mathcal{B}$ denote fuzzy sets in $\mathcal{X}$ and $\mathcal{X} \times \mathcal{Y}$ respectively. Then the composition rule of inference asserts that the solution of the relational assignment equations*

$$R(x) \quad = \quad \mathcal{A} \qquad \textit{and} \qquad \mathcal{R}(x,y) \quad = \quad \mathcal{B}$$

is given by

$$\mathcal{R}(y) \quad = \quad \mathcal{A} \circ \mathcal{B} \quad = \quad C \tag{3.3}$$

where $\mathcal{A} \circ \mathcal{B}$ is the max-min composition of $\mathcal{A}$ and $\mathcal{B}$.

Example 3.3 : Let the universe be $\mathcal{X} = \{1, 2, 3, 4\}$.

$\mathcal{A} \equiv$ little $= \{(1, 1.0), (2, 0.6), (3, 0.2), (4, 0.0)\}$

$\mathcal{B} \equiv$ *approximately equal* be a fuzzy relation defined by

	1	2	3	4
1	1.0	0.5	0.0	0.0
2	0.5	1.0	0.5	0.0
3	0.0	0.5	1.0	0.5
4	0.0	0.0	0.5	1.0

Applying the max-min composition, $C(y) = \mathcal{A} \circ \mathcal{B}$ yields

$$\begin{aligned} C(y) &= \max_x \min\{\mu_{\mathcal{A}}(x), \mu_{\mathcal{B}}(x, y)\} \\ &= \{(1, 1.0), (2, 0.6), (3, 0.5), (4, 0.2)\} \\ &\equiv \textit{approximately little.} \quad \spadesuit \end{aligned}$$

3.3 Multivalued Recognition System

3.3.1 Concept of representing a pattern

Based on the theories of fuzzy set and approximate reasoning, let us now describe a recognition system which is capable of handling input pattern having feature information in *quantitative form*, *linguistic form*, *mixed form* and *set form*. To handle various imprecise/ambiguous information in these forms, three primary linguistic terms or properties are considered, namely *small*, *medium* and *high* so that the feature information in any of the aforesaid forms can be viewed to have these properties to some degree. Therefore, a pattern

$X = [x_1, x_2, \ldots, x_N]'$ can be represented as

$$X = \begin{bmatrix} \mu_{small}(x_1) & \mu_{medium}(x_1) & \mu_{high}(x_1) \\ \mu_{small}(x_2) & \mu_{medium}(x_2) & \mu_{high}(x_2) \\ \vdots & \vdots & \vdots \\ \mu_{small}(x_N) & \mu_{medium}(x_N) & \mu_{high}(x_N) \end{bmatrix} \tag{3.4}$$

Representation of the imprecise input X through their primary properties *small*, *medium* and *high* basically implies that the entire dynamic range of each feature has been divided into three overlapping *feature subdomains* corresponding to these primary properties. So the whole feature space is decomposed into 3^N overlapping *space subdomains*. Note that the regions in the individual feature axis are referred to as the *feature subdomains* and the regions in the whole feature space, which are combinations of the *feature subdomains*, are referred to as the *space subdomains*.

Let us now define a *Property Combination Vector* (henceforth PCV) consisting of 3^N components, which represent the various combinations of the basic properties *small*, *medium* and *high* as possessed by the features of X. The components *viz.* (F_1 is *small*, F_2 is *small*, ..., F_N is *small*), (F_1 is *small*, F_2 is *small*, ..., F_N is *medium*), ..., (F_1 is *high*, F_2 is *high*, ..., F_N is *high*), correspond to the aforementioned 3^N *space subdomains* of the feature space. In other word, we can write for $N = 2$

$$PCV = \begin{bmatrix} F_1 \text{ is } small & , & F_2 \text{ is } small \\ F_1 \text{ is } small & , & F_2 \text{ is } medium \\ F_1 \text{ is } small & , & F_2 \text{ is } high \\ \vdots & , & \vdots \\ F_1 \text{ is } high & , & F_2 \text{ is } high \end{bmatrix} \tag{3.5}$$

having $3^2 = 9$ components. These are explained in Fig. 3.1, where the entire feature space has been decomposed into nine overlapping (fuzzy) *space subdomains* in order to represent the impreciseness in input feature information. Each component of PCV corresponds to one of the nine regions and

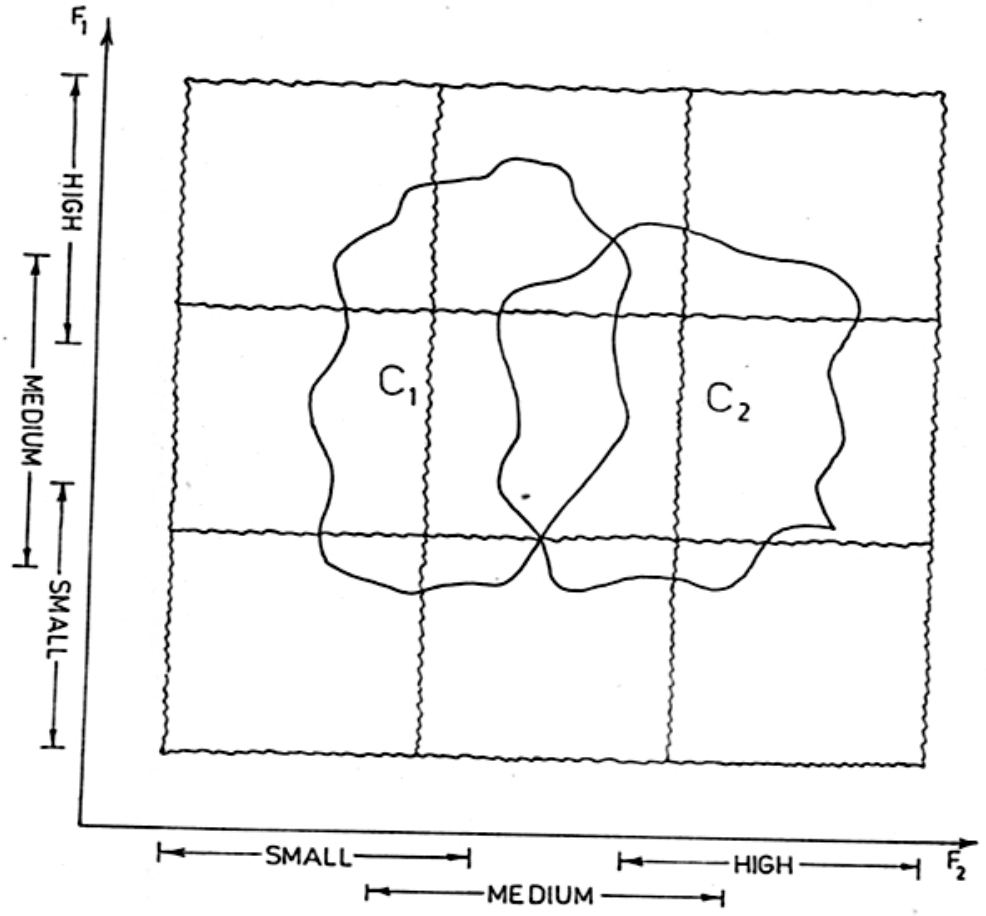


Figure 3.1 : Feature space showing nine overlapping *space subdomains* in terms of the properties *small*, *medium* and *high*. Curl lines denote fuzzy boundaries.

represents a combination of the fuzzy property sets "F_1 is p_1^h" and "F_2 is p_2^h", where h $(h = 1, 2, \ldots, 9)$ stands for the *space subdomains* and depending on the value of h, p_i^h $(i = 1, 2)$ represents one of the *feature subdomains* {*small*, *medium*, *high*} of the individual feature F_i. Value of the hth component of PCV therefore denotes the degree (joint possibility) to which F_1 and F_2 possess the properties p_1^h and p_2^h respectively i.e., the degree of belonging to the hth *space subdomain* of the feature space.

Now a pattern class can be viewed as consisting of all these property combinations to some extent. Depending on the significance (weight) of individual

components of PCV in characterizing a class, a *characteristic vector*

$$CV_j(X) = \left(cv_{j_1}(X), cv_{j_2}(X), \ldots, cv_{j_h}(X), \ldots, cv_{j_{sN}}(X)\right)' \qquad (3.6)$$

corresponding to the pattern class C_j $(j = 1, 2, \ldots, M)$ is determined. Here M denotes the number of classes C_1, C_2, ..., C_M. The hth element of $CV_j(X)$ denotes the degree of possessing the hth property combination by X given that the X is from class C_j.

Example 3.4 : Consider the problem of identifying whether a student is from science stream (C_1) or humanities stream (C_2), from his marks in mathematics (F_1) and literature (F_2) in a combined examination. Assume that the marks are characterized by the linguistic properties *small* (poor), *medium* and *high* (good).

Consider a particular property combination (marks(math) is good, marks(literature) is good). Then obviously, the degree of possessing this property combination is higher if the student is from science stream because the mark in mathematics is more important (i.e., has more weight) than that in literature, in characterizing a science student. The converse is true if the student is from the humanities stream. Therefore, the characteristic vector $[CV(X)]$ for the science stream will be different from that of the humanities. •

The above example shows the varying weight concepts of different features in characterizing different classes. Before describing the block diagram of the system, let us explain the membership functions considered for assigning membership values corresponding to the *feature subdomains* representing the property sets *small*, *medium* and *high* of an individual feature axis. ♣

3.3.2 Membership functions

It is known that the data in *linguistic form* contains summarized information. So for the recognition purpose, the primary properties *small*, *medium* and *high* of a feature which reflect its linguistic information must be defined in such a way that they represent properly a set of data and provide a summarized information on them. Keeping this in mind, the sets *small*, *medium* and *high* have all been represented by π functions [18,67]. The functional form of a standard π-function is given below.

$$\pi(x;\beta,\gamma) = \begin{cases} S(x;\gamma-\beta,\gamma-\beta/2,\gamma) & for\ x \leq \gamma \\ 1 - S(x;\gamma,\gamma+\beta/2,\gamma+\beta) & for\ x \geq \gamma \end{cases} \tag{3.7}$$

$$\text{where} \quad S(x;\alpha,\beta,\gamma) = \begin{cases} 0 & for\ x \leq \alpha \\ 2[(x-\alpha)/(\gamma-\alpha)]^2 & for\ \alpha \leq x \leq \beta \\ 1 - 2[(x-\gamma)/(\gamma-\alpha)]^2 & for\ \beta \leq x \leq \gamma \\ 1 & for\ x \geq \gamma \end{cases} \tag{3.8}$$

where β is the crossover point and in general, $\beta = \frac{\alpha+\gamma}{2}$. Note that the π function is a combination of S and (1-S) functions. S function denotes the compatibility function for the set "x is large" whereas, π function denotes the compatibility function for the set "x is γ". Fig. 3.2 shows the general structures of S and π functions.

Fig. 3.3 shows the coexistence structure of the various membership functions under one particular feature as used in our recognition system. Depending on the problem, different values for β and γ are assigned to represent the primary property sets with respect to each feature.

It is to be noted that consideration of π function as the compatibility functions for *small* and *high* is a major deviation from the standard approach in the fuzzy set theory where these are usually represented by (1-S) and S functions respectively. The reason for this deviation is explained below.

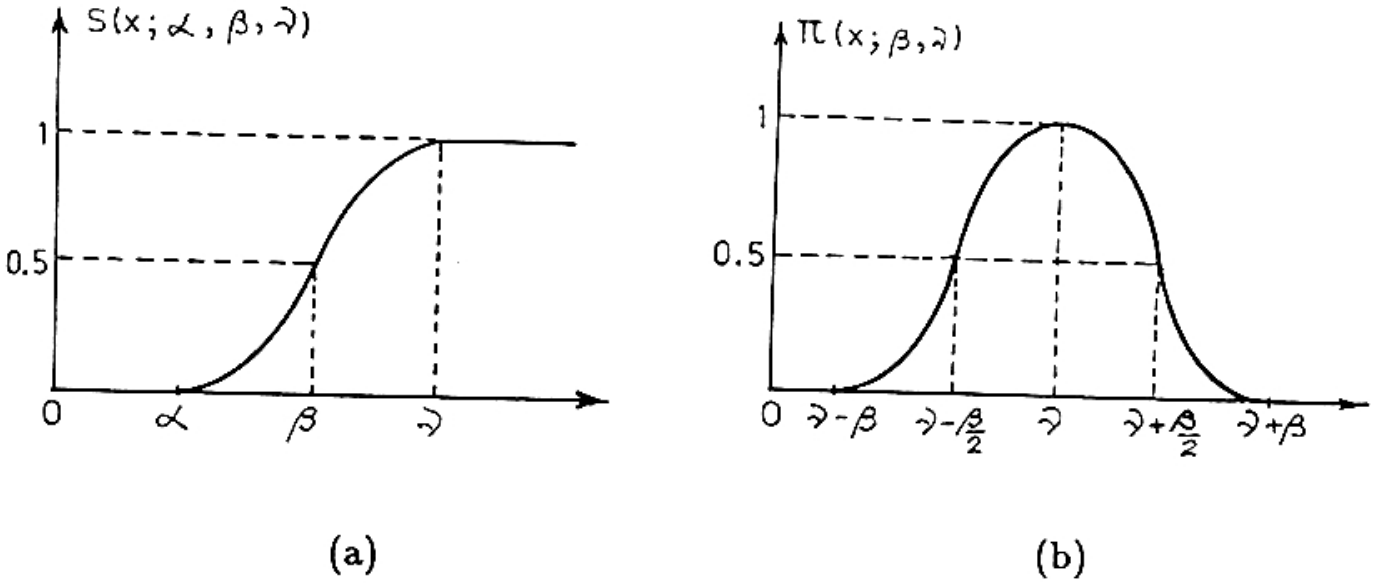


Figure 3.2 : (a) S function and (b) π function.

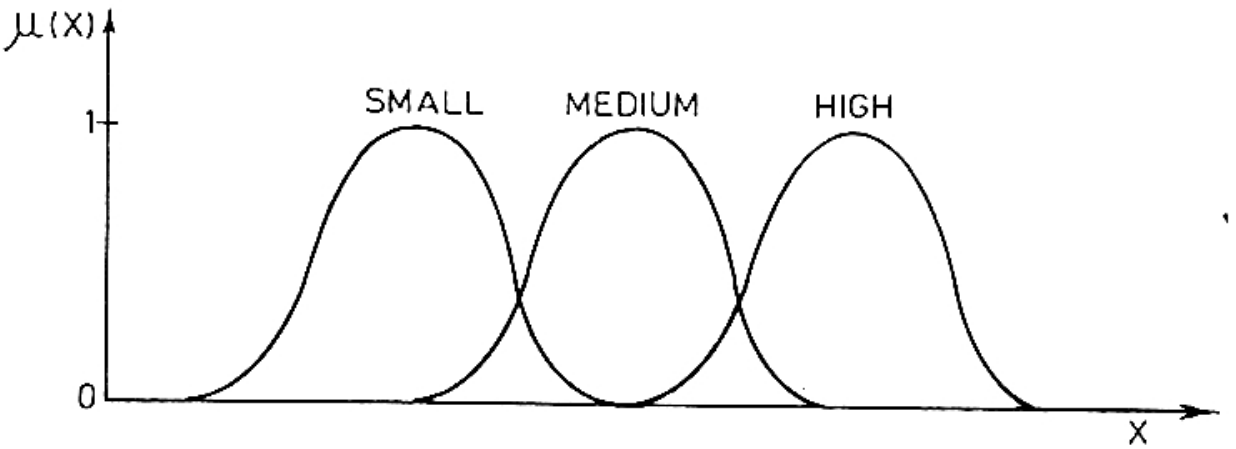


Figure 3.3 : Coexistence structure of the membership functions for the properties *small*, *medium* and *high*.

In fuzzy set theory, the set *very small* is a subset of the set labeled *small* i.e., *very small* possesses a relatively high membership value in the class *small*. But this is not always intuitively appealing. For example, consider the problem of scoring in a class. We usually use the term *very bad* to indicate marks between 10% to 30%, say, and *bad* to indicate marks between 20% to 40%, say. So by *bad* and *very bad* humans highlight two different data sets although one overlaps other. Therefore, it is reasonable here to decrease the membership value for "very small" in the class *small*. To incorporate this view, we have considered the membership function for *small* as a π function, instead of (1-S) function. Similar argument holds for the class *high*. As an example, by *tall*, it indicates the height roughly $5.5ft$ to $6.5ft$, say, but by *very tall*, it indicates the height between $6ft$ to $7ft$, say. ♣

3.3.3 Block diagram

The block diagram of the multivalued recognition system is shown in Fig. 3.4. It consists of two sections, namely *learning* and *fuzzy processor*. The learning section uses the training samples and outputs a relational matrix and weight matrices to the *processor*. The fuzzy processor uses the relational matrix and the weight matrices to give a natural decision output regarding the classes from which the pattern X may come. Note that both the sections are having the block linguistic feature extractor (LFE), which takes the input pattern X and outputs *characteristic vectors*. During learning, it outputs a *characteristic vector* $CV(X)$ corresponding to the input X from the training samples. On the other hand, in the processor it works as a weighted LFE so that it outputs M *characteristic vectors* ($CV_j(X)$'s) with the help of weight matrices corresponding to an unknown pattern X.

The relational matrix R denotes the compatibility for the various pattern classes corresponding to the elements of PCV. Relational matrix is obtained

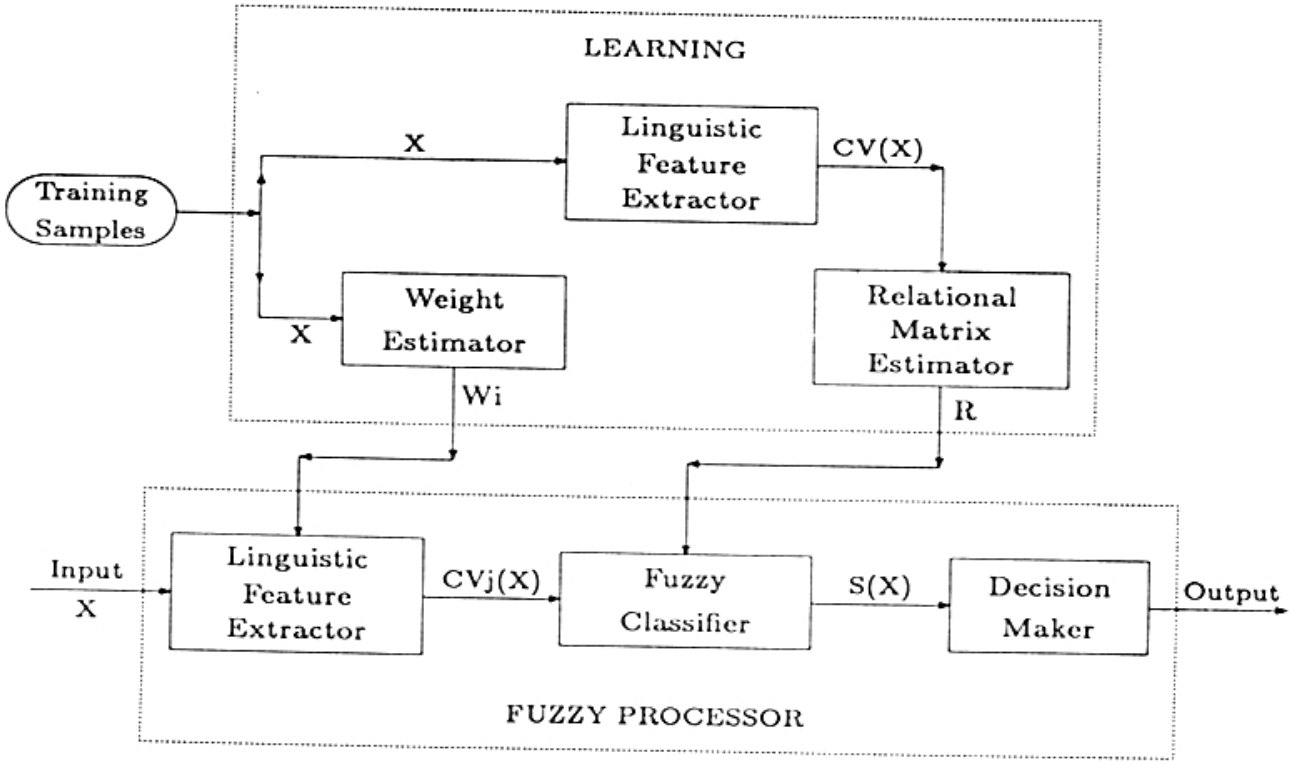


Figure 3.4 : Block diagram of the multivalued recognition system.

based on the training samples. Each column of R corresponds to a class and each row of that column denotes the degree to which a class should be characterized (based on training samples) by the corresponding component of PCV.

As mentioned earlier, all the features and even all the properties may not be of equal importance in characterizing a class. So it is reasonable to provide weights corresponding to feature properties in order to determine the *characteristic vectors* $CV_j(X)$ $(j = 1, 2, \ldots, M)$ of X.

The $CV_j(X)$'s are the input to the fuzzy classifier. It uses the relational matrix R to determine the degree of belonging the unknown pattern X to different pattern classes. This is done using the composition rule of inference taken between $CV_j(X)$'s and R. The fuzzy classifier therefore gives as output

a *class similarity vector*

$$S(X) = (s_1(X), s_2(X), \ldots, s_j(X), \ldots, s_M(X))' \tag{3.9}$$

where $s_j(X)$ denotes the degree of similarity of a pattern X to the jth class.

Ambiguity (uncertainty) in the fuzzy decision, provided by $S(X)$, is then determined by computing CF (*Confidence or Certainty Factor*). The higher the value of CF, the greater is the contrast between the $\max_j\{s_j\}$ and the remaining, and hence the stronger is the decision. Depending on the value of CF, the final output of the recognition system is given in linguistic form.

The linguistic recognition system described here is referred to as multivalued because it normally gives multiple output choices for classes with their preferences. It may also be viewed as a generalized classifier providing natural (fuzzy and/or hard) output from both fuzzy and deterministic input. The operations in different blocks of Fig. 3.4 are discussed in the following sections. ♠

3.4 Linguistic Feature Extractor (LFE)

The input pattern of the recognition system as mentioned earlier can be any of the four forms, namely *quantitative form*, *linguistic form*, *mixed form* and *set form*. First of all, each feature information is considered separately to determine its membership value corresponding to the properties *small*, *medium* and *high*. The way it has been done for various forms of input is furnished below.

3.4.1 Quantitative form

The information in this form is in exact numerical terms, like "F_1 is 500". The membership values of an input information in this form for different

linguistic feature property sets (i.e., *small*, *medium*, *high*) are determined directly from their corresponding membership functions as

$$\mu_{ip}(F_i \text{ is } v) = \pi(v; \beta, \gamma) \tag{3.10}$$

where $\mu_{ip}(.)$ represents the membership value corresponding to a property set p ($\in \{small, medium, high\}$) of the feature F_i. Note that the parameter values of the π functions are assigned before hand according to the nature of various features. •

3.4.2 Linguistic form

Here the input pattern information is provided in terms of linguistic variables and/or hedges e.g., "F_1 is small" or "F_1 is more or less medium" or "F_1 is very high" *etc.* A short description on the linguistic terms and hedges is provided in section 3.2.1.

The linguistic variables, under consideration, are the three primary terms *small*, *medium* and *high*. When the input statement contains only the primary terms, its membership values for the sets *small*, *medium* and *high* are assigned as

$$\begin{aligned} \text{small} &\equiv \{\ 0.8/small,\ \ 0.2/medium,\ \ 0.0/high\ \} \\ \text{medium} &\equiv \{\ 0.2/small,\ \ 0.8/medium,\ \ 0.2/high\ \} \\ \text{high} &\equiv \{\ 0.0/small,\ \ 0.2/medium,\ \ 0.8/high\ \} \end{aligned} \tag{3.11}$$

For the statements with linguistic hedges, say *very*, *more or less*, *slightly* *etc.*, we proceed as follows. When the statement contains "very small', its membership value for the property *small*, as explained in section 3.3.2, will be decreased and the membership value for the property *medium* will further be decreased. On the other hand, by "more or less small", the membership value for the set *small* will be decreased, but the membership value for *medium* will be increased.

Similar is the case with "high" also, but the converse is true for "medium". Here we will increase the membership value for *medium* and decrease the membership value for both *small* and *high*, when the input statement contains "very medium". Similarly, by "more or less medium" the membership value for *medium* is decreased and the membership values for both *small* and *high* are increased.

The modifications of the membership values may be carried out in a similar manner for other possible linguistic hedges. To increase the membership value, the dilation operation DIL [Eq. (3.1)] is used and to decrease the membership value, the concentration operation CON [Eq. (3.1)] is used. •

3.4.3 Mixed form

The information is provided in the form of a mixture of linguistic hedges and quantitative terms such as, "F_1 is *about* 400" or "F_1 is *more or less* 400".

Since the linguistic term increases the impreciseness in the information, the membership values of the statement as a whole, for different primary properties, should be lower than that of the quantitative term alone. The amount of decrease will be determined according to the linguistic hedges. As an example, for the information "F_i is about v", the membership value is decreased from that of the information "F_i is v" as

$$\mu_{ip}(F_i \text{ is about } v) = \{\mu_{ip}(v)\}^{1.25} \tag{3.12}$$

where $\mu_{ip}(.)$ represents the membership value corresponding to a property set p ($\in \{small, medium, high\}$) of the feature F_i.

The aforementioned modification of the membership value will be reflected in the *confidence factor* (CF) of the classifier output. •

3.4.4 Set form

Like the *mixed form* the information here is also a mixture of linguistic hedges and quantitative terms. The only difference lies with the nature of linguistic hedges. The linguistic hedges considered in the *set form* are *less than, more than, between etc.*, such that the data reflected is a set and at least one boundary of the data set is known. The examples of this form are "F_1 is *less than* 400", or "F_1 is *more than* 400" or "F_1 is *between* 400 and 500".

First the membership values for the various primary properties with respect to the quantitative terms (e.g., 400) are calculated. We know that the compatibility functions considered for the primary sets are all standard π functions of the form $\pi(x, \beta, \gamma)$ where γ is the ideal point i.e., the point where the membership value is 1.0. The membership value of the information "F_i is less than v" is obtained by modifying the membership value of the information "F_i is v" as

$$\mu_{ip}(F_i \text{ is } less\ than\ v) = \begin{cases} [\mu_{ip}(v)]^{1/2} & if\ v \geq \gamma\ \&\ \mu_{ip}(v) > 0 \\ [\mu_{ip}(v)]^2 & if\ v \leq \gamma\ \&\ \mu_{ip}(v) > 0 \\ 0.2 & if\ v > \gamma\ \&\ \mu_{ip}(v) = 0 \\ 0 & Otherwise \end{cases} \tag{3.13}$$

corresponding to a primary property p ($\in \{small, medium, high\}$) of the feature F_i.

As an example, consider the statement "Ram is less than 25 years old". This means Ram's age is more likely to be around 20 years. Therefore if $25 > \gamma$ for a primary property set, the μ value of the statement "less than 25" will be higher than that of $\mu(25)$, because "around 20 years" is more towards the value γ. The reverse is true for $25 < \gamma$. Eq. (3.13) reflects these facts.

For the linguistic hedges like *greater than* or *more than* where exactly

one boundary of the reflected data set is known, the membership values are similarly decided.

Now there may be information with statements using the connectors *and, but etc.* (e.g., "F_i is less than v_1 and/but greater than v_2") where the reflected data set are both way bounded. In such cases, initially the two statements are considered separately and correspondingly two membership values (as explained earlier) are obtained. The resultant membership value is computed from their geometric mean i.e.,

$$\mu_{ig}(F_i \text{ is } less\ than\ v_1\ and\ greater\ than\ v_2)$$

$$\doteq [\mu_{ip}(F_i \text{ is } less\ than\ v_1) \times \mu_{ip}(F_i \text{ is } greater\ than\ v_2)]^{1/2} \qquad (3.14)$$

If there is a statement like "F_i is between v_1 and v_2" which is equivalent to the statement "F_i is greater than v_1 and less than v_2", then we proceed as in the previous case. Hence, to put it concisely, rules for the calculation of membership values can be found out if the provided information is in the *set form.* •

It may happen that the information about some particular feature is fully missing. In this case it is reasonable to assign some low (say, 0.2) membership value to all the primary linguistic properties i.e.,

$$\text{no information} \equiv (0.2/small,\quad 0.2/medium,\quad 0.2/high) \qquad (3.15)$$

The aforementioned discussion shows a way how the impreciseness/uncertainty in the input feature information can be handled by providing/modifying membership value heuristically to a great extent. The logic behind the assignment of membership value is also intuitively appealing.

3.4.5 Characteristic vectors (CV)

After obtaining the membership values of features for the properties *small*, *medium* and *high*, class memberships of a pattern X ($CV_j(X)$ $j = 1, 2, \ldots, M$ [Eq. (3.6)]) corresponding to all property combinations (i.e., the elements of PCV) are then computed. Let us consider the hth component of PCV which represents the property combination

$$\left(p_1^h,\ p_2^h,\ \ldots,\ p_N^h\right) \tag{3.16}$$

where depending on the value of h, p_i^h denotes one of the primary properties *small*, *medium*, *high* in ith feature F_i and represents the set "F_i is p_i^h". So the hth element of the characteristic vector, $CV_j(X)$, i.e., the membership value corresponding to the hth element of the PCV, is defined as

$$cv_{j_h}(X) = \sum_{i=1}^{N} \left(\mu_{i_h}(X) \times w_{i_{jh}}\right) \tag{3.17}$$

where $\mu_{i_h}(X)$ is the membership value of X to the set "F_i is p_i^h". $w_{i_{jh}}$ is the weight of the ith feature for the jth pattern class corresponding to the hth *property combination (i.e., hth element of PCV) and its computational aspect* is explained in section 3.5.1. Note that $cv_{j_h}(X)$ [Eq. (3.17] is the weighted arithmetic mean of the membership values of the individual feature properties. Output of LFE will have M *characteristic vectors* ($CV_j(X)$, $j = 1, 2, \ldots, M$) and each vector contains 3^N elements. ♣

As mentioned in section 3.3.3, the LFE in the processor works basically as an weighted LFE, and that is what has been described in this section to result in M *characteristic vectors* $CV_j(X)$. On the other hand, the LFE in the learning section does not take weight matrices into consideration and thus provides a single output vector irrespective of the classes from the training samples. ♠

3.5 Learning

This section takes input from the training samples and estimates weight matrices and a relational matrix. It has three blocks, namely *linguistic feature extractor (LFE)*, *weight estimator* and *relational matrix estimator*. The LFE determines a single *characteristic vector* $CV(X)$ corresponding to the sample X of the training patterns. The weight matrices and the relational matrix are similarly estimated from the training samples as follows.

3.5.1 Weight matrices

It is the fact that all the features and hence the properties are not of equal importance to determine a pattern class. For example, for the first element of PCV, all the properties (i.e., "F_1 is *small*", "F_2 is *small*", ..., "F_N is *small*") may not have equal importance in characterizing a class. So it leads us to define some weights corresponding to various feature properties to find the membership value of a pattern, corresponding to the elements in PCV, for a class. For N features and M class problem, we determine N weight matrices of order $3^N \times M$ where rows stand for elements of PCV, columns stand for different pattern classes, and each matrix corresponds to a particular feature. N such matrices can therefore be represented as $W_1, W_2, \ldots, W_N$, where the ith matrix W_i, corresponds to the ith feature i.e., it represents the weights (importance) of the ith feature information to determine the membership value of a pattern to the various classes corresponding to the entries in PCV.

In order to illustrate this, let us assume that the (j,h)th elements (i.e., the elements corresponding to the class C_j and the hth *property combination* $(p_1^h, p_2^h, \ldots, p_N^h)$) of the weight matrices $W_1, W_2, \ldots, W_N$ are $\omega_{1_{jh}}, \omega_{2_{jh}}, \ldots, \omega_{N_{jh}}$ respectively. Then $\omega_{1_{jh}}$ represents the weight of the property "F_1 is p_1^h", $\omega_{2_{jh}}$ represents the weight of the property "F_2 is p_2^h", ..., $\omega_{N_{jh}}$ represents the weight of the property "F_N is p_N^h", to determine the hth element of the

characteristic vector corresponding to the class C_j. Then N weight matrices may therefore be viewed as a single matrix of order $3^N \times M$, where the (j,h)th element is $(\omega_{1_{jh}}, \omega_{2_{jh}}, \ldots, \omega_{N_{jh}})$.

Determination of the weight matrices : Weight matrices $W_1, W_2, \ldots, W_N$ are determined from training samples. Consider the ith element of PCV [Eq. (3.16)]. Let n_j be the number of training samples in the jth pattern class (C_j) and X_{j_k} denote the kth training sample of the class C_j. Find

$$\theta_{i_{jh}} = \sum_{k=1}^{n_j} \Delta_{i_{jh_k}} \ , \qquad i = 1, 2, \ldots, N \tag{3.18}$$

where

$$\Delta_{i_{jh_k}} = (1 - \mu_{i_h}(X_{j_k})) \ , \qquad k = 1, 2, \ldots, n_j.$$

$\mu_{i_h}(X_{j_k})$ is the membership value of X_{j_k} with respect to the set "F_i is p_i^h"; $\Delta_{i_{jh_k}}$ denotes the absolute deviation of $\mu_{i_h}(X_{j_k})$ from 1 (i.e., from ideal membership value) and $\theta_{i_{jh}}$ is the sum of $\Delta_{i_{jh_k}}$ for all n_j samples.

So, $w_{i_{jh}}$, the entry corresponding to the jth row (i.e., jth pattern class) and hth column (i.e., hth element of PCV) of the ith weight matrix W_i (i.e., the weight matrix for ith feature), is

$$w_{i_{jh}} = \frac{\frac{1}{\theta_{i_{jh}}}}{\sum_{i=1}^{N} \frac{1}{\theta_{i_{jh}}}} \tag{3.19}$$

Varying i, j and h, all the entries of the weight matrices $W_1, W_2, \ldots, W_N$ are determined. ♣

3.5.2 Relational matrix

The relational matrix R denotes the compatibility of the various pattern classes corresponding to the elements of PCV. The order of R is $3^N \times M$. Here columns correspond to the pattern classes and rows correspond to the

property combinations i.e., the components of PCV. In other word, we can write for $N = 2$ and $M = 3$

$(F_1\ ,\ F_2)$	C_1	C_2	C_3
$(small\ ,\ small)$	μ_{11}	μ_{12}	μ_{13}
$(small\ ,\ medium)$	μ_{21}	μ_{22}	μ_{23}
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$(high\ ,\ high)$	μ_{91}	μ_{92}	μ_{93}

If a pattern has the property "F_1 is *small* and F_2 is *small*", then μ_{11} will denote the membership value of the pattern to be in the class C_1 based on that property. Similar is the case for all other entries of the relational matrix. The concept of the relational matrix has already been provided in section 3.2.2.

Determination of the relational matrix : The relational matrix R is determined from training samples. Consider here also the hth element of PCV [Eq. (3.16)]. Let $cv_h(X_{j_k})$ be the membership value (independent of any class) of X_{j_k} corresponding to the hth element of PCV. So

$$cv_h(X_{j_k}) = \frac{1}{N}\sum_{i=1}^{N}\mu_{i_h}(X_{j_k})\ , \qquad k = 1, 2, \ldots, n_j. \tag{3.20}$$

Note that the weight matrices have not been considered here.

So r_{hj}, the (h,j)th $(h = 1, 2, \ldots, 3^N,\ j = 1, 2, \ldots, M)$ element of the relational matrix R, is

$$r_{hj} = \frac{1}{n_j}\sum_{k=1}^{n_j} cv_h(X_{j_k}) \tag{3.21}$$

where n_j is the number of training samples taken from jth pattern class. ♣

Note that, in order to estimate the weight matrices and the relational

matrix, the training samples which make non-zero contribution toward the components of PCV are only considered. ♠

3.6 Fuzzy Processor

It consists of three blocks, namely *linguistic feature extractor (LFE)*, *fuzzy classifier* and *decision maker*. The LFE gives characteristic vectors $\tilde{CV}_j(X)$ $(j = 1, 2, \ldots, M)$ as output for an input X. Its function has already been described in section 3.4. The $CV_j(X)$'s are used to determine the *class similarity vector* $S(X)$ which denotes the degree of similarity of the input pattern X to the various pattern classes. The *decision maker* block gives a natural output along with its degree of certainty based on the *similarity vector* $S(X)$.

3.6.1 Fuzzy classifier

The classifier incorporates the *max-min composition rule of inference* (described in section 3.2.2) in a slightly modified manner. Here the *min* operator (which is an connective operator) of *max-min* operation in the Zadeh's composition rule [Eq. (3.3)] has been replaced by *geometric mean* (GM) (which gives collective information). The *class similarity vector* $S(X)$ [Eq. (3.9)] is determined as

$$\begin{aligned} s_j(X) &= CV_j(X) \circ R \\ &= \max_{h=1,2,\ldots,3^N} \{cv_{j_h}(X) \times r_{hj}\}^{\frac{1}{2}} \\ & \qquad\qquad j = 1, 2, \ldots, M \end{aligned} \tag{3.22}$$

where $cv_{j_h}(X)$ is the hth element of the *characteristic vector* $CV_j(X)$ for the unknown pattern X and r_{hj} is the (h, j)th entry of the relational matrix R.

So a *class similarity vector* $S(X)$ (with M entries in terms of fuzzy membership value for different pattern classes) comes out from the classifier block for an input pattern X.

Example **3.5 :** Suppose we have only one feature, say F and three pattern classes, say C_1, C_2 and C_3. Assume that the weights of all the feature properties are same for these classes i.e., there is a unique *characteristic vector.* Let the *characteristic vector* be $CV(X) = [0.7, 0.3, 0.0]$ for an input pattern X and the relational matrix R be

F	C_1	C_2	C_3
small	0.2	0.7	0.4
medium	0.8	0.1	0.1
high	0.4	0.3	0.8

The *similarity vector* S(X) will then be

$$\begin{aligned} S(X) &= CV(X) \circ R \\ &= [0.49 \quad 0.70 \quad 0.53] \end{aligned}$$

From $S(X)$ it appears that the pattern X is inclined to the class C_2. ♣

3.6.2 Decision maker

The *class similarity vector* $S(X)$ is analysed here. In the way of analysis, it finds the *confidence factor* (CF), and depending on the value of CF, it gives a linguistic output which is indeed the output of the recognition system.

Confidence factor (CF) : Now we need to find out some measure which will reflect the amount of difficulty in arriving at a single output by minimizing ambiguity in the *similarity vector* $S(X)$. It is to be mentioned here that the impreciseness in the input information has been reflected in the *characteristic vectors* $CV_j(X)$. The concept of difficulties in arriving at a single output from

the $S(X)$ will be clear from the following set of *similarity vectors*. Consider the following four output *similarity vectors* with three pattern classes

$$\begin{aligned} S^1(X) &= [0.9 \;\; 0.2 \;\; 0.0] \\ S^2(X) &= [0.9 \;\; 0.5 \;\; 0.2] \\ S^3(X) &= [0.9 \;\; 0.8 \;\; 0.2] \\ S^4(X) &= [0.9 \;\; 0.9 \;\; 0.7] \end{aligned}$$

It is clear from these output vectors that the difficulty in deciding the class C_1 is increasing in the order $S^1(X)$, $S^2(X)$, $S^3(X)$ and $S^4(X)$. It can be said that the difficulties in assigning a particular pattern class depends not only on the highest entry in the similarity vector $S(X)$ but also on its difference with other entries in $S(X)$. Based on this, a measurement of *confidence factor* (CF) is defined as

$$CF = \frac{1}{2}\left[\{s_{max}(X)\}^{f_{max}} + \frac{1}{(M-1)}\sum_{j=1}^{M}\{s_{max}(X) - s_j(X)\}\right] \qquad (3.23)$$

$$0 \leq CF \leq 1$$

where $s_j(X)$ is jth entry of the $S(X)$; $s_{max}(X)$ is the highest entry in $S(X)$; f_{max} is the frequency (number of occurrence) of $s_{max}(X)$ in $S(X)$; and M is the number of pattern classes. From Eq. (3.23) it is seen that the higher the value of CF, the lower is the difficulty in deciding a class and hence the greater is the degree of certainty of the output decision. For the aforementioned example, the CF values are 0.850, 0.725, 0.650, 0.505 for $S^1(X)$, $S^2(X)$, $S^3(X)$, $S^4(X)$ respectively. ♣

3.6.3 Output

Based on the value of CF, the system makes the following decisions in order to give output in *linguistic (natural) form.*

(i) The assigned pattern class is *definitely true* if CF is in $[0.8, 1.0]$ and there is no *second choice* of pattern class.

(ii) The assigned pattern class is *true* and there is a *second choice* of pattern class if CF lies in $[0.6, 0.8)$.

(iii) The assigned pattern class is *more or less true* and there is a *second choice* of pattern class if CF lies in $[0.4, 0.6)$.

(iv) The assigned pattern class is *not false* for CF lying in $[0.1, 0.4)$ and there is no *second choice* of pattern class.

(v) The classifier is *unable to recognize* the pattern class from the given input information if CF lies in $[0.0, 0.1)$. •

To give a *second choice* of pattern class, we find the *confidence factor* (CF_2) for the second highest entry in the *similarity vector* $S(X)$ by the same formula as in Eq. (3.23). We will give the second choice of the pattern class, if $CF_2 \geq 0.2$.

Some typical output forms are:

(i) This is *very likely* to be C_1 $(CF = 0.89)$.

(ii) This is *likely* to be C_1 $(CF = 0.72)$ but *not very unlikely* to be C_2 $(CF = 0.32)$.

(iii) This is *more or less likely* to be C_1 $(CF = 0.45)$ but *not unlikely* to be C_2 $(CF = 0.35)$.

(iv) This is *not unlikely* to be C_1 $(CF = 0.28)$.

(v) Sorry, it is *not recognizable*.

There may be some cases where there are multiple entry with highest value in the similarity vector $S(X)$. In that case, there will not be a *second choice* of pattern class. The form of the output here is

(vi) This is likely to be C_1 or C_2 $(CF = 0.52)$. ♠

3.7 Implementation and Results

The aforementioned algorithm was implemented on a set of Indian Telugu vowel sounds (/ð/, /a/, /i/, /u/, /e/, /o/). The typical feature space of the six vowel classes (having ill-defined boundaries) consisting of 871 deterministic data corresponding to F_1 and F_2 has been shown in Fig. 2.9 (chapter 2). Recall that F_1 and F_2 denote the first and second vowel formant frequencies respectively.

The testing data set consists of 871 deterministic and 102 imprecise data. These imprecise (F_1, F_2) information were coded to various *linguistic forms*, *viz.* (*small*, *more or less medium*), (700, *between 1800 to 2200*), (*about 600*, *high*), (*small*, –) *etc.* by trained personnel. It is to be mentioned here that these imprecise samples were ignored in the earlier work [26,139,235,236] which were incapable of handling them.

The compatibility functions assigned corresponding to F_1 and F_2 are as follows:

For F_1 feature:

$$\begin{aligned} \mu_{1_{small}}(x) &= \pi(x;\ 200,\ 325) \\ \mu_{1_{medium}}(x) &= \pi(x;\ 200,\ 525) \\ \mu_{1_{high}}(x) &= \pi(x; 200, 725). \end{aligned}$$

For F_2 feature:

$$\begin{aligned} \mu_{2_{small}}(x) &= \pi(x;\ 450,\ 1000) \\ \mu_{2_{medium}}(x) &= \pi(x;\ 450,\ 1600) \\ \mu_{2_{high}}(x) &= \pi(x; 450, 2200). \end{aligned}$$

The overall recognition score for various sizes of samples is shown in Fig. 3.5 by divided-bar diagram. The recognition score shown are obtained by averaging those corresponding to five different training sets of a specified size. The individual recognition scores are shown, as an illustration, in Fig.

3.6 only for 10% training samples. The recognition scores are grouped in four categories, namely *first correct choice*, *combined correct choice*, *second correct choice* and *fully wrong choice*. Here the *first correct choice set* includes those samples for which classifier's *first choice* agrees with their actual class. *Combined correct choice set* includes those samples while one of the *combined choice* is correct. The *second correct choice set* includes those samples, for which their *second choice* corresponds to the actual vowel class. Samples not falling under the aforementioned categories are termed as *misclassified* or *fully wrong choice*.

A list of some typical output is given below for illustration.

- (*300, 900*) : This is most likely to be $/u/$ $(CF = 0.82)$.
- (*700, 1300*) : This is more or less likely to be $/a/$ $(CF = 0.49)$ but not unlikely to be $/e/$ $(CF = 0.32)$.
- (*900, 1400*) : This is not very unlikely to be $/a/$ $(CF = 0.25)$.
- (*250, 1550*) : Sorry, it is difficult to recognize.
- (*high, more or less small*) : This is more or less likely to be $/a/$ $(CF = 0.52)$.
- (*between 500 to 600, 1600*) : This is likely to be $/e/$ $(CF = 0.63)$, but not unlikely to be $/\delta/$ $(CF = 0.45)$.
- (*about 350, —*) : This is likely to be either $/i/$ or $/u/$ $(CF = 0.48)$.

These linguistic outputs confirm the vowel diagram in Fig. 2.9. Note that for the input information (250, 1550), the system is unable to recognize the vowel, as this information has a very insignificant similarity with the vowel classes. This has been regarded as misclassified while computing the recognition score of the system. Further, for the input information (*about*

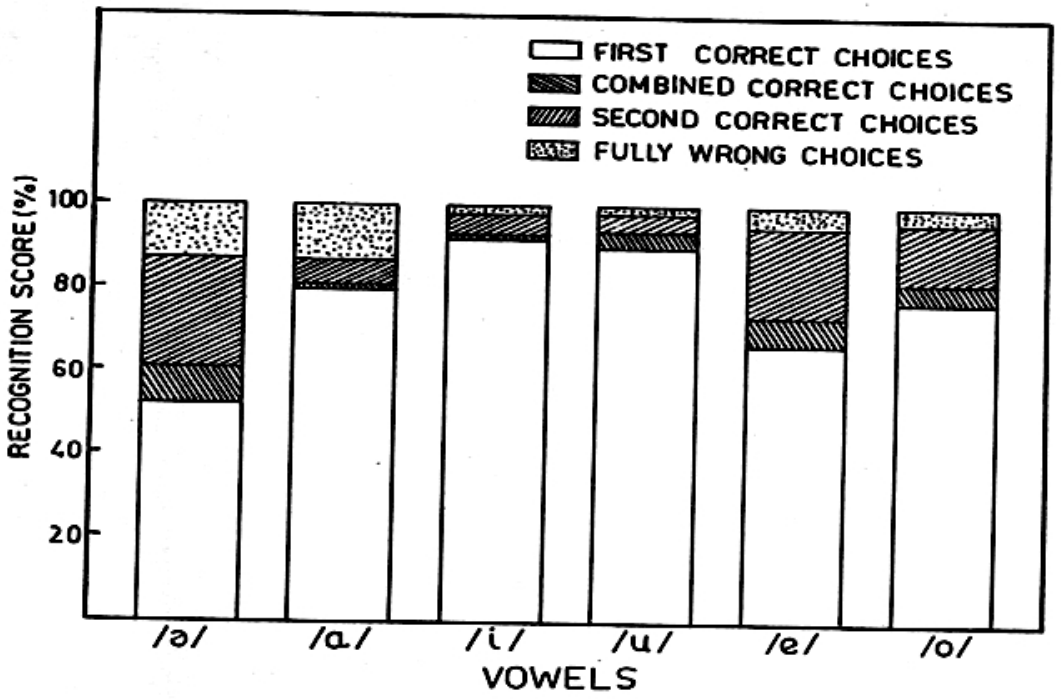


Figure 3.5 : Pie diagram showing the overall recognition score for different sizes of training samples.

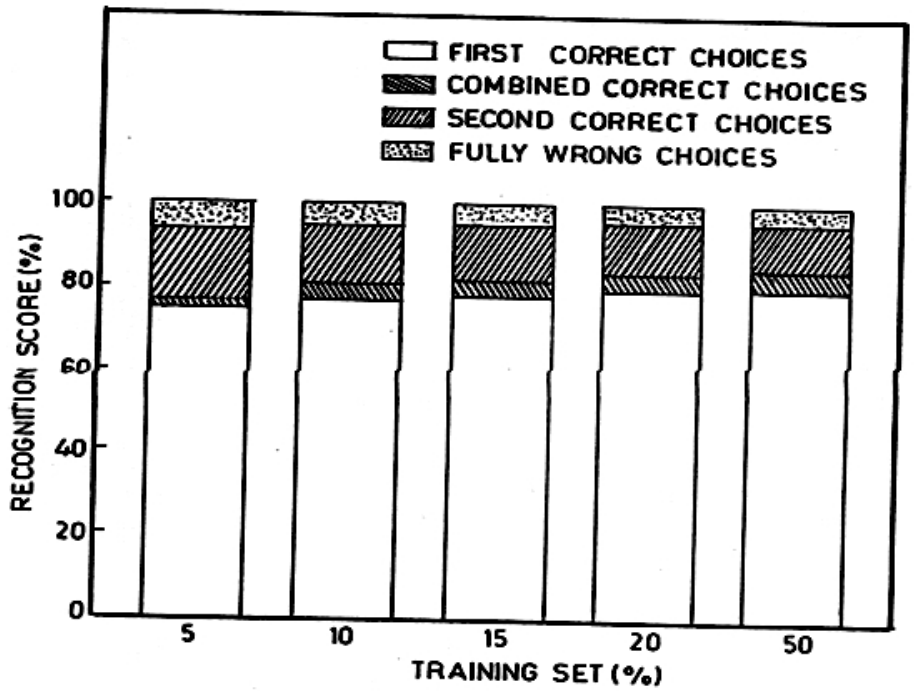


Figure 3.6 : Pie diagram showing the recognition score of the individual vowels for 10% training samples.

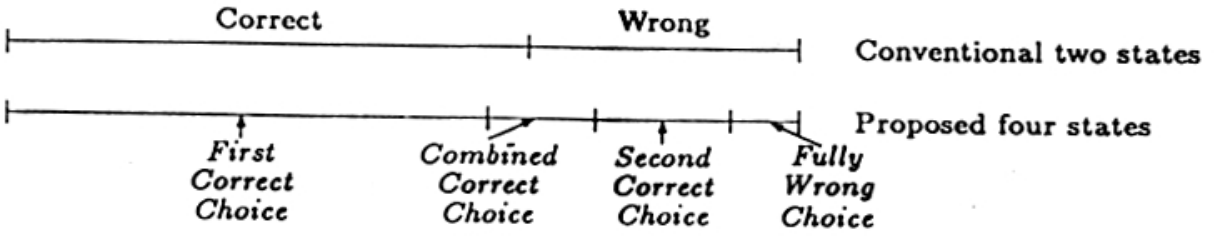


Figure 3.7 : Conventional two state versus proposed four state output.

350, —) (here "—" indicates that there is no information about F_2 feature), the system finds some similarity of this information with the vowel classes */i/* and */u/*, on the basis of the F_1 feature information only.

It is observed that the confusion in recognizing a sample considering the *first choice* lies, in general, only with the neighboring classes constituting a vowel triangle. The similar findings were also obtained with the previous investigations [26,139,235,236], considering deterministic input/output. The overall recognition score corresponding to *first choice* is quite satisfactory considering the fact that it accepts approximate feature information and the information relates only to F_1 and F_2. Feature F_3, which was incorporated in [26,139,235,236], has not been considered here.

It is to be noted that the proposed linguistic recognition system provides natural output in four states. Thus, it has very low ($\approx 5\%$) misclassification rate as compared to those ($\approx 20\%$) in [26,139,235,236] which give two state hard decision like *correct* or *wrong*. (For computing the misclassification rate of the proposed system, we refer only to the category *fully wrong choice*.) This is explained in Fig. 3.7 where the category *wrong*, in two state conventional system, has been decomposed into three categories in the proposed system. Because of the flexibility, the proposed system has therefore a provision of improving its efficiency significantly by incorporating *combined* and *second choices* under the control of a supervisory scheme.

The result of a fuzzy set theoretic decision making system usually depends, to some extent, on the choice of membership functions. We have considered here membership functions of *small*, *medium*, *high* to be symmetric and occupying equally the entire feature range of the training samples. Moreover, significant overlapping has been incorporated among them so that the obtained results become more meaningful and less susceptible to the choice of membership functions.

It is further to be mentioned here that the system has been programmed considering only three primary properties *small*, *medium* and *high*. Incorporation of additional subset property (e.g., *very small*, *more or less small etc.*) will definitely improve the system performance because it will lead to reduction of the impreciseness in input linguistic information by generating more subregions in Fig. 3.1. For example, if we include another property, say *very small*, it will result in 4^N *space subdomains*. Again, the primary properties considered need not necessarily be the same for all the features.

In the proposed scheme with N features and 3 linguistic properties corresponding to each feature, the entire feature space is decomposed in 3^N overlapping *space subdomains*. If the value of N is large, the number of *space subdomains* becomes very high and this leads to serious practical difficulty in implementing/executing the proposed scheme. Again, increasing the number of linguistic properties (hedges) will enhance this problem further for large N. ♠

Chapter 4

MULTIVALUED RECOGNITION SYSTEM USING GEOMETRIC STRUCTURE BASED DECOMPOSITION

Contents

4.1 Introduction

In the previous chapter, a recognition system has been described which is capable of handling various imprecise/ambiguous input patterns either in *quantitative form* or *linguistic form* or *mixed form* or *set form* and of providing the output decision in linguistic (natural) form. The system has considered imprecise data both in its training and testing phases and used three linguistically phrased property sets (namely, *small*, *medium* and *high*) for the input patterns so that each feature information can be viewed to have these properties to some degree. As a result, the entire feature space is decomposed into 3^N (N denotes the number of features) overlapping *space subdomains* representing various property combinations. It is to be noticed that the structures and relative positions of the pattern classes were not taken into account in the aforementioned decomposition process.

In the present chapter, another recognition system has been described for providing improved performance. Initially, each individual feature range is divided into some *feature subdomains* depending on the geometric structure (as described in chapter 2) and the relative position of the pattern classes found in the training samples. To handle the impreciseness of the input feature information and to incorporate the portions possibly uncovered by the training samples, each of the *feature subdomains* is extended to some extent using triangular membership functions. As a result, the entire feature range is decomposed into few overlapping *space subdomains*. Because of the use of the concept of the extended regions, the output forms are different to some extent from those used in the previous chapter. The system is trained with deterministic data, but in the testing phase it can handle input data in either of the four forms (*quantitative* or *linguistic* or *mixed* or *set form*). There is no concept of weight matrices as used in the recognition system in chapter 3.

A relational matrix corresponding to the *space subdomains* and the pattern classes has been considered in the Zadeh's compositional rule of inference [72] (described in section 3.2.2) in order to recognize the samples. The recognition system provides multiple output choices for classes with their preferences and accordingly, it considers four output forms namely, *single choice*, *combined choice*, *first-second choice* and *null choice*. The final output decision is given in *linguistic form* using a confidence factor. The effectiveness of the system has been demonstrated on some artificially generated pattern sets and also on the real life speech data.

In section 4.2, the basic features and the block diagram of the recognition system are provided. The description of different blocks is furnished in sections 4.3 and 4.4. Results are discussed in section 4.5. ♠

4.2 Multivalued Recognition System

4.2.1 Concept of representing a pattern

For describing the recognition system, let us consider an M class (C_1, C_2, ..., C_j, ..., C_M) and N feature (F_1, F_2, ..., F_i, ..., F_N) problem. The system assumes that every pattern class is a collection of nearly parallelepiped shaped sets. Thus, initially the training sample set of every pattern class is divided into few groups of nearly parallelepiped shapes (as described in chapter 2). Further, depending on the relative positions of the sample groups in the feature space, the sample groups are again subdivided. As a result, the training sample set of the class C_j ($j = 1, 2, \ldots, M$) is decomposed into m_j sample groups. Let $\hat{M}$ denote the total number of sample groups i.e., $\hat{M} = \sum_{j=1}^{M} m_j$.

Accordingly each feature domain is partitioned into some *feature subdo-*

mains to highlight the sample groups so that each feature information can be converted as the belongingness to the *feature subdomains* to some degree. Let n_i denote the number of *feature subdomains* in the ith feature axis. To handle the uncertainty of the input information and to incorporate the portions (of the pattern classes) possibly uncovered by the training samples, each of the *feature subdomains* is extended to some extent using triangular membership functions. Thus the whole feature space is decomposed into some overlapping *space subdomains*. The number of *space subdomains* is denoted by $\hat{N}$ i.e., $\hat{N} = \prod_{i=1}^{N} n_i$. Note that the feature regions in the individual feature axes are referred to as the *feature subdomains* and the regions in the whole feature space, which are the combinations of the *feature subdomains*, are referred to as the *space subdomains* here. The *feature subdomains* in the ith ($i = 1, 2, \ldots, N$) feature axis are denoted as $D_{i_1}, D_{i_2}, \ldots, D_{i_q}, \ldots, D_{i_{n_i}}$ and the *space subdomains* are denoted as $SD_1, SD_2, \ldots, SD_h, \ldots, SD_{\hat{N}}$.

The aforementioned decomposition of the sample groups and the dynamic ranges of the features constitute the preprocessing part of the recognition system. It is to be noted that the preprocessing is completely based on the training samples.

To explain the preprocessing concept, let us consider a two class (A and B) and two feature (F_1 and F_2) problem (i.e., $M = 2$ and $N = 2$) as shown in Fig. 4.1(a). Based on the geometric structure (as described in chapter 2), the sample set of class A is initially decomposed into two groups (denoted by A_1 and A_2) of nearly rectangular shapes as shown in Fig. 4.1(b). Then depending on the relative positions of the sample groups, the sample group A_1 is again subdivided into 2 subgroups A_{11} and A_{12} and the sample set of B is divided into two groups B_1 and B_2 [Fig. 4.1(c)]. Hence there are five sample groups i.e., $\hat{M} = 5$. Now in order to distinguish all the sample groups, the feature spaces F_1 and F_2 have been decomposed into three and two overlapping *feature subdomains* respectively. Thus, there are six ($\hat{N} = 3 \times 2 = 6$) *space subdomains*

that highlight all the five sample groups (and hence the two actual pattern classes). The *space subdomains* with reflected sample groups are shown in Fig. 4.1(*d*). •

For a given feature information X, let us define a *characteristic vector* $CV(X)$ as

$$CV(X) = (cv_1(X), cv_2(X), \ldots, cv_h(X), \ldots, cv_{\hat{N}}(X))' \qquad (4.1)$$

where the hth element $cv_h(X)$ $(h = 1, 2, \ldots, \hat{N})$ denotes the degree of belonging of X to the hth *space subdomain*. This is introduced keeping in mind the notion of fuzzy set theory where a pattern can have a finite membership to any *space subdomain* in the feature space. This $CV(X)$ is considered as the representation of a pattern X for the proposed system and its computational aspect is explained in the section 4.4.1.

The membership or compatibility functions used for characterizing the *feature subdomains* are described below.

4.2.2 Membership functions

After the preprocessing operation, each individual feature range gives rise to a few overlapping *feature subdomains*. As mentioned in section 2.4.1, for a given feature point the possibility of its being a member of a *feature subdomain* is maximum if it lies in the centre of the *feature subdomain*. As its distance from the centre increases, the membership value decreases.

Considering the aforementioned property, the gth *feature subdomain* along ith feature axis is characterized by $\pi_{ig}(x, \alpha_{ig}, \beta_{l_{ig}}, \beta_{u_{ig}}, \gamma_{l_{ig}}, \gamma_{u_{ig}})$ in which α_{ig} is the central point where the membership value is 1.0 ; $\beta_{l_{ig}}$ and $\beta_{u_{ig}}$ are the lower and upper most ambiguous (crossover) points where the membership values are 0.5 ; $\gamma_{l_{ig}}$ and $\gamma_{u_{ig}}$ are the lower and upper end points

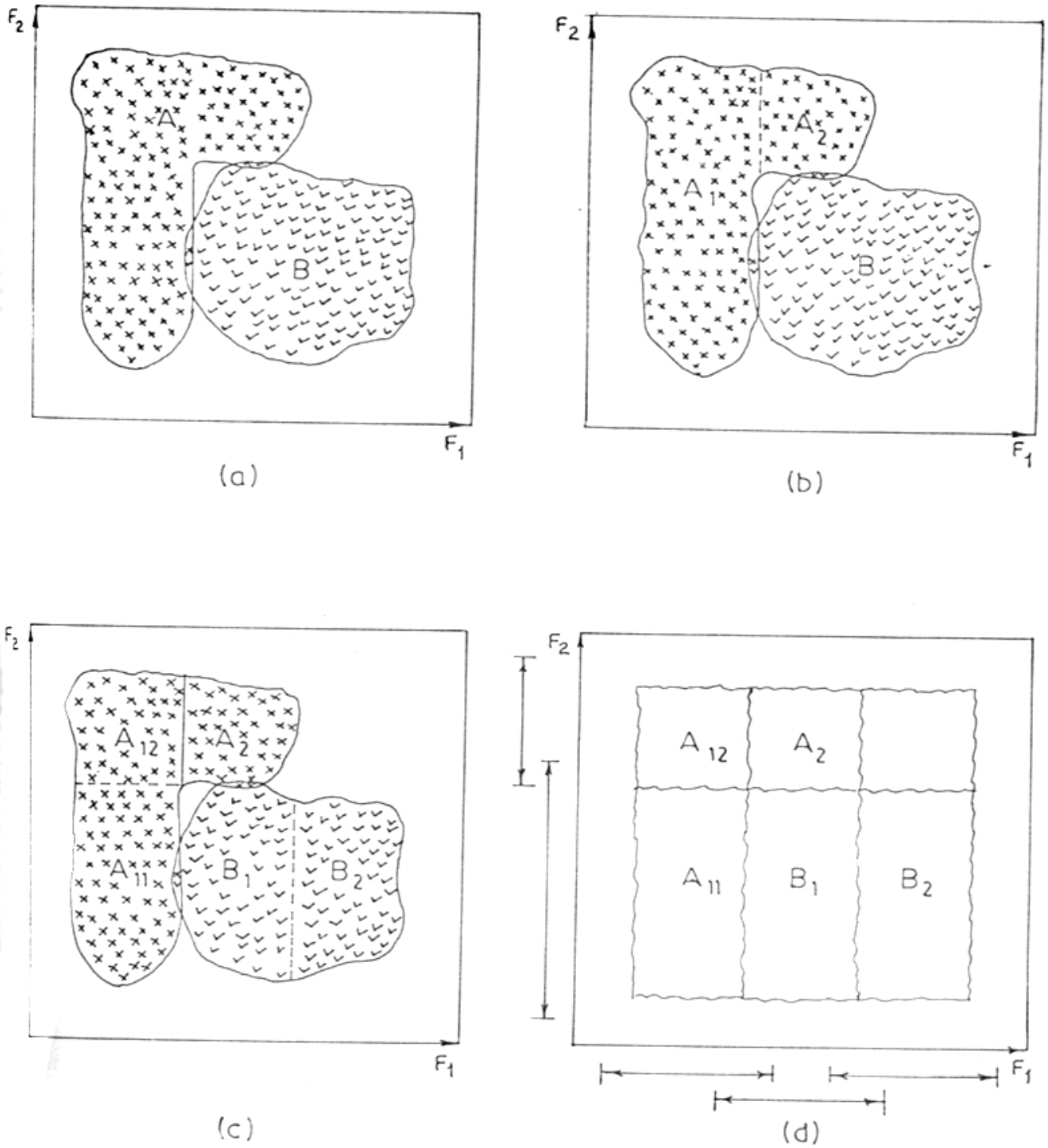


Figure 4.1 : Illustrating the preprocessing concept.

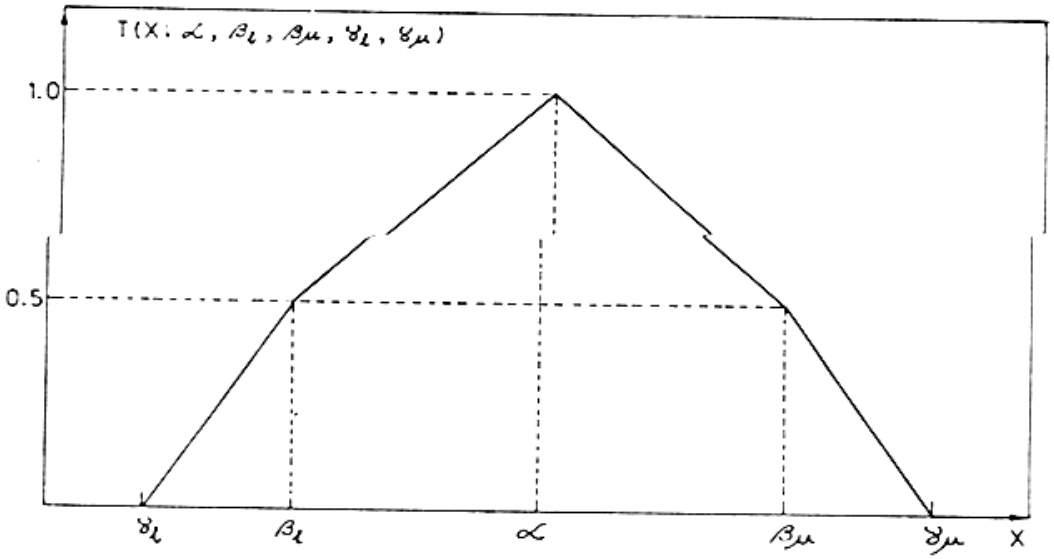


Figure 4.2 : Piecewise linear triangular function.

beyond which the membership values are zero. The functional form of such a π function has been provided in Eq. (2.5) [chapter 2] and it has been shown graphically in Fig. 2.5.

Note that the π function is a quadratic function. For simplicity, a piecewise linear triangular function (T) may also be considered. The functional form of such a linear triangular function is stated below.

$$T(x;\alpha,\beta_l,\beta_u,\gamma_l,\gamma_u) = \begin{cases} \frac{1}{2}\left(\frac{x-\gamma_l}{\beta_l-\gamma_l}\right) & \text{if } \gamma_l < x \leq \beta_l \\ \frac{1}{2}+\frac{1}{2}\left(\frac{x-\beta_l}{\alpha-\beta_l}\right) & \text{if } \beta_l \leq x \leq \alpha \\ \frac{1}{2}+\frac{1}{2}\left(\frac{x-\beta_u}{\alpha-\beta_u}\right) & \text{if } \alpha \leq x \leq \beta_u \\ \frac{1}{2}\left(\frac{x-\gamma_u}{\beta_u-\gamma_u}\right) & \text{if } \beta_u \leq x < \gamma_u \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

This is shown in Fig. 4.2. The parameters have the same meaning as in the case of the π function. ♣

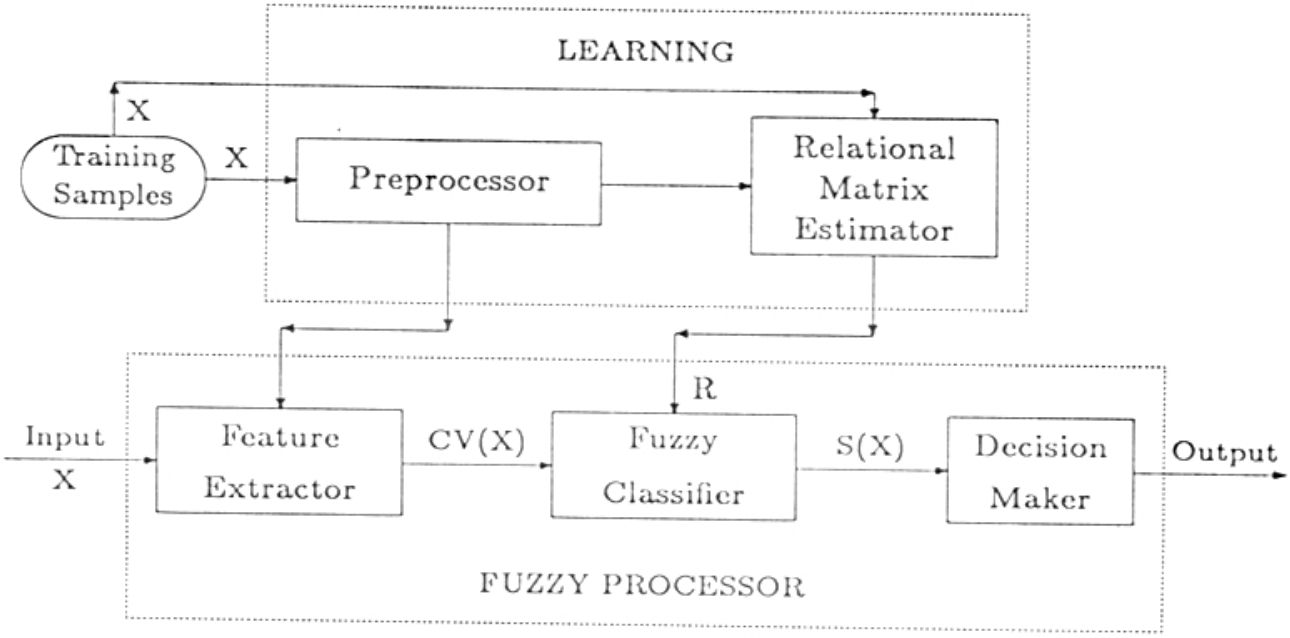


Figure 4.3 : Block diagram of the multivalued recognition system.

4.2.3 Block diagram

The block diagram of the proposed recognition system is shown in Fig. 4.3. It consists of two parts namely, *learning* and *fuzzy processor*. Learning section uses only the training sample information and finds the representative *space subdomains* and a relational matrix. The fuzzy processor uses the relational matrix in the Zadeh's compositional rule of inference (described in section 3.2.2 of chapter 3) to give a natural or linguistic output decision regarding the class or classes to which an unknown pattern X may belong. The preprocessing task of the learning section has been explained before. It decomposes the whole feature space into some overlapping *space subdomains*. The *relational matrix estimator* block finds a relational matrix R which denotes the compatibility of various pattern classes corresponding to the *space subdomains*. The order of R is $\hat{N} \times M$.

The *feature extractor* block takes a pattern X as input and outputs a

characteristic vector $CV(X)$. The *fuzzy Classifier* block uses $CV(X)$ and R in the Zadeh's compositional rule of inference to find a *class similarity vector* $S(X)$ as

$$S(X) = (s_1(X), s_2(X), \ldots, s_j(X), \ldots, s_M(X))' \tag{4.3}$$

where the jth ($j = 1, 2, \ldots, M$) element $s_j(X)$ denotes the degree of similarity of a pattern X to the jth class C_j.

Ambiguity (uncertainty) in the fuzzy decision, provided by $S(X)$, is then determined by computing CF (*confidence or certainty factor*). The higher the value of CF, the stronger is the validity of the decision. Depending on the value of CF, the final output of the recognition system is given in linguistic form. The description of various blocks of Fig. 4.3 is furnished in the following sections. ♠

4.3 Learning

Based on the training samples (only deterministic), the learning section decomposes the whole feature space into some overlapping *space subdomains* in order to handle the ambiguous information and estimates a relational matrix. It has two blocks, namely *preprocessing* and *relational matrix estimator*. The *feature subdomains* and *space subdomains* in the feature space are obtained in the preprocessing block. The relational matrix estimator block finds a relational matrix R.

4.3.1 Preprocessing

In this block, geometric structure (as described in chapter 2) and the relative position of the given pattern classes are considered one after another to decompose the training sample set of the pattern classes into some groups.

Accordingly each individual feature range is partitioned into some *feature subdomains* to highlight the sample groups. An intuitive idea of these concepts has been provided earlier in section 4.2.1 and these are explained below in detail.

A. Geometric complexity

The system assumes every pattern class as a collection of nearly parallelepiped sets. In order to determine from the training samples whether a pattern class is of nearly parallelepiped or not, the window based approach described in chapter 2 is utilized here. Based on the number of available training samples (say, t), it initially finds an *accuracy factor* (δ_t). The value of δ_t is decided by Eq. (2.1) [chapter 2] so that as $t \rightarrow \infty$, $\delta_t \rightarrow 0$ and $t\delta_t^N \rightarrow \infty$. Consequently using δ_t and the spans of the training sample set in the individual feature axes, it finds the corresponding *coverage factors* (ϵ_i, $i = 1, 2, \ldots, N$) using Eq. (2.3). Windows with training samples are then generated and subsequently the boundary variations of the sample set are found and analyzed. Ultimately, each of the training sample sets is decomposed into some groups of nearly parallelepiped shapes. The details of the approach have already been explained in chapter 2. •

B. Relative positions of pattern classes

The sample groups/sets thus generated are nearly parallelepiped in shape. The relative positions of the sample sets in the feature space are then considered to *divide (if necessary) further the sample groups so that the spans* (ranges) of the sample sets in a group are more or less same in an individual feature axis. This will facilitate (as described in the next section) to decompose the entire feature space into various *feature subdomains*. This concept of

relative position has been explained earlier by pattern diagrams in Fig. 4.1.

Here each feature axis is considered separately. Let us assume that there are $\hat{M}$ sample groups (initially $\hat{M} = M$). Let l_{ik} and u_{ik} ($i = 1, 2, \ldots, N$; $k = 1, 2, \ldots, \hat{M}$) be the lower and upper limits of the training samples corresponding to ith feature and kth sample group. Now follows an algorithm to decompose the sample groups based on their relative positions along the ith feature axis.

Algorithm 4.1 :

Here two temporary sets of the sample groups namely, *cv-set* and *l-set* are used.

STEP 1 : (Global initialization)

$cv\text{-}set = \text{NULL}$; $\hat{M} = M$;

STEP 2 : $l\text{-}set = \text{NULL}$;

$$L_1 = \min_{\substack{j=1,2,\ldots,\hat{M} \\ k \notin cv\text{-}set}} \{l_{ik}\} ;$$

$$l\text{-}set = l\text{-}set + [k] \quad \text{if} \quad l_{ik} = L_1$$

$$\text{for } k = 1, 2, \ldots, \hat{M} \quad \text{and} \quad k \notin cv\text{-}set ;$$

$$U_1 = \min \left\{ \min_{\substack{k=1,2,\ldots,\hat{M} \\ k \notin cv\text{-}set \\ k \in l\text{-}set}} l_{ik} \ , \ \min_{\substack{k=1,2,\ldots,\hat{M} \\ k \notin cv\text{-}set}} u_{ik} \right\} ;$$

$$cv\text{-}set = cv\text{-}set + [k] \quad \text{if} \quad u_{ik} = U_1$$

$$\text{for } k = 1, 2, \ldots, \hat{M} \quad \text{and} \quad k \notin cv\text{-}set ;$$

STEP 3 : The sample groups belonging to the *l-set* and not to the *cv-set* are decomposed into two groups. In such cases, the training samples with ith feature value less than or equal to U_1 are kept in the original sample group and include this sample group in the *cv-set*. Then a new sample group is generated (i.e., $\hat{M} = \hat{M} + 1$) and the residual samples are put in this new group.

STEP 4 : If *cv-set* includes all the sample groups, then the algorithm terminates. Otherwise go to STEP 2. •

The aforementioned algorithm decomposes the training sample set of a pattern class into groups according to the relative positions of the sample sets along the ith feature axis. This algorithm is repeated for all the features i.e., for $i = 1, 2, \ldots, N$. As a result, the training sample set of jth pattern class C_j $(j = 1, 2, \ldots, M)$ is decomposed into m_j sample groups and finally, $\hat{M}$ $(= \sum_{j=1}^{M} m_j)$ sample groups/sets are obtained. ♣

C. Decomposition of feature space

In order to highlight the sample sets, each individual feature space is divided into some overlapping *feature subdomains*. It is not difficult to group the sample sets so that the sample sets in each group correspond to one particular *feature subdomain* along an individual feature axis under consideration. The *feature subdomains* are then extended to some extent to incorporate the portions (of the pattern classes) possibly uncovered by the training samples and to handle the overlapping regions between the pattern classes. These *feature subdomains* are characterized by different π functions [Eq. (2.5)] of the form $\pi_{ig}(x, \alpha_{ig}, \beta_{l_{ig}}, \beta_{u_{ig}}, \gamma_{l_{ig}}, \gamma_{u_{ig}})$ (as described in section 2.2 of chapter 2). Note that, the piecewise linear triangular functions T [Eq. (4.2)] may also be used in characterizing the *feature subdomains*.

Here one feature axis is considered at a time. Let us assume that there are $\hat{M}$ sample groups which are obtained by decomposing the M training sample sets corresponding to the M pattern classes. Recall that m_j denotes the number of sample groups obtained by decomposing the training samples of the jth class C_j. Suppose l_{ik} and u_{ik} $(i = 1, 2, \ldots, N$ and $k = 1, 2, \ldots, \hat{M})$ are the lower most and upper most training samples corresponding to ith feature

and kth sample group. Recall also that n_i denotes the number of *feature subdomains* in the ith feature space. An algorithm is described below to find *feature subdomains* (and hence the corresponding membership functions) along the ith feature axis.

Algorithm 4.2 :

Here two temporary sets of the sample sets namely, *cv-set* and *l-set* are used. Let δ_t be the *accuracy factor* [Eq. (2.1)] and ϵ_{ig} be the extension-factor for the gth ($g = 1, 2, \ldots, \hat{N}$) *feature subdomain* along the ith ($i = 1, 2, \ldots, N$) feature axis.

STEP 1 : (Global initialization)

$cv\text{-}set = \text{NULL}$; $\quad g = 0$;

STEP 2 : $\quad l\text{-}set = \text{NULL}$; $\quad g = g + 1$;

$$L_1 = \min_{\substack{k=1,2,\ldots,\hat{M} \\ k \notin cv\text{-}set}} \{l_{ik}\} ;$$

$$l\text{-}set = l\text{-}set + [k] \qquad \text{if} \quad l_{ik} = L_1$$

$$\text{for } k = 1, 2, \ldots, \hat{M} \quad \text{and} \quad k \notin cv\text{-}set ;$$

$$U_1 = \max_{\substack{k=1,2,\ldots,\hat{M} \\ k \in l-set}} \{u_{ik}\} ;$$

$$cv\text{-}set = cv\text{-}set + l\text{-}set ;$$

STEP 3 : (Finding the parameters of π function)

$$\epsilon_{ig} = (U_1 - L_1) * \delta_t ;$$

$$\alpha_{ig} = (U_1 + L_1) * 0.5 ;$$

$$\beta_{l_{ig}} = L_1 \quad ; \quad \beta_{u_{ig}} = U_1 ;$$

$$\Gamma_{l_{ig}} = L_1 - \epsilon_{ig} \quad \text{and} \quad \Gamma_{u_{ig}} = U_1 + \epsilon_{ig} ;$$

STEP 4 : If *cv-set* includes all the sample groups, then the algorithm terminates and assign $n_i = g$. Otherwise go to STEP 2. •

This algorithm decomposes the ith feature space into n_i overlapping *feature subdomains*. This algorithm is repeated for all the feature axes i.e., for

$i = 1, 2, \ldots, N$. As a result, the total feature space is decomposed into $\hat{N}$ $(= \prod_{i=1}^{N} n_i)$ *space subdomains* that reflect all the sample groups. Note that $\hat{M} \neq \hat{N}$, in general.

4.3.2 Relational matrix estimator

The relational matrix R denotes the compatibility of various pattern classes corresponding to the *space subdomains*. The order of R is $\hat{N} \times M$, where $\hat{N}$ is the number of *space subdomains* and M is the number of pattern classes. Each column of R corresponds to a class and each row of that column denotes the degree to which a class should be characterized (based on the training samples) by the corresponding *space subdomains*. In other words, for a three class (C_1, C_2 and C_3) and two feature (F_1 and F_2) problem (i.e., $M = 3$ and $N = 2$) with three *feature subdomains* (p_{ig}, $i = 1, 2$ and $g = 1, 2, 3$) corresponding to each feature space (i.e., with $\hat{N} = 3 \times 3 = 9$ *space subdomains* which are denoted as $SD_h \equiv (p_{ig}, p_{ig'})$ where $g, g' = 1, 2, 3$ and $h = 1, 2, \ldots, 9$), a relational matrix R can be written as

(F_1, F_2)	SD_h	C_1	C_2	C_3
(p_{11}, p_{21})	1	r_{11}	r_{12}	r_{13}
(p_{11}, p_{22})	2	r_{21}	r_{22}	r_{23}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
(p_{13}, p_{23})	9	r_{91}	r_{92}	r_{93}

If a pattern belongs to the first *space subdomain* SD_1 (i.e., "F_1 is p_{11} and F_2 is p_{21}") then the entry r_{11} will denote the possibility value of the pattern to be in the class C_1. Similar is the case for all other entries of the relational

matrix.

Determination of R : The relational matrix R is estimated from the training samples in the relational matrix estimator block. Let r_{hj} denote the (h,j)th element of R i.e., the element corresponding to the hth *space subdomain* and jth pattern class. The value of r_{hj} is decided as

$$r_{hj} = \begin{cases} 0 & \text{if } SD_h \text{ does not highlight } C_j \text{ ;} \\ 1 & \text{if } SD_h \text{ highlights only } C_j \text{ ;} \\ (0.8)^{\frac{NS_h}{NG_h NC_j^h}} & \text{if } SD_h \text{ highlights } C_j \text{ along with } C_{j'},\ j' \neq j. \end{cases} \quad (4.4)$$

Here NG_h is the number of training sample groups highlighted by the *space subdomain* SD_h; NC_j^h is the number of training samples from the class C_j in SD_h, and NS_h is the total number of training samples in SD_h i.e., $NS_h = \sum_{j=1}^{M} NC_j^h$. If $NG_h = 0$ then $r_{hj} = 0 \ \forall\ j = 1,2,...,M$. If $NG_h = 1$ and SD_h highlights the class C_j then $r_{hj} = 1$ and $r_{hj'} = 0$ for $j' \neq j$. Otherwise i.e., if $NG_h > 1$, then SD_h is an overlapping *space subdomain* according to the training samples. The factor $\frac{NS_h}{NG_h NC_j^h}$ is used as a density factor for the class C_j in the hth (overlapping) *space subdomain* SD_h.

So the block relational matrix estimator provides R which is utilized in the fuzzy classifier block to find the output of the recognition system. ♠

4.4 Fuzzy Processor

This section consists of three parts, namely *feature extractor*, *fuzzy classifier* and *decision maker*. The feature extractor gives a *characteristic vector* $CV(X)$ [Eq. (4.1)] as output corresponding to an input X. The $CV(X)$ along with the relational matrix R is used in the fuzzy classifier to determine the degree of similarity of the input pattern X to the various pattern classes. The decision maker block gives a linguistic output along with its degree of certainty.

4.4.1 Feature extractor

Like the recognition system described in chapter 3, the input patterns considered here are in any of the four forms namely, *quantitative form*, *set form*, *mixed form* and *linguistic form*. Each feature value is considered separately to determine its membership values corresponding to various *feature subdomains* along that feature axis. The way it has been done is furnished below.

A. Quantitative form : The parameters of the membership functions corresponding to different *feature subdomains* have been determined in the preprocessing block [section 4.3]. So for the information in *quantitative form* (as exact numerical terms), the membership values to different *feature subdomains* in the feature range in consideration are computed directly from the corresponding membership functions. •

B. Mixed form : As mentioned in section 3.4.3 [chapter 3], the membership value of the input information in *mixed form* corresponding to various *feature subdomains* has been computed using Eq. (3.12). Note that the forms of the membership functions used in chapter 3 are changed here. The property set p in Eq. (3.12) is to be replaced here by the *feature subdomain* g. •

C. Set form : The way of processing the input information in *set form* is more or less the same as that mentioned in section 3.4.4 [chapter 3]. The only difference is that the property sets of chapter 3 are to be replaced here by the *feature subdomains*. •

D. Linguistic form : To handle linguistic information, the system assumes only three primary linguistic variables, namely *small*, *medium* and *high* which are represented by (1-S), π and S functions respectively. (Note that all these variables were represented by π functions in the previous chapter). Using some apriori knowledge, the values of the parameters of the membership functions are assigned.

As long as the membership functions are chosen properly, one recovers [237] the entirety of the classical logic for the designations of *true* and *false*. This implies that the system finds two truth values to indicate the interval of the truth values corresponding to a linguistic feature information. Here the system assumes for the two linguistic variables *small* and *high* that *true* $\equiv$ $[0.5, 1]$, *false* $\equiv$ $[0, 0.5)$ and extend this particular logic by adding

$$\begin{aligned} \textit{very true} &\equiv [0.8, 1.0] \\ \textit{more or less true} &\equiv [0.6, 0.8) \\ \textit{neither true nor false} &\equiv [0.4, 0.6) \\ \textit{more or less false} &\equiv [0.2, 0.4) \\ \textit{very false} &\equiv [0.0, 0.2) \end{aligned} \tag{4.5}$$

So corresponding to the aforementioned interval based truth value logic, one can find an interval of feature values which can be considered to be an equivalent of any linguistic information. That means, corresponding to a linguistic feature information, the system finds an interval of feature values. In other words, the system converts the linguistic information in *set form*. Then finds the membership value for various *feature subdomains* of the feature space depending on the converted information in *set form* from *linguistic form*.

This interval based truth value logic can not be directly used for the primary linguistic variable *medium* which is represented by a π function. Here the aforementioned interval based truth value reflects two different data sets and accordingly two different membership values for various *feature subdomains* are obtained. Finally, the maximum of these two membership values is retained as the membership value corresponding to each *feature subdomain* in the feature space. •

It may happen that the information about a particular feature is unavailable or missing. In such cases, it is reasonable to assign some low (say, 0.2) membership value to all the *feature subdomains* in that particular feature

axis. This will enable the system to handle the missing information and to make decision based on the available partial (or incomplete) information.

The aforementioned discussion shows a way, how the impreciseness/uncertainty in the input feature information can be handled by providing/ modifying the membership values heuristically to a great extent. The logic behind the assignment of membership values is also intuitively appealing. These modifications of the membership values will be reflected in the *confidence factor* (CF) i.e., in the final output of the recognition system.

E. Characteristic vector : The membership values corresponding to any input pattern X to be in the *space subdomains* are denoted by a vector, named as *characteristic vector* $CV(X)$ [Eq. (4.1)]. A typical pattern X consists of the individual feature information i.e., $X = (x_1, x_2, \ldots, x_N)/$. Initially, each individual feature information is considered separately to find the membership values to the *feature subdomains* of the individual feature axis. The approaches to determine the membership values from the feature information have been discussed earlier.

Let us consider a typical, say hth $(h = 1, 2, \ldots, \hat{N})$ *space subdomain* SD_h which consists of the following *feature subdomains*

$$SD_h \equiv \left\{ g_1^h, g_2^h, \ldots, g_i^h, \ldots, g_N^h \right\} \tag{4.6}$$

where depending on h, g_i^h represents a particular *feature subdomain* in the ith $(i = 1, 2, \ldots, N)$ feature axis. Suppose $\mu_{g_i^h}(X)$ represents the membership of X to belong in the g_i^hth *feature subdomain*. So the hth element of $CV(X)$ (i.e., the membership value corresponding to *space subdomain* SD_h) is defined as the arithmetic mean of the membership values of its constituent *feature*

subdomains i.e.,

$$cv_h(X) = \begin{cases} \frac{1}{N}\sum_{i=1}^{N} \mu_{g_i^h}(X) & \text{if } \mu_{g_i^h}(X) > 0 \\ & \forall\, i = 1,2,\ldots,N \\ 0 & \text{otherwise} \end{cases} \qquad (4.7)$$

$$h = 1,2,\ldots,\hat{N}.$$

Suppose the membership values $[cv_h(X)]$ *of a feature point are positive* for two or more neighboring *space subdomains*, and one particular pattern class is highlighted by the said *space subdomains*. Then it indicates that the said feature point lies in two or more sample groups of a pattern class. This in turn increases the possibility for the said feature point to belong in the actual pattern class. We call this notion as the *neighboring effect* and to incorporate this effect in the characteristic vectors $CV(X)$, M neighboring characteristic vectors corresponding to M pattern classes are defined as

$$CNV_j(X) = \left(cnv_{j1}(X), cnv_{j2}(X), \ldots, cnv_{jh}(X), \ldots, cnv_{j\hat{N}}(X)\right)' \qquad (4.8)$$

$$j = 1,2,\ldots,M$$

where the hth element $cnv_{jh}(X)$ denotes the degree of belonging to the jth class C_j based on hth *space subdomain* SD_h and incorporating the effect of neighboring *space subdomains* of SD_h on jth class. It is defined as

$$cnv_{jh}(X) = \begin{cases} \min\{1, cv_h(X) + \frac{1}{2\hat{N}} cv_{\hat{h}}(X)\} & \\ & \text{when } r_{hj} > 0 \text{ and } r_{\hat{h}j} > 0 \text{ and} \\ & \quad cv_{\hat{h}}(X) \text{ is the maximum among} \\ & \quad \text{all the neighboring } \textit{subdomains} \\ & \quad \text{of } SD_h \text{ highlighting } C_j \\ cv_h(X) & \text{otherwise} \end{cases} \qquad (4.9)$$

$$j = 1,2,\ldots,M \;;\; h = 1,2,\ldots,\hat{N}.$$

So the feature extractor block finds the *neighboring characteristic vectors*

with $\hat{N}$ elements (for $\hat{N}$ *space subdomains*) corresponding to each input pattern X. These $CNV_j(X)$'s along with the relational matrix R are utilized in the fuzzy classifier to find the degree of similarity of the input X to the various pattern classes. ♣

4.4.2 Fuzzy classifier

This block incorporates the Zadeh's compositional rule of inference (described in section 3.2.2) in a slightly modified way. The *min* operator (which gives connective information) of *max-min* operation in Eq. (3.3) is replaced by *arithmetic mean (AM)* operator (which gives collective information). That is, the *max-AM composition rule of inference* is incorporated here. Finally the *class similarity vector* $S(X)$ [Eq. (4.3)] is determined as

$$s_j(X) = \begin{cases} \max\limits_{h=1,2,\ldots,\hat{N}} \left\{\frac{1}{2}\left[cnv_{jh}(X) + r_{hj}\right]\right\} \\ \qquad\qquad \text{if } cnv_{jh}(X) > 0 \text{ and } r_{hj} > 0 \\ 0 \qquad\qquad \text{Otherwise} \end{cases} \tag{4.10}$$

$$j = 1, 2, \ldots, M$$

where $cnv_{jh}(X)$ is the hth element of $CNV_j(X)$; r_{hj} is the (h,j)th entry of R and $\hat{N}$ is the number of *space subdomains*. Hence the block fuzzy classifier finds a *class similarity vector* $S(X)$ corresponding to an unknown input X.

Note that we considered here AM in Eq. (4.10) because of the convenience of theoretical analysis of the system which will be made in the next chapter. One could have also used GM (geometric mean) as in Eq. (3.22) of chapter 3. ♣

4.4.3 Output

The *similarity vector* $S(X)$ is analyzed in the decision maker block. The system always tries to provide multiple output choices for classes with their preferences corresponding to any input X. So, the system considers the following four output forms:

(i) *Single choice* : If the entry in $S(X)$ corresponding to only one class, say C_j, is positive then the class C_j is considered as the output with *single choice.*

(ii) *Combined choice* : If the entries in $S(X)$ are positive corresponding to more than one class and if the said entries are nearly same (difference ≤ 0.05) then the corresponding classes are considered as output with *combined choice.*

(iii) *First-second choice* : If the entries in $S(X)$ are positive corresponding to more than one class and the said entries do not satisfy the criterion for *combined choice* then *first-second choice* is considered. The highest two entries in $S(X)$ are taken as the *first choice* and *second choice* respectively.

(iv) *Null choice* : If all the entries in $S(X)$ are zero then the system refuses to assign the unknown sample to any class i.e., *null choice* is given.

It is to be mentioned here that the *single choices* correspond to the non-overlapping regions in the feature space whereas the *combined* and *first-second choices* reflect the overlapping regions. *Null choices* reflect the regions uncovered by the training samples (with extended portions) and also the regions not represented by any class.

In order to give the final output decision in linguistic form regarding the class or classes to which the unknown input pattern X may belong, the *confidence factor* (CF) as defined in Eq. (3.23) [chapter 3] is calculated from

the elements of $S(X)$. In the case of *single choice*, the linguistic variable *surely* is attached to the final output decision. Otherwise, the linguistic variable *likely* is attached to the output. Based on the CF values, the linguistic hedges *very*, *more* or *less*, *not etc.* are assigned with the linguistic variables of the output decision as follows:

$$
\begin{aligned}
(i) \quad & \textit{very true} && : \quad \text{if } 0.8 \leq CF \leq 1.0 \\
(ii) \quad & \textit{true} \text{ (only)} && : \quad \text{if } 0.6 \leq CF < 0.8 \\
(iii) \quad & \textit{more or less true} && : \quad \text{if } 0.4 \leq CF < 0.6 \\
(iv) \quad & \textit{not false} && : \quad \text{if } 0.0 < CF < 0.4.
\end{aligned}
$$

In the case of *null choice*, the system gives the linguistic output decision as *unable to recognize*. The CF values are also attached to the linguistic output decisions.

It is to be mentioned here that the recognition system as described in the previous chapter provides output in linguistic form but the significance of the linguistic decisions is different than that adopted here. In the previous case, the linguistic decisions were fully based on the CF values. In the present case, the output form (either in *single choice* or *combined choice* or *first-second choice* or *null choice*) is initially decided based on the entries in the *similarity vector* $S(X)$. Depending on the output choices, some linguistic variables *surely*, *likely etc.* are assigned to the output decisions. Finally based on the CF values, linguistic hedges (*very*, *more or less*, *not*) *etc.* are attached with the linguistic variables of the output decisions.

4.5 Implementation and Results

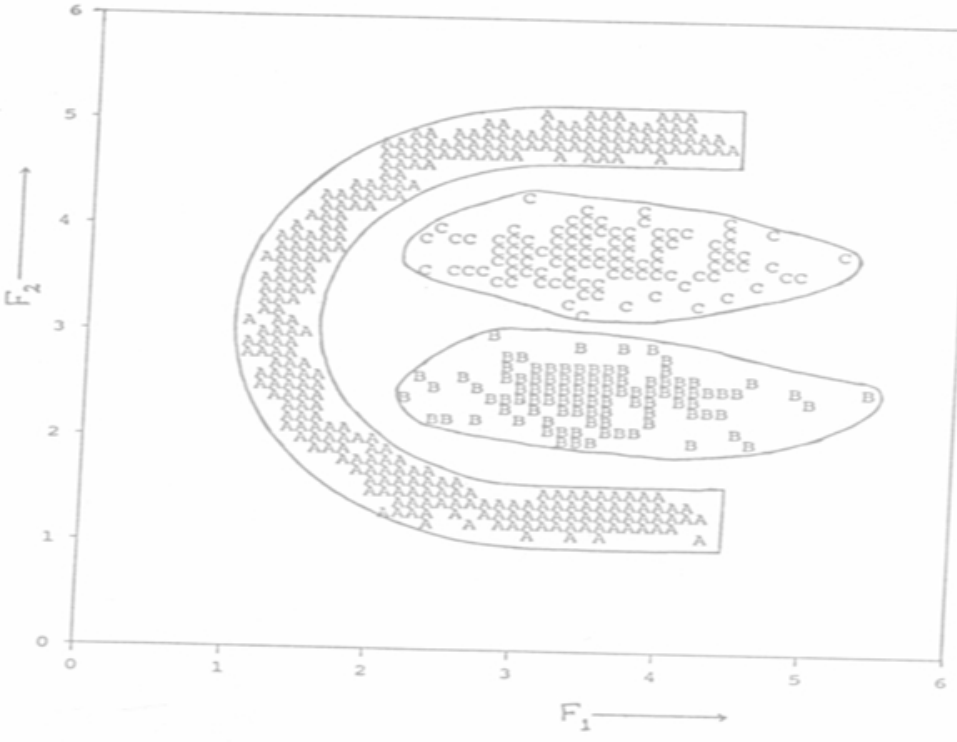
4.5.1 Artificially generated data

To demonstrate the effectiveness of the multivalued recognition system, different possible pattern sets were generated artificially and the proposed algorithm was implemented on them. The recognition scores have been found to be quite satisfactory in all the cases. Figures 4.4(a)-(d) show four typical pattern sets in two dimensional feature space. In Fig. 4.4(a), there are six pattern classes (denoted by A, B, C, D, E and F) with 120, 120, 90, 90, 180 and 120 samples respectively. In Fig. 4.4(b), there are three pattern classes (denoted by A, B and C) with 300, 100 and 100 samples respectively. In Fig. 4.4(c), there are two pattern classes (denoted by A and B) with 100 samples in each of the classes. In Fig. 4.4(d), there are two pattern classes (denoted by A and B) with 100 and 150 samples respectively.

To implement the proposed algorithm, five different sets of 10% training samples were chosen randomly from each of the four sets of pattern classes. The recognition scores for the considered four cases are shown in Tables 4.1(a)-(d) respectively. The scores shown are obtained by averaging those corresponding to five different training sets. The membership functions of various *feature subdomains* are considered as the π functions [Eq. (2.5)]. Assuming the piecewise linear triangular functions T [Eq. (4.2)], more or less same results are obtained.

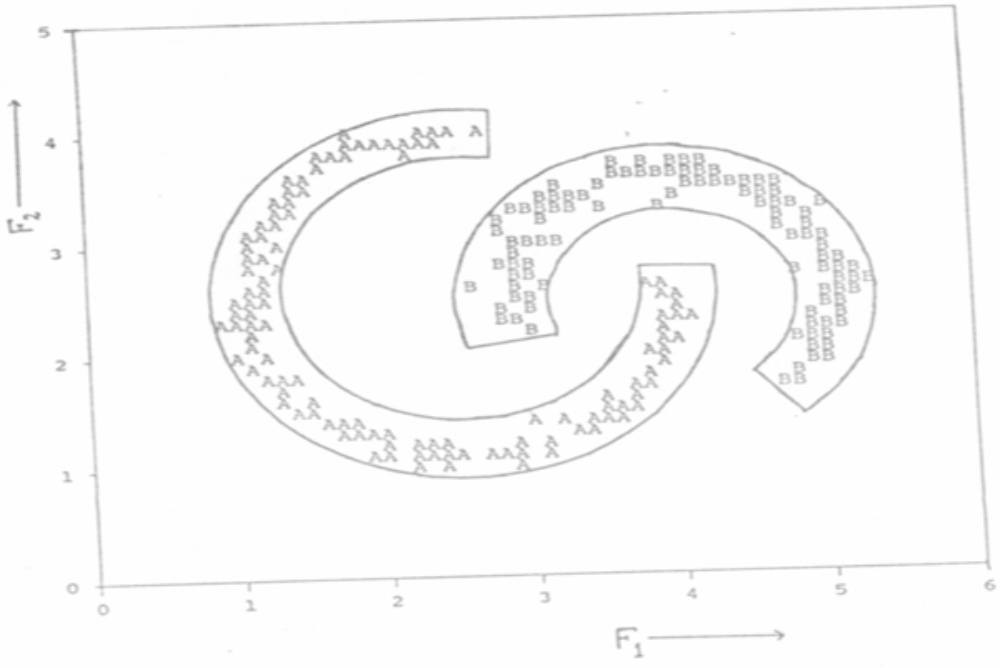
Note that the recognition scores are grouped into five categories, namely *single correct choice*, *first correct choice*, *combined correct choice*, *second correct choice* and *fully wrong choice*. The *single correct choice* set includes those samples for which the system's *single choice* corresponds to their actual class. The *first correct choice* set includes those samples for which the system provides *first-second choice* with *first choice* as their actual class. The

(a)

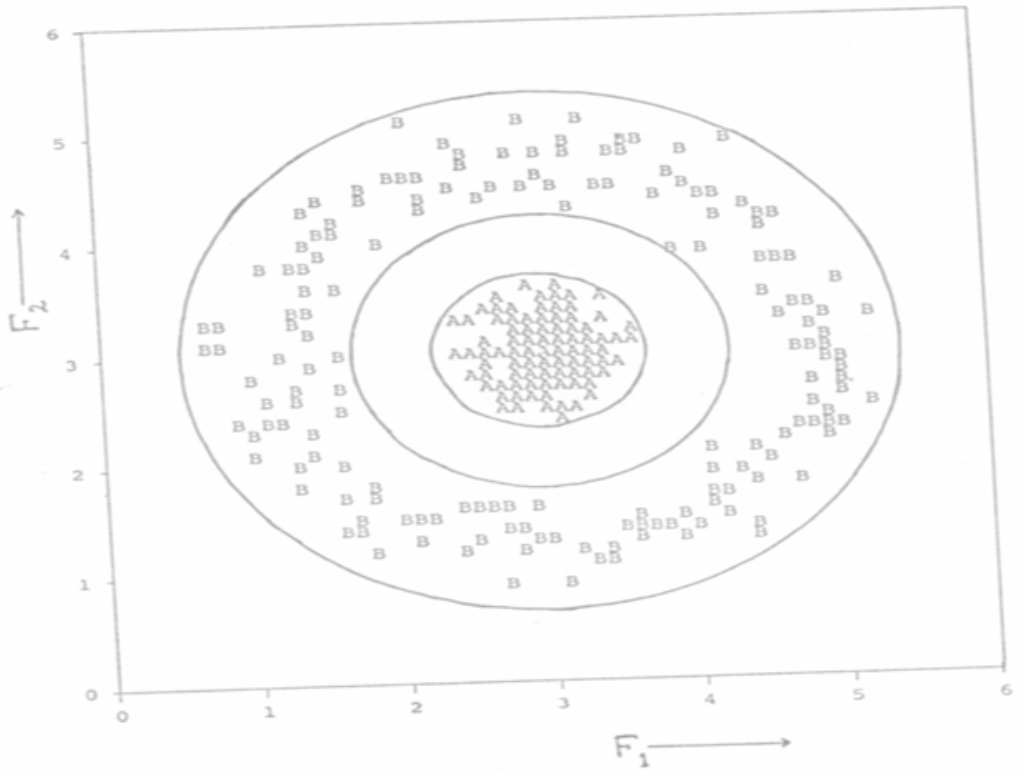


(b)

Figure 4.4 : (a)-(d) Four sets of pattern classes.



(c)



(d)

Figure 4.4 : (Continued).

Table 4.1(a) : Recognition score for the pattern classes in Fig. 4.4(a).

VARIOUS GROUP OF CHOICES	% RECOGNITION SCORE						
	ACTUAL CLASSES						OVERALL SCORE
	A	B	C	D	E	F	
Single correct choice	75.00	50.83	77.78	66.67	72.79	67.50	68.67
First correct choice	19.17	24.17	16.66	17.77	22.77	16.67	19.80
Combined correct choice	4.17	7.50	1.11	6.67	2.22	3.33	4.03
Second correct choice	1.66	16.67	3.33	8.89	1.67	11.67	6.94
Fully wrong choice	0.00	0.83	1.11	0.00	0.56	0.83	0.56

Table 4.1(b) : Recognition score for the pattern classes in Fig. 4.4(b).

VARIOUS GROUP OF CHOICES	% RECOGNITION SCORE			
	ACTUAL CLASSES			OVERALL SCORE
	A	B	C	
Single correct choice	95.33	94.00	97.00	95.40
First correct choice	3.00	3.00	2.00	2.80
Combined correct choice	1.66	3.00	1.00	1.80
Second correct choice	0.00	0.00	0.00	0.00
Fully wrong choice	0.00	0.00	0.00	0.00

Table 4.1(c) : Recognition score for the pattern classes in Fig. 4.4(c).

VARIOUS GROUP OF CHOICES	% RECOGNITION SCORE		
	ACTUAL CLASSES		OVERALL SCORE
	A	B	
Single correct choice	98.00	100.00	99.00
First correct choice	1.00	0.00	0.50
Combined correct choice	1.00	0.00	0.50
Second correct choice	0.00	0.00	0.00
Fully wrong choice	0.00	0.00	0.00

Table 4.1(d) : Recognition score for the pattern classes in Fig. 4.4(d).

VARIOUS GROUP OF CHOICES	% RECOGNITION SCORE		
	ACTUAL CLASSES		OVERALL SCORE
	A	B	
Single correct choice	99.00	100.00	99.60
First correct choice	1.00	0.00	0.40
Combined correct choice	0.00	0.00	0.00
Second correct choice	0.00	0.00	0.00
Fully wrong choice	0.00	0.00	0.00

combined correct choice set includes those samples for which the classifier provides *combined choice* and one of the choices corresponds to the actual class. The *second correct choice* set includes those for which the system provides *first-second choice* with *second choice* as the actual class. Samples not falling under the aforementioned categories are termed as *misclassification* or *fully wrong choice*. It is to be noticed that the states *first correct choice* and *second correct choice* are generated from the output form *first-second choice*. Hence the four output forms of the recognition system are categorized in the aforementioned five states.

Observe that the pattern classes in Fig. 4.4(a) are of regular (elliptical) shape and there exists overlapping between two or more classes. On the other hand, the pattern classes in figures 4.4(b)-(d) are of irregular shapes and they are mutually non-overlapping. In these three cases, most (95.4% to 99.6%) of the samples are seen to be recognized by *single choices* and the remaining samples are recognized either by *first choices* or *combined choices*. There is no samples falling under the sets *second choice* and *fully wrong choice*. In the case of Fig. 4.4(a), there are some samples which are found under the sets *second correct choice* and *fully wrong choice*. Note that when the overlapping is between two classes then the samples are recognized either by *single choice* or *first choice* or *combined choice* or *second choice* and so there will not be any sample falling under the set *fully wrong choice*. In case the overlapping exists between more than two classes, some samples will obviously fall under *fully wrong choice*. In such cases, it may be possible to avoid this situation by providing a higher choice namely, *third choice*. Note that for the pattern classes in figures 4(b)-(d), the 1-nearest neighbor decision rule [2,7] will give very satisfactory results. However, the purpose of considering the pattern classes in figures 4.4(a)-(d) is to show that the proposed multi-state system can be applied satisfactorily to different possible structures of pattern classes.

4.5.2 Speech data

To examine the practical applicability of the system, the algorithm has been implemented on the speech data as used in chapters 2 and 3. The vowel

Table 4.2 : Recognition score for the vowel classes in Fig. 2.9.

VARIOUS GROUP OF CHOICES	% RECOGNITION SCORE						
	ACTUAL CLASSES						OVERALL SCORE
	/δ/	/a/	/i/	/u/	/e/	/o/	
Single correct choice	40.05	59.47	68.72	67.92	52.19	55.37	58.92
First correct choice	11.34	22.55	24.88	26.12	20.27	22.41	21.76
Combined correct choice	20.83	11.24	2.91	1.99	13.04	8.33	8.63
Second correct choice	19.44	6.74	3.49	3.97	12.56	13.89	9.56
Fully wrong choice	8.34	0.00	0.00	0.00	1.94	0.00	1.13

set contains 871 deterministic samples with six vowel classes (/δ/, /a/, /i/, /u/, /e/, /o/). The feature space in $F_1 \times F_2$ plane, showing ill-defined class boundaries, is shown in Fig. 2.9 [chapter 2]. F_1 and F_2 denote the first and second formant frequencies.

As mentioned in the previous chapter, the test set consists of 871 deterministic and 102 imprecise (incomplete) data. These imprecise data on F_1 and F_2 were coded to various linguistic forms *viz.*, *(*700, *between* 1800 *to* 2200*)*, *(about* 600, *more or less high), (small, -) etc.* by the trained personnel. The recognition score of the vowel data is shown in Table 4.2 where the classifier is trained with a set of 10% samples drawn randomly from 871 deterministic data. The membership functions of various *feature subdomains* are considered as the π functions [Eq. (2.5)]. Assuming the piecewise linear triangular functions T [Eq. (4.2)], more or less same results are obtained.

A list of some typical output is given for illustration.

- (300, 900) : This is *surely* to be /u/ ($CF = 0.78$).
- (700, 1300) : This is *more or less likely* to be /a/ ($CF = 0.55$) but *not*

unlikely to be /δ/ ($CF = 0.32$).

- (900, 1400) : This is *more or less sure* to be /a/ ($CF = 0.25$).
- (250, 1550) : *unable to recognize* this sample.
- (*between* 500 and 600, 1600) : This is *likely* to be /i/ ($CF = 0.69$) but *not unlikely* to be /e/ ($CF = 0.32$).
- (*about* 350, –) : This is *likely* to be either /i/ or /u/ ($CF = 0.28$).

Like chapter 3, the outputs confirm the vowel diagram in Fig. 2.9. Note that for the input (250, 1550), the system is unable to recognize the vowel, as this information has very insignificant similarity with the vowel classes. This has been regarded as *misclassification* while computing the recognition score. Further, for the information (*about* 350, –) (here "–" indicates that there is no information for F_2 feature), the system finds some similarity with the vowel classes /i/ and /u/, on the basis of the F_1 feature information alone.

4.5.3 Some remarks

Unlike the conventional classifiers which usually involve two state (correct or wrong) decisions, the proposed recognition system provide multi-state (Fig. 4.5) output reflecting the overlapping, non-overlapping and no-class regions in the feature space. The distribution of the pattern classes need not be known here and the system can be applied to a wide range of situations. The only assumption made is that the training samples more or less should represent the classes.

The effectiveness of most of the existing statistical classifiers depends on the distribution functions of the pattern classes. Bayes classifier (as described in section 1.2.5*B*) is the most well known and established classifier, and we

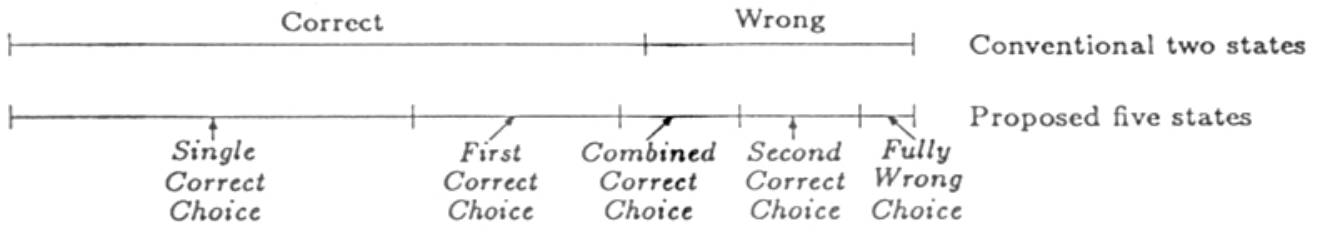


Figure 4.5 : Conventional two state versus proposed five state output.

have tried to apply it on the artificially generated pattern sets for the comparison purpose. If the classes are of regular shape and if their distribution functions can be obtained accurately, the performance of the Bayes classifier is more or less same with our system (considering *single correct*, *first correct* and half of *combined correct choices*). For example, the classes in Fig. 4.4(a) are of regular (elliptical) shape and the recognition score of the Bayes classifier was found to be 90.64% whereas, the recognition score for our classifier with the *single*, *first* and half of the *combined correct choices* is 90.485%. Note that the error rate (9.36%) of the Bayes classifier has been split up into *fully wrong choices*, *second correct choices* and a part of *combined correct choices* in the proposed system. This gives a provision of improving its efficiency significantly by incorporating the *combined* and *second choices* under the control of a supervisory scheme. Therefore the output decisions showing multiple choices are more natural and justified.

When the pattern classes are not of regular shape [figures 4(d)-(d)], it is extremely difficult to find their distribution functions [2, 7]. However, in such cases the densities may be estimated by some non-parametric methods [266-271] (e.g., Parzen [267] and Cacoullos [268] window methods, k-nearest neighbor density estimation method [271]) for the application of Bayes classifier, and the empirical Bayes classifier based on estimated densities may provide good results. On the other hand, the proposed system provides good results [Tables 4.1(a)-(d)] irrespective of the shapes and densities of the data sets.

As in the previous chapter, the confusion in recognizing a vowel data considering the *single choices* and *first choices* lies, in general, only with the neighboring classes constituting a vowel triangle. As expected, the overall recognition score is seen to be better than those obtained in [26,139,235,236] because of the consideration of geometric structures of the classes. As in the previous chapter, the classifier described here has also serious practical difficulties for its implementation in higher dimensions. ♠

Chapter 5

THEORETICAL ANALYSIS OF MULTIVALUED RECOGNITION SYSTEM

Contents

5.1 Introduction

The multivalued recognition system, described in the previous chapter, decomposes the feature space based on the geometric structure and the relative position of the pattern classes, and provides output as multiple choice for classes with their preferences. It classifies a sample either as a *single choice* (possibility to be only in one class) or as a *combined choice* (possibility of belonging to two classes with same preference) or as a *first-second choice* (possibility of belonging to two classes with different preferences) or as a *null choice* (possibility of not belonging to any of the classes). The *single choices* correspond to the non-overlapping regions whereas, the overlapping regions are reflected by the *first-second* and *combined choices*. The *null choices* reflect the regions outside the pattern classes and/or the regions of the pattern classes uncovered by the training samples (even with its extended portions). Thus, the system has the ability of discriminating the overlapping and no-class (i.e., ambiguous/doubtful) regions and analyzing the associated uncertainties by providing output in four states.

In this chapter, a theoretical analysis of the aforementioned characteristics and performance of the recognition system is provided. Initially, the regions corresponding to the four output forms are determined analytically and the estimates of the non-overlapping, overlapping and no-class regions in the feature space are then determined. Various possible situations in one and two dimensional feature spaces with two classes are considered here. It has been shown theoretically that with the increase in the size of the training samples, the estimates of the overlapping, non-overlapping and no-class (null choice) regions tend to their actual sizes.

All the analytical findings have been substantiated with experimental results. For a comparative study of the recognition system with a conventional one, the Bayes classifier (as described in section 1.2.5B of chapter 1) is implemented on the considered various data sets. The Bayes thresholds between classes are always found to lie within the *combined choice region* of the mul-

tivalued recognition system. The present investigation, in turn, analytically establishes the justification of considering output decision in four states for managing uncertainties arising from ambiguous regions.

Section 5.2 provides some theorems which are used in the subsequent sections. Sections 5.3 and 5.4 deal with the theoretical analysis of the system in one and two dimensional feature spaces respectively. ♠

5.2 Some Theorems

A formal definition of a pattern class has been provided in chapter 2 (section 2.2.1*A*) where it is considered as a compact and bounded set in $\mathbb{R}^N$, $N \geq 2$. In one dimensional space, the pattern class is a compact interval. In this section some theorems are stated and also verified. These theorems are used in the following sections for the proposed theoretical analysis.

Definition 5.1 : *P is said to be probability measure on $\mathcal{A}$, $\mathcal{A} \in \mathbb{B}$ if P satisfies the following properties [61,238,239]*

(*i*) $P \ll \lambda$. *[λ is the Lebesgue measure on $\mathbb{R}^N$]*

(*ii*) *Let* $f = \frac{dP}{d\lambda}$ *be the density on $\mathcal{A}$. Then* $f(x) > 0 \quad \forall x \in Int(\mathcal{A})$. •

Note that Definition 5.1 restricts the probability measures under consideration on $\mathcal{A}$.

Result : Let $X_1, X_2, \ldots, X_t$ be independent and identically distributed random variables with density f. If $\mathcal{A}$ is a class and P is a probability measure on $\mathcal{A}$

then $\forall B \subseteq \mathcal{A}$, B is open,

$\Rightarrow \quad P(X_1 \notin B, X_2 \notin B, \ldots, X_t \notin B) \to 0$ as $t \to \infty$.

Proof : Proof is obvious. •

Theorem 5.1 : *Let $X_1, X_2, \ldots, X_t$ be independent and identically dis-*

tributed random variables taking values in $[a, b]$ *with continuous density* f.

Let $X_{(1)} = \min\ \{X_1, \ldots, X_t\}$ *and* $X_{(t)} = \max\ \{X_1, \ldots, X_t\}$.

Then $X_{(1)}$ *goes to* a *in probability as* $t \to \infty$

and $X_{(t)}$ *goes to* b *in probability as* $t \to \infty$.

Proof : Let F be the density function i.e., $F(x) = \int_a^x f(y)dy$, $a \leq x \leq b$.

Then $F(a) = 0$ and $F(b) = 1$.

$$\begin{aligned} P\left(X_{(1)} > y\right) &= P\left(X_1 > y, X_2 > y, \ldots, X_t > y\right) \\ &= [P\left(X_1 > y\right)]^t \\ &= [1 - P\left(X_1 \leq y\right)]^t \\ &= [1 - F(y)]^t \qquad a \leq y \leq b \end{aligned} \tag{5.1}$$

Let $\varepsilon > 0$.

$P\left(|\, X_{(1)} - a| > \varepsilon\right) \to 0$ for any $\varepsilon > 0$.

$\Leftrightarrow\ P\left(X_{(1)} - a > \varepsilon\right) \to 0$

$\Leftrightarrow\ P\left(X_{(1)} > a + \varepsilon\right) \to 0$

$\Leftrightarrow\ [1 - F(a + \varepsilon)]^t \to 0$ [by Eq. (5.1)] which is true as $t \to \infty$.

So $X_{(1)}$ goes to a in probability.

$$\begin{aligned} P\left(X_{(t)} \leq y\right) &= P\left(X_1 \leq y, X_2 \leq y, \ldots, X_t \leq y\right) \\ &= [F(y)]^t \qquad a \leq y \leq b \end{aligned} \tag{5.2}$$

Let $\varepsilon > 0$.

$P\left(|\, X_{(t)} - b| \geq \varepsilon\right) \to 0$ for any $\varepsilon > 0$.

$\Leftrightarrow\ P\left(b - X_{(t)} \geq \varepsilon\right) \to 0.$

$\Leftrightarrow\ P\left(X_{(t)} \leq b - \varepsilon\right) \to 0.$

$\Leftrightarrow\ [F(b - \varepsilon)]^t \to 0$ [by Eq. (5.2)] which is true as $t \to \infty$.

So $X_{(t)}$ goes to b in probability as $t \to \infty$.

Hence the theorem. •

Theorem 5.2 : *Let* $X_1, X_2, \ldots, X_t$ *be independent and identically distributed random variables with continuous density* f *in* $[a, b]$.

Let $\mathcal{A} = [a, b]$ *and* $\delta_t \to 0$ *as* $t \to \infty$.

Let $\mathcal{B}_t = \left[X_{(1)} - \delta_t\ ,\ X_{(t)} + \delta_t\right]$.

training samples are either overlapping or non-overlapping. Note that if the training samples are non-overlapping then the membership functions of the feature *feature subdomains* may be overlapping. All these situations have been considered in the proposed analysis. ♠

5.3 Analysis in 1-D Feature Space

To provide the theoretical analysis of the multivalued recognition system in one dimensional feature (F) space, a two class (C_1 and C_2) problem is considered here. Since there is only one feature, the *space subdomains* are same as the *feature subdomains*. Suppose $[L_1, U_1]$ and $[L_2, U_2]$ [Fig. 5.1(a)] denote the actual class ranges corresponding to the classes C_1 and C_2 respectively. Suppose also, $[l_1, u_1]$ and $[l_2, u_2]$ [Fig. 5.1(a)] are the ranges of training samples corresponding to C_1 and C_2 for a particular training set. To find the regions corresponding to *single*, *first-second*, *combined* and *null choices*, and to verify *Proposition* 5.1, different possible cases are considered.

5.3.1 Non-overlapping pattern classes

This case is shown in Fig. 5.1(a). Initially, the system finds two distinct *feature subdomains* $[l_1, u_1]$ and $[l_2, u_2]$ corresponding to the classes C_1 and C_2. The *feature subdomains* are denoted by D_1 and D_2. Now these *feature subdomains* are extended to some extent using piecewise linear triangular functions [Eq. (4.2)] T_1 and T_2 respectively. The *feature subdomains* with their characterizing membership functions are shown in Fig. 5.1(b). Note that

$$\beta_{l_g} = l_g \quad ; \quad \beta_{u_g} = u_g \quad ;$$
$$\Gamma_{l_g} = l_g - \varepsilon_g \quad \text{and} \quad \beta_{u_g} = u_g + \varepsilon_g \quad \text{for g} = 1, 2.$$

Here ε_g [Eq. (2.3)] and δ_g [Eq. (2.1)] are the extended portion and the accuracy factor respectively for the gth *feature subdomain* D_g ($g = 1, 2$).

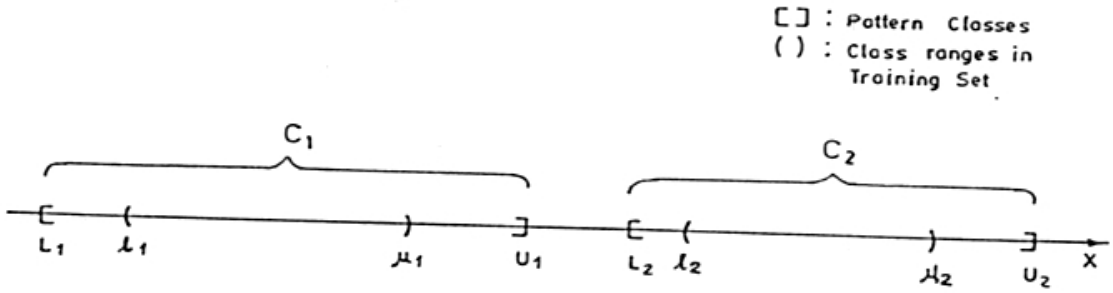


Figure 5.1(a) : Two non-overlapping classes (1-D) with a training set.

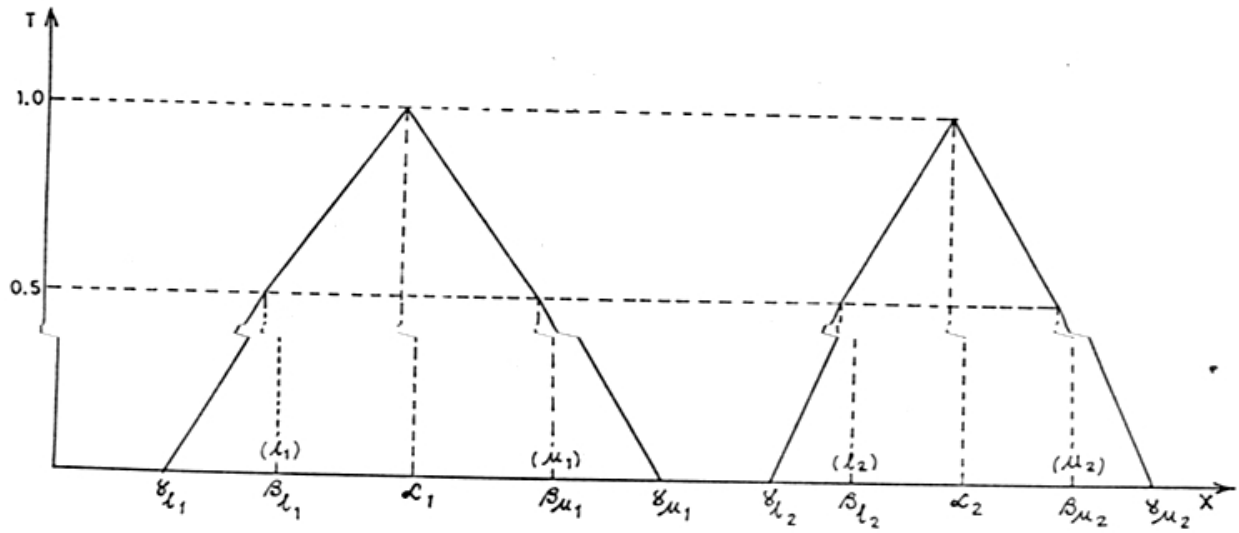


Figure 5.1(b) : *Subdomains* (disjoint) with their membership functions for the pattern classes in Fig. 5.1(*a*).

The relational matrix R in this case will be

$$R = ((r_{gj}))_{g=1,2;\ j=1,2}$$

$$\text{where} \quad r_{gj} = \begin{cases} 1.0 & \text{for } g = j \\ 0 & \text{for } g \neq j. \end{cases}$$

Case 1 : *Feature subdomains* are disjoint

This case is shown in Fig. 5.1(b) where the *feature subdomains* D_1 and D_2 are disjoint. Here, it is easy to show that the *single choice region* for C_1 is $(\Gamma_{l_1}, \Gamma_{u_1})$; the *single choice region* for C_2 is $(\Gamma_{l_2}, \Gamma_{u_2})$ and the remaining points in the feature space represent the *null choice region.* •

Case 2 : *Feature subdomains* are overlapping

This case is demonstrated in Fig. 5.1(c) where the *feature subdomains* D_1 and D_2 are overlapping in their extended portions. It is clear from Fig. 5.1(c) that $(\Gamma_{l_1}, \Gamma_{l_2}]$ and $[\Gamma_{u_1}, \Gamma_{u_2})$ represent the *single choice regions* for C_1 and C_2 respectively. For $X \in (\Gamma_{l_2}, \Gamma_{u_1})$, the elements in the *similarity vector* $S(X)$ are all positive i.e., $s_1(X) > 0$ and $s_2(X) > 0$. This implies that $(\Gamma_{l_2}, \Gamma_{u_1})$ is the overlapping region and we will concentrate now only on this region.

Here two points, say τ_1 and τ_2, can be found such that $\Gamma_{l_2} < \tau_1 < \tau_2 < \Gamma_{u_1}$, and

$$\begin{aligned} &\text{for} \quad X \in (\Gamma_{l_2}, \tau_1), \quad [s_1(X) - s_2(X)] > 0.05; \\ &\text{for} \quad X \in (\tau_2, \Gamma_{u_1}), \quad [s_2(X) - s_1(X)] > 0.05 \quad \text{and} \\ &\text{for} \quad X \in [\tau_1, \tau_2], \quad | s_1(X) - s_2(X) | \leq 0.05. \end{aligned}$$

The values of τ_1 and τ_2 are given by

$$\begin{aligned} \tau_1 &= \frac{u_1\epsilon_2 + l_2\epsilon_1 - 1.1\epsilon_1\epsilon_2}{\epsilon_1 + \epsilon_2} \\ \text{and} \qquad \tau_2 &= \frac{u_1\epsilon_2 + l_2\epsilon_1 - 0.9\epsilon_1\epsilon_2}{\epsilon_1 + \epsilon_2} \end{aligned} \tag{5.3}$$

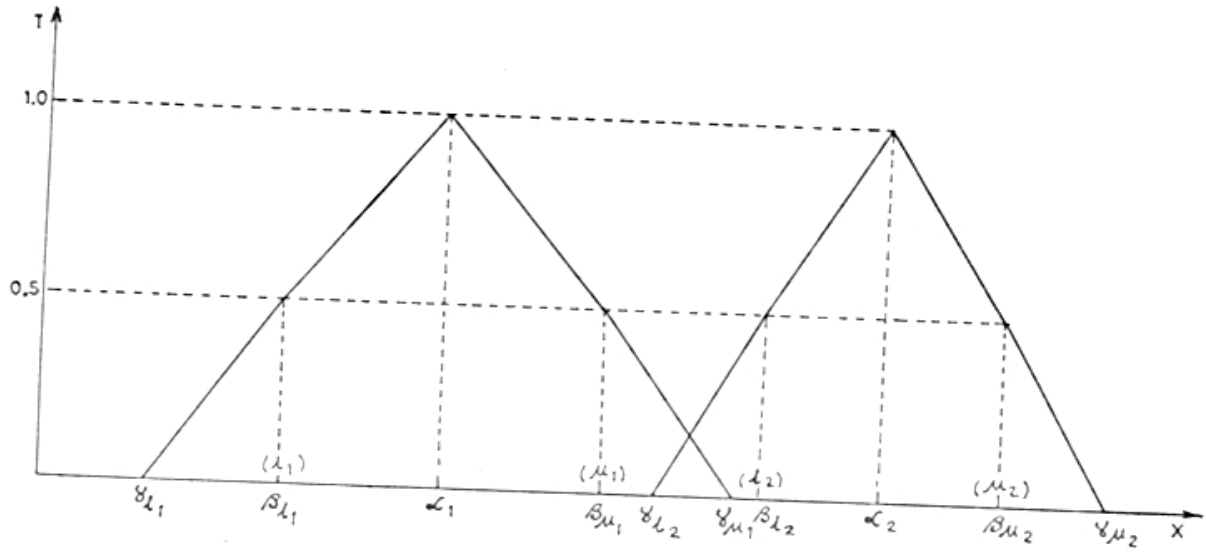


Figure 5.1(c) : *Subdomains* (overlapping) with their membership functions for the pattern classes in Fig. 5.1(a).

where ϵ_g [Eq. (2.3)] is the extended portion of the *feature subdomain* D_g ($g = 1, 2$), and l_j and u_j are the lowest and highest values respectively among the training samples from the class C_j ($j = 1, 2$).

So, as a summary, we can state the following :

The *single choices* for C_1 is $(\Gamma_{l_1}, \Gamma_{l_2}]$; the *first-second choice region* with first choice as C_1 is (Γ_{l_2}, τ_1) ; the *combined choice region* is $[\tau_1, \tau_2]$; the *first-second choice region* with first choice as C_2 is (τ_2, Γ_{u_1}) ; the *single choices* for C_2 is $[\Gamma_{u_1}, \Gamma_{u_2})$, and the remaining points in the feature space correspond to the no-class (*null choice*) region. However, if $T_1(\Gamma_{l_2}) \leq 0.05$ i.e., if $l_2 - u_1 \geq \epsilon_2 + 0.9\epsilon_1$ then (Γ_{l_2}, τ_1) becomes a *combined choice region* instead of a *first-second choice* region. Again, if $T_1(\Gamma_{l_2}) \leq 0.05$ i.e., if $l_2 - u_1 \geq \epsilon_1 + 0.9\epsilon_2$ then (τ_2, Γ_{u_1}) will be a *combined choice region* instead of a *first-second choice region.* If both the conditions are satisfied i.e., if $l_2 - u_1 \geq \epsilon_2 + 0.9\epsilon_1$ and $l_2 - u_1 \geq \epsilon_1 + 0.9\epsilon_2$ then the complete set $(\Gamma_{l_2}, \Gamma_{u_1})$ will represent a *combined choice region.* •

Proof of Proposition 5.1 :

By Theorem 5.1, it can be stated that with the increase of training sample size,

$$l_j \rightarrow L_j \text{ and } u_j \rightarrow U_j \text{ in probability for } j = 1, 2. \quad (5.4)$$

With the increase of training sample size, the sample sizes in the *feature subdomains* also increase. Thus, by Theorem 5.2,

$$\Gamma_{l_g} \rightarrow \beta_{l_g} \text{ and } \Gamma_{u_g} \rightarrow \beta_{u_g} \text{ in probability for } g = 1, 2. \quad (5.5)$$

This implies that the uncovered portions of the feature space (by the training sample set) decrease in probability with the increase in sample size.

In case 1, there are no overlapping region, and the *single choice region* for C_j $(j = 1, 2)$ is $(\Gamma_{l_j}, \Gamma_{u_j})$ which tends to the set (L_j, U_j) (the actual non-overlapping region) in probability.

In case 2, the estimated overlapping region is $(\Gamma_{l_2}, \Gamma_{u_1})$. By equations (5.4) and (5.5), this overlapping region tends to (L_2, U_1) in probability. But $U_1 < L_2$. This implies that the overlapping region decreases in probability.

The *single choice regions* for C_1 and C_2 are $(\Gamma_{l_1}, \Gamma_{l_2}]$ and $[\Gamma_{u_1}, \Gamma_{u_2})$ respectively. By equations (5.4) and (5.5) these regions converge to (L_1, L_2) and (U_1, U_2) in probability. As $U_1 < L_2$, the regions under *single choices* for C_1 and C_2 tend to (L_1, U_1) and (L_2, U_2) (actual non-overlapping regions) in probability.

Hence the proof. •

Experimental results :

To verify the aforementioned analytical results, a two class problem with class ranges $[2, 6]$ and $[7, 11]$ is considered. It may be noted that the pattern classes are non-overlapping. To implement the recognition system, five

training sample sets with 50, 100, 150, 200 and 250 samples from each of the classes are chosen randomly.

The ranges of the training sample sets for the classes C_1 and C_2, and for the *feature subdomains* D_1 and D_2 corresponding to the five sample sets are shown in Table 5.1(a). It is to be noticed that for first two sample sets (with 50 and 100 samples from each class), the *feature subdomains* obtained are overlapping whereas, for the remaining three sets (with 150, 200 and 250 samples from each class), the *feature subdomains* obtained are non-overlapping. So the first two sample sets represent the case 2 i.e., the *feature subdomains* are overlapping when the pattern classes are non-overlapping. The remaining three sample sets represent the case 1 i.e., the *feature subdomains* are non-overlapping when the pattern classes are non-overlapping. The various regions obtained for the first two sample sets are shown in Table 5.1(b). The regions obtained for the remaining three sample sets are shown in Table 5.1(c).

These results verify that with the increase in sample size, the training classes (i.e., portions covered by the training samples) tend to the actual classes (Theorem 5.1) and the *feature subdomains* with the extended portions tend to their actual sizes (Theorem 5.2). It is to be noticed that the overlapping regions (i.e., corresponding to *combined* and *first-second choices*), the non-overlapping regions (i.e., corresponding to *single choices*) and no-class regions (i.e., corresponding to *null choices*) are tending to their actual sizes with the increase in sample sizes. Thus *Proposition* 5.1 has been verified experimentally for the case of non-overlapping pattern classes in one dimensional feature space.

To show the recognition performance of the system experimentally, a test sample set with 1000 samples from each of the considered pattern classes is generated. With all the aforementioned five training sets, the system recognized the total test sample set correctly under *single choices*.

For comparison, the Bayes classifier is also applied on the same pattern classes (assuming normal distributions). The threshold points found between

Table 5.1(a) : Ranges of training sets and *feature subdomains* for the pattern classes [2,6] and [7,11]

	SAMPLE SIZES IN EACH CLASS				
	50	**100**	**150**	**200**	**250**
Class C_1	2.09 - 5.92	12.07 - 5.94	12.06 - 5.97	12.02 - 5.98	12.01 - 5.99
Class C_2	7.16 - 10.89	7.05 - 10.92	7.04 - 10.93	7.02 - 10.95	7.01 - 10.98
Subdomain D_1	1.29 - 6.74	1.37 - 6.59	1.52 - 6.44	1.67 - 6.32	1.78 - 6.19
Subdomain D_2	6.49 - 11.65	6.42 - 11.62	6.56 - 11.47	6.71 - 11.33	6.82 - 11.22

Table 5.1(b) : Various regions and Bayes threshold points for the first two training sets of the pattern classes [2,6] and [7,11]

VARIOUS GROUP OF CHOICES	SAMPLE SIZES IN EACH CLASS	
	50	**100**
Null choice	$-\infty$ - 1.2908	$-\infty$ - 1.3727
Single choice (C_1)	1.2909 - 6.4947	1.3728 - 6.4263
First-second choice (C_1)	6.4948 - 6.5402	6.4264 - 6.4494
Combined choice	6.5403 - 6.6966	6.4495 - 6.5748
First-second choice (C_2)	6.6967 - 6.7494	6.5749 - 6.5984
Single choice (C_2)	6.7495 - 11.6507	6.5985 - 11.6236
Null choice	11.6508 - $+\infty$	11.6237 - $+\infty$
Bayes threshold point	6.4859152	6.4435577

Table 5.1(c) : Various regions and Bayes threshold points for the remaining three training sets of the pattern classes [2,6] and [7,11]

VARIOUS GROUP OF CHOICES	SAMPLE SIZES IN EACH CLASS		
	150	**200**	**250**
Null choice	$-\infty$ - 1.5250	$-\infty$ - 1.6690	$-\infty$ - 1.7779
Single choice (C_1)	1.5251 - 6.4461	1.6691 - 6.3288	1.7480 - 6.2519
Null choice	6.4462 - 6.5638	6.3289 - 6.6969	6.2520 - 6.7619
Single choice (C_2)	6.5639 - 11.4736	6.6970 - 11.3288	6.7620 - 11.2496
Null choice	11.4737 - $+\infty$	11.3289 - $+\infty$	11.2197 - $+\infty$
Bayes threshold point	6.3621473	6.4193249	6.3900852

the classes C_1 and C_2 are shown in Tables 5.1(b) and 5.1(c) corresponding to the first two and the remaining three training sets respectively. These thresholds are seen to lie in the *combined choice region* of our recognition system. The Bayes classifier also recognized all the test samples correctly. ♣

5.3.2 Overlapping pattern classes

This case is shown in Fig. 5.2(a). Here the actual overlapping region between the classes C_1 and C_2 is (L_2, U_1). In the training samples, the overlapping region is (l_2, u_1).

Initially the algorithm decomposes the feature space into three *feature subdomains* as $[l_1, l_2)$, $[l_2, u_1]$, $(u_1, u_2]$ denoted by D_1, D_2 and D_3 respectively. Here the *feature subdomain* D_1 reflects only C_1, the *feature subdomain* D_3 reflects only C_2 and the *feature subdomain* D_2 is overlapping by reflecting both C_1 and C_2. These *feature subdomains* are extended to some extent and are characterized by piecewise linear triangular functions [Eq. (4.2)] T_1, T_2 and T_3. The *feature subdomains* with their membership functions are shown in Fig. 5.2(b).

The relational matrix R in this case will be

$$R = ((r_{gj}))_{\ g=1,2,3;\ j=1,2}$$

$$\text{where} \quad \begin{array}{llll} r_{11} = 1.0 & ; & r_{12} = 0 & ; \\ r_{21} > 0 & ; & r_{22} > 0 & ; \\ r_{31} = 0 & \textit{and} & r_{32} = 1.0. & \end{array}$$

In the feature region $(\Gamma_{l_1}, \Gamma_{l_2}]$, $T_1(X) > 0$ and $T_2(X) = T_3(X) = 0$. This implies that $s_1(X) > 0$ and $s_2(X) = 0$. So $(\Gamma_{l_1}, \Gamma_{l_2}]$ is the *single choice region* for C_1.

Similarly, $[\Gamma_{u_2}, \Gamma_{u_3})$ is the *single choice region* for C_2.

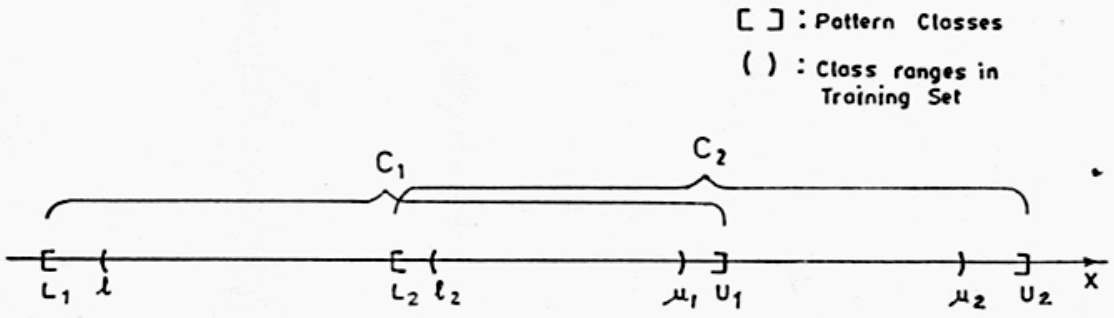


Figure 5.2(a) : Two overlapping classes(1-D) with a training set.

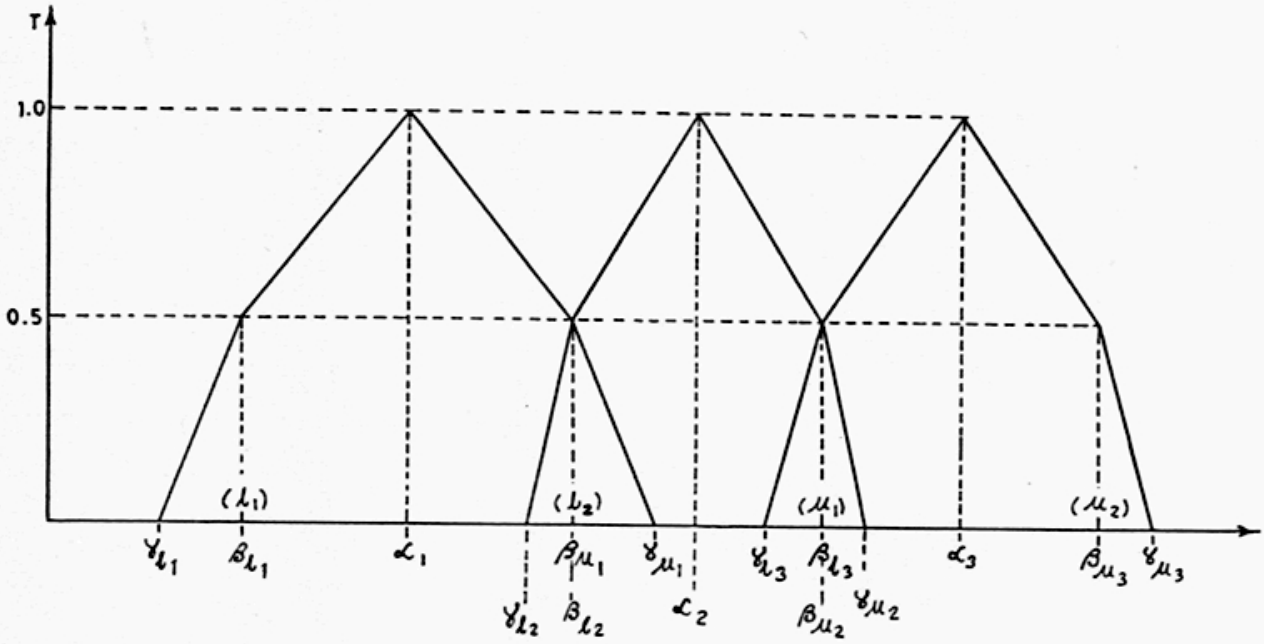


Figure 5.2(b) : *Subdomains* with their membership functions for the pattern classes in Fig. 5.2(*a*).

Let us now consider the feature region $(\Gamma_{l_2}, \beta_{l_2}]$. Here $T_1(X) \geq 0.5 \geq T_2(X) > 0 = T_3(X)$. Hence

$$s_1(X) = \tfrac{1}{2}\left[T_1(X) + \tfrac{1}{2}T_2(X) + 1.0\right] \quad \text{as } r_{11} = 1.0$$

and $$s_2(X) = \tfrac{1}{2}\left[T_2(X) + r_{22}\right].$$

$$\Rightarrow \quad s_1(X) - s_2(X) = \tfrac{1}{2}\left[T_1(X) - \tfrac{1}{2}T_2(X) + 1.0 - r_{22}\right].$$

As here $T_1(X) \geq 0.5 \geq T_2(X) > 0$ and $r_{22} < 1.0$, $[s_1(X) - s_2(X)] > 0.05$. Thus, $(\Gamma_{l_2}, \beta_{l_2}]$ comes under a *first-second choice region* with first choice as C_1.

Similarly, $[\beta_{l_3}, \Gamma_{u_2})$ comes under a *first-second choice region* with first choice as C_2.

Now consider the region $[\Gamma_{u_1}, \Gamma_{l_3}]$. Here $T_1(X) \geq 0.5$ and $T_2(X) = T_3(X) = 0$. Hence $s_1(X) = \frac{1}{2}[T_2(X) + r_{21}]$ and $s_2(X) = \frac{1}{2}[T_2(x) + r_{22}]$.
Now $\quad | s_1(X) - s_2(X) | \leq 0.05 \quad \Rightarrow \quad | r_{21} - r_{22} | \leq 0.1.$
So, if $| r_{21} - r_{22} | \leq 0.1$ then $(\Gamma_{u_1}, \Gamma_{l_3}]$ will be a *combined choice region.* If $r_{21} > r_{22} + 0.1$, then $(\Gamma_{u_1}, \Gamma_{l_3}]$ will be a *first-second choice region* with first choice as C_1. Otherwise (i.e., if $r_{22} > r_{21} + 0.1$), it becomes a *first-second choice region* with first choice as C_2.

Now in the region $(\beta_{u_1}, \Gamma_{u_1})$, $T_2(X) \geq 0.5 \geq T_1(X) \geq 0 = T_3(X)$.
Hence $\quad s_1(X) = \frac{1}{2}\left[T_2(X) + \frac{1}{2}T_1(X) + r_{21}\right]$ and $s_2(X) = \frac{1}{2}[T_2(X) + r_{22}]$.
$\Rightarrow \quad s_1(X) - s_2(X) = \frac{1}{2}\left[\frac{1}{2}T_1(X) + r_{21} - r_{22}\right]$.
Here a point τ_1 may be found such that $[s_1(\tau_1) - s_2(\tau_1)] = 0.05$ where

$$\tau_1 = l_2 + \varepsilon_1 [4(r_{21} - r_{22}) + 0.6] \tag{5.6}$$

in which ε_1 [Eq. (2.3)] is the extended portion of the *feature subdomain* D_1 and l_2 is the lowest among the training samples from C_2.

Thus, (β_{u_1}, τ_1) is a *first-second choice region* with first choice as C_1 and $[\tau_1, \Gamma_{u_1})$ is a *combined choice region.* If $r_{22} \geq r_{21} + 0.15$, then the complete set $(\beta_{u_1}, \Gamma_{u_1})$ will be a *combined choice region.* Again, if $r_{21} \geq r_{22} + 0.1$, then $(\beta_{u_1}, \Gamma_{u_1})$ will be a *first-second choice region* with first choice as C_1.

Similarly in the region $(\Gamma_{l_3}, \beta_{l_3})$, a point τ_2 may be found such that $[s_2(\tau_2) - s_1(\tau_2)] = 0.05$ where

$$\tau_2 = u_1 - \epsilon_3 [4(r_{22} - r_{21}) + 0.6] \tag{5.7}$$

in which ϵ_3 is the extended portion of the *feature subdomain* D_3 and u_1 is the highest among the training samples from C_1.

So, (τ_2, β_{l_3}) is a *first-second choice region* with first choice as C_2 and $(\Gamma_{l_3}, \tau_2]$ is a *combined choice region.* If $r_{21} \geq r_{22} + 0.15$, then the complete set $(\Gamma_{l_3}, \beta_{l_3})$ will be a *combined choice region.* Again, if $r_{22} \geq r_{21} + 0.1$, then $(\Gamma_{l_3}, \beta_{l_3})$ will be a *first-second choice region* with first choice as C_2.

The conclusions, after combining all the previous results, are given below:

The *single choice region* for C_1 is $(\Gamma_{l_1}, \Gamma_{l_2}]$; the *single choice region* for C_2 is $[\Gamma_{u_2}, \Gamma_{u_3})$; the overlapping region (i.e., corresponding to *combined* and *first-second choices*) is $[\Gamma_{l_2}, \Gamma_{u_2}]$ and the remaining portion in the feature space represent the no-class region.

Note that in the overlapping region, $(\Gamma_{l_2}, \beta_{l_2}]$ is a *first-second choice region* with first choice as C_1 and $[\beta_{u_2}, \Gamma_{u_2}]$ is a *first-second choice region* with first choice as C_2.

In the region $(\Gamma_{u_1}, \Gamma_{l_3}]$, if $| r_{21} - r_{22} | \leq 0.1$, then it becomes a *combined choice region.* If $r_{21} > r_{22} + 0.1$, then $(\Gamma_{u_1}, \Gamma_{l_3}]$ becomes a *first-second choice region* with first choice as C_1, and if $r_{22} > r_{21} + 0.1$, then $(\Gamma_{u_1}, \Gamma_{l_3}]$ becomes a *first-second choice region* with first choice as C_2.

In the region $(\beta_{u_1}, \Gamma_{u_1})$, if $r_{22} \geq r_{21} + 0.15$, then it becomes a *combined choice region,* and if $r_{21} \geq r_{22} + 0.1$, then it comes under a *first-second choice region* with first choice as C_1. Otherwise (i.e., if $r_{21} - 0.1 < r_{22} < r_{21} + 0.15$), a point τ_1 [Eq. (5.6)] is found such that (β_{u_1}, τ_1) becomes a *first-second choice region* with first choice as C_1 and $[\tau_1, \Gamma_{u_1})$ becomes a *combined choice*

region.

In the region $(\Gamma_{l_3}, \beta_{l_3})$, if $r_{21} \geq r_{22} + 0.15$, then it becomes a *combined choice region.* If $r_{22} \geq r_{21} + 0.1$, then $(\Gamma_{l_3}, \beta_{l_3})$ will be a *first-second choice region* with first choice as C_2. Otherwise (i.e., if $r_{22} - 0.1 < r_{21} < r_{22} + 0.15$), a point τ_2 [Eq. (5.7)] is found such that (τ_2, β_{l_3}) will be a *first-second choice region* with first choice as C_2 and $(\Gamma_{l_3}, \tau_2]$ will be a *combined choice region.*

Proof of Proposition 5.1 :

Here, the actual overlapping region is $[L_2, U_2]$ and the non-overlapping regions for C_1 and C_2 are (L_1, L_2) and (U_1, U_2) respectively. In the training set, the region $[\Gamma_{l_2}, \Gamma_{u_1}]$ is overlapping between C_1 and C_2, and the regions $(\Gamma_{l_1}, \Gamma_{l_2})$ and $(\Gamma_{u_1}, \Gamma_{u_2})$ are non-overlapping for C_1 and C_2 respectively.

By Theorem 5.1, it can be stated that as the training sample size increases

$$
\begin{aligned}
& l_1 = \beta_{l_1} \rightarrow L_1 && \text{in probability} \\
& l_2 = \beta_{u_1} = \beta_{l_2} \rightarrow L_2 && \text{in probability} \\
& u_1 = \beta_{l_3} = \beta_{u_2} \rightarrow U_1 && \text{in probability} \\
\text{and} \quad & u_2 = \beta_{u_3} \rightarrow U_1 && \text{in probability.}
\end{aligned}
$$

With the increase in the size of training samples, the number of samples in the *feature subdomains* also increases. Then, by Theorem 5.2,

$$
\begin{aligned}
& \Gamma_{l_g} \rightarrow \beta_{l_g} \text{ and } \Gamma_{u_g} \rightarrow \beta_{u_g} && \text{in probability} && \text{for } g = 1, 2, 3. \\
\Rightarrow \quad & \Gamma_{l_2} \rightarrow \beta_{l_2} = \beta_{u_1} = l_2 \rightarrow L_2 && \text{in probability} \\
\text{and} \quad & \Gamma_{u_2} \rightarrow \beta_{u_2} = \beta_{l_3} = u_1 \rightarrow U_1 && \text{in probability.}
\end{aligned}
$$

Therefore, with the increase of the training sample size, the estimated overlapping region $[\Gamma_{l_2}, \Gamma_{u_1}]$ tends toward the actual overlapping region $[L_2, U_2]$ in probability. At the same time, the estimated non-overlapping (*single choice*) regions $(\Gamma_{l_1}, \Gamma_{l_2})$ and $(\Gamma_{u_1}, \Gamma_{u_2})$ for the classes C_1 and C_2 tend

toward their actual sizes (L_1, L_2) and (U_1, U_2) respectively in probability. The no-class region also tends to its actual size.

Hence the proof. •

Experimental results :

To substantiate the analytical results, a two class problem with classes $[2, 7]$ and $[5, 10]$ is considered. Here, the actual non-overlapping regions for the classes C_1 and C_2 are $[2, 5]$ and $[7, 10]$ respectively, and the actual overlapping region between the classes is $[5, 7]$. To implement the recognition system, five training sample sets with 50, 100, 150, 200 and 250 samples from each of the pattern classes are chosen randomly.

The ranges of the training samples of the classes C_1 and C_2, and the ranges of the obtained three *feature subdomains* D_1, D_2 and D_3 for the considered five sample sets are shown in Table 5.2(a). The regions obtained corresponding to various output choices are shown in Tables 5.2(b) and 5.2(c) for the first two (with 50 and 100 samples from each class) and the remaining three (with 150, 200 and 250 samples from each class) sample sets respectively.

These results substantiate that with the increase in sample size, the training classes (i.e., the portions covered by the training samples) tend to their actual classes (Theorem 5.1) and the *feature subdomains* with the extended portions tend to their actual sizes (Theorem 5.2). It is to be noticed that the overlapping and the non-overlapping regions are tending to their actual sizes with the increase in sample sizes. Hence *Proposition* 5.1 is verified experimentally for the case of overlapping pattern classes in one dimensional feature space.

For analyzing the recognition performance of the system, 1000 test samples from each of the pattern classes are generated. The recognition scores

Table 5.2(a) : Ranges of training sets and *feature sub-domains* for the pattern classes [2,7] and [5,10]

	SAMPLE SIZES IN EACH CLASS				
	50	**100**	**150**	**200**	**250**
Class C_1	2.11 - 6.88	2.08 - 6.93	2.06 - 5.93	2.04 - 5.96	2.02 - 5.99
Class C_2	5.13 - 9.89	5.12 - 9.92	5.07 - 9.94	5.03 - 9.95	5.01 - 9.99
Subdomain D_1	1.34 - 5.64	1.43 - 5.58	1.56 - 5.44	1.69 - 5.31	1.77 - 5.21
Subdomain D_2	4.57 - 7.35	4.68 - 7.32	4.76 - 7.24	4.83 - 7.16	4.86 - 7.13
Subdomain D_3	6.26 - 10.63	6.40 - 10.54	6.55 - 10.40	6.68 - 10.29	6.79 - 10.22

Table 5.2(b) : Various regions and Bayes threshold points for the first two training sets of the pattern classes [2,7] and [5,10]

VARIOUS GROUP OF CHOICES	SAMPLE SIZES IN EACH CLASS	
	50	**100**
Null choice	$-\infty$ - 1.3402	$-\infty$ - 1.4274
Single choice (C_1)	1.3403 - 4.5741	1.4275 - 4.6752
First-second choice (C_1)	4.5742 - 5.3090	4.6753 - 5.3175
Combined choice	5.3091 - 6.4549	5.3176 - 6.6060
First-second choice (C_2)	6.4550 - 7.3457	6.6061 - 7.3173
Single choice (C_2)	7.3458 - 10.6274	7.3174 - 10.5438
Null choice	10.6275 - $+\infty$	10.5439 - $+\infty$
Bayes threshold point	5.7445097	5.8655739

Table 5.2(c) : Various regions and Bayes threshold points for the remaining three training sets of the pattern classes [2,7] and [5,10]

VARIOUS GROUP OF CHOICES	SAMPLE SIZES IN EACH CLASS		
	150	**200**	**250**
Null choice	$-\infty$ - 1.5625	$-\infty$ - 1.6921	$-\infty$ - 1.7684
Single choice (C_1)	1.5626 - 4.7565	1.6922 - 4.8287	1.7685 - 4.8549
First-second choice (C_1)	4.7566 - 5.2758	4.8288 - 5.1997	4.8550 - 5.1318
Combined choice	5.2759 - 6.7329	5.1998 - 6.8013	5.1319 - 6.8505
First-second choice (C_2)	6.7330 - 7.2361	6.8014 - 7.1621	6.8506 - 7.1279
Single choice (C_2)	7.2362 - 10.3957	7.1622 - 10.2906	7.1280 - 10.2228
Null choice	210.3958 - $+\infty$	110.2907 - $+\infty$	10.2229 - $+\infty$
Bayes threshold point	5.9670115	5.9635067	5.9716177

Table 5.2(d) : Recognition score for the pattern classes [2,7] and [5,10]

VARIOUS GROUP OF CHOICES		% RECOGNITION SCORE SAMPLE SIZES IN EACH CLASS				
		50	100	150	200	250
Single Correct Choice		55.40	58.30	60.15	61.60	63.05
First Correct Choice		16.85	15.50	13.15	9.60	7.25
Combined Correct Choice		18.60	18.55	21.10	24.10	26.50
Second Correct Choice		9.15	7.65	5.60	4.70	3.20
Fully Wrong Choice		0.00	0.00	0.00	0.00	0.00
BAYES CLASSIFIER	*Correct*	79.00	79.35	79.30	79.30	79.25
	Wrong	21.00	20.65	20.70	20.70	20.75

corresponding to the considered five training sets are shown in Table 5.2(*d*). Note that the recognition scores are grouped into five categories, namely *single correct choice*, *first-correct choice*, *combined correct choice*, *second correct choice* and *fully wrong choice*. The meaning of these five categories has already been explained in the previous chapter (section 4.5.1).

For comparison, the Bayes classifier is implemented on the same pattern classes (assuming normal distributions). The threshold points found between the classes C_1 and C_2 are shown in Tables 5.2(*b*) and 5.2(*c*) corresponding to the considered five training sets. The recognition scores of the Bayes classifier are included in Table 5.2(*d*).

Note from Tables 5.2(*b*) and 5.2(*c*) that the Bayes threshold points lie in the *combined choice region* of our recognition system. By adding up the scores corresponding to *single correct*, *first-second correct* and half of the *combined correct choices* in Table 5.2(*d*), the recognition score becomes higher than the corresponding correct choices of Bayes classifier. Again, the wrong choices of Bayes classifier are found to be distributed in the *combined correct* and *second correct choices* of the proposed system. Therefore, the proposed recognition system has a provision of improving its efficiency significantly by

incorporating *combined* and *second choices* under the control of a supervisory scheme. ♠

5.4 Analysis in 2-D Feature Space

Let us consider a two-class (C_1 and C_2) problem to analyze the performance of the multivalued recognition system in 2-dimensional feature space ($F_1 \times F_2$). For the sake of convenience, the classes are initially assumed to be of rectangular shape. Then the results are extended to circular pattern classes. These results can easily be generalized to the pattern classes of any shape.

5.4.1 Rectangular classes

Suppose $[L_{11}, U_{11}] \times [L_{21}, U_{21}]$ and $[L_{12}, U_{12}] \times [L_{22}, U_{22}]$ denote the classes C_1 and C_2 respectively. Here $[L_{12}, U_{11}] \times [L_{22}, U_{21}]$ is the overlapping portion between the classes. Suppose also that $[l_{11}, u_{11}] \times [l_{21}, u_{21}]$ and $[l_{12}, u_{12}] \times [l_{22}, u_{22}]$ denote the training sets corresponding to C_1 and C_2 respectively where $[l_{12}, u_{11}] \times [l_{22}, u_{21}]$ is the overlapping portion in the training set. Such pattern classes with the span of their training sets are shown in Fig. 5.3(a).

Initially, based on the training set, each individual feature domain is partitioned into three *feature subdomains* [Fig. 5.3(b)]. Recall that the gth ($g = 1, 2, 3$) *feature subdomain* in the ith ($i = 1, 2$) feature axis is denoted by D_{ig}. As a result, the total feature space is decomposed into nine *space subdomains*, which are denoted by SD_1, SD_2, ..., SD_9 respectively [Fig. 5.3(b)]. Here SD_1, SD_2 and SD_4 uniquely correspond to the class C_1; SD_6, SD_8 and SD_9 uniquely correspond to C_2; SD_5 is overlapping by reflecting both C_1 and C_2, and SD_3 and SD_5 are the no-class regions i.e., they reflect neither C_1 nor C_2.

Thus, the *feature subdomains* along F_1 and F_2 are extended to some extent

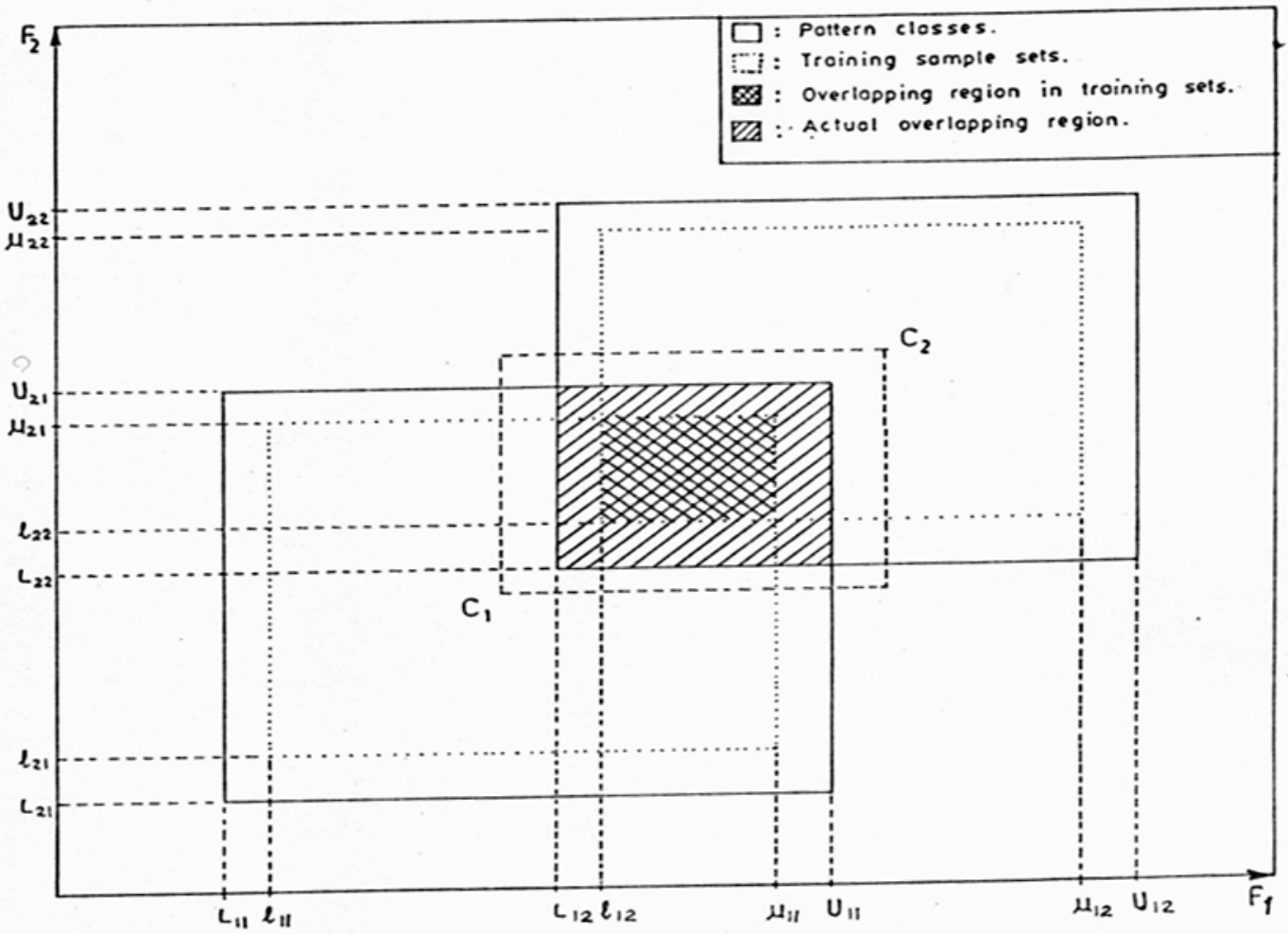


Figure 5.3(a) : Two rectangular classes with a training set.

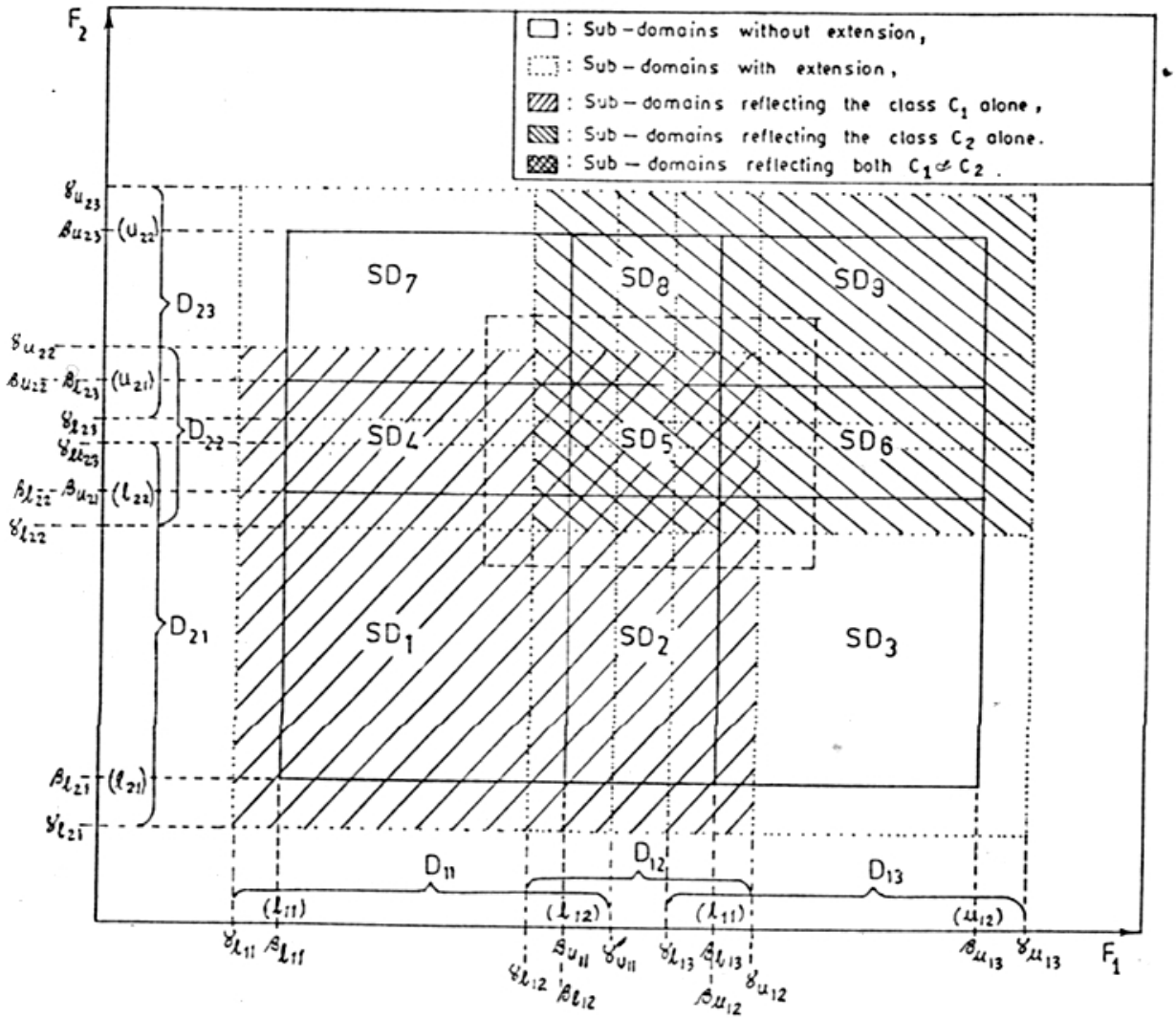


Figure 5.3(b) : *Subdomains* and choice regions for the pattern classes in Fig. 5.3(*a*).

to highlight the portions possibly uncovered by the training samples. These *feature subdomains* are characterized by different piecewise linear triangular functions of the form $T_{ig}(x_i;\ \alpha_{ig},\ \beta_{l_{ig}},\ \beta_{u_{ig}},\ \Gamma_{l_{ig}},\ \Gamma_{u_{ig}})$ $(i = 1,2;\ g = 1,2,3)$ [Eq. (4.2)]. The *feature subdomains* and the *space subdomains* are shown in Fig. 5.3(b). Note that

$$\begin{aligned} \beta_{l_{i1}} &= l_{i1} \quad ; & \beta_{u_{i3}} &= u_{i2} \quad ; \\ \beta_{l_{i2}} &= \beta_{u_{i1}} = l_{i2} \quad ; & \beta_{l_{i3}} &= \beta_{u_{i2}} = u_{i1} \quad ; \\ \text{and} \quad \varepsilon_{ig} &= (\beta_{u_{ig}} - \beta_{l_{ig}})\delta_{ig} \end{aligned}$$

where ε_{ig} [Eq. (2.3)] and δ_{ig} [Eq. (2.1)] are the extended portion and accuracy factor respectively for the gth $(g = 1,2,3)$ *feature subdomain* in the ith $(i = 1,2)$ feature axis. It is obvious from diagram 5.3(b) that $[\Gamma_{l_{11}},\ \Gamma_{l_{12}}] \times [\Gamma_{l_{21}},\ \Gamma_{u_{22}}]$ and $[\Gamma_{l_{12}},\ \Gamma_{u_{12}}] \times [\Gamma_{l_{21}},\ \Gamma_{l_{22}}]$ are the *single choice regions* for C_1, and $[\Gamma_{l_{12}},\ \Gamma_{l_{13}}] \times [\Gamma_{u_{22}},\ \Gamma_{u_{23}}]$ and $[\Gamma_{u_{12}},\ \Gamma_{u_{13}}] \times [\Gamma_{l_{22}},\ \Gamma_{u_{22}}]$ are the *single choice regions* for C_2.

If $X \in [\Gamma_{l_{12}},\ \beta_{l_{12}}] \times [\Gamma_{l_{22}},\ \beta_{u_{22}}]$ or $X \in [\Gamma_{l_{12}},\ \beta_{u_{12}}] \times [\Gamma_{l_{22}},\ \beta_{l_{22}}]$, the elements in $S(X)$ are positive for both the classes i.e., $s_1(X) > 0$ and $s_2(X) > 0$. It can be easily shown that $[s_1(X) - s_2(X)] > 0.05$ for X's lying in these regions. This implies that $[\Gamma_{l_{12}},\ \beta_{l_{12}}] \times [\Gamma_{l_{22}},\ \beta_{u_{22}}]$ and $[\Gamma_{l_{12}},\ \beta_{u_{12}}] \times [\Gamma_{l_{22}},\ \beta_{l_{22}}]$ are the *first-second choice regions* with first choice as C_1. Similarly, $[\Gamma_{l_{12}},\ \beta_{l_{12}}] \times [\Gamma_{l_{22}},\ \beta_{u_{22}}]$ and $[\Gamma_{l_{12}},\ \beta_{u_{12}}] \times [\Gamma_{l_{22}},\ \beta_{l_{22}}]$ are the *first-second choice regions* with first choice as C_2.

Now consider the feature region $[\Gamma_{l_{12}},\ \beta_{l_{12}}] \times [\beta_{u_{22}},\ \Gamma_{u_{22}}]$ which is overlapping. That is, $s_1(X) > 0$ and $s_2(X) > 0$. In this region, two lines Z_1 and Z_2 may be found so that

for X lying below the line Z_1, $[s_1(X) - s_2(X)] > 0.05$,
for X lying above the line Z_2, $[s_2(X) - s_1(X)] > 0.05$ and
for X lying between Z_1 and Z_2, $| s_1(X) - s_2(X) | \leq 0.05$.

The equations of Z_1 and Z_2 are

$$\begin{aligned} A_1x_1 + B_1x_2 - \zeta_1 &= 0 \\ \text{and} \quad A_2x_1 + B_2x_2 - \zeta_2 &= 0 \end{aligned} \tag{5.8}$$

where $A_1 = A_2 = k_2k_3k_4 - k_1k_3k_4$;

$B_1 = B_2 = k_1k_2k_4 - k_1k_2k_3$

$$\begin{aligned}\varsigma_1 = \beta_{u_{11}}k_2k_3k_4 &- \Gamma_{l_{12}}k_1k_3k_4 + \Gamma_{u_{22}}k_1k_2k_4 \\ &- \beta_{l_{23}}k_1k_2k_3 + 0.4k_1k_2k_3k_4 \ ; \\ \varsigma_2 = \beta_{u_{11}}k_2k_3k_4 &- \Gamma_{l_{12}}k_1k_3k_4 + \Gamma_{u_{22}}k_1k_2k_4 \\ &- \beta_{l_{23}}k_1k_2k_3 - 0.4k_1k_2k_3k_4\end{aligned}$$

with $k_1 = \alpha_{11} - \beta_{u_{11}}$; $k_2 = \beta_{l_{12}} - \Gamma_{l_{12}}$;

$k_3 = \beta_{u_{22}} - \Gamma_{u_{22}}$ and $k_4 = \alpha_{23} - \beta_{l_{23}}$.

Similarly, $[\beta_{u_{12}}, \Gamma_{u_{12}}] \times [\Gamma_{l_{22}}, \beta_{l_{22}}]$ is a overlapping region. Here also, two lines Z_3 and Z_4 may be found such that

for X lying below the line Z_3, $[s_1(X) - s_2(X)] > 0.05$,

for X lying above the line Z_4, $[s_2(X) - s_1(X)] > 0.05$ and

for X lying between Z_3 and Z_4, $| s_1(X) - s_2(X) | \leq 0.05$.

The equations of Z_3 and Z_4 are

$$\begin{aligned} A_3x_1 + B_3x_2 - \varsigma_3 &= 0 \\ \text{and} \quad A_4x_1 + B_4x_2 - \varsigma_4 &= 0 \end{aligned} \tag{5.9}$$

where $A_3 = A_4 = k_2k_3k_4 - k_1k_3k_4$;

$B_3 = B_4 = k_1k_2k_4 - k_1k_2k_3$;

$$\begin{aligned}\varsigma_3 = \beta_{u_{12}}k_2k_3k_4 &- \Gamma_{l_{13}}k_1k_3k_4 + \beta_{u_{21}}k_1k_2k_4 \\ &- \Gamma_{l_{22}}k_1k_2k_3 + 0.4k_1k_2k_3k_4 \ ; \\ \varsigma_4 = \beta_{u_{12}}k_2k_3k_4 &- \Gamma_{l_{13}}k_1k_3k_4 + \beta_{u_{21}}k_1k_2k_4 \\ &- \Gamma_{l_{22}}k_1k_2k_3 - 0.4k_1k_2k_3k_4\end{aligned}$$

with $k_1 = \beta_{u_{12}} - \Gamma_{u_{12}}$; $k_2 = \alpha_{13} - \beta_{l_{13}}$;

$k_3 = \alpha_{21} - \beta_{u_{21}}$ and $k_4 = \beta_{l_{22}} - \Gamma_{l_{22}}$.

Let us now consider the region $[\beta_{u_{11}}, \Gamma_{u_{11}}] \times [\Gamma_{l_{23}}, \beta_{l_{23}}]$ which is overlapping. In this region, two lines Z_5 and Z_6 may be found so that

for X lying below the line Z_5, $[s_1(X) - s_2(X)] > 0.05$,

for X lying above the line Z_6, $[s_2(X) - s_1(X)] > 0.05$ and

for X lying between Z_5 and Z_6, $| s_1(X) - s_2(X) | \leq 0.05$.

The equations of Z_5 and Z_6 are

$$\begin{aligned} A_5x_1 + B_5x_2 - \varsigma_5 &= 0 \\ \text{and} \quad A_6x_1 + B_6x_2 - \varsigma_6 &= 0 \end{aligned} \tag{5.10}$$

where
$$\begin{aligned} A_5 &= A_6 = k_2k_3k_4 - k_1k_3k_4 \ ; \\ B_5 &= B_6 = k_1k_2k_4 - k_1k_2k_3 \ ; \\ \varsigma_5 &= \Gamma_{u_{11}}k_2k_3k_4 - \beta_{l_{12}}k_1k_3k_4 + \beta_{u_{22}}k_1k_2k_4 \\ &\quad - \Gamma_{l_{23}}k_1k_2k_3 + [1.6 - 16(r_{51} - r_{52})]k_1k_2k_3k_4 \ ; \\ \varsigma_6 &= \Gamma_{u_{11}}k_2k_3k_4 - \beta_{l_{12}}k_1k_3k_4 + \beta_{u_{22}}k_1k_2k_4 \\ &\quad - \Gamma_{l_{23}}k_1k_2k_3 - [1.6 + 16(r_{51} - r_{52})]k_1k_2k_3k_4 \end{aligned}$$

with
$$\begin{aligned} k_1 &= \beta_{u_{11}} - \Gamma_{u_{11}} \ ; \quad k_2 = \alpha_{12} - \beta_{l_{12}} \ ; \\ k_3 &= \alpha_{22} - \beta_{l_{22}} \quad \text{and} \quad k_4 = \beta_{l_{23}} - \Gamma_{l_{23}} \ . \end{aligned}$$

Similarly, in the overlapping region $[\Gamma_{l_{13}}, \beta_{l_{13}}] \times [\beta_{u_{21}}, \Gamma_{u_{21}}]$, two lines Z_7 and Z_8 may be found so that

for X lying below the line Z_7, $[s_1(X) - s_2(X)] > 0.05$,

for X lying above the line Z_8, $[s_2(X) - s_1(X)] > 0.05$ and

for X lying between Z_7 and Z_8, $| s_1(X) - s_2(X) | \leq 0.05$.

The equations of Z_7 and Z_8 are

$$\begin{aligned} A_7x_1 + B_7x_2 - \varsigma_7 &= 0 \\ \text{and} \quad A_8x_1 + B_8x_2 - \varsigma_8 &= 0 \end{aligned} \tag{5.11}$$

where
$$\begin{aligned} A_7 &= A_8 = k_2k_3k_4 - k_1k_3k_4 \ ; \\ B_7 &= B_8 = k_1k_2k_4 - k_1k_2k_3 \ ; \\ \varsigma_7 &= \beta_{u_{12}}k_2k_3k_4 - \Gamma_{l_{13}}k_1k_3k_4 + \Gamma_{u_{21}}k_1k_2k_4 \\ &\quad - \beta_{l_{22}}k_1k_2k_3 + [1.6 - 16(r_{51} - r_{52})]k_1k_2k_3k_4 \ ; \\ \varsigma_8 &= \beta_{u_{12}}k_2k_3k_4 - \Gamma_{l_{13}}k_1k_3k_4 + \Gamma_{u_{21}}k_1k_2k_4 \\ &\quad - \beta_{l_{22}}k_1k_2k_3 - [1.6 + 16(r_{51} - r_{52})]k_1k_2k_3k_4 \end{aligned}$$

with
$$\begin{aligned} k_1 &= \alpha_{12} - \beta_{u_{12}} \ ; \quad k_2 = \beta_{l_{13}} - \Gamma_{l_{13}} \ ; \\ k_3 &= \beta_{u_{21}} - \Gamma_{u_{21}} \quad \text{and} \quad k_4 = \alpha_{22} - \beta_{l_{22}} \ . \end{aligned}$$

Now consider the region $[\beta_{u_{11}}, \Gamma_{u_{11}}] \times [\alpha_{22}, \Gamma_{l_{23}}]$. Here, a line Z_9 may be found so that

for X lying below the line Z_9, $[s_1(X) - s_2(X)] > 0.05$ and
for X lying above the line Z_9, $| s_1(X) - s_2(X) | \leq 0.05$.

The equation of Z_9 is

$$A_9 x_1 + B_9 x_2 - \varsigma_9 = 0 \tag{5.12}$$

where $A_9 = k_2$; $B_9 = k_1$;

$$\varsigma_9 = \Gamma_{u_{11}} k_2 + \beta_{u_{22}} k_1 + [0.6 - 16(r_{51} - r_{52})] k_1 k_2$$

with $k_1 = \beta_{u_{11}} - \Gamma_{u_{11}}$ and $k_2 = \alpha_{22} - \beta_{u_{22}}$.

Similarly, in the region $[\beta_{u_{11}}, \Gamma_{u_{11}}] \times [\Gamma_{u_{21}}, \alpha_{22}]$, a line Z_{10} may be found such that

for X lying above the line Z_{10}, $[s_1(X) - s_2(X)] > 0.05$ and
for X lying below the line Z_{10}, $| s_1(X) - s_2(X) | \leq 0.05$.

The equation of Z_{10} is

$$A_{10} x_1 + B_{10} x_2 - \varsigma_{10} = 0 \tag{5.13}$$

where $A_{10} = k_2$; $B_{10} = k_1$;

$$\varsigma_{10} = \Gamma_{u_{11}} k_2 + \beta_{l_{22}} k_1 + [0.6 - 16(r_{51} - r_{52})] k_1 k_2$$

with $k_1 = \beta_{u_{11}} - \Gamma_{u_{11}}$ and $k_2 = \alpha_{22} - \beta_{u_{22}}$.

In the regions $[\beta_{u_{11}}, \alpha_{12}] \times [\beta_{u_{21}}, \Gamma_{u_{21}}]$ and $[\alpha_{12}, \Gamma_{l_{13}}] \times [\beta_{u_{21}}, \Gamma_{u_{21}}]$, two lines Z_{11} and Z_{12} are found such that

for X lying below both Z_{11} and Z_{12}, $[s_1(X) - s_2(X)] > 0.05$ and
for X lying above Z_{11} or Z_{12}, $| s_1(X) - s_2(X) | \leq 0.05$.

The equations of Z_{11} and Z_{12} are

$$\begin{aligned} A_{11} x_1 + B_{11} x_2 - \varsigma_{11} &= 0 \\ \text{and} \quad A_{12} x_1 + B_{12} x_2 - \varsigma_{12} &= 0 \end{aligned} \tag{5.14}$$

where $A_{11} = A_{12} = k_2$;

$B_{11} = k_1$ and $B_{12} = \hat{k}_1$;

$$\varsigma_{11} = \beta_{l_{12}} k_2 + \Gamma_{u_{21}} k_1 + [0.6 - 16(r_{51} - r_{52})] k_1 k_2 \ ;$$

$$\varsigma_{12} = \beta_{u_{12}} k_2 + \Gamma_{u_{21}} \hat{k}_1 + [0.6 - 16(r_{51} - r_{52})] \hat{k}_1 k_2$$

with $k_1 = \alpha_{12} - \beta_{l_{12}}$; $k_2 = \beta_{u_{21}} - \Gamma_{u_{21}}$

and $\hat{k}_1 = \alpha_{12} - \beta_{u_{12}}$.

In the regions $[\Gamma_{l_{13}}, \beta_{l_{13}}] \times [\Gamma_{u_{21}}, \alpha_{22}]$ and $[\Gamma_{l_{13}}, \beta_{l_{13}}] \times [\alpha_{22}, \Gamma_{l_{23}}]$, two lines Z_{13} and Z_{14} may be found in their respective regions such that

for X lying right to both Z_{13} and Z_{14}, $[s_2(X) - s_1(X)] > 0.05$ and

for X lying left to Z_{13} or Z_{14}, $| s_1(X) - s_2(X) | \leq 0.05$.

The equations of Z_{13} and Z_{14} are

$$\begin{aligned} A_{13}x_1 + B_{13}x_2 - \varsigma_{13} &= 0 \\ \text{and} \quad A_{14}x_1 + B_{14}x_2 - \varsigma_{14} &= 0 \end{aligned} \tag{5.15}$$

where $A_{13} = k_2$; $A_{14} = \hat{k}_2$;

$B_{13} = B_{14} = k_1$;

$\varsigma_{13} = \Gamma_{l_{13}} k_2 + \beta_{l_{22}} k_1 + [0.6 - 16(r_{52} - r_{51})] k_1 k_2$;

$\varsigma_{14} = \Gamma_{l_{13}} \hat{k}_2 + \beta_{u_{22}} k_1 + [0.6 - 16(r_{52} - r_{51})] k_1 \hat{k}_2$

with $k_1 = \beta_{l_{13}} - \Gamma_{l_{13}}$; $k_2 = \alpha_{22} - \beta_{l_{22}}$

and $\hat{k}_2 = \alpha_{22} - \beta_{u_{22}}$.

Similarly, in the regions $[\Gamma_{u_{11}}, \alpha_{12}] \times [\Gamma_{l_{23}}, \beta_{l_{23}}]$ and $[\alpha_{12}, \Gamma_{l_{13}}] \times [\Gamma_{l_{23}}, \beta_{l_{23}}]$, two lines Z_{15} and Z_{16} may be found in their respective regions such that

for X lying above both Z_{15} and Z_{16}, $[s_2(X) - s_1(X)] > 0.05$ and

for X lying below Z_{15} or Z_{16}, $| s_1(X) - s_2(X) | \leq 0.05$.

The equations of Z_{15} and Z_{16} are

$$\begin{aligned} A_{15}x_1 + B_{15}x_2 - \varsigma_{15} &= 0 \\ \text{and} \quad A_{16}x_1 + B_{16}x_2 - \varsigma_{16} &= 0 \end{aligned} \tag{5.16}$$

where $A_{15} = A_{16} = k_2$;

$B_{15} = k_1$; $B_{16} = \hat{k}_1$;

$\varsigma_{15} = \beta_{l_{12}} k_2 + \Gamma_{l_{23}} k_1 + [0.6 - 16(r_{52} - r_{51})] k_1 k_2$;

$\varsigma_{16} = \beta_{u_{12}} k_2 + \Gamma_{l_{23}} \hat{k}_1 + [0.6 - 16(r_{52} - r_{51})] \hat{k}_1 k_2$

with $k_1 = \alpha_{12} - \beta_{l_{12}}$; $k_2 = \beta_{l_{23}} - \Gamma_{l_{23}}$

and $\hat{k}_1 = \alpha_{12} - \beta_{u_{12}}$.

Let us now consider the region $[\beta_{u_{11}}, \Gamma_{u_{11}}] \times [\beta_{u_{21}}, \Gamma_{u_{21}}]$. Here two lines Z_{17} and Z_{18} may be found such that

for X lying below both Z_{17} and Z_{18}, $[s_1(X) - s_2(X)] > 0.05$ and
for X lying above Z_{17} or Z_{18}, $| s_1(X) - s_2(X) | \leq 0.05$.

The equations of Z_{17} and Z_{18} are

$$\begin{aligned} A_{17}x_1 + B_{17}x_2 - \varsigma_{17} &= 0 \\ \text{and} \quad A_{18}x_1 + B_{18}x_2 - \varsigma_{18} &= 0 \end{aligned} \tag{5.17}$$

where $A_{17} = k_2$; $A_{18} = \hat{k}_2$;
$B_{17} = k_1$; $B_{18} = \hat{k}_1$;
$\varsigma_{17} = \Gamma_{u_{11}} k_2 + \beta_{l_{22}} k_1 + [0.6 - 16(r_{51} - r_{52})] k_1 k_2$;
$\varsigma_{18} = \beta_{l_{12}} \hat{k}_2 + \Gamma_{u_{21}} \hat{k}_1 + [0.6 - 16(r_{51} - r_{52})] \hat{k}_1 \hat{k}_2$
with $k_1 = \beta_{u_{11}} - \Gamma_{u_{11}}$; $k_2 = \alpha_{22} - \beta_{l_{22}}$;
$\hat{k}_1 = \alpha_{12} - \beta_{l_{12}}$ and $\hat{k}_2 = \beta_{u_{21}} - \Gamma_{u_{21}}$.

Again, in the region $[\Gamma_{l_{13}}, \beta_{l_{13}}] \times [\Gamma_{l_{23}}, \beta_{l_{23}}]$, two lines Z_{19} and Z_{20} may be found such that

for X lying above both Z_{19} and Z_{20}, $[s_2(X) - s_1(X)] > 0.05$ and
for X lying below Z_{19} or Z_{20}, $| s_1(X) - s_2(X) | \leq 0.05$.

The equations of Z_{19} and Z_{20} are

$$\begin{aligned} A_{19}x_1 + B_{19}x_2 - \varsigma_{19} &= 0 \\ \text{and} \quad A_{20}x_1 + B_{20}x_2 - \varsigma_{20} &= 0 \end{aligned} \tag{5.18}$$

where $A_{19} = k_2$; $A_{20} = \hat{k}_2$;
$B_{19} = k_1$; $B_{20} = \hat{k}_1$;
$\varsigma_{19} = \Gamma_{l_{13}} k_2 + \beta_{u_{22}} k_1 + [0.6 - 16(r_{52} - r_{51})] k_1 k_2$;
$\varsigma_{20} = \beta_{u_{12}} \hat{k}_2 + \Gamma_{u_{23}} \hat{k}_1 + [0.6 - 16(r_{52} - r_{51})] \hat{k}_1 \hat{k}_2$
with $k_1 = \beta_{l_{13}} - \Gamma_{l_{13}}$; $k_2 = \alpha_{22} - \beta_{u_{22}}$;
$\hat{k}_1 = \alpha_{12} - \beta_{u_{12}}$ and $\hat{k}_2 = \beta_{l_{23}} - \Gamma_{l_{23}}$.

At last, let us consider the region $[\Gamma_{u_{11}}, \Gamma_{l_{13}}] \times [\Gamma_{u_{21}}, \beta_{l_{23}}]$. Here $s_1(X) > 0$ and $s_2(X) > 0$ and there does not exist any effect from the neighboring *feature subdomains*. In this region, if $r_{51} > r_{52} + 0.1$ then $[s_1(X) - s_2(X)] > 0.05$ i.e., the region will be a *first-second choice region* with first choice as C_1. If $r_{52} > r_{51} + 0.1$ then $[s_2(X) - s_1(X)] > 0.05$ i.e., the region

will be a *first-second choice region* with first choice as C_2. Otherwise (i.e., if $| r_{51} - r_{52} | \leq 0.1$ then $| s_1(X) - s_2(X) | \leq 0.05$), the region will become a *combined choice region.*

The conclusions, after combining all the previous cases, are given below :

The *single choice regions* for C_1 are $[\Gamma_{l_{11}}, \Gamma_{l_{12}}] \times [\Gamma_{l_{21}}, \Gamma_{u_{22}}]$ and $[\Gamma_{l_{12}}, \Gamma_{u_{12}}] \times [\Gamma_{l_{21}}, \Gamma_{l_{22}}]$. The *single choice regions* for C_2 are $[\Gamma_{l_{12}}, \Gamma_{l_{13}}] \times [\Gamma_{u_{22}}, \Gamma_{u_{23}}]$ and $[\Gamma_{u_{12}}, \Gamma_{u_{13}}] \times [\Gamma_{l_{22}}, \Gamma_{u_{22}}]$. The overlapping (corresponding to *combined* and *first-second choices*) regions are $[\Gamma_{l_{12}}, \beta_{l_{12}}] \times [\Gamma_{l_{22}}, \beta_{u_{22}}]$ and $[\Gamma_{l_{12}}, \beta_{u_{12}}] \times [\Gamma_{l_{22}}, \beta_{l_{22}}]$. The remaining portion in the feature space is termed as the no-class (*null choice*) region.

In the overlapping region, $[\Gamma_{l_{12}}, \beta_{l_{12}}] \times [\Gamma_{l_{22}}, \beta_{u_{22}}]$ and $[\Gamma_{l_{12}}, \beta_{u_{12}}] \times [\Gamma_{l_{22}}, \beta_{l_{22}}]$ correspond to the *first-second choice regions* with first choice as C_1. Again, $[\Gamma_{l_{12}}, \beta_{l_{12}}] \times [\Gamma_{l_{22}}, \beta_{u_{22}}]$ and $[\Gamma_{l_{12}}, \beta_{u_{12}}] \times [\Gamma_{l_{22}}, \beta_{l_{22}}]$ correspond to the *first-second choice regions* with first choice as C_2. The region surrounded by the lines Z_1, Z_2, ... and Z_{20} [Fig. 5.3(d)] is the *combined choice region.* The portions in the overlapping region, not falling in the *combined choice region*, are the *first-second choice regions.*

The pattern classes C_1 and C_2 are shown in Fig. 5.3(a). A typical training set is also shown in this figure. The overlapping portions in the pattern classes and also in the training set are marked here. Fig. 5.3(b) shows the typical *feature subdomains* and *space subdomains* corresponding to the training set [Fig. 5.3(a)]. The regions corresponding to *single, first-second, combined* and *null choices* are also shown in Fig. 5.3(b). Note that the regions are drawn based on the previous analytical findings.

To show the overlapping regions more prominently, the rectangular portion (shown by dotted lines in Fig. 5.3(a)) which includes the overlapping regions, are enlarged in Fig. 5.3(c). The corresponding rectangular portion in Fig. 5.3(b) is also enlarged in Fig. 5.3(d). Figures 5.3(b) and 5.3(d) show the various choice regions corresponding to the pattern classes in fig-

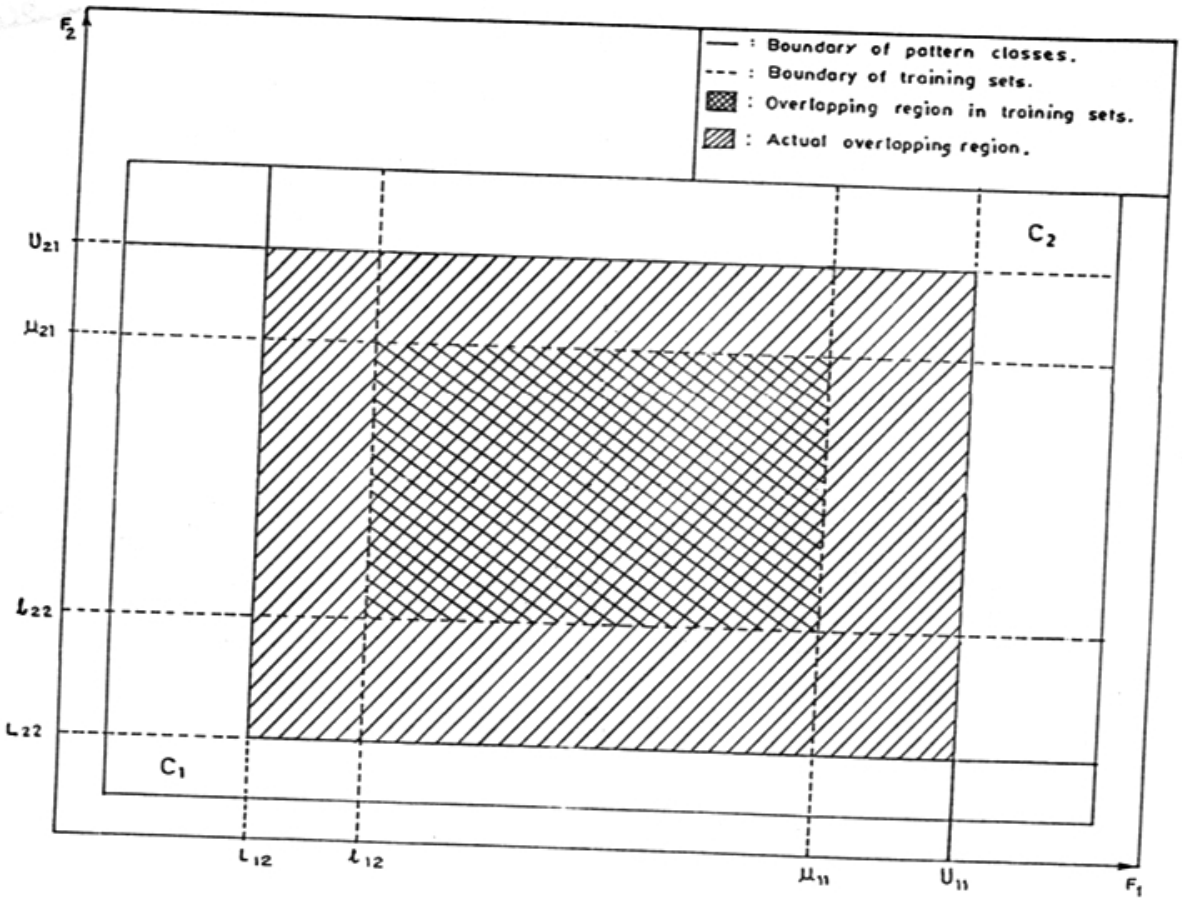


Figure 5.3(c) : Enlarged version of the rectangular portion (enclosed by dotted lines) in Fig. 5.3(a).

ures 5.3(a) and 5.3(c) respectively. The lines Z_1, Z_2, ..., Z_{20} corresponding to equations (5.8)-(5.18) are shown in Fig. 5.3(d). •

Proof of Proposition 5.1 :

In the present case, the actual overlapping region between the classes C_1 and C_2 is $[L_{12}, U_{11}] \times [L_{22}, U_{21}]$. The non-overlapping regions for C_1 are $[L_{11}, U_{11}] \times [L_{21}, L_{22}]$ and $[L_{11}, L_{12}] \times [L_{22}, U_{21}]$. The non-overlapping regions for C_2 are $[L_{12}, U_{12}] \times [U_{21}, U_{22}]$ and $[U_{11}, U_{12}] \times [L_{22}, U_{21}]$. In the training samples, the feature region $[l_{12}, u_{11}] \times [l_{22}, u_{21}]$ is overlapping. Based on the training set, $[\Gamma_{l_{12}}, \Gamma_{u_{11}}] \times [\Gamma_{l_{22}}, \Gamma_{u_{21}}]$ is the estimated overlap-

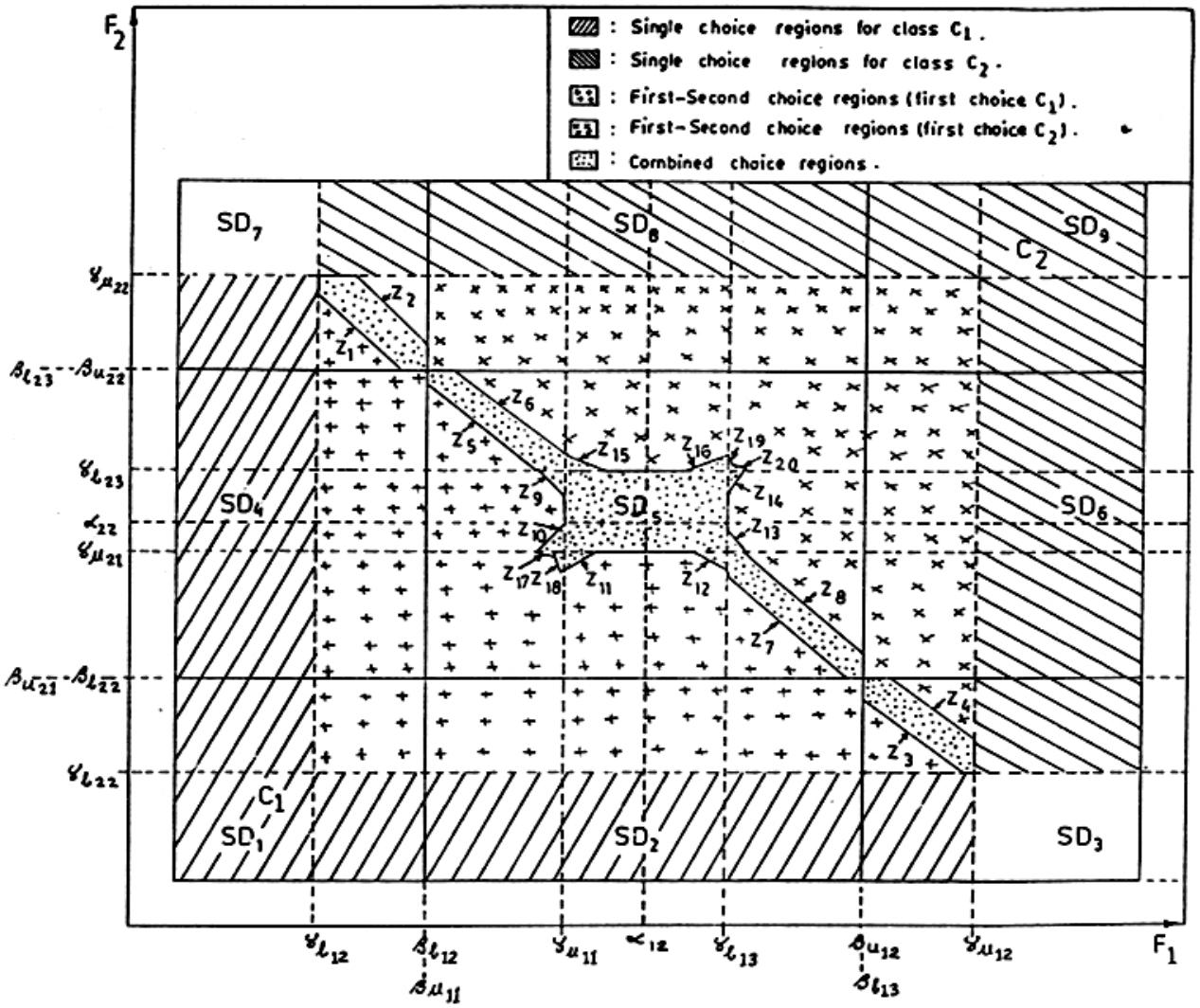


Figure 5.3(d) : Enlarged version of the rectangular portion (enclosed by dotted lines) in Fig. 5.3(*b*).

ping region. The estimated non-overlapping regions for C_1 are $[\Gamma_{l_{11}}, \Gamma_{u_{12}}] \times [\Gamma_{l_{21}}, \Gamma_{u_{22}}]$ and $[\Gamma_{l_{11}}, \Gamma_{u_{12}}] \times [\Gamma_{l_{22}}, \Gamma_{u_{22}}]$. On the other hand, the estimated non-overlapping regions for C_2 are $[\Gamma_{l_{12}}, \Gamma_{u_{13}}] \times [\Gamma_{u_{22}}, \Gamma_{u_{23}}]$ and $[\Gamma_{u_{12}}, \Gamma_{u_{13}}] \times [\Gamma_{l_{22}}, \Gamma_{u_{22}}]$.

By Theorem 5.1, it can be stated that as the size of the training samples increases

$$\begin{aligned} l_{ij} &\to L_{ij} \quad \text{in probability} \\ \text{and } u_{ij} &\to U_{ij} \quad \text{in probability} \quad (i = 1,2;\ j = 1,2). \end{aligned} \tag{5.19}$$

With the increase of the size of the training samples, the samples in the *feature subdomain* also increase. Thus, by Theorem 5.2,

$$\begin{aligned} \Gamma_{l_{ig}} &\to \beta_{l_{ig}} \quad \text{in probability} \\ \text{and } \Gamma_{u_{ig}} &\to \beta_{u_{ig}} \quad \text{in probability} \quad (i = 1,2;\ g = 1,2,3). \end{aligned} \tag{5.20}$$

From Fig. 5.3(c), we have

$$\begin{aligned} \beta_{l_{i1}} &= l_{i1} \quad ; \quad \beta_{u_{i1}} = \beta_{l_{i2}} = l_{i2} \quad ; \\ \beta_{l_{i3}} &= \beta_{u_{i2}} = u_{i1} \quad \text{and} \quad \beta_{u_{i3}} = u_{i2} \quad (i = 1,2). \end{aligned} \tag{5.21}$$

Combing equations (5.19), (5.20) and (5.21), one gets

$$\begin{aligned} \Gamma_{l_{i1}} \to \beta_{l_{i1}} = l_{i1} \to L_{i1} \quad &\text{in probability} \\ \Gamma_{u_{i1}} \to \beta_{u_{i1}} = l_{i2} \to L_{i2} \quad &\text{in probability} \\ \Gamma_{l_{i2}} \to \beta_{l_{i2}} = l_{i2} \to L_{i2} \quad &\text{in probability} \\ \Gamma_{u_{i2}} \to \beta_{u_{i2}} = u_{i1} \to U_{i1} \quad &\text{in probability} \\ \Gamma_{l_{i3}} \to \beta_{l_{i3}} = u_{i1} \to U_{i1} \quad &\text{in probability} \\ \Gamma_{u_{i3}} \to \beta_{u_{i3}} = u_{i2} \to U_{i2} \quad &\text{in probability} \quad (i = 1,2). \end{aligned}$$

Therefore as the size of the training sample increases, the estimated overlapping region $[\Gamma_{l_{12}}, \Gamma_{u_{11}}] \times [\Gamma_{l_{22}}, \Gamma_{u_{21}}]$ tends to the actual overlapping region $[L_{12}, U_{11}] \times [L_{22}, U_{21}]$ in probability. At the same time, the estimated non-overlapping regions for C_1 and C_2 go to their actual sizes respectively in probability. Using the previous results, it can also be concluded that the no-class (*null choice*) region also tends to their actual size in probability.

Hence the proposition. •

Experimental results :

To substantiate the analytical results, a two-class problem with classes $[2,7] \times [2,7]$ and $[5,10] \times [5,10]$ is considered [Fig. 5.4(a)]. Here, the non-overlapping regions for C_1 are $[2,7] \times [2,5]$ and $[2,5] \times [5,7]$, and the non-overlapping regions for C_2 are $[5,10] \times [7,10]$ and $[7,10] \times [5,7]$. The overlapping region is $[5,7] \times [5,7]$.

To implement the recognition system, five training sample sets with 50, 100, 150, 200 and 250 samples from each of the classes are chosen randomly. Figures 5.4(b)-(f) show the regions corresponding to various output choices for the five sample sets. Here the character 'A' represents the *single choice* for C_1; the character 'B' represents the *single choice* for C_2; the character 'a' represents the *first-second choice* with first choice as C_1; the character 'b' represents the *first-second choice* with first choice as C_2; the character 'C' represents the *combined choice* reflecting both C_1 and C_2, and the blank character ' ' represents the *null choice.*

The results in figures 5.4(b)-(f) verify that with the increase of sample sizes, the training classes tend to the actual classes (Theorem 5.1) and the *feature subdomains* with the extended portions tend to their actual sizes (Theorem 5.2). The overlapping (*combined* and *first-second choices*), non-overlapping (*single choice*) and no-class (*null choice*) regions are seen to tend to their actual sizes with the increase in the size of training samples. Hence *Proposition* 5.1 is verified experimentally for the case of rectangular pattern classes. The experimental results in figures 5.4(b)-(f) also support the analytical findings and diagrams in figures 5.3(b) and 5.3(d).

For analyzing the performance of the proposed recognition system, a test set with 1000 samples from each of the classes is generated. Table 5.3 provides the recognition scores under various choice groups corresponding to the aforesaid five training sample sets.

The Bayes classifier is also applied on the same pattern classes assuming

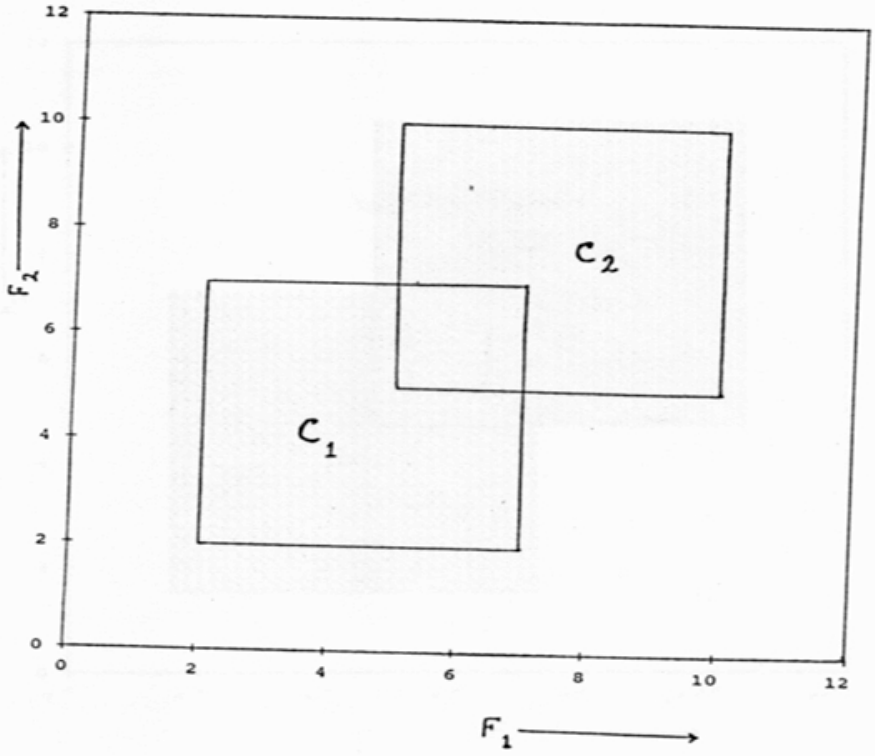
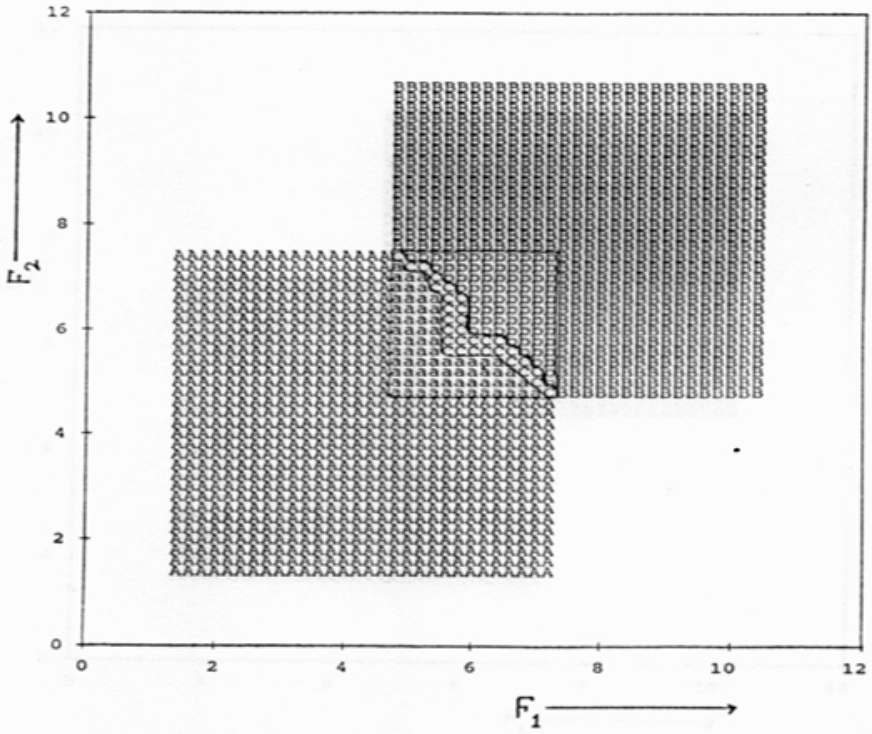
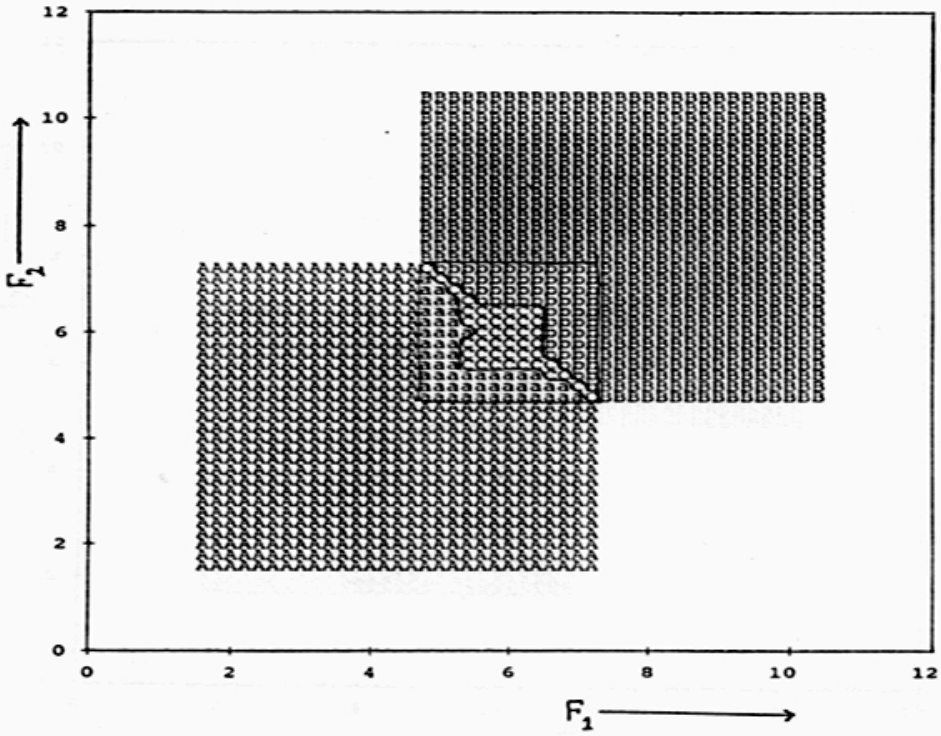


Figure 5.4(a) : Two rectangular classes with a training set.

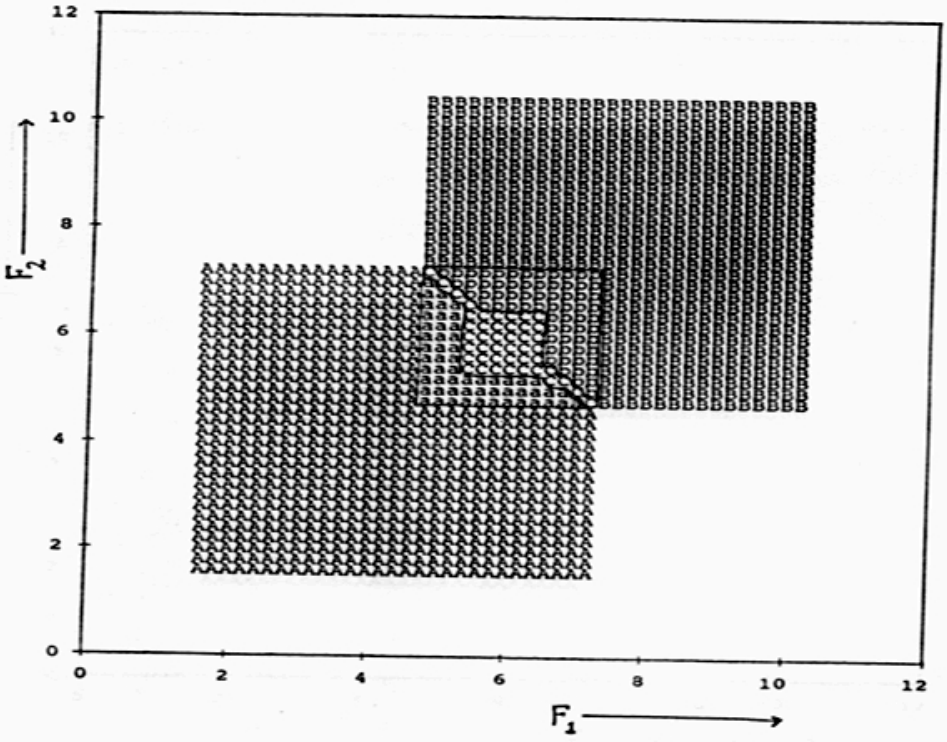


(b)

Figures 5.4 (b)-(f) : Various regions corresponding to five training sets (with 50, 100, 150, 200 and 250 samples from each class) for the pattern classes in Fig. 5.4(*a*).

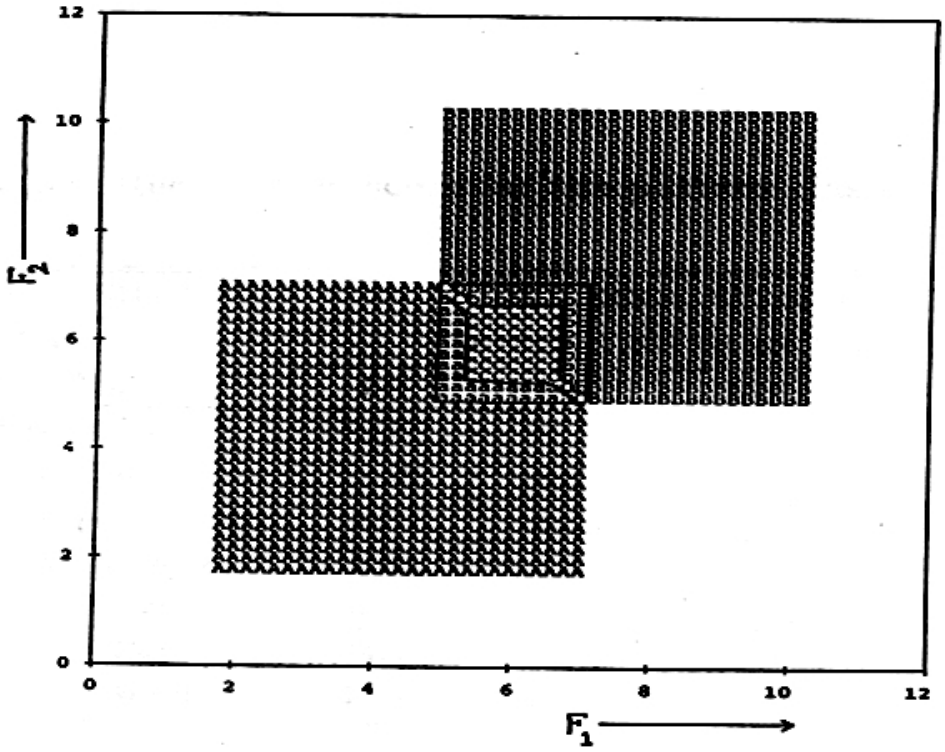


(c)

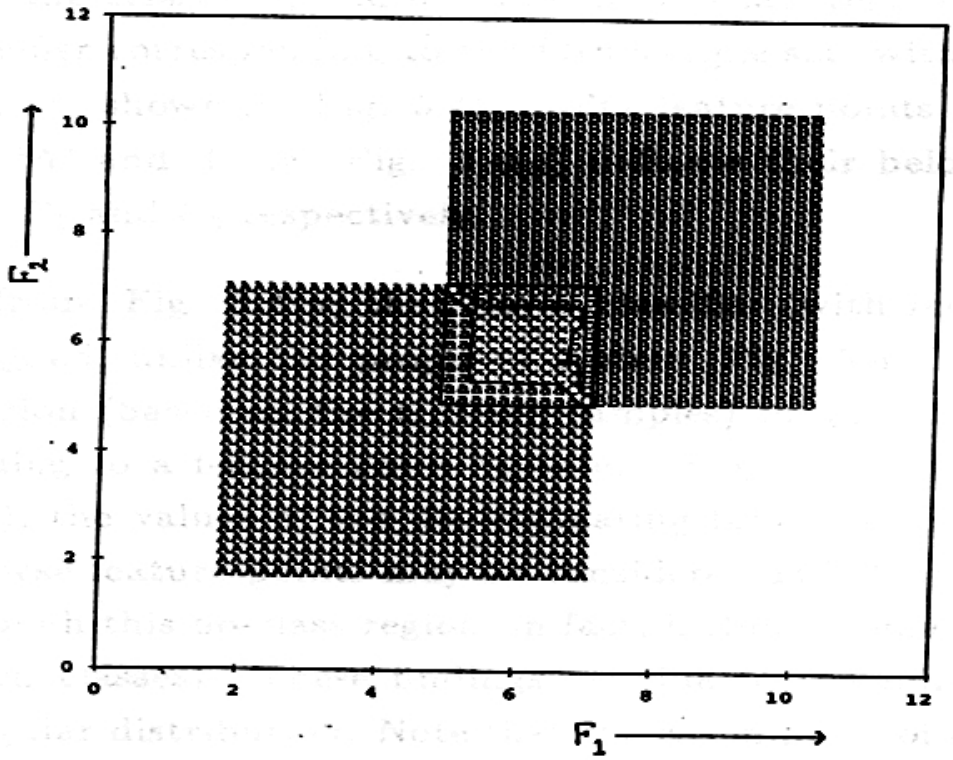


(d)

Figures 5.4 (b)-(f) : (continued)



(e)



(f)

Figures 5.4 (b)-(f) : (continued)

Table 5.3 : Recognition score for the pattern classes in Fig. 5.4(a)

VARIOUS GROUP OF CHOICES		% RECOGNITION SCORE				
		SAMPLE SIZES IN EACH CLASS				
		50	100	150	200	250
Single Correct Choice		79.60	80.35	80.60	81.60	82.45
First Correct Choice		11.05	8.75	8.35	6.45	4.80
Combined Correct Choice		3.70	6.05	6.60	8.15	9.95
Second Correct Choice		5.65	4.85	4.45	3.80	2.80
Fully Wrong Choice		0.00	0.00	0.00	0.00	0.00
BAYES CLASSIFIER	*Correct*	84.25	88.00	88.35	88.55	88.90
	Wrong	15.75	12.00	11.65	11.45	11.10

rectangular distributions of the classes. For a comparative study, the hard regions for the classes C_1 and C_2, and the no-class region obtained with the Bayes classifier corresponding to the fifth sample set (with 250 samples from each class) are shown in Fig. 5.4(g). The feature points represented by the characters 'A' and 'B' in Fig. 5.4(g) indicate their belonging to the hard regions for C_1 and C_2 respectively.

Note from Fig. 5.4(g) that Bayes classifier (with rectangular distributions) assigned, unlike the proposed system [Fig. 5.4(f)], the whole overlapping region (based on the training samples) to one class i.e., C_2. Again, corresponding to a feature point represented by the blank characters ' ' in Fig. 5.4(g), the values of the discriminating function are zero for both the classes. These feature points may be considered as belonging to the no-class region (though this no-class region, in fact, includes some portions of the actual pattern classes). These findings are due to the inherent properties of the rectangular distribution. Note that the assumption of any other distribution is not valid here. On the other hand, the output decisions, as shown in figures 5.4(b)-(f), of our multivalued recognition system are seen to be very appropriate.

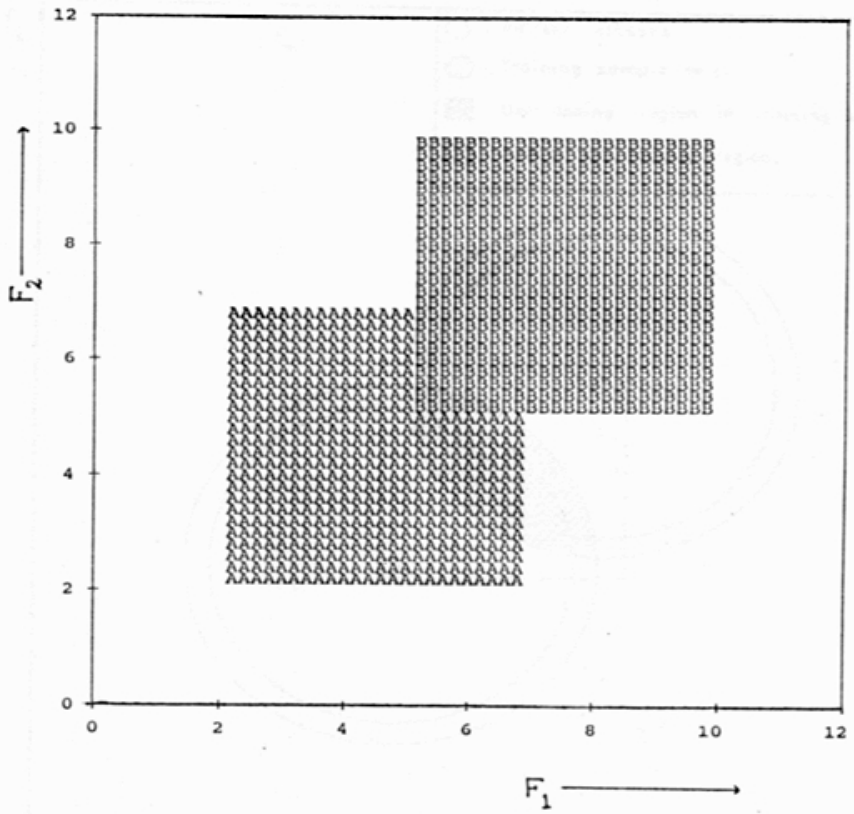


Figure 5.4(g) : Various regions of the Bayes classifier corresponding to the fifth training set (with 250 samples from each class) for the pattern classes in Fig. 5.4(a).

The recognition scores of the Bayes classifier (with rectangular distributions) corresponding to the five training sets are included in Table 5.3. Note that the score obtained by adding *single correct*, *first correct* and half of the *combined correct choices* becomes much higher than the correct recognition score of the Bayes classifier. Again, a significant portion of the wrong choices of Bayes classifier is seen to be corrected by the proposed system and the remaining portion is distributed among the *combined* and *second correct choices*. Because of this flexibility, the proposed system has, therefore, a provision of improving its efficiency significantly by incorporating *combined* and *second choices* under the control of a supervisory scheme. ♣

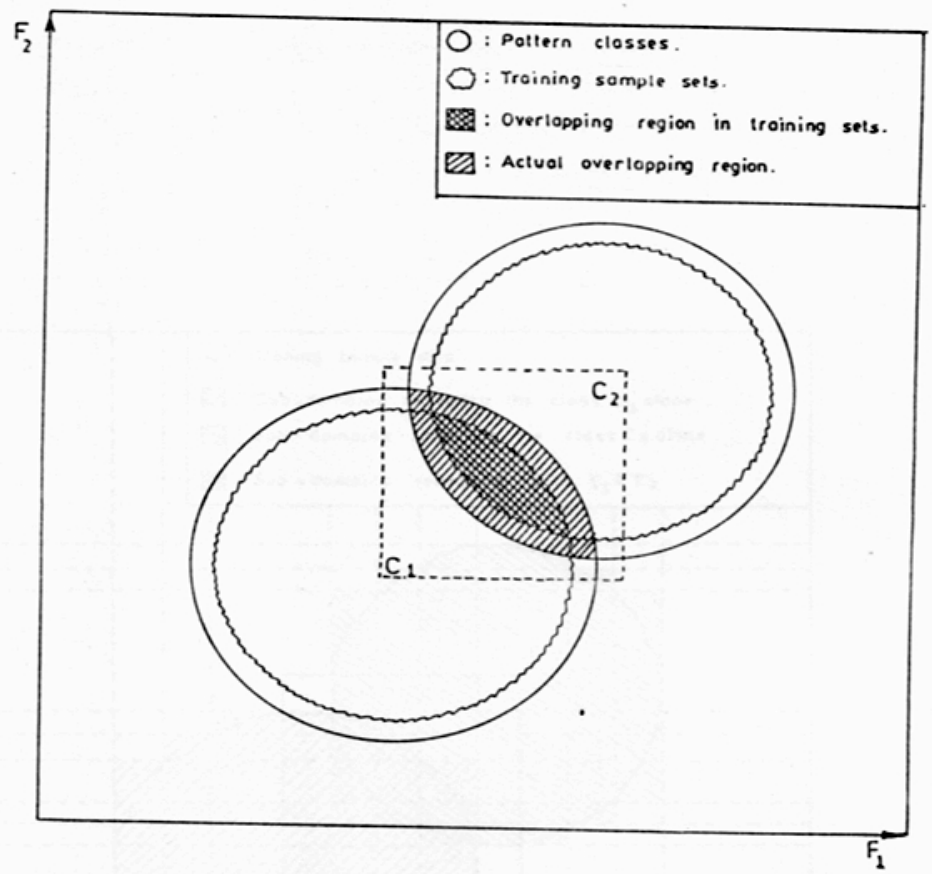


Figure 5.5(a) : Two circular classes with a training set.

5.4.2 Circular classes

The analytical results of rectangular classes obtained in the previous section are extended here to circular pattern classes [Fig. 5.5(a)]. A typical training sample set is assumed for carrying out the theoretical analysis of the proposed multivalued recognition system on the circular classes in Fig. 5.5(a). The actual overlapping portions and the overlapping portions from the training set are distinctly marked in Fig. 5.5(a).

Some typical *feature subdomains* and *space subdomains* corresponding to the training samples [Fig. 5.5(a)] are drawn in Fig. 5.5(b). Based on these *feature subdomains* and *space subdomains*, and using the results obtained for the rectangular pattern classes [figures 5.3(a)-(d)], the regions corresponding to the *single*, *first-second*, *combined* and *null choices* are shown in Fig. 5.5(b).

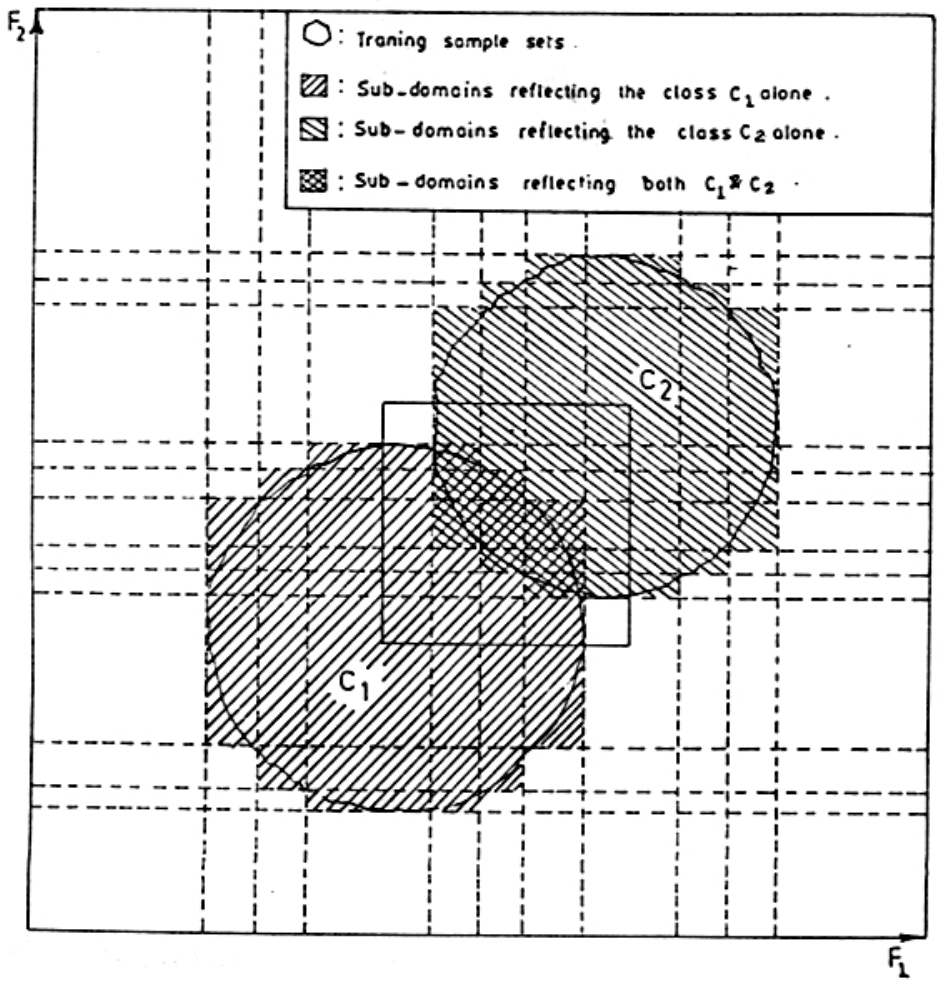


Figure 5.5(b) : *Subdomains* and choice regions for the pattern classes in Fig. 5.5(*a*).

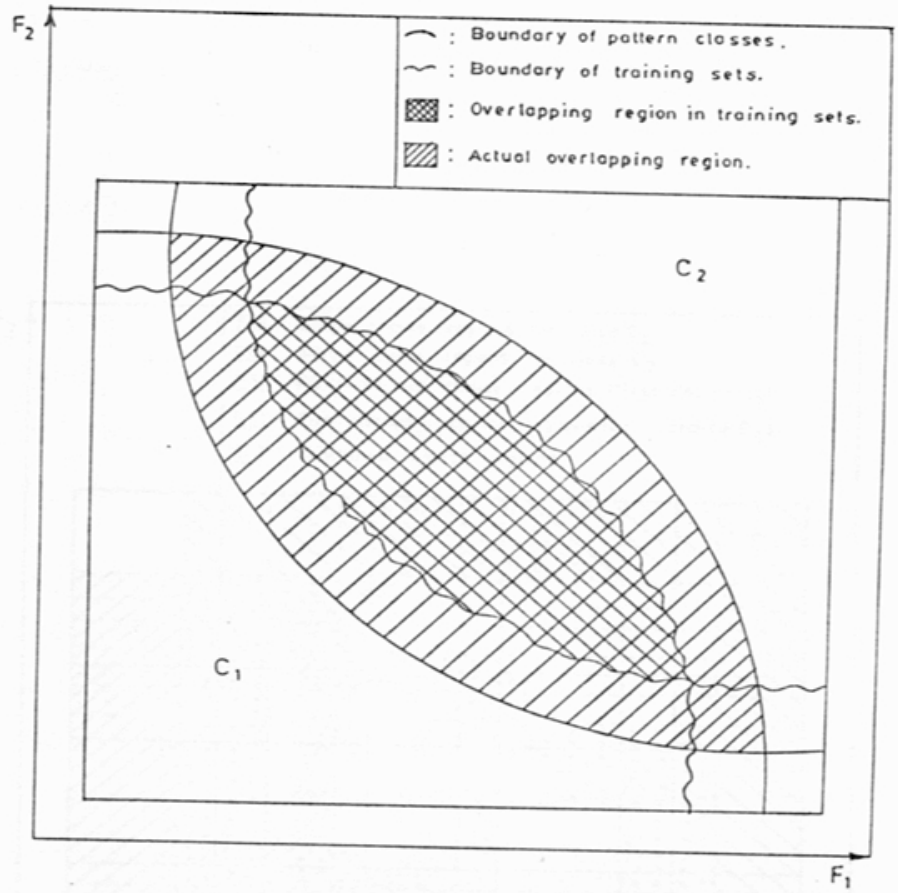


Figure 5.5(c) : Enlarged version of the rectangular portion (enclosed by dotted lines) in Fig. 5.5(a).

To show the overlapping regions more prominently, the rectangular portion (enclosed by dotted lines) in Fig. 5.5(a) which includes the overlapping region, is enlarged in Fig. 5.5(c). The corresponding rectangular portion (enclosed by thick lines) in Fig. 5.5(b) is also enlarged in Fig. 5.5(d). The figures 5.5(b) and 5.5(d) show the regions corresponding to various output choices for the pattern classes in figures 5.5(a) and 5.5(c) respectively.

When the size of the training samples increases, the values of the accuracy factor [Eq. (2.1)] decrease, and correspondingly the number of *feature subdomains* and *space subdomains* increases. Therefore, the sizes of the *feature subdomains* and *space subdomains* decrease with the increase in the size of the training samples. Hence, it can be concluded that the increase in sam-

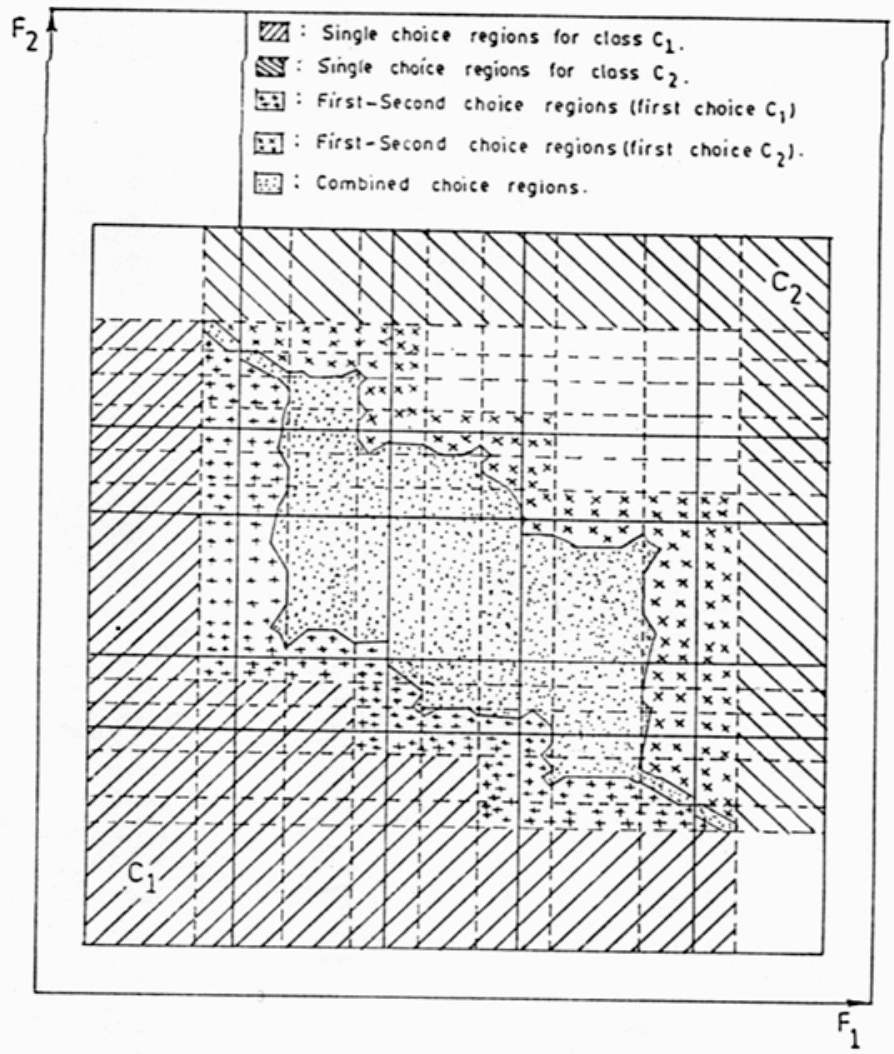


Figure 5.5(d) : Enlarged version of the rectangular portion (enclosed by solid lines) in Fig. 5.5(*b*).

ple sizes results in increase in the accuracy of the choice regions with respect to their actual sizes. Therefore, *Proposition* 5.1 is claimed for the circular shaped pattern classes. •

Experimental results :

To substantiate the analytical findings, a two-class problem with circular classes is considered. The centres of the classes C_1 and C_2 are taken to be $(5, 5)$ and $(8.5, 8.5)$ respectively and their radii are considered to be 3.5 and 3 respectively [Fig. 5.6(a)].

To implement the recognition system, five training sample sets with 50, 100, 150, 200 and 250 samples from each of the pattern classes are chosen randomly. Figures 5.6(b)-(f) show the regions corresponding to various output choices for the five sample sets. Here the character 'A' represents the *single choice* for C_1; the character 'B' represents the *single choice* for C_2; the character 'a' represents the *first-second choice* with first choice as C_1; the character 'b' represents the *first-second choice* with first choice as C_2; the character 'C' represents the *combined choice* for both C_1 and C_2, and the blank character ' ' represents the *null choice*. These results demonstrate that with the increase of sample sizes, the estimated classes tend to the actual classes (Theorem 5.1) and the *feature subdomains* (with the extended portions) tend to their actual sizes (Theorem 5.2). The estimated non-overlapping, overlapping and no-class regions are seen to tend to their actual sizes with the increase in the size of training samples. Hence the claim of *Proposition* 5.1 is justified experimentally. The experimental results in figures 5.6(b)-(f) also support the analytical findings in figures 5.5(b) and 5.5(d).

The distribution functions of the aforesaid pattern classes are assumed to be Gaussian for applying the Bayes classifier. The hard regions of the Bayes classifier for the classes C_1 and C_2 corresponding to the fifth sample set (with 250 samples from each class) are shown in Fig. 5.6(g). The feature points represented by the characters 'A' and 'B' in Fig. 5.6(g) indicate their

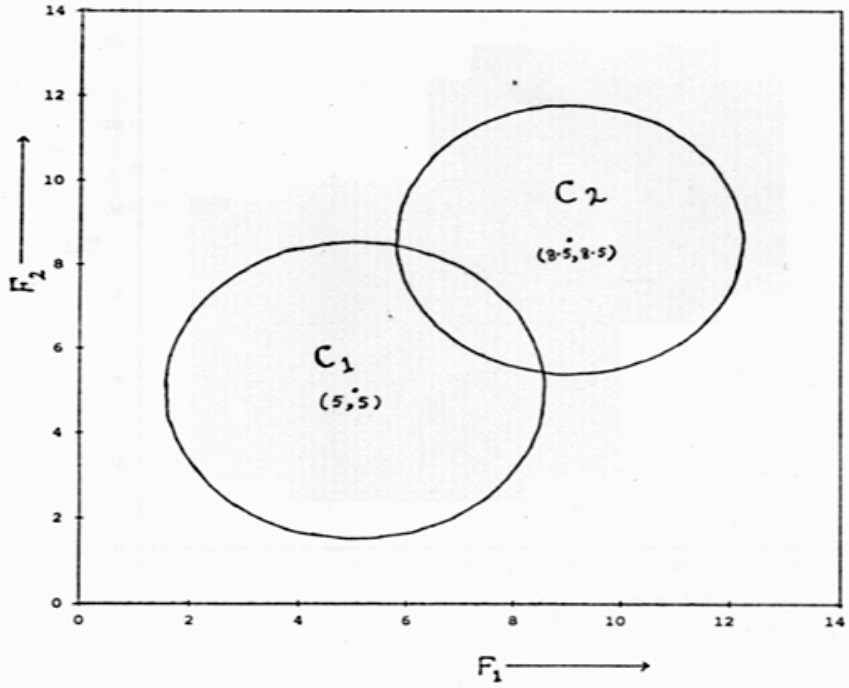
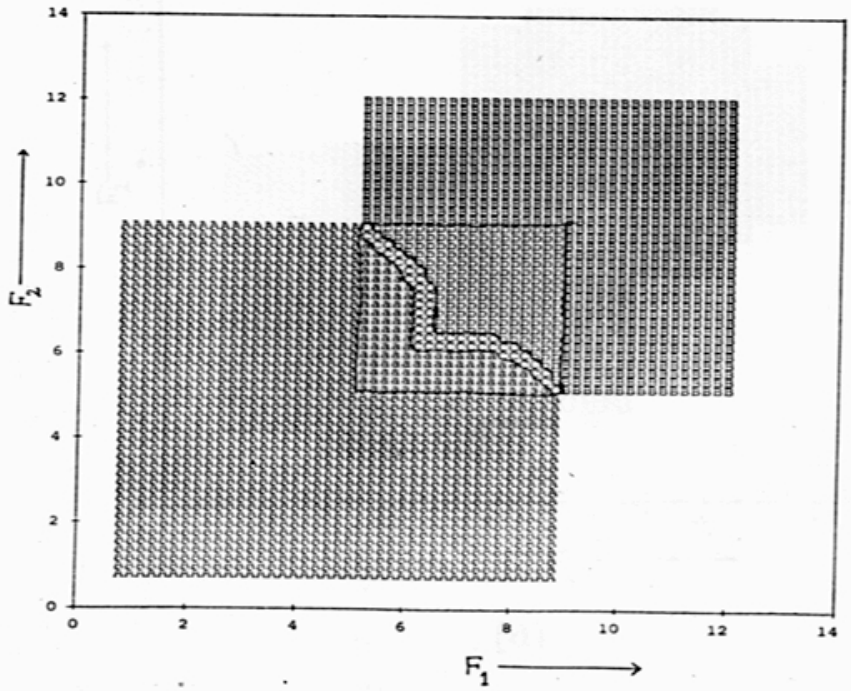
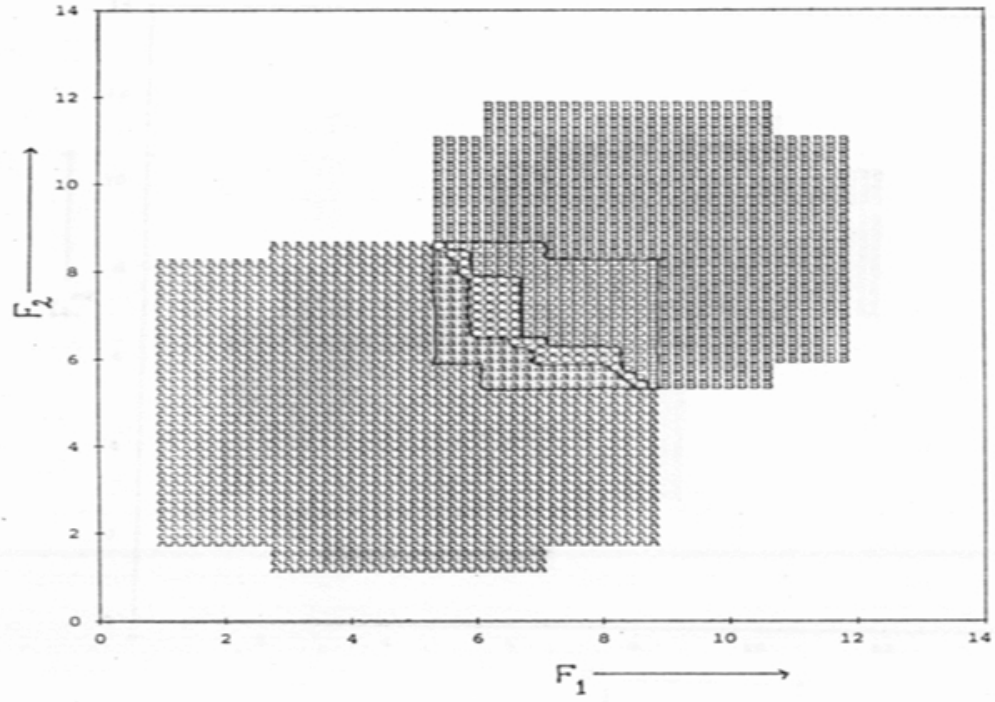


Figure 5.6(a) : Two circular classes with centres at (5.5, 5.5) and (8.5, 8.5), and with radii 3.5 and 3 respectively.

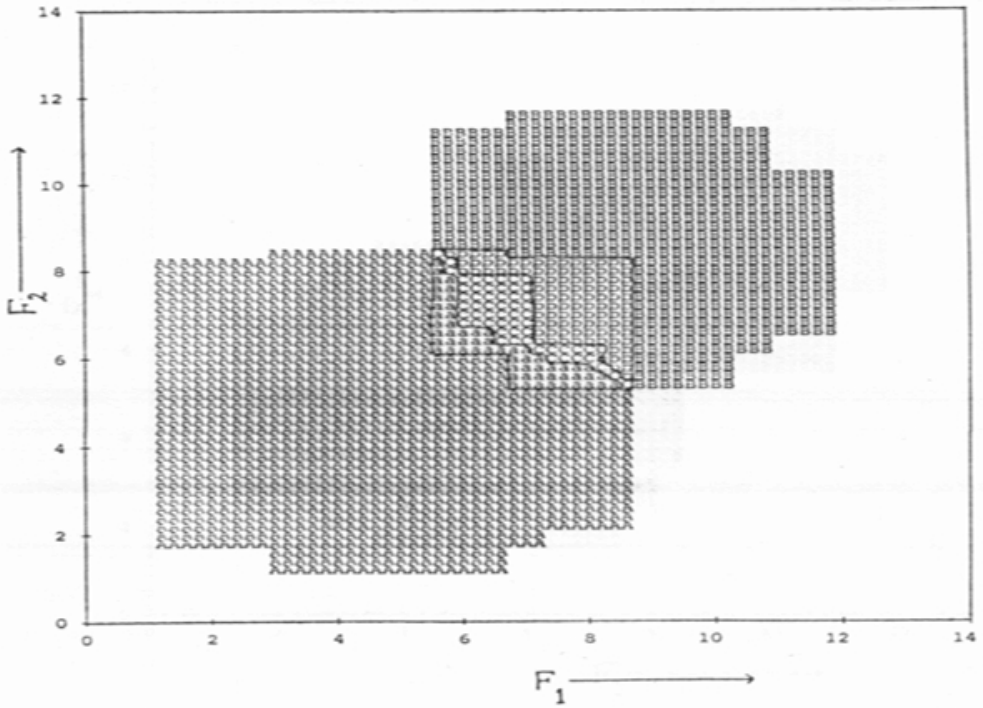


(b)

Figures 5.6 (b)-(f) : Various regions corresponding to five training sets (with 50, 100, 150, 200 and 250 samples from each class) for the pattern classes in Fig. 5.6(a).

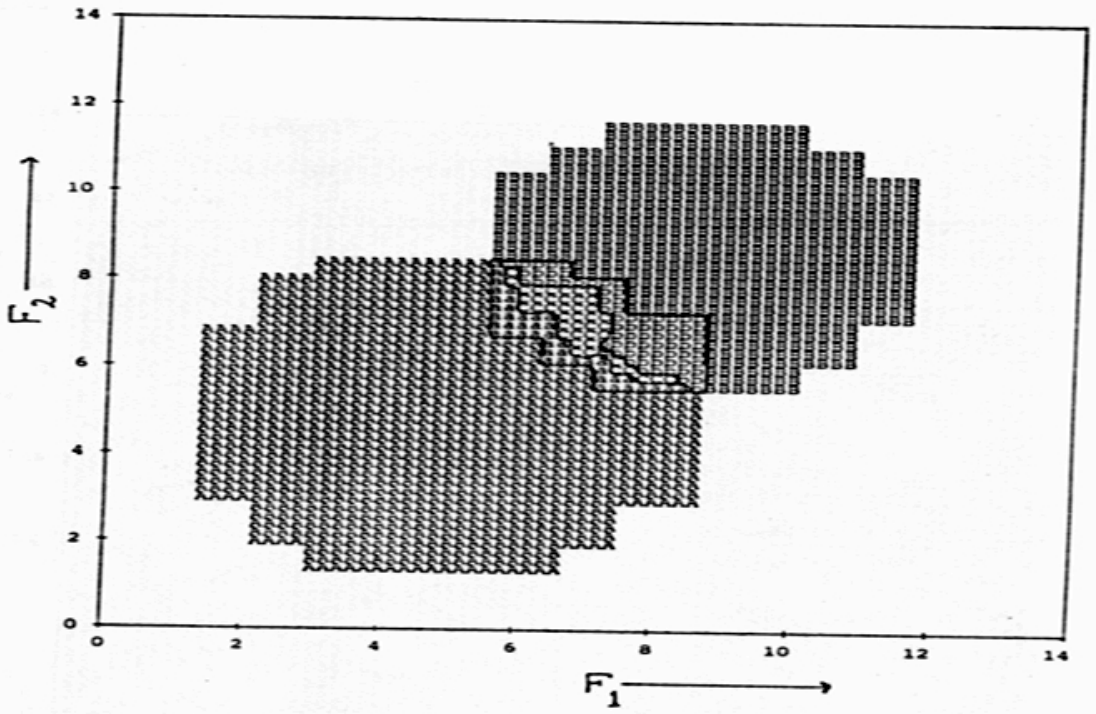


(c)

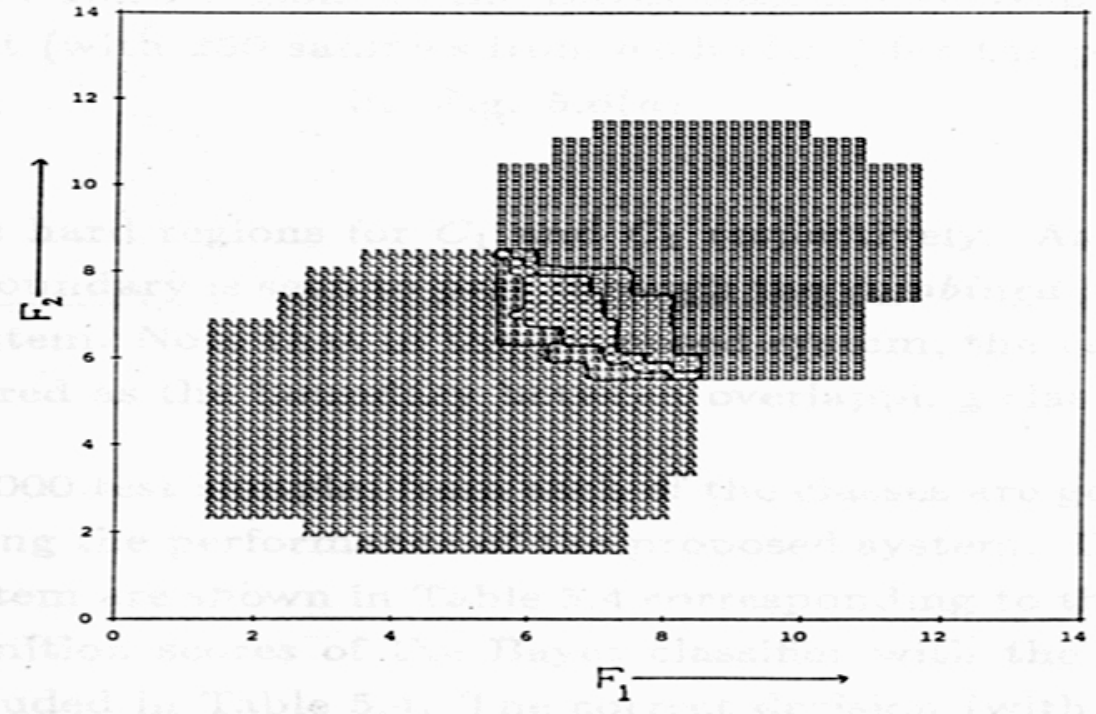


(d)

Figures 5.6 (b)-(f) : (continued)



(e)



(f)

Figures 5.6 (b)-(f) : (continued)

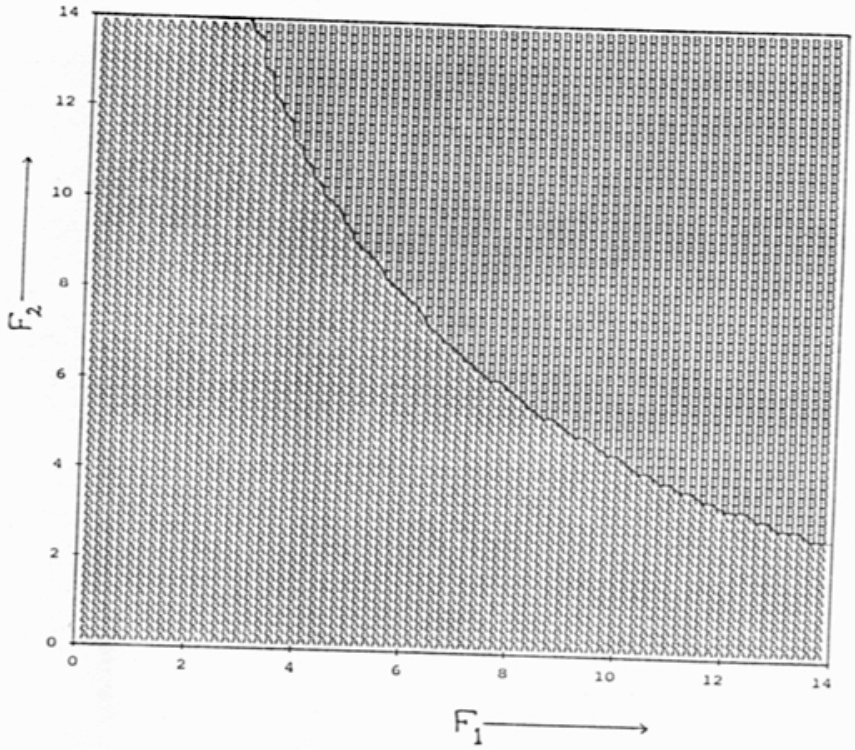


Figure 5.6(g) : Hard regions of the Bayes classifier corresponding to the fifth training set (with 250 samples from each class) for the pattern classes in Fig. 5.6(a).

belonging to the hard regions for C_1 and C_2 respectively. As expected, the Bayes decision boundary is seen to pass through the *combined choice region* of the proposed system. Note that in the proposed system, the *combined choice region* is considered as the boundary between overlapping classes.

As before, 1000 test samples from each of the classes are generated artificially for analyzing the performance of the proposed system. The recognition scores of our system are shown in Table 5.4 corresponding to the five training sets. The recognition scores of the Bayes classifier with the same training sets are also included in Table 5.4. The correct decision (with *single correct*, *first correct* and half of the *combined correct choices*) of the proposed system is seen to be much higher than that of the Bayes classifier. In other words, among the samples which were misclassified by Bayes some are found to be corrected by our system and the remaining samples are distributed among

Table 5.4 : Recognition score for the pattern classes in Fig. 5.6(a)

VARIOUS GROUP OF CHOICES		% RECOGNITION SCORE SAMPLE SIZES IN EACH CLASS				
		50	**100**	**150**	**200**	**250**
Single Correct Choice		73.95	80.70	84.40	88.70	89.90
First Correct Choice		20.80	13.85	10.20	6.45	4.80
Combined Correct Choice		3.10	3.85	4.00	2.85	3.35
Second Correct Choice		2.15	1.60	1.60	1.95	1.95
Fully Wrong Choice		0.00	0.00	0.00	0.00	0.00
BAYES CLASSIFIER	*Correct*	93.00	94.10	94.25	94.50	94.45
	Wrong	7.00	5.90	5.75	5.50	5.55

the proposed *combined* and *second correct choices.* ♣

5.4.3 Some remarks

- The four state decisions of the proposed multivalued recognition system reflect the characteristics of the overlapping, non-overlapping and no-class regions of the feature space. It has been verified that with the increase in the size of the training samples, the estimated versions of these regions tend to their actual sizes (*Proposition* 5.1).

- In all the experimental results it has been observed that Bayes decision boundaries lie within the *combined choice regions* of the multivalued recognition system. Among the wrong decisions made by the Bayes classifier, a part is seen to be corrected by the *single* and *first choices* of the proposed system. The remaining portion is found to be distributed among the *combined correct* and *second correct* choices. The proposed system has, therefore, a provision of improving its efficiency significantly by incorporating the *second* and *combined* choices under the control of a supervisory scheme.

- The present investigation, in turn, establishes analytically the justification of considering output decisions in four states for managing uncertainties arising from ambiguous regions.

- Although the analysis has been made assuming two class problems in one and two dimensional feature spaces, it can be extended to a general M class N feature problem with classes of any shape. ♠

Chapter 6

ANALYSIS OF SATELLITE IMAGERY FOR IDENTIFYING ILL-DEFINED OBJECT REGIONS

Contents

6.1 Introduction

In chapter 4, we have developed a multivalued recognition system (using geometrical structure based decomposition) and demonstrated its performance on some artificially generated pattern sets and also on the real life speech data. In this chapter, we consider another real life application of the system to a different domain, namely analysis of remotely sensed satellite imagery data for detecting ill-defined object regions.

Analysis of remotely sensed images for detecting ill-defined object regions has been a topic of research for the past two decades. Initially, various image processing techniques [52-55,240,241] have been used in analyzing remotely sensed images. In order to identify certain targets, some knowledge about the targets and the scene needs to be incorporated in the algorithm [242].

A good proportion of the existing research work in the field of remote sensing has addressed the problem of road detection. Fischer *et al.* [243] have presented a method where locally evaluated evidence of road presence from multiple sources is combined. Bajcy and Tavakoli [244] used a world model of roads for detecting road and roadlike features from ERST-1 satellite data. They used a single band, namely *green* band image for this purpose. Groch [245] has described a semi-automatic method of following line shaped objects in aerial images. Mckeown and Pane [246] have addressed the problem of alignment and connection of linear feature fragments in aerial imagery. Zhu and Yeh [247] have proposed a method for road network detection where image processing techniques are used to detect linear segments. In order to predict missing links between broken segments, they used the facts about the road network and perceptual continuity. Bezdek *et al.* [248] described a parallel line detector algorithm which compares the contrast of neighboring pixels and checks for uniformity along the tracking direction, to detect road like structures from radar images. Vasudevan *et al.* [249] have developed an intermediate processing stage that addresses tasks of partitioning and connecting road like fragments.

Mckeown *et al.* [250] used aerial imagery and a data base of man-made objects for the interpretation of airport scenes. Wharton [251] proposed a prototype expert system for multi-spectral remotely sensed data on the basis of spectral knowledge. Mckeown [252] suggested an extension of the concept of a geographical information system to include remotely sensed data along image-to-map correspondence capacity. He also described a research system MAPS to justify his claim. Goodenough *et al.* [253], and Nazif and Levine [254] proposed some expert system concepts to analyze remotely sensed images. Silberberg [255] described an image interpretation system to detect objects like submarine, airplanes etc. from aerial images using multiple image resolution. Mohan and Nevatia [256] proposed collated features as the representation of the primitive image elements which are useful for stereo and the generation of shape description and object segmentation. Most of these approaches are restricted to a particular remotely sensed imagery with limited targets.

There has been an attempt [257,258] to detect airports from IRS (Indian Remote Sensing) images based on *green* and *infrared* images. The approach initially partitioned an image into six land cover types using a minimum distance classifier. Then spatial knowledge was incorporated on the partitioned image for finding the possible locations of airports. Again, there was another attempt [259] to extract the linear features from IRS images based on only infrared band image and considering spatial information.

The relevance of fuzzy set theory in image processing/analysis problem has been addressed in section 1.4.7. In a remotely sensed image, the regions (objects) are usually ill-defined because of both grayness and spatial ambiguities. Moreover, the gray value assigned to a particular pixel of a remotely sensed image is the average reflectance of different types of ground covers present in the corresponding pixel area ($36.25m \times 36.25m$ for the IRS imagery). Therefore, a pixel may represent more than one class with a varying degree of belongingness. Thus, the approaches based on fuzzy set theory can be very effective in analyzing remotely sensed images.

Few attempts have been made in the remote sensing image analysis using fuzzy set theory. Jeansoulin *et al.* [260] used the concept of fuzzy set theory to combine criteria for automatic multitemporal segmentation. Cannon *et al.* [194] proposed a fuzzy c-means clustering algorithm to develop unsupervised classification on a Thematic Mapper (TM) image. Zenzo *et al.* [261] formulated a fuzzy relaxation algorithm for contexual classification. Kent and Mardia [262] used fuzzy membership concepts to perform classification on Landsat data. Wang [263] developed a fuzzy supervised classification method in which geographical information is presented as fuzzy sets. The existing approaches are unable to extract the majority of information dormants within digital remotely sensed data and accuracy levels of image classification are quite often unsatisfactory [264]. To improve the analysis, it was suggested that studies are required on resolutions of sensor systems, physical principles of remote sensing and image processing algorithms [240].

The present chapter describes an application of the multivalued recognition system (described in chapters 4 and 5) for analyzing Indian Remote Sensing (IRS) satellite imagery for detecting various ill-defined (fuzzy) object regions, namely *roads*, *bridges*, *islands*, *sandbeds*, *beaches*, *city area* and *township/industrial areas*. The system is initially applied on an IRS image to classify (based on the spectral knowledge of the image) its pixels into six classes, namely *pond water*, *turbid water*, *concrete structure*, *habitation*, *vegetation* and *open space*. The *green* and *infrared* band information are used for the classification. The clustered images are then processed for detecting various object regions.

In order to identify certain targets, some spatial knowledge about them and their inter-relationships are incorporated on the clustered images. For example, an *island* is a compact land region surrounded completely by water bodies. Similar considerations are incorporated for discriminating other targets. The proposed algorithms have been implemented on some IRS images where the pixel resolution is $36.25m \times 36.25m$. Some heuristic constraints are considered solely because of the resolution value. For example, the maximum

width of the *road* is restricted here to 3 pixels for their detection.

The concept of multiple choices of the recognition system is found to be very effective in identifying the *road* patterns from concrete structure pixels. Because of the low pixel resolution of the IRS imagery, many portions of the roads may not be classified as concrete structures. In order to identify them, a traversal algorithm through reflected *road* pixels has been proposed here. Some of the movements in this algorithm are governed by only the *second* and *combined choices* of the multivalued recognition system. A thinning algorithm [228] is applied in order to facilitate the traversal algorithm. Some morphological operations [53,55] are used on the clustered image for locating the *city* and *township/industrial* areas. To demonstrate the effectiveness of the algorithms, two images corresponding to the scenes of two cities in India, namely Bombay and Calcutta are considered. The results are found to be very encouraging.

The classes corresponding to major land cover types in the IRS imagery and the overall detection strategy are furnished in section 6.2. Section 6.3 deals with the method of detecting the *islands*, *sandbeds* and *beaches*. The algorithms for finding the locations of *roads*, *bridges*, *city area* and *township/industrial areas* are discussed in section 6.4. The results are demonstrated in section 6.5. ♠

6.2 Land Cover Types and Overall Detection Strategy

6.2.1 Land cover types

The IRS (Indian Remote Sensing) satellite images corresponding to scenes of Bombay and Calcutta are considered in the present work. These scenes primarily consist of six different land cover types. These six classes are *pond*

water, *turbid water*, *concrete structure*, *habitation*, *vegetation* and *open space*. In fact in India, these happen to be the major land cover types. The constituents of the six classes are furnished below.

1. *Pond water* : This class contains pond water, fisheries etc.
2. *Turbid water*: This class contains sea water, river water etc. where the soil content is more.
3. *Concrete structure* : This class contains buildings, railways, roads, air strips etc. The signature of sandbeds in remote sensing images also belongs to this class.
4. *Habitation* : This class basically consists of suburban and rural habitation i.e., concrete structures but comparatively less in density than the previous class (concrete structure).
5. *Vegetation* : This class essentially represents crop and forest areas.
6. *Open space* : This class contains basically the barren land. More specifically, a pixel with less greenery and less concrete structures falls into this class. The beaches come under this class. •

In remotely sensed data, the gray value that is assigned to a particular pixel is the average reflectance of different types of ground covers present in the corresponding pixel area ($36.25m \times 36.25m$ for IRS imagery). If a pixel has a big building and some open space, it is more likely to fall into the class "habitation" than "concrete structure". Considering the concept of *second* and *combined choices* on output decisions of the recognition system, the information about the building and the open space may be obtained.

It is to be observed that the same region may fall in different classes in different seasons. As for example,

- A cultivated land, which is a vegetation area, after harvestation becomes open space.

- A river bed during summer when it is dried up falls under the class open space.
- Some land portions during flood fall under water bodies. ♠

6.2.2 Overall detection strategy

The multivalued recognition system, as described in chapter 4, is initially applied on an IRS image to classify its pixels into the aforementioned six classes. The system basically uses the spectral knowledge of the classes and the scene found in the training samples. The clustered image is processed further to detect some ill-defined target regions which include *roads*, *bridges*, *islands*, *sandbeds*, *beaches*, *city area* and *township/industrial areas*. The overall strategy to discriminate hierarchically these targets is demonstrated in a block diagram in Fig. 6.1.

Before elaborating the diagram, the meaning of the targets considered here is described below.

- Water : The pond and turbid water pixels constitute the water bodies in a scene.
- Land : The remaining pixels (i.e., not belonging to water bodies) are primarily categorized as the land.
- *Island* : An *island* is a land portion surrounded completely by water bodies.
- *Sandbed* : The sand portions (which belong to concrete structure) adjacent to the water bodies (sea, river etc.) are referred to as *sandbeds*. Note that if all the pixels in an *island* represent concrete structure, then the *island* is referred to as a *sandbed*.
- *Beach* : A *beach* is an open space region (with a significant area) adjacent to the water bodies and/or *sandbeds*.

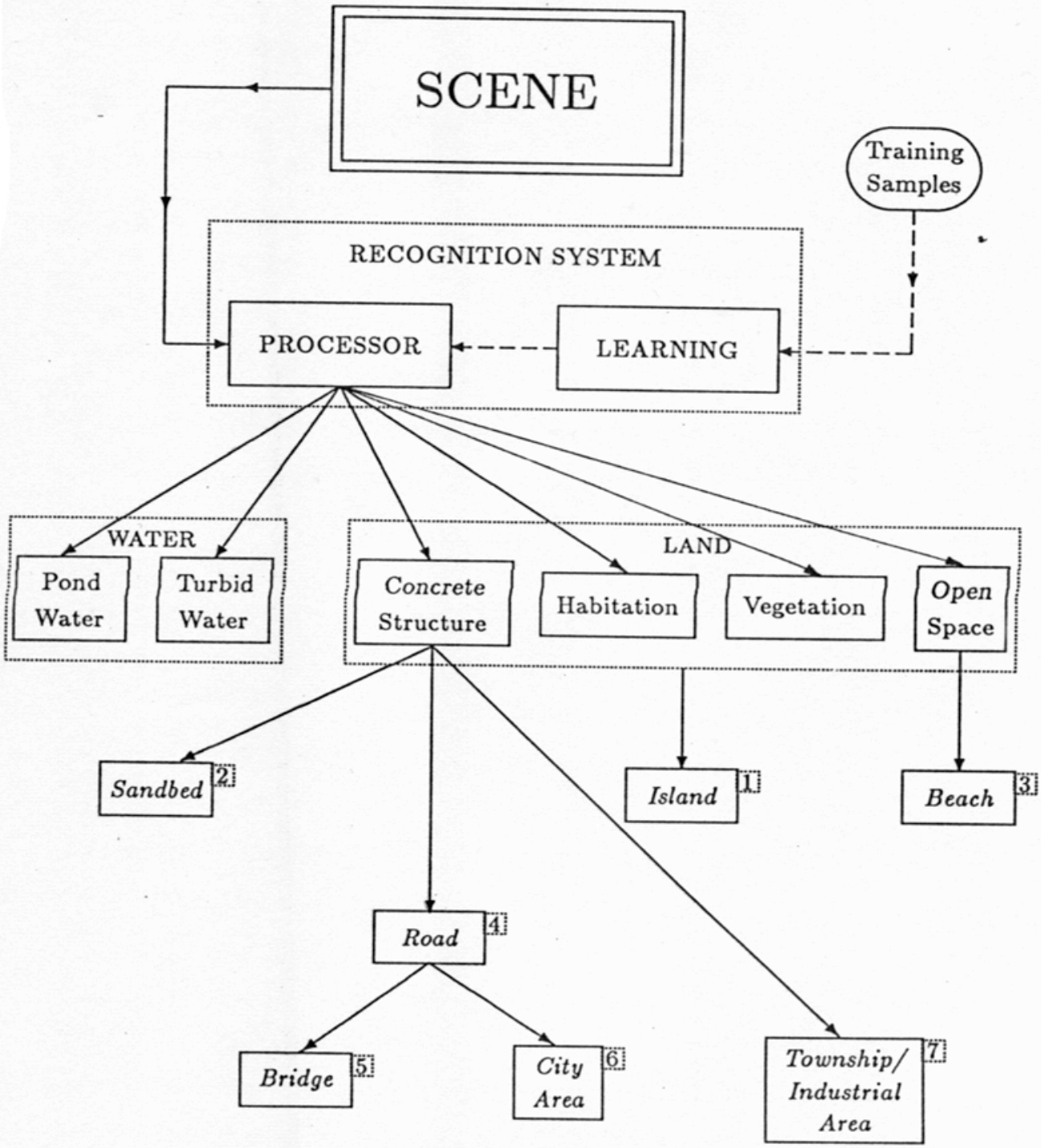


Figure 6.1 : Overall strategy diagram for detecting various target regions.

- *Road* : The *roads* are the strips of varied width or single dotted curves, locally approximated by straight lines or parabolas, although globally they are arbitrary curves of concrete structure pixels. The railway tracks and airport runways also come as *roads* here.

- *Bridge* : The only *bridges* considered here are the bridges on the water bodies. If a narrow linear concrete structure segment lies in between two disjoint water bodies (i.e., subdivides a water body into two distinct parts) and is connected with a *road* then it is referred to as a *bridge*. If the segment is not attached with any *road* then it is categorized as a *sandbed*.

- *City area and township/industrial areas* : The *city area* is the largest densed concrete structure region connected with many *roads*. By filling up all the non-concrete structure pixels with concrete structure pixels, the *city areas* are detected. The other densed concrete structure regions with moderate sizes represent *townships* and/or *industrial areas* and so these are classified as *township/industrial areas*. In case there are major townships on the opposite sides of a river connected with *bridge(s)*, those townships are considered as a single entity and the intermediate water bodies are taken as a part of the township. As there exist many buildings as well as many *roads* in a *city area*, it is extremely difficult to differentiate them inside the *city area*.

Corresponding to any scene, there are four available IRS band images, namely *blue*, *green*, *red* and *infrared*. It has been observed that the *green* and *infrared* band images are more sensitive than other band images to discriminate various land cover types [242]. Thus the data corresponding to these two band images are considered here as the features. Initially 50 pixels corresponding to each of the aforementioned six classes are chosen as the training samples. These training samples provide the spectral knowledge about the classes and the scene to the recognition system. Thus the system classifies all the pixels of the scene into six classes. The clustered image is processed further to detect various target regions. The sequence (numerals shown in

the small dashed boxes adjacent to the target boxes in Fig. 6.1) in which the targets are considered for identification is furnished below.

0. Initial processing : Initially, all the disjoint water bodies are identified through their boundary pixels. The narrow concrete structure segments (with maximum 3 pixel width) which lie between two different water bodies are found and at this stage, these segments are marked as the candidate regions for *bridges*.

1. *Island* : The land portions surrounded completely by water bodies are found and these are identified as *island*.

2. *Sandbed* : As mentioned earlier, the signature of the *sandbed* pixels in satellite imagery belong to the class *concrete structure*. The narrow concrete structure regions (with maximum 3 pixel width) adjacent to water bodies are recognized as *sandbeds*. Note that the candidate *bridge* pixels are not considered here to find such *sandbeds*. The *islands* consisting of only concrete structure pixels are also categorized as *sandbeds*.

(These *sandbed* pixels are eliminated from the concrete structure pixel set for detection of other attributes, namely *roads*, *city area*, *township/industrial areas* etc. in the following stages.)

3. *Beach* : The open space region with a moderate size (at least 25 pixels) adjacent to the *sandbeds* and/or water bodies are categorized as the *beaches*.

4. *Road* : Here we use three steps. In the first step, the clustered image is scanned horizontally, vertically and diagonally to extract some pixels as the candidate pixels for the possible *road* segments. The concrete structure pixel-runs with maximum 3 pixels are marked as the candidate pixels for *roads*. In the second step, the candidate pixel patterns are thinned. Finally, these thinned pixels are traversed and some obvious connections between the existing *road* segments are made. The traversed pixels provide the *road* patterns for a scene.

5. *Bridge* : The candidate pixels for *bridges* (which were identified during initial processing) are now checked whether they belong to the *roads* or not. If they belong to the *roads*, then they are confirmed as the *bridge* pixels. Otherwise, they are categorized as *sandbeds*.

(All the original concrete structure pixels connected with the *roads* are now put back. The resultant image is now referred to as the *extended roadmap*. This *extended roadmap* image includes *city area* and some *township/industrial areas*.)

6. *City area* : Using morphological dilation and erosion operations [53,55] on the *extended roadmap* image, the *city area* in the scene is detected. The largest dense concrete structure regions with a significant area ($\geq$ 625 pixels) in the *extended roadmap* image is identified as the *city area*. This is to be mentioned here that because of the intermixing of *roads* and buildings, it is quite difficult to differentiate them in the *city area*.

7. *Town/Industry area* : The *township/industrial areas* are detected from the initial clustered image. The dense concrete structure regions with significant sizes ($>$ 80 pixels) in the image are categorized as the possible *township/industrial areas*. Morphological dilation and erosion operations are also used to find such regions. As the characteristics of the *townships* and the *industrial areas* are more or less the same in the satellite images, we could not differentiate the two. ♣

8-directional Code :

To find various water bodies and the *road* patterns, the traversal algorithms used for connecting the pixels involve *8-directional chain code* (Fig. 6.2) to keep track of the direction. The pixels P_1, P_2, ..., P_8 in Fig. 6.2 are the 8 direct neighbors of the pixel P_0 so that the direction of the pixel P_i with respect to P_0 is coded as i ($i = 1, 2, \ldots, 8$). •

Some remarks :

The overall strategy and the sequence, in which various objects are detected,

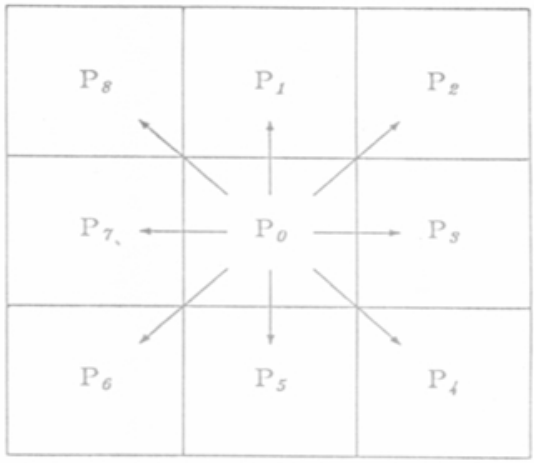


Figure 6.2 : 8-directional code.

have been stated above. The descriptions of the methods are provided in detail in the subsequent sections. It is to be mentioned here that the characteristics for various objects and the algorithms to detect those objects have been formulated keeping in view the Indian environment and the resolution of IRS images ($36.25m \times 36.25m$). In many developing countries like India, the construction of object regions and their interconnections are not, in general, on the same scale nor of the same specifications as in the developed countries in many cases. For example, the *roads* do not always have significant width in order to be reflected in the corresponding satellite images. Thus, it is possible to get *townships* not connected by any *road* which, in reality, should not be the case.

On the other hand, the pixels in the boundaries of homogeneous regions do not possess the exact match with a single land cover type and so such pixels sometimes reflect wrong land cover types in the corresponding satellite imagery. For example, the boundary pixels of various water bodies, in general, reflect concrete structures, which are not always found to be true. Because of such inconsistencies and the resolution of IRS imagery, we could not use some of the usual constraints about the objects to detect them. Many of the constraints have been applied very loosely and a few heuristic constraints have also been used in our proposed methods. These heuristics may sometime seem

to be redundant, but they are helpful in obtaining the actual target mixed with some noise. The heuristic rules are mentioned in the subsequent sections along with the description of the detection procedures. ♠

6.3 Islands, Sandbeds and Beaches

The approaches taken to detect the *islands*, *sandbeds* and *beaches* are described here in detail. As mentioned in the previous section, various water bodies are first of all identified through their boundary pixels. Then the concrete structure pixels lying between two different water bodies are marked as the candidate pixels for *bridges*. The *islands*, *sandbeds* and *beaches* are then detected.

6.3.1 Determining various water bodies

From the classified (with *single* and *first choices* of the multivalued recognition system) image, all the possible boundary pixels of water bodies are initially identified. To perform this, the image is scanned horizontally from left to right starting with the first row. The water pixels either followed by or preceded by non-water (land) pixels are considered as the boundary water pixels. Then the image is traversed vertically from top to bottom starting with the first column to find similarly the boundary water pixels in the vertical direction.

With the boundary water pixels, various distinct water bodies are then identified. To find this, the aforesaid marked image is scanned horizontally from left to right starting with the first row to search for a boundary water pixel. Starting with that, all the connected boundary water pixels are traversed using octal codes [Fig. 6.2]. The neighboring pixels of the starting pixel are searched for obtaining another boundary water pixel. In case, only one boundary water pixel is found in its 8 neighboring positions [Fig. 6.2], the pixel is taken for the traversal and the direction of traversal is noted. In case,

more than one boundary water pixel is found in its neighboring positions, one of them is taken for next traversal and the remaining boundary water pixels with their directions are kept reserved. In the traversed direction, either the left or right pixel of the next selected pixel has to be land (non-water). In case the left (right) one is land, then the selection of all following traversal pixels must have land pixels in the left (right) pixel positions of the traversed directions for the traversal with the same starting point. These descriptions are also noted for all the reserved pixels.

To continue the traversal, the neighboring pixels with respect to the current pixel are searched for the boundary water pixels (not already traversed). If only one traversable pixel is found, then that pixel is taken for next traversal. In case, multiple traversable pixels are found, the next movement is taken by choosing that direction which most closely matches that of the previous description of the path [190]. The remaining traversable pixels are kept reserved. This enables one to use these pixels for continuing the description of other contours connected to them.

The traversal procedure is continued until there is no non-traversed boundary pixel found in the 8 neighbors of the current pixel. After that, a pixel is taken from the reserved set provided it is not already traversed, and the traversal procedure is similarly continued. The traversal procedure is terminated for a water body when there is no more non-traversed reserve pixels. Therefore, the traversed pixels constitute the boundary of a water body.

The image is again scanned horizontally from left to right to search for a new non-traversed boundary water pixel. Following the same procedure, another boundary of a water body is found and the boundary pixels are assigned a new value. In this way, the boundaries of all the distinct water bodies are identified. ♣

6.3.2 Observations related to certain targets

The method of identifying different water bodies has been described in the previous section. Different possible cases arising out of different types of water boundaries are shown in figures 6.3(a)-(i). With the help of these figures, certain observations are made below.

- It is already mentioned earlier that the pixels corresponding to the *sandbeds* belong to the concrete structure class. The narrow concrete structure regions adjacent to the water pixels are categorized as *sandbeds*. The width of the *sandbeds* along the water bodies is, in general, not significant [figures 6.3(a)-(i)]. The width of such *sandbeds* is restricted to 3 pixels for practical reasons.

- In figures 6.3(a)-(c), each of the boundaries is seen to make an open path. In figures 6.3(d)-(f), each of the boundaries is seen to make a closed path whereas in figures 6.3(g)-(i), each of the boundaries is seen to make a multiple closed path. In the case of a closed path, it may enclose a water body [figures 6.3(d)&(g)] or a land portion [figures 6.3(e),(h)& (i)] and/or a concrete structure region [figures 6.3(f)&(i)]. The enclosed land portions are identified as *islands* and the enclosed concrete structure regions are categorized as *sandbeds*.

- There may be a narrow concrete structure region between two distinct water bodies (water boundaries), as seen in figures 6.3(b)&(c) (enclosed by dark boundaries). In such cases, the concrete region may be visualized to subdivide a water body into two distinct parts. Two possibilities may arise with such a concrete structure region. The region may represent a *bridge* and in such a case, there should be a *road* passing through the *bridge* [Fig. 6.3(b)]. The other possibility is that the region may represent a *sandbed* and in that case, the region will not be connected with any *road* [Fig. 6.3(c)]. So, for the time being such concrete structure regions are marked as the candidate region for *bridges*. Final decision

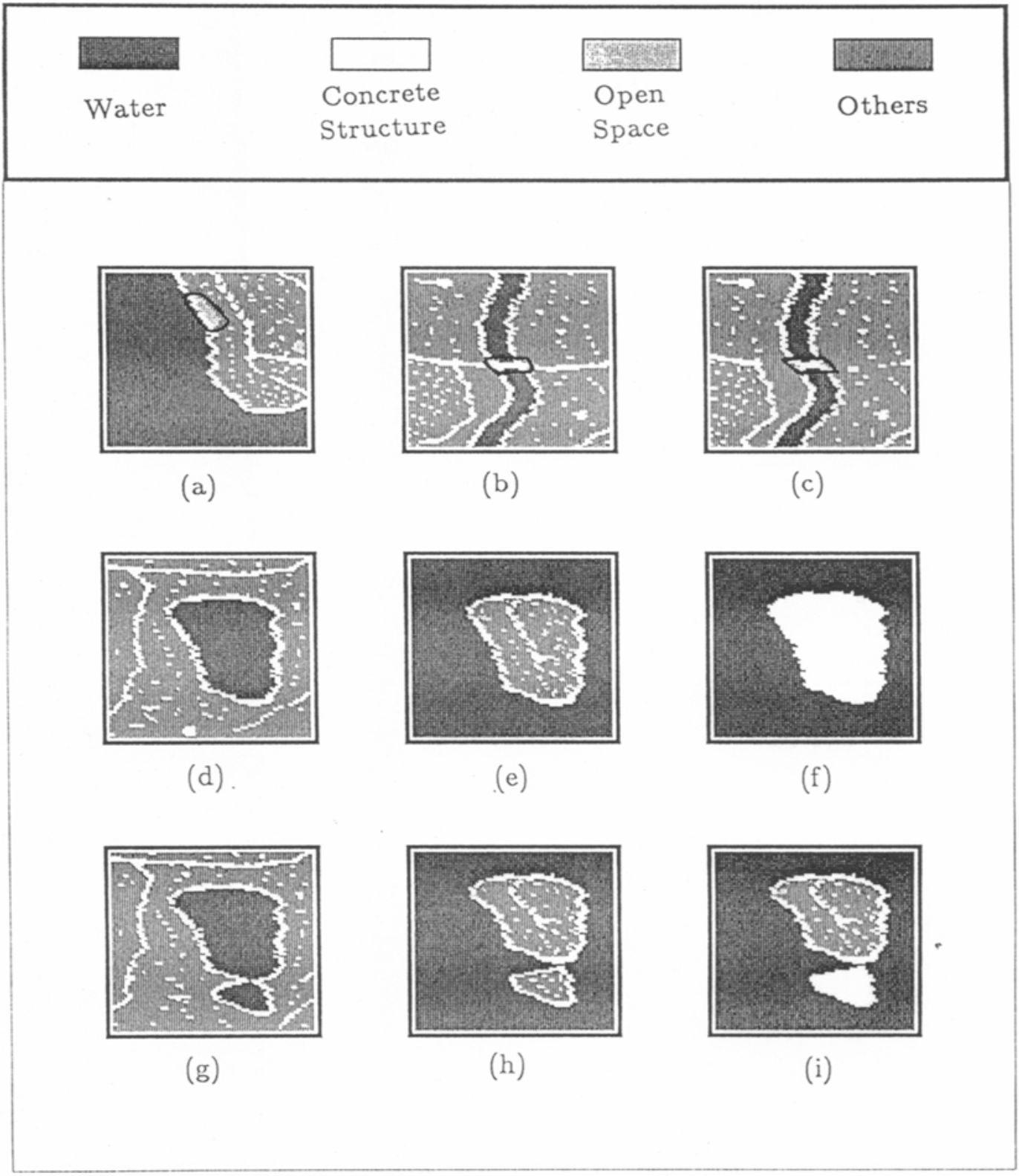


Figure 6.3 : Different targets related to water bodies.

on these regions is taken in section 6.4 after the detection of *roads* of the scene.

- If an open space region with a moderate size ($\geq$ 25 pixels) is found adjacent to a water body and/or *sandbeds*, as seen in Fig. 6.3(a) (enclosed by dark boundaries), the open space region is identified as a *beach*.

The aforementioned observations are utilized in our experiments for detecting the *islands*, *sandbeds*, *beaches* and the candidate regions for *bridges*. ♠

6.4 Roads, Bridges, City Area and Township/Industrial Areas

The method of identifying the *road* patterns is initially described here from concrete structure pixels. As the *bridges* are part of the *roads*, these are obtained from the *road* patterns. Then the *road* patterns are analyzed to find the possible *city area* in the image. The *township/industrial areas* are detected from the originally classified concrete structure pixels.

6.4.1 Detection of roads

To find the *roads*, the concrete structure pixels except the *sandbeds* (as detected in the previous section) are analyzed. The width of the *roads* has an upper bound which is considered here to be 108.75m (3 pixels) for practical reasons. So all the pixels lying on concrete curves with width not more than 3 pixels are initially considered as the candidate set for *roads*. As the size of a pixel is 36.25m, all the portions of actual *roads* may not be reflected as concrete structures and as a result, the candidate pixels may constitute some broken curve segments. In order to identify them, a traverse through the candidate pixels is required. Before traversing, one also needs to thin the candidate

road patterns so that a unique traverse can be made through them. The total procedure to find *road* patterns therefore consists of three parts - *(i)* selecting the candidate pixels for *roads*, *(ii)* thinning the candidate *road* patterns and *(iii)* traversing the thinned patterns to make some obvious connections between the curve segments. These are sequentially described below.

(For a better understanding of the total procedure for finding *roads*, we have included Fig. 6.4 as an example. Fig. 6.4(a) represents an input (concrete structure patterns) of dimension 80×80. The successive results are shown in figures 6.4(b)-(i).)

A. Selection of candidate road pixels

To find the candidate pixel set for *roads*, the image is initially scanned horizontally from left to right starting with the first row and the concrete structure runs of maximum 3 pixels are marked as the candidate pixels. Note that this horizontal scan can identify only the vertically elongated patterns and some diagonally elongated patterns [Fig. 6.4(b)]. The image is then scanned vertically from top to bottom starting with the first column in order to identify the horizontally elongated pattern along with some diagonally elongated patterns [Fig. 6.4(c)]. The aforesaid scans may not still be able to identify all the diagonally elongated patterns and some junction points of the *roads* [Fig. 6.4(c)]. For this purpose, the image is scanned in both the diagonal directions. The resulting candidate pixels for *roads* are shown in Fig. 6.4(d) corresponding to the concrete structure patterns in Fig. 6.4(a). •

B. Thinning

The parallel thinning algorithm proposed by Zhang and Suen [228] is applied here for thinning the pattern. In this procedure, the new value given to a point at the kth iteration depends on its own value as well as those of its eight neighbors [Fig. 6.2] at the (k-1)th iteration. The method consists of

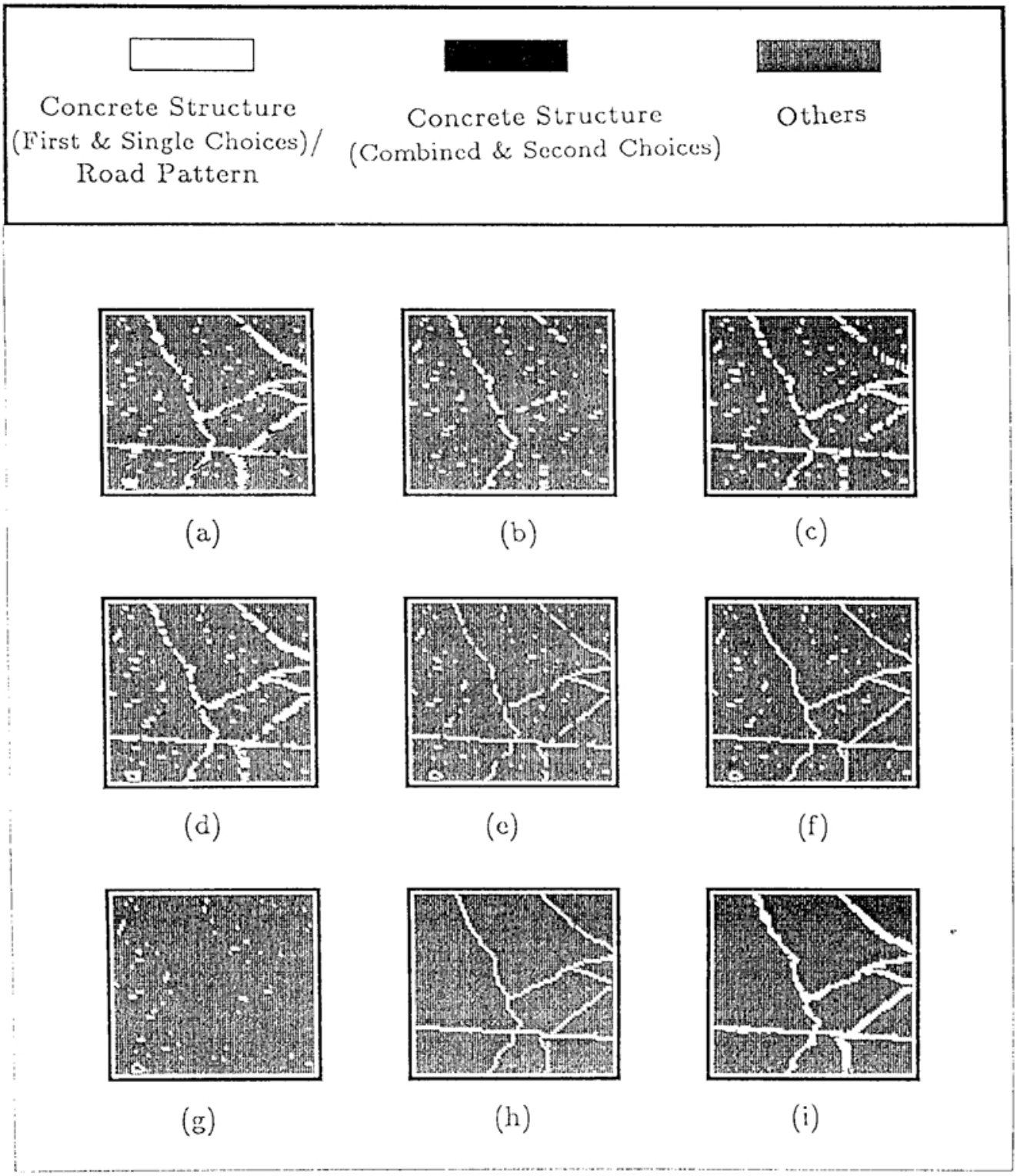


Figure 6.4 : Illustrating the method of finding *roads*.

removing all the contour points of the pattern except those points that belong to the skeleton. The algorithm has two subiterations as discussed below.

In the first subiteration, the contour point P_0 [Fig. 6.2] is deleted (i.e., 0 is assigned) from the digital pattern if it satisfies the following conditions

$$(a) \quad 2 \leq B(P_0) \leq 6$$
$$(b) \quad A(P_0) = 1$$
$$(c) \quad P_2 \times P_4 \times P_6 = 0$$
$$(d) \quad P_2 \times P_4 \times P_8 = 0$$

where $A(P_0)$ is the number of 01 patterns in the ordered set P_1, P_2, ..., P_8 that is eight neighbors of P_0 [Fig. 6.2] and $B(P_0)$ is the number of non-zero neighbors of P_0, that is

$$B(P_0) = P_1 + P_2 + \ldots + P_8.$$

In the second subiteration, only condition *(c)* and *(d)* are changed as follows

$$(c') \quad P_2 \times P_6 \times P_8 = 0$$
$$(d') \quad P_4 \times P_6 \times P_8 = 0$$

and the rest remaining the same.

The iterations continue until no more points can be removed. •

For the purpose of thinning, all the candidate pixels of the image are assigned value 1 and the remaining pixels are considered to be 0. So the image matrix consists of 1 and 0. Using the aforementioned thinning algorithm [228], the candidate curve patterns are thinned. The resultant thinned patterns provide some broken curve segments [Fig. 6.4(*e*)]. •

C. Traversal and joining

The narrow concrete structure curves consist of *roads* and road like structures (e.g., railway tracks, airport runways). In India, the *roads* (or some of its portions) are sometimes very narrow. In some places, the *roads* are

surrounded by big trees. Because of such unavoidable circumstances, some pixels on the real *roads* may not be reflected as concrete structures and will make the curves discontinuous. The information about many of such ambiguous/distorted curves can be obtained from the *second* and *combined choices* provided by the multivalued recognition system (described in chapter 4). When there is a big building adjacent to a *road*, the corresponding *road* portion in the image will look wider with the concrete structure pixels. Therefore, the pixels on that *road* portion may not come under the candidate pixel set; thereby resulting in a discontinuation in the candidate road patterns [Fig. 6.4(d)]. Again, some of the candidate pixels belonging actually to the *roads* may be removed during thinning process [Fig. 6.4(e)].

To overcome these problems, a new traversal algorithm (different from that used in section 6.3.1) through the thinned *road* patterns is adopted here. During traversal, it always looks forward to the obvious joining between the *road* segments patterns. One of the inherent properties of the proposed traversal procedure is that the tracing of *road* segments proceeds more or less in the same direction. So to keep track of the direction, the 8-directional code [Fig. 6.2] is used during the traversal. After analyzing the potentialities of the pixels to be in the *road* segments, conceptually, 5 different potential groups (coded by A-E) are assumed. These are

- **A**: The pixels present in the thinned concrete structure patterns.
- **B**: The candidate concrete structure pixels removed during thinning.
- **C**: The pixels already traversed.
- **D**: Other concrete structure pixels decided either by the *single* or *first* or *second* or *combined choices* of the multivalued recognition system.
- **E**: The pixels not belonging to the aforementioned categories.

Obviously the pixels in group **A** have the highest potentiality than the pixels in other groups to be in the *road* segments.

The thinned candidate *road* patterns provide some broken *road* segments. The middle most pixel of each of the disjoint *road* segments is chosen as the starting pixel for traversing the corresponding *road* segment. The intention behind this choosing is to allow the growth of the *roads* in all the possible directions.

First of all, the longest thinned *road* segment is chosen and the traversal begins with the starting pixel of that segment. To find the next pixel to be traversed, 8 neighboring positions of the starting pixel are analyzed. One of the neighboring pixels with potential category **A** is taken for the traversal and the directional code of the pixel with respect to the starting pixel is noted. Other neighboring pixels which having the potential code **A** are marked as reserved for the future traversal and the directional codes of the pixels with respect to the starting pixel are also noted.

Now at any point of time, there may be many possible choices for the next traversal depending on the potential categories of the neighboring pixels and the current traversal direction. Twelve such situations are considered in our experiment and are shown in Fig. 6.5. These situations are arranged in an order so that if one fails, then the next one is considered for choosing the next traversal pixel. (One may always think of some more situations but we restrict ourselves to these twelve cases). The current traversal directions are shown here by thick lines, the next traversal directions are shown by thin lines, and the intermediate movements are shown by dotted lines. The solid circles inside a box, dotted box and solid line box represent the current, intermediate and final traversal positions of the pixels respectively. Note that the situations in (5) and (8)-(12) of Fig. 6.5 are considered for the possible extension (i.e., looking forward to join the present curve segment with other curves) of the existing curve segment, and these situations are considered in case the pixel-count of the current traversing curve segment is at least five.

Now a non-traversed reserved pixel is taken and the traversing of the curve is continued assuming its directional code as the current traversing direction. When all the reserved pixels are exhausted (and/or traversed) and

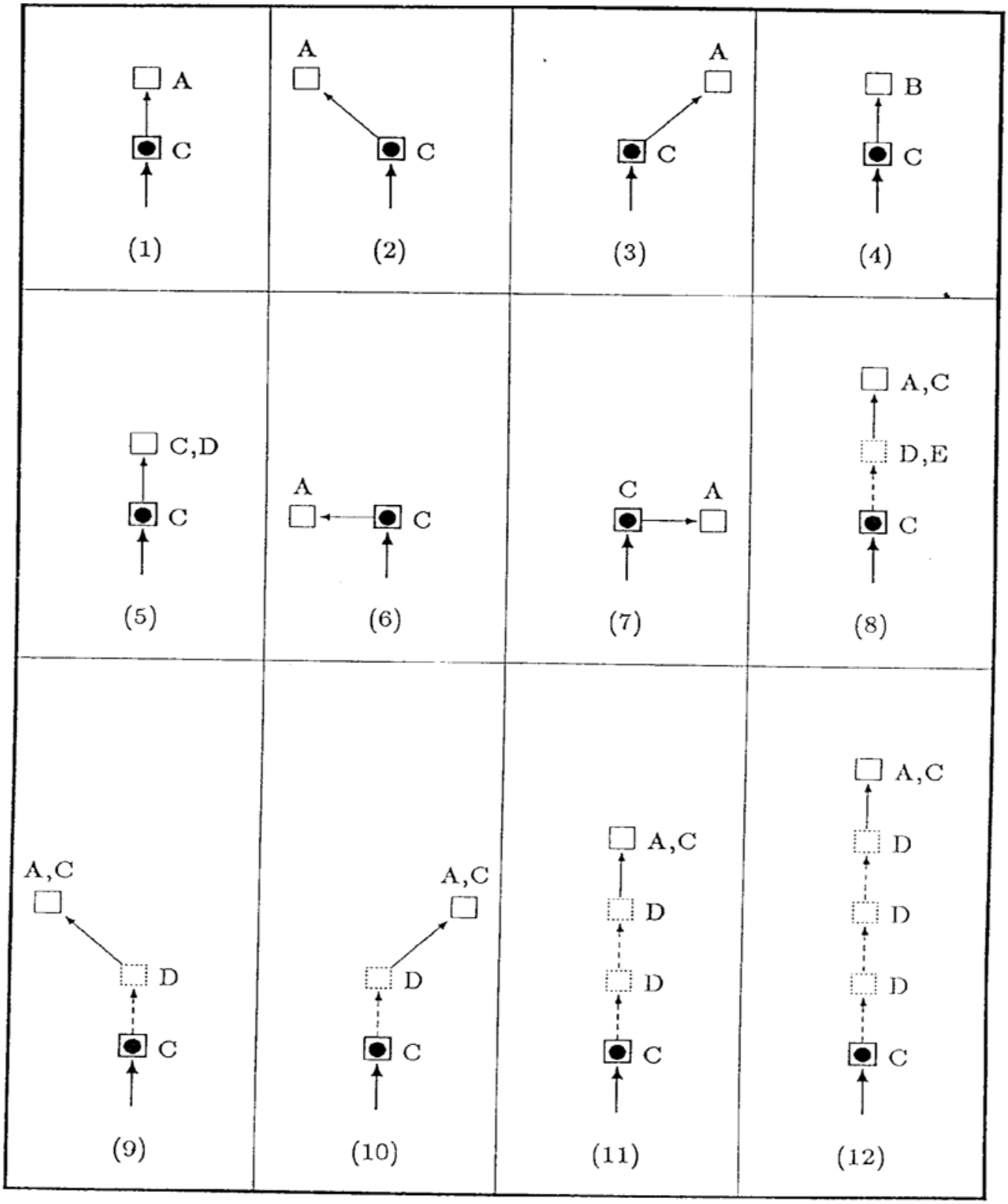


Figure 6.5 : Various movements considered for traversing road segments.

none of the aforementioned twelve situations [Fig. 6.5] is found, the traversing is stopped for the present curve. The same procedure is repeated for other isolated *road* segments in the scene. ♣

D. Use of multiple choices

It has already been mentioned earlier that one pixel may represent more than one land cover type. As the recognition system provides multiple choices, the information about all the classes present in a particular pixel can be obtained. For completing the task of identification of *road* patterns, the use of multiple choices is directly realized. Out of the twelve movements considered [Fig. 6.5] for traversing the *road* segments, the movements in (5), (8)-(12) are governed by the *second* and *combined choices*. Note that after using *second* and *combined choices* in traversing the segments in Fig. 6.4(e), we have more complete information as in Fig. 6.4(f). •

The aforementioned traversal procedure results in narrow concrete structure curves. The curve patterns may also include some spurious wiggles and some isolated noisy segments. These segments with insignificant lengths (< 10 pixels) are discarded from the curve patterns. The removal of the stray points as seen in Fig. 6.4(g) makes the resultant curve patterns [Fig. 6.4(h)] further prominent. The curve segments thus obtained represent the skeleton version of the *roads*. To get the *roads* back from the skeletons, we now put back all the concrete structure pixels lying in the 8 neighboring positions corresponding to the pixels on the previously obtained skeletons. The final *road* pattern corresponding to the image in Fig. 6.4(a) is displayed in Fig. 6.4(i). Note that the *bridges* are included in the *road* patterns. ♣

E. Bridges

It may be recalled here that some narrow concrete structure regions lying between two different water bodies were marked earlier (section 6.3) as the

candidate regions for *bridges*. According to the characteristics of *bridges*, as defined in section 6.2.2, the candidate regions belonging to the *roads* are confirmed as the *bridges*. The remaining candidate regions (not belonging to the *roads*) are categorized as the *sandbeds*. ♣

6.4.2 City area and township/industrial areas

The concrete structure pixels connected with the *road* patterns are now put back to represent the *extended roadmap*. To find the *city area*, this *extended roadmap* image is analyzed. The *township/industrial areas* are detected from the original concrete structure patterns. The largest compact concrete structure region with a significant size (≥ 625) present in the scene is referred to as the *city area*. Other dense concrete structure regions with more than 80 pixels are assumed to represent either *townships* or *industrial areas*. The morphological dilation and erosion operations [53,55] are found very effective to find these target areas. Before describing their detection procedures, a short introduction to the mathematical morphology is furnished below.

A. Mathematical morphology [53,55]

Mathematical morphology is a tool for image processing which provides a system of operators. This system of operators when acting on complex shapes, decomposes them into their meaningful parts and filter out extraneous parts. They also serve to highlight spatial pattern and to remove spatial noise.

The primary morphological operations are dilation and erosion. Other morphological operations, such as opening and closing, are composed from dilation and erosion.

Let A and B be two sets in N-space (E^N) with elements a and b respectively, $a = (a_1, a_2, \ldots, a_N)$ and $b = (b_1, b_2, \ldots, b_N)$ being N-tuple of element

coordinates. The dilation of $\mathcal{A}$ by $\mathcal{B}$ is denoted by $\mathcal{A} \bigoplus \mathcal{B}$ and is defined by

$$\mathcal{A} \bigoplus \mathcal{B} = \left\{ c \in E^N / c = a + b \quad for\ some\ a \in \mathcal{A} \quad and \quad b \in \mathcal{B} \right\} \tag{6.1}$$

Here $\mathcal{A}$ is associated with the image underlying morphologic processing and $\mathcal{B}$ is referred to as the structuring element, that shape which acts on $\mathcal{A}$ through the dilation operation to produce the result $\mathcal{A} \bigoplus \mathcal{B}$.

Erosion is the morphological dual to dilation. The erosion of $\mathcal{A}$ by $\mathcal{B}$ is denoted by $\mathcal{A} \bigodot \mathcal{B}$ and is defined by

$$A \bigodot B = \left\{ x \in E^N \ / \ x + b \in A \quad for\ every \quad b \in B \right\} \tag{6.2}$$

The dilation and erosion operations have been combined in a number of ways to detect the *city area* and the *township/industrial areas*. The same structuring element is considered throughout this paper which is of size 5×5, all with the concrete structure pixels and origin at the centre i.e., in the position (3,3). •

B. City area

The *city area* is obtained from the *extended roadmap* image, as the *cities* are always connected with the other significant regions in the scene by some major *roads*. The dilation operation is initially applied 3 times on the *extended roadmap* which fills up the intermediate non-concrete structure pixels by concrete structure pixels and results in few isolated compact concrete structure regions. The dilation operations also expand the outer boundaries of the obtained compact regions beyond its size. So to bring back these expanded regions more or less to their previous sizes, three erosion operations are applied with the same structuring element. Next, for deletion of all the insignificant regions (regions with sizes less than 25×25 pixels), the erosion operations has been applied five times. Five dilation operations bring back the residual regions to their previous boundaries.

After the aforesaid processing, the largest compact region available in the image is considered to represent the *city area*. The other remaining compact regions may be viewed as various *township/industrial areas*. As all the *roads* are not detectable from the satellite images, many *township/industrial areas* may not be obtained from the *extended roadmap* image. For this reason, we used the original set of concrete structure pixels to extract them. The way of extraction is described below. •

C. Township/industrial areas

Various dense concrete structure regions with a moderate size (> 80 pixels) are considered here as *township/industrial areas*. From the original set of concrete structure pixels, the *sandbed* pixels are first of all discarded. The concrete structure pixels belonging to the *city area* are also deleted from the concrete structure pattern set. The erosion operation is applied twice on this concrete structure pattern image, which results in deletion of all the concrete regions with size less than 9×9. Now the dilation operation is applied four times to replace some of the intermediate non-concrete structure pixels by concrete structure pixels. Then the erosion operation is applied twice to bring back the various isolated concrete structure regions more or less to their previous sizes. The residual compact concrete structure regions are categorized as the *township/industrial areas*. •

D. Some remarks

It is to be mentioned here that the minimum width of the *road* that can be detected by the proposed algorithm is decided by the pixel resolution of the image. In the present case, it is $36.25m$ (i.e., 1 pixel). The maximum width of the *road* has been considered here to be 3 pixels i.e., $108.75m$. Similarly a *beach area*, as we considered, consists of minimum 25 pixels; the *city area* consists of minimum 625 pixels; the *township area* consists of minimum 81 pixels. These ill-defined object regions do not have any restriction on their maximum sizes. Note that the aforesaid restrictions are decided heuristically and can always be modified according to needs. ♠

6.5 Implementation and Results

Data sets : The data used for the present work are acquired from Indian Remote Sensing Satellite (IRS). The IRS data is obtained through satellite IRS-1A which has been launched from a cosmodrome at Baikanaur, in the republic of Kazhakstan in the USSR in March, 1988. This is a circular sun-synchronous satellite, rotating around the earth at the rate of 14 orbits per day, at an altitude of 904 km and a repetition cycle of 22 days [265]. This satellite is equipped with two different sensors: $LISS\text{-}I$ and $LISS\text{-}II$. $LISS\text{-}I$ has a resolution of $72.5m \times 72.5m$ while $LISS\text{-}II$ has a resolution of $36.25m \times 36.25m$. Data used for this work was obtained from $LISS\text{-}II$ sensor. $LISS\text{-}II$ has a focal length of $324.4m$ with the spectral range between $0.45\mu m - 0.86\mu m$ (micrometer). The radiometric resolution is 128. The whole spectral range has been decomposed into four bands namely *blue* $(0.45\mu m - 0.52\mu m)$, *green* $(0.52\mu m - 0.59\mu m)$, *red* $(0.62\mu m - 0.68\mu m)$, and *infrared* $(0.77\mu m - 0.86\mu m)$ [265]. •

To demonstrate the effectiveness of the proposed methodologies, these have been implemented on some IRS image frames corresponding to various scenes in India. The results were found to be quite satisfactory in all the cases. The results are demonstrated here on two of these IRS frames. These two frames represent two major cities in India, namely Bombay and Calcutta. The image frames of Bombay and Calcutta were taken on 28th January, 1990 and 27th November, 1989 respectively. Each of these frames consists of 512 rows and 512 columns. For every scene, we have images corresponding to four spectral bands. Since the *green* and *infrared* images were found to be more sensitive than other band images to discriminate various land cover types [242], these two band images have been used as features here. Figures 6.6(a)&(b) show the *green* and *infrared* band images respectively of the city Bombay. The *green* and *infrared* images of Calcutta are shown in figures 6.7(a)&(b) respectively. In these input band images, the numbers, as mentioned with the 16 different gray labels, represent the actual values by the corresponding gray values in the displayed images.

The multivalued recognition system (described in chapter 4) has been applied on each of the IRS image frames to classify its pixels into six classes, namely *pond water, turbid water, concrete structure, habitation, vegetation* and

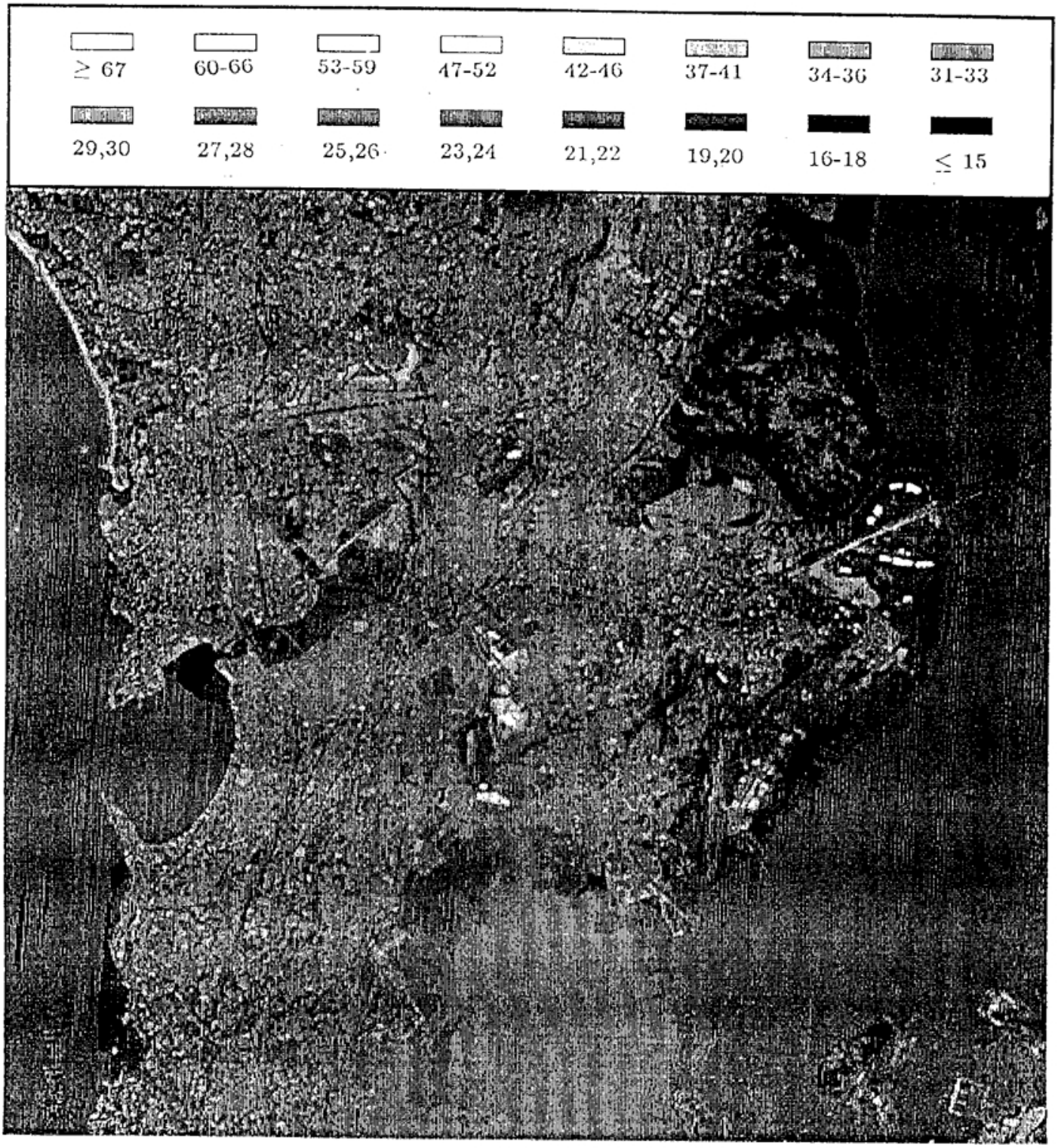


Figure 6.6(a) : IRS Bombay Band-2 (*green*) image.

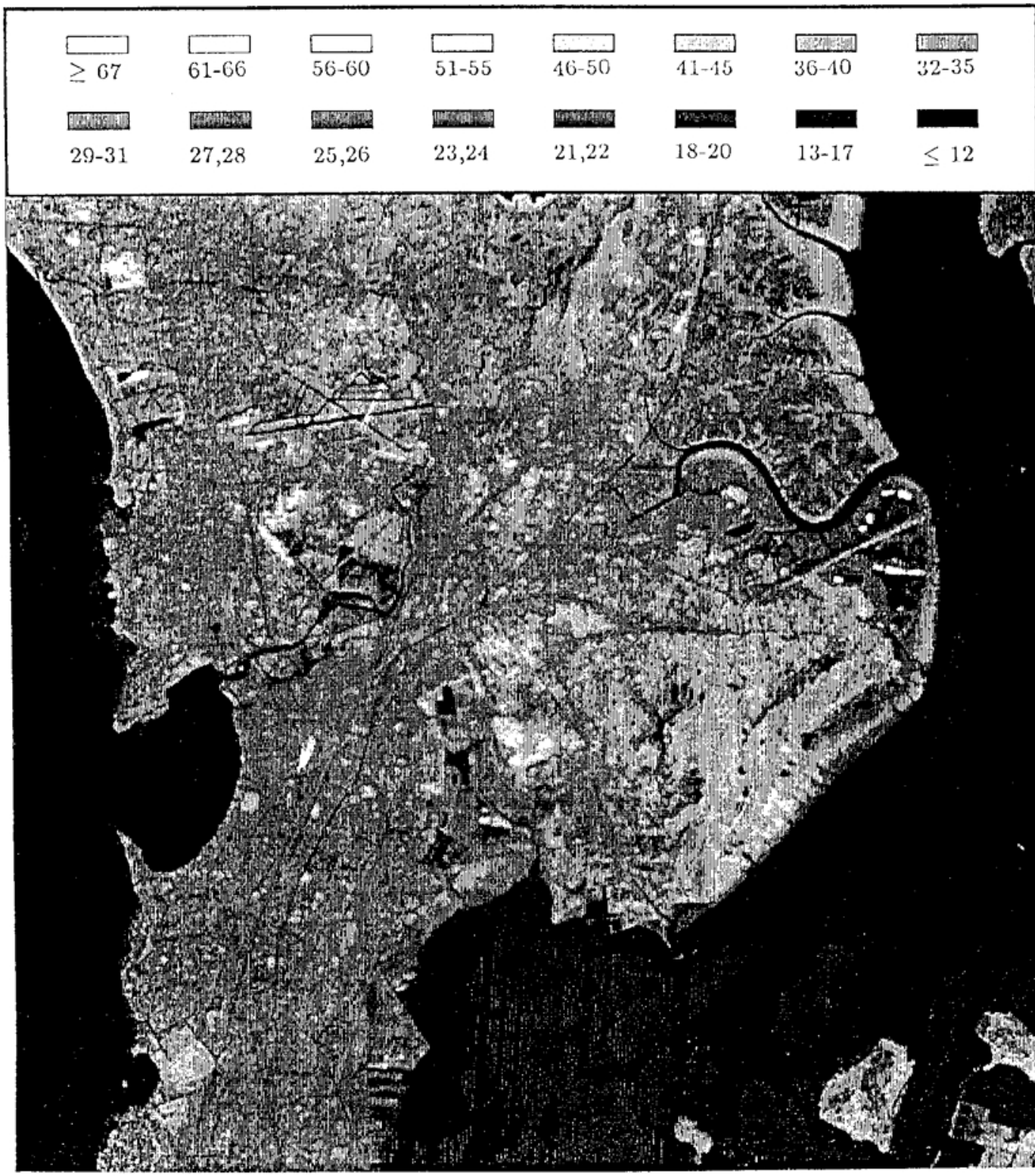


Figure 6.6(b) : IRS Bombay Band-4 (*infrared*) image.

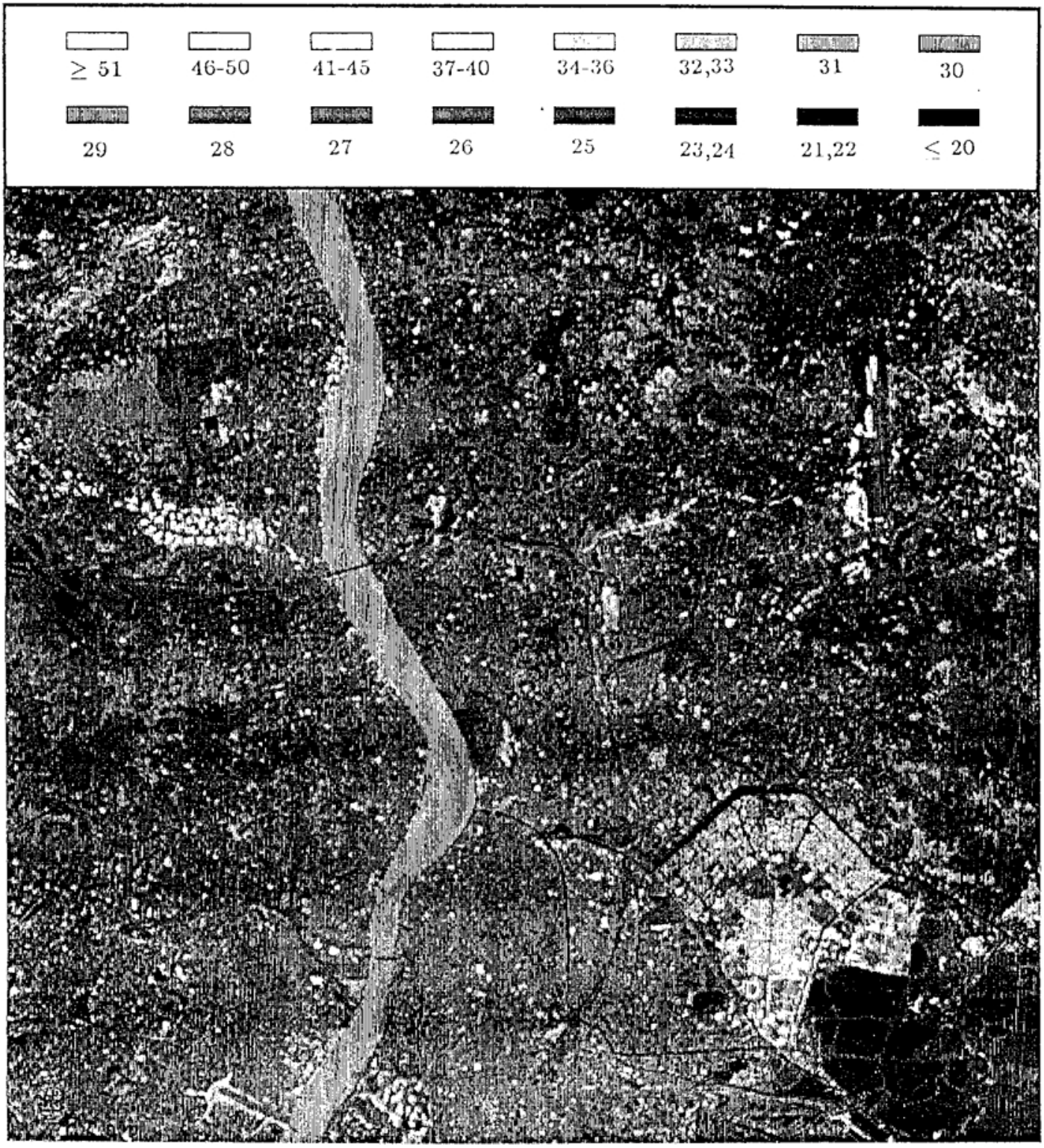


Figure 6.7(a) : IRS Calcutta Band-2 (*green*) image.

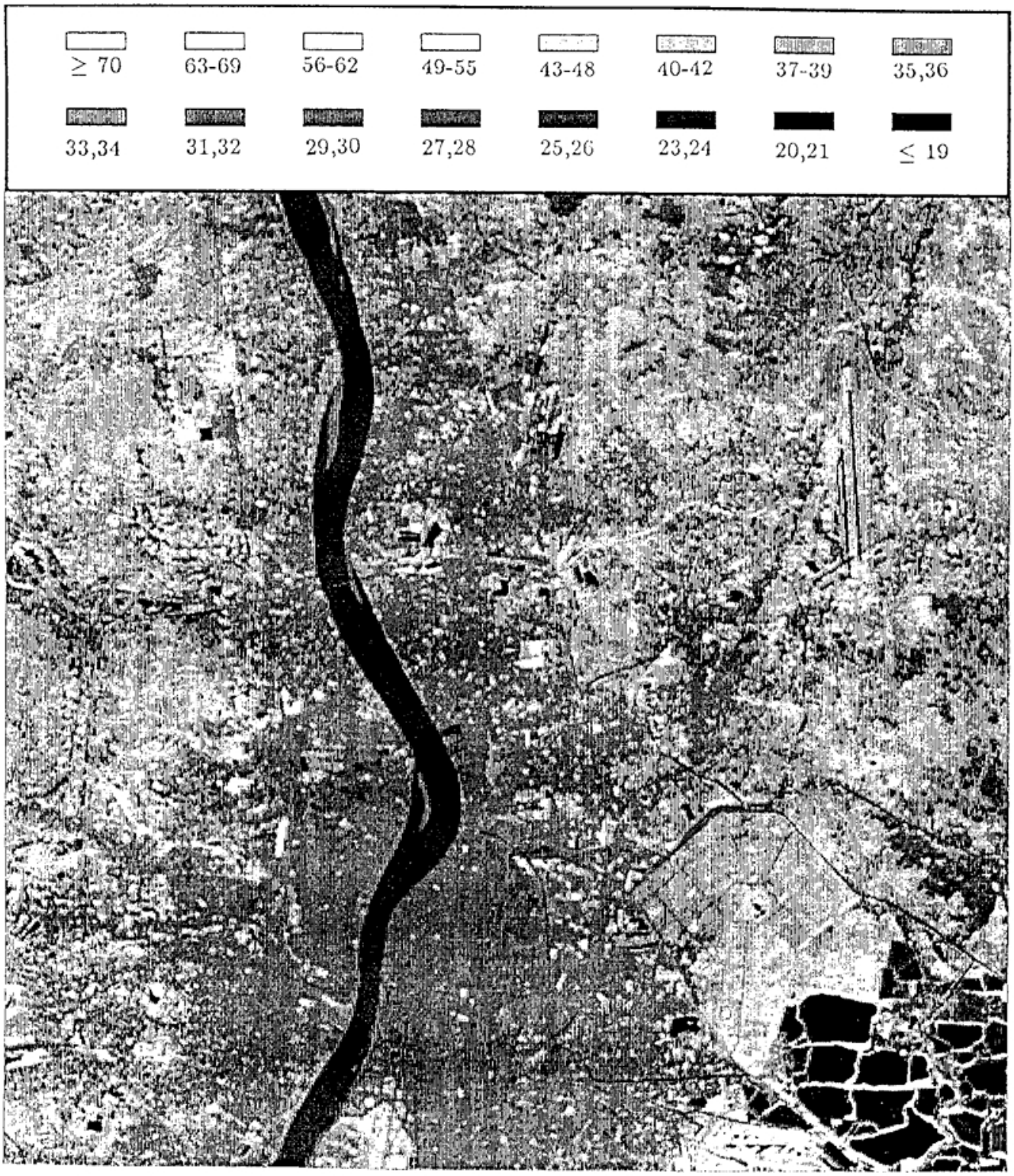


Figure 6.7(b) : IRS Calcutta Band-4 (*infrared*) image.

open space. The classified Bombay and Calcutta images are shown in figures 6.8(a)&(b) respectively. As the recognition system is multivalued, a pixel may be classified into more than one class. In the clustered images 6.8(a)&(b), each of the pixels reflects only the class having maximum similarity value. This is shown for the convenience of representation.

The huge turbid water bodies [Fig. 6.8(a)], above which the Bombay city is situated, represent the Arabic sea. The vertically elongated turbid water body [Fig. 6.8(b)], which bisects the Calcutta image frame, corresponds to the river Ganges. The concrete structure pixels (only in the *single* and *first choices*) of Bombay and Calcutta frames are shown in figures 6.9(a)&(b) respectively. It is to be mentioned here that the concrete structure pixels provide the useful information for detecting various ill-defined object regions. To visualize the concept of multivalued classification, the concrete structure pixels in *single*, *first* and other (*second* and *combined*) choices are displayed in figures 6.10(a)&(b) corresponding to Bombay and Calcutta scenes respectively.

Various ill-defined object regions, namely *roads*, *bridges*, *islands*, *sandbeds*, *beaches*, *city area*, *township/industrial areas* etc. are then detected from the clustered images. The detected *sandbeds*, *islands* and *beaches* in Bombay frame are shown in Fig. 6.11(a). There is no *beach* or *island* in the Calcutta frame. Thus, only the *sandbeds* in Calcutta image are displayed in Fig. 6.11(b). The *sandbeds* which are completely surrounded by water bodies, the *islands* and the *beaches* are marked with white boundaries in the above figures. The *roads* of Bombay and Calcutta are shown in figures 6.12(a)&(b) respectively. The *bridges* are shown (marked with dark boundaries) in the *roadmap* images. The detected *city areas* and *township/industrial areas* in Bombay and Calcutta frames are displayed in figures 6.13(a)&(b) respectively. In order to provide better idea of locations (or better reference) of various detected objects, the water bodies are always shown in all the image frames.

To illustrate the performance of the system in detecting various target regions, some of the important findings are listed below:

- Two nearby *beach areas* [Fig. 6.11(a)] detected from the Bombay image frame correspond to a single *beach*, known as Jhuhu beach.

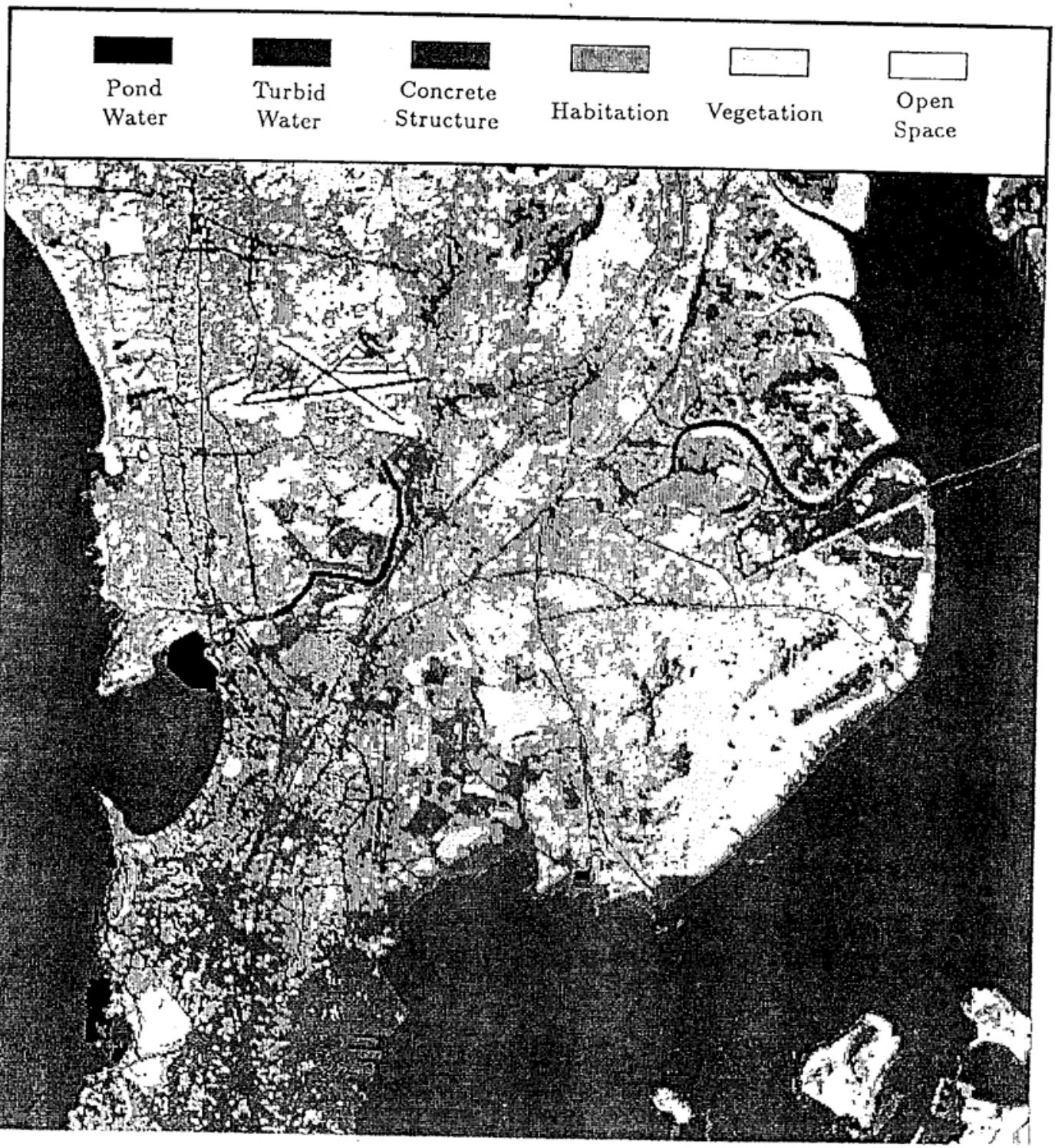


Figure 6.8(a) : IRS Bombay classified (clustered) image.

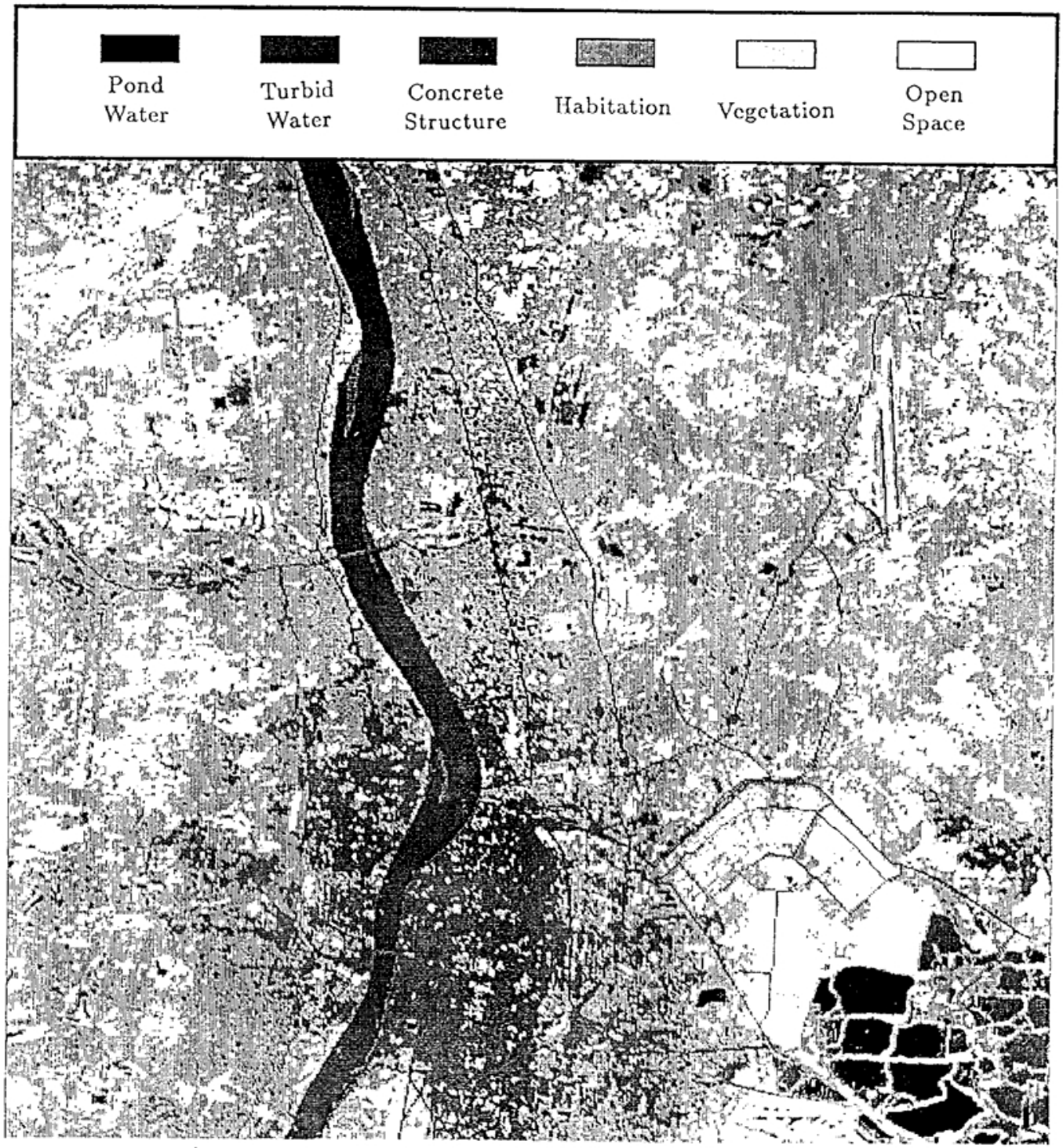


Figure 6.8(b) : IRS Calcutta classified (clustered) image.

Figure 6.9(a) : Concrete structures in *single* and *first choices* of IRS Bombay image.

Figure 6.9(b) : Concrete structures in *single* and *first choices* of IRS Calcutta image.

Figure 6.10(a) : Concrete structures in multiple choices of IRS Bombay image.

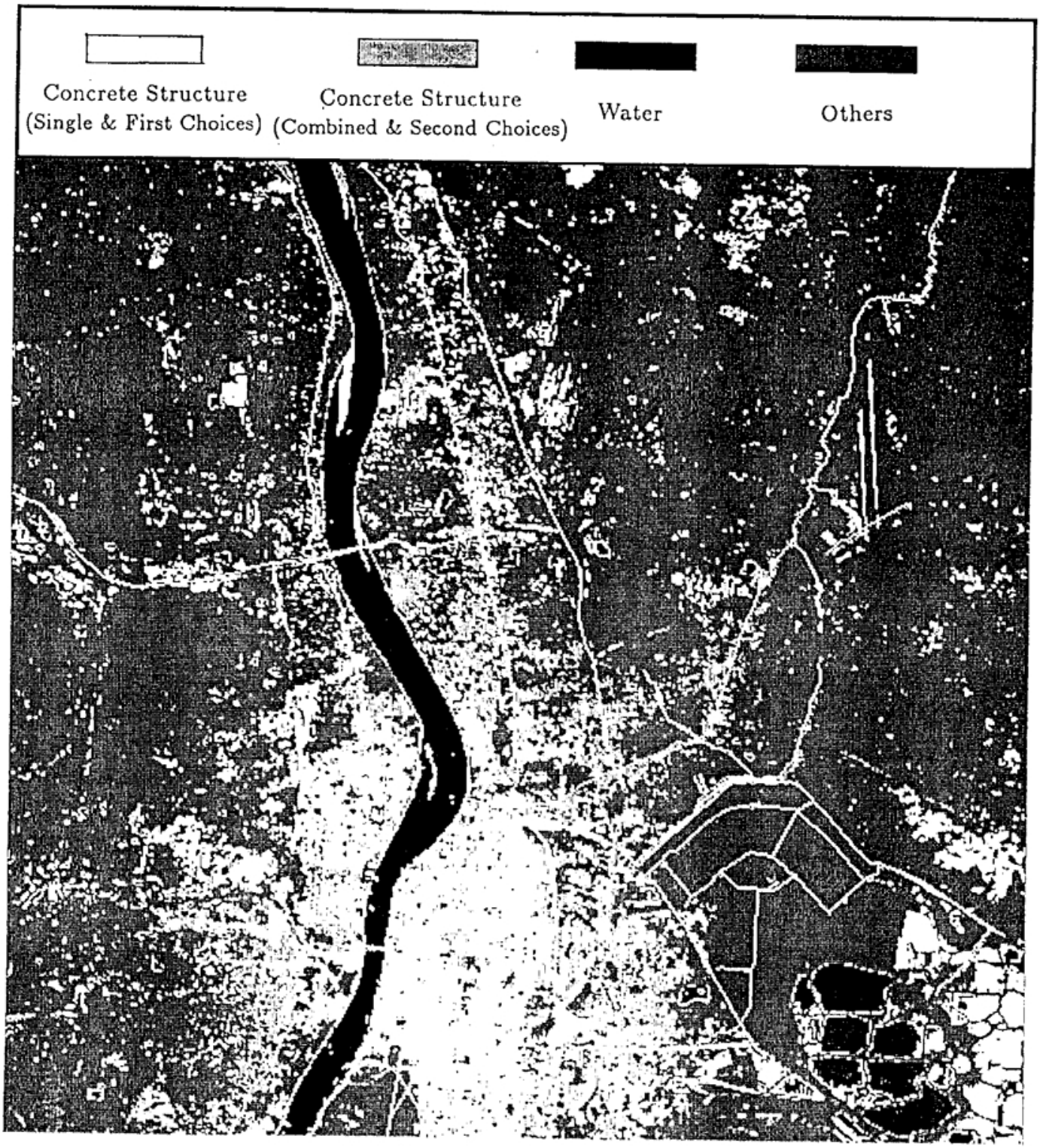


Figure 6.10(b) : Concrete structure in multiple choices of IRS Calcutta image.

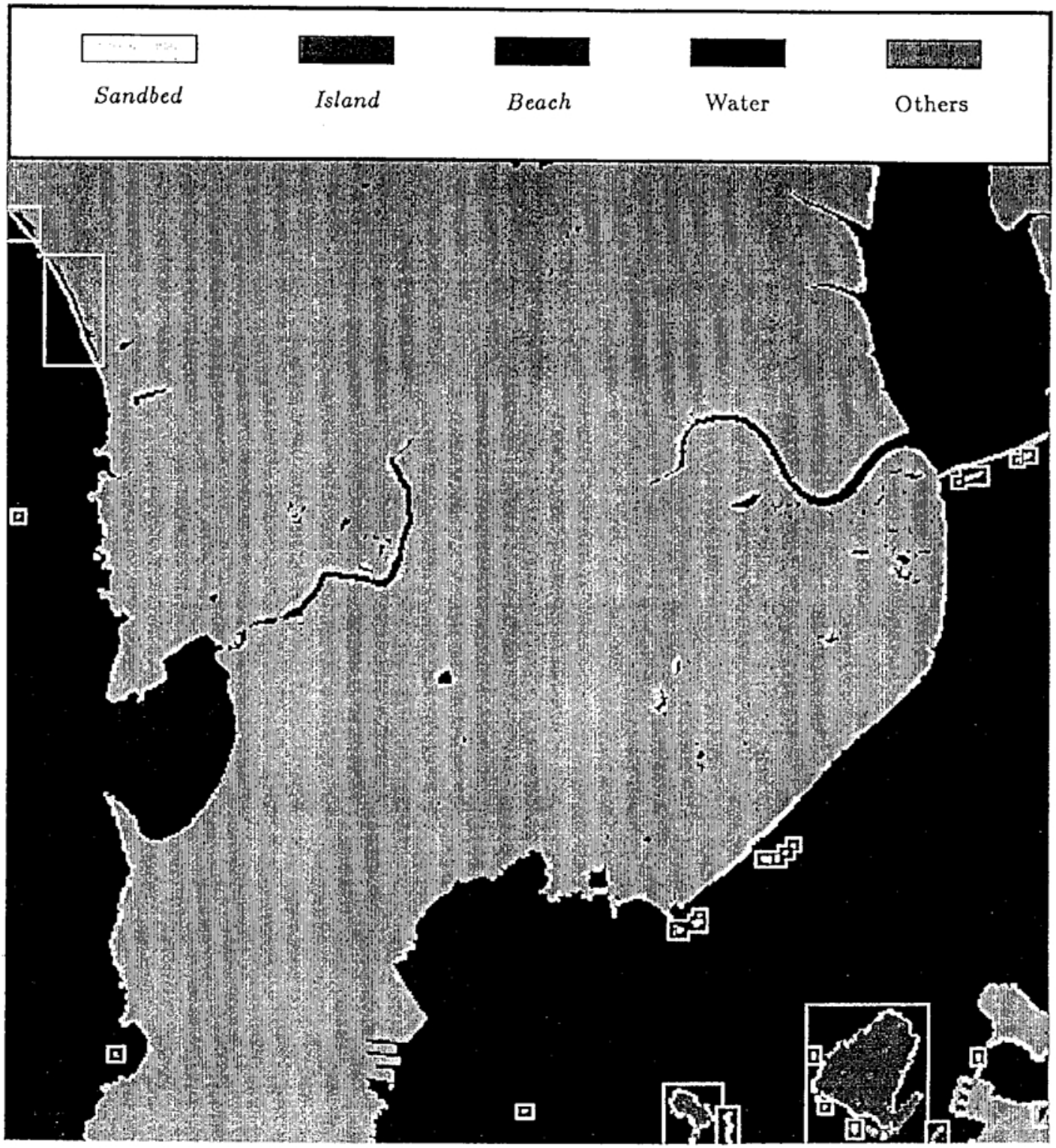


Figure 6.11(a) : *Sandbeds, islands & beaches* in IRS Bombay image.

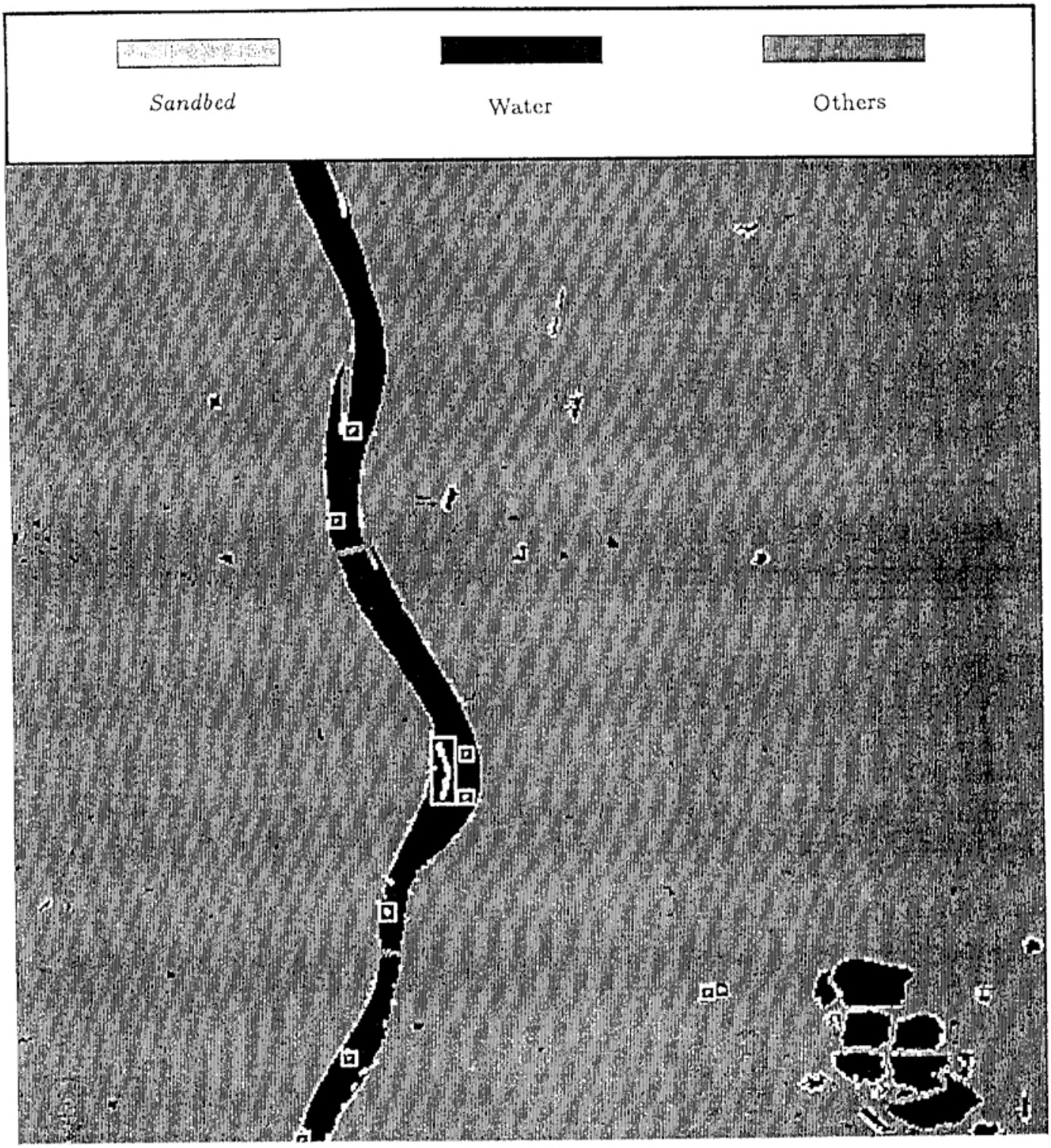


Figure 6.11(b) : *Sandbeds* with water bodies in IRS Calcutta image.

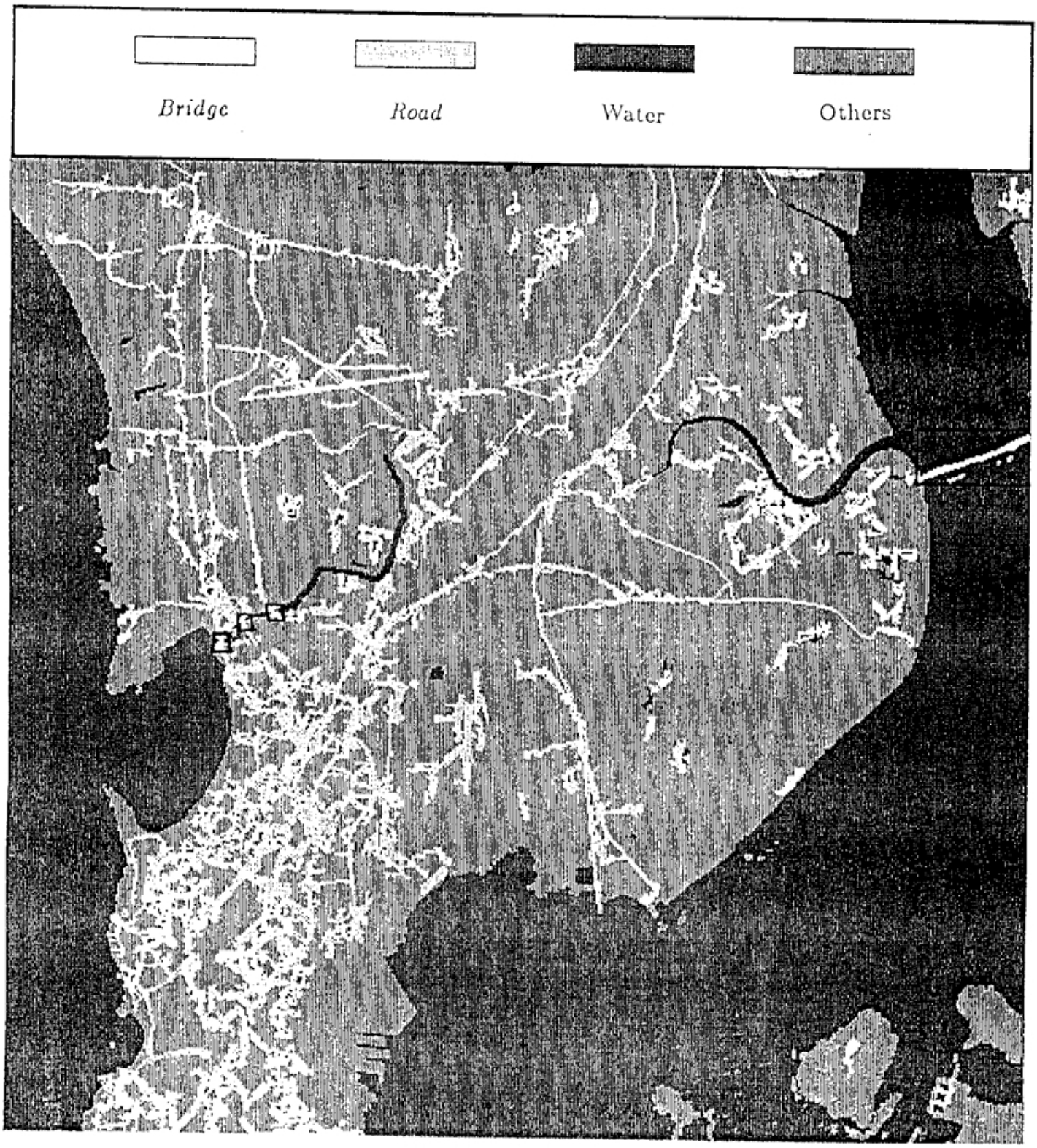


Figure 6.12(a) : *Roads* in IRS Bombay image.

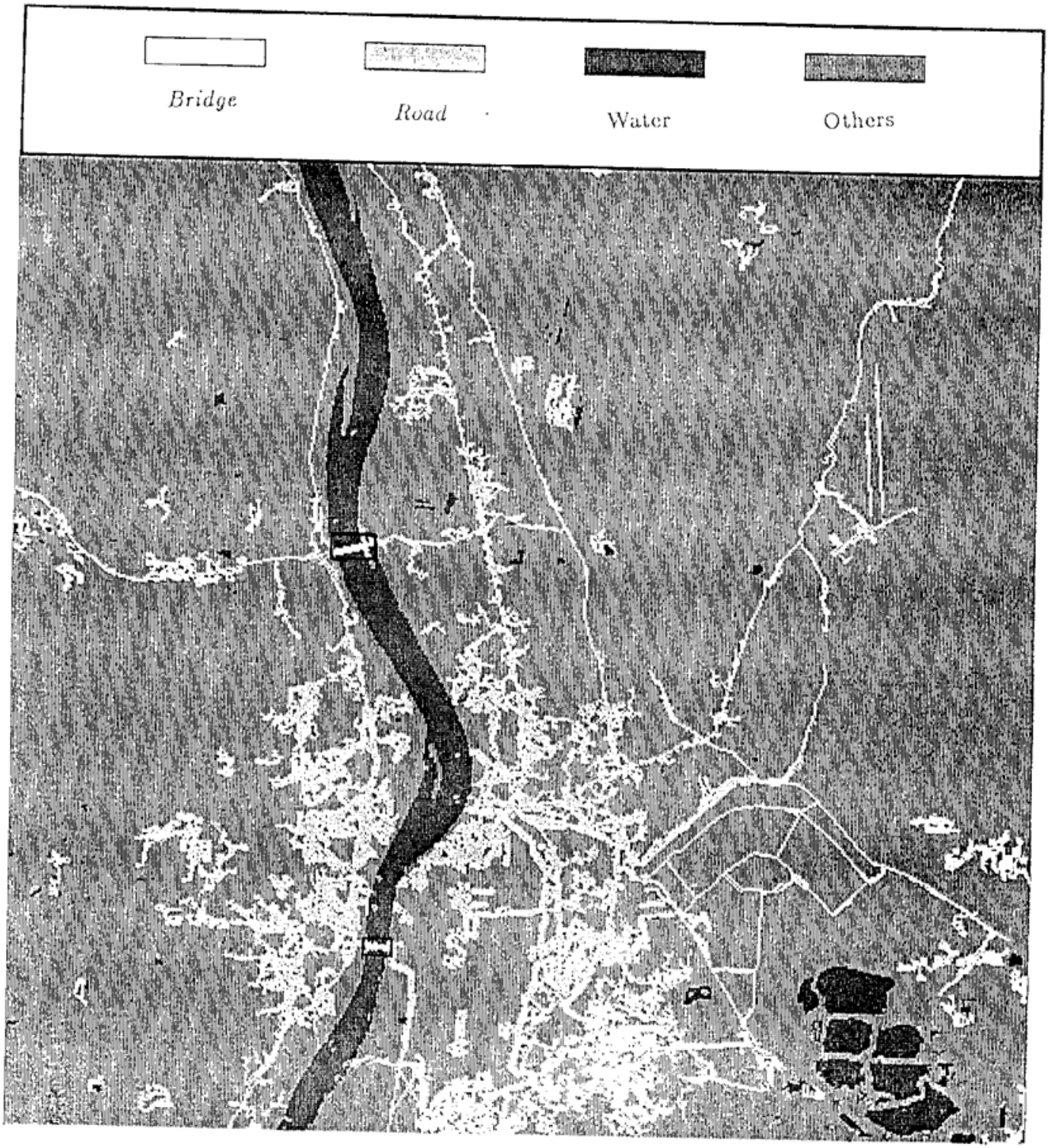


Figure 6.12(b) : *Roads* in IRS Calcutta image.

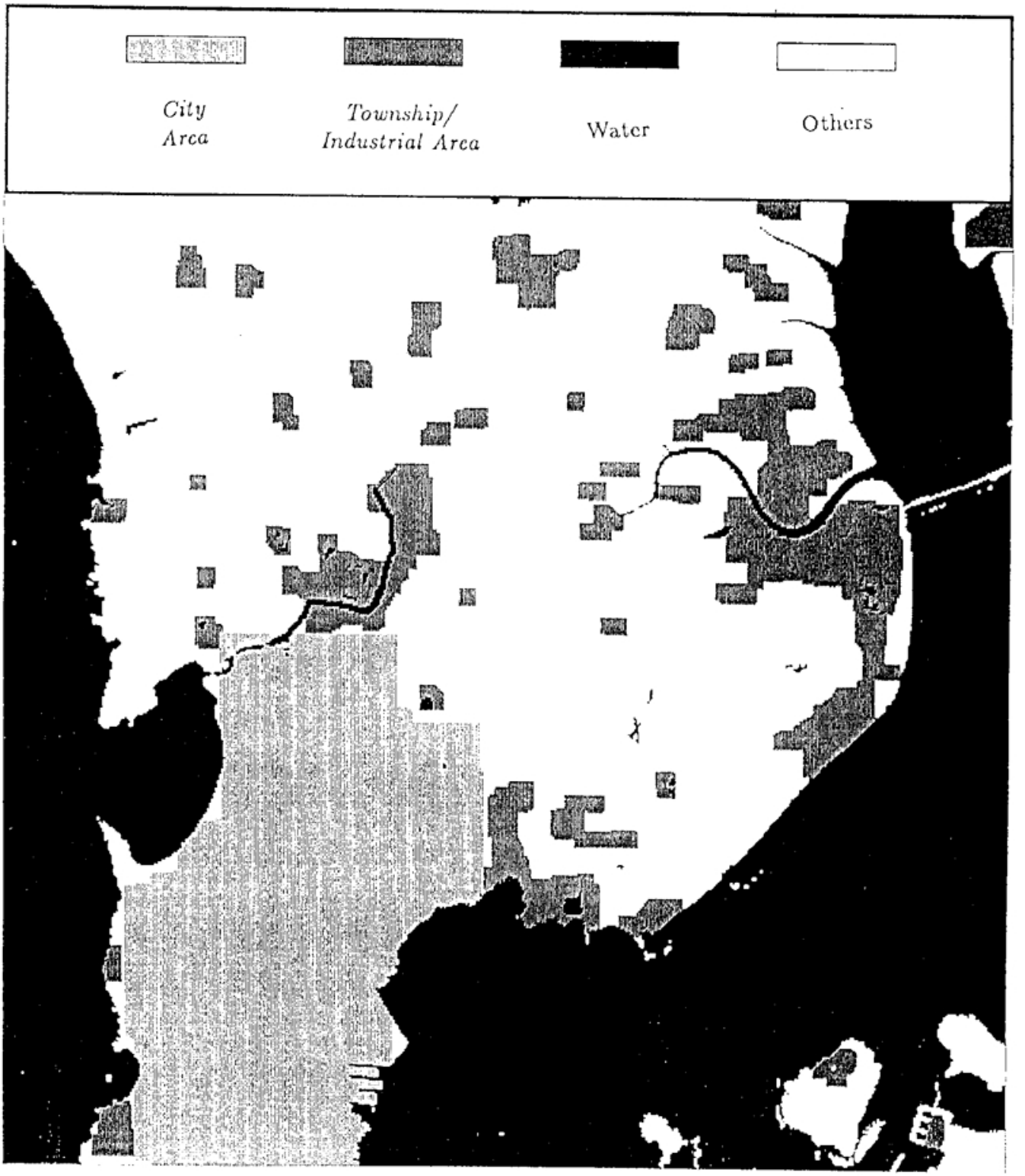


Figure 6.13(a) : *City & township/industrial areas* in IRS Bombay image.

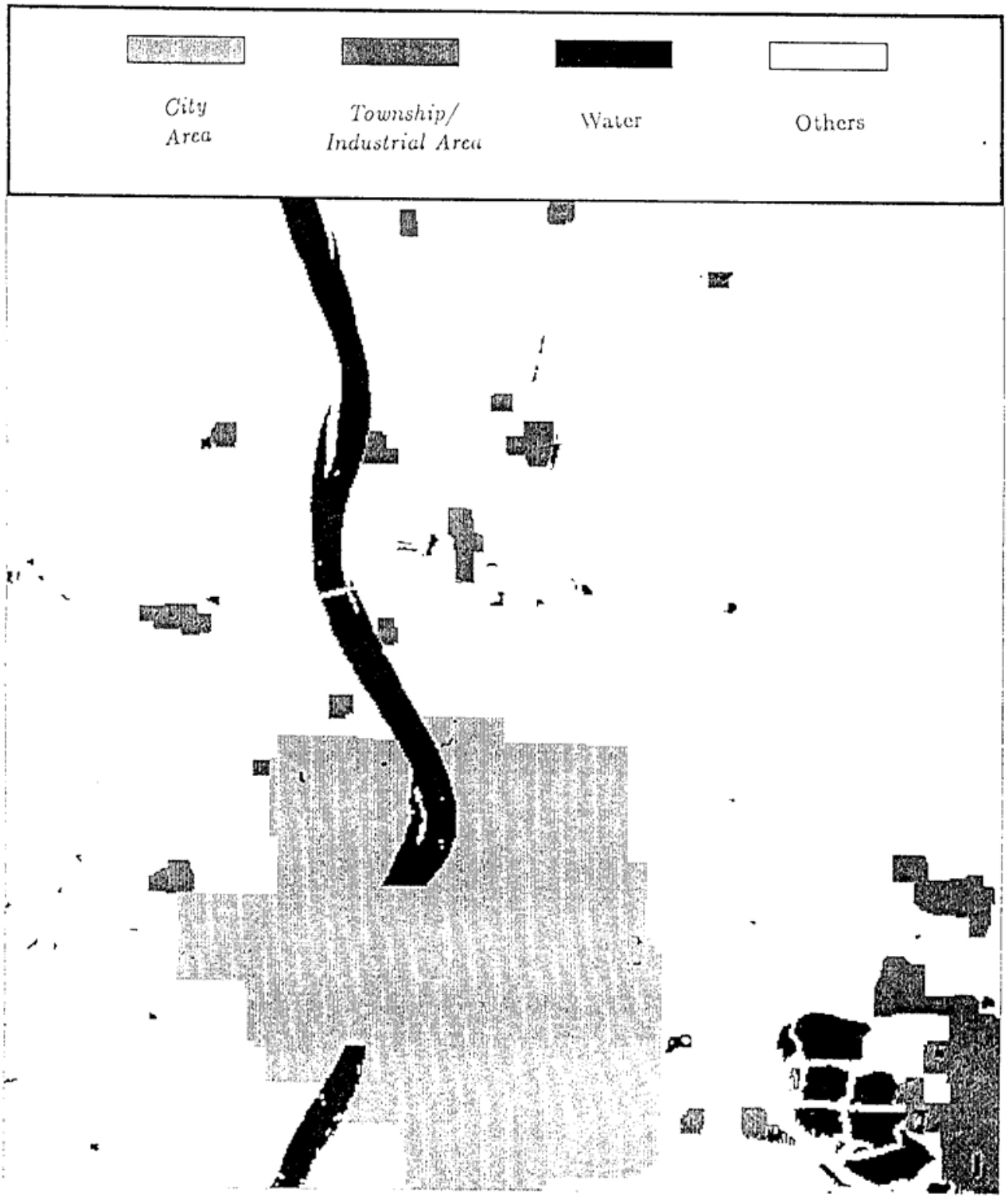


Figure 6.13(b) : *City & township/industrial areas* in IRS Calcutta image.

- All the major *roads* in both Bombay and Calcutta images have been found to be detected [figures 6.12(a)&(b)].

- Out of 4 *bridges* as identified [Fig. 6.12(a)] from the Bombay scene, the largest one corresponds to the Vashi bridge. The others are three small road bridges, one of them is a railway bridge.

- Two bridges detected [Fig. 12(b)] from the Calcutta scene correspond to the Howrah and Vivekananda bridges respectively.

- The *city area* as identified [Fig. 6.13(b)] from the Calcutta frame includes two major cities, namely Calcutta and Howrah, lying opposite sides of the river Ganges and they are connected by Howrah bridge. According to our definition of the *city area* (provided in section 6.2.2), these two are identified under one *city*.

(Note that the experiment showing the percentage of pixels belonging to each one of the land use categories is an elaborate time consuming process and this has not been done here. We did not have also the opportunity to verify the results with any human photointerpreter. We have visually checked the detected regions (general structures and locations) with the corresponding geographical maps.)

The major credit for the exact detection of the aforementioned targets is due to the recognition system providing multiple choices for making decision. It is to be mentioned here that some experiments [257-259] have already been carried out on the Calcutta image frame. The results obtained by the present method are found to be significantly improved over the earlier ones. For example, most of the pixels on the Howrah bridge [Fig. 6.7(a)&(b)] were classified by them [257-259] as turbid water instead of concrete structure, but the multi-valued recognition system (described in chapter 4) have been able to recognize them correctly by the *single* and *first choices*. The classification accuracy of our recognition system is not only found to be better, but also its ability of providing multiple choices in making decisions is found to be very effective in detecting/analyzing various ill-defined targets from IRS images. ♠

Chapter 7

CONCLUSIONS AND SCOPE OF FURTHER RESEARCH

Contents

7.1 Conclusions

We have presented in this thesis some results of investigation, both theoretical and experimental, that demonstrate the effectiveness of the theory of fuzzy sets in formulating some methodologies for handling uncertainties (which may arise from imprecise, incomplete and linguistic input information, and overlapping/ill-defined class boundaries) in certain tasks of pattern recognition. Initially, a procedure for providing multivalued (fuzzy) shape of a pattern class from its sampled points has been described. Then the design aspects of two multivalued recognition systems are presented along with the theoretical analysis of their performances. The effectiveness of the proposed systems has been demonstrated on some artificially generated pattern sets as well as some data sets relating to real life problems such as speech recognition and remote sensing image analysis for identifying ill-defined object regions.

The concept of fuzzy sets has been found to be appropriate in determining the multivalued shape/boundary of a pattern class from its sampled points. In estimating the shape, the portions not covered by the sampled points are assigned some fuzzy membership values denoting the degrees of their belongingness to the actual class. Therefore, unlike the conventional approaches the proposed method does not attempt to provide crisp boundary from incomplete sample set. The effectiveness of the methodology has been demonstrated on some artificially generated data sets (in $I\!R^2$ and $I\!R^3$) and also on the real life speech data. The multivalued shapes can be converted to the usual crisp versions by considering only the feature points with possibility value(θ) ≥ 0.5 to be within the classes. The convergence of the estimated shape to the original one has been verified both experimentally and analytically. In this context, a new metric has been defined which can be used to find the similarity between any two finite sets.

The parameters (like *accuracy factor*, *coverage factors* etc.) required for

determining the shape of a class are determined automatically from the sampled points. One of the major underlying features of the procedure is that any pattern class can be represented as a collection of some nearly parallelepiped shaped subclasses. This concept has been found to be very useful in developing a multivalued recognition system (described in chapter 4).

As mentioned in chapter 2, the complexity of the algorithm for determining the multivalued shape of a pattern class increases exponentially with the dimension (N) of the feature space. Again, it is very difficult to use the Hausdorff metric as well as the newly defined metric *sim* to verify the goodness of the estimated shape of a pattern class for higher dimensions ($N > 3$). It becomes computationally expensive for its implementation in higher dimensional feature spaces, although we could formulate the algorithm for any dimension.

The multivalued recognition systems (described in chapters 3 and 4) have the flexibility of accepting various imprecise input patterns and of providing multi-state output decisions in natural (linguistic) form along with their confidence values. One of them considers three linguistically phrased property sets (namely *small*, *medium* and *high*) at the input so that each feature information can be viewed to have these properties to some degree. As a result, the entire feature space is decomposed into 3^N (N being the number of features) overlapping *space subdomains*, thereby providing more local information of the feature space for making decisions.

The second system, on the other hand, considers decomposition of the entire feature space into some (not predefined or fixed as in the previous case) overlapping *space subdomains* depending on the geometric structure and the relative position of the pattern classes found in the training samples. It provides improved performance. Although the system is trained (unlike the previous case) with deterministic data, it can handle various imprecise input information in the recognizing (testing) phase.

Both the recognition systems have the capability of discriminating the overlapping, non-overlapping and no-class (null choice) regions of the feature space

by providing output decisions in multiple states. The *single choices* correspond to the non-overlapping regions, whereas the overlapping regions are reflected by the *combined* and *first-second choices*. The *null choices* reflect the portions outside the pattern classes and/or the portions of the pattern classes uncovered by the training samples. With the increase in the size of the training samples, the estimates of these regions tend to their actual sizes.

Based on an experiment with artificially generated pattern sets, it has been observed that the Bayes decision boundaries always lie within the *combined choice* region of the recognition system (chapter 4). Among the wrong choices made by the Bayes classifier, a part is seen to be corrected by the *single* and *first choices* of the proposed system. The remaining portion is found to be distributed among the *combined* and *second correct choices*. The multivalued recognition system has, therefore, a provision of improving its efficiency significantly by incorporating the *combined* and *second choices* under the control of a supervisory scheme.

Note that the effectiveness of most of the statistical classifiers (e.g., Bayes classifier) depends on the distribution functions of the pattern classes. If the classes are of regular shape and if the distribution functions can be obtained accurately, the Bayes classifier gives similar performance with the proposed systems, although the output decisions of our systems showing multiple choices are more natural and justified. When the classes are not of regular shape, it is extremely difficult to find their distribution functions. In such cases, the densities may be estimated by some non-parametric methods for the application of Bayes classifier. On the other hand, the distribution functions of the classes need not be known to apply the proposed systems and the systems can be applied to a wide range of situations. The only assumption made is that the training samples, more or less, should represent the classes.

Again, the recognition system (described in chapter 3) has been programmed considering only three primary properties *small*, *medium* and *high*. Incorporation of additional subset properties (e.g., *very small*, *more or less small* etc.) will definitely improve the system performance because it will lead to reduction

in the impreciseness in input information by generating more *subdomains* in the feature space. For example, if we include another property *very small*, say, it will result in 4^N *space subdomains*. Again, the primary properties considered need not necessarily be the same for all the features.

The effectiveness of the multivalued recognition systems has been demonstrated on two real life problems namely, recognition of vowel sounds (in the consonant-vowel-consonant context) and analysis of satellite image for identifying ill-defined object regions. Speech is a pattern of biological origin and the pattern manifests a considerable amount of fuzziness. It has been observed that the confusion in recognizing a sample lies, in general, only with the neighboring classes constituting a vowel triangle. Similar findings were also obtained with previous investigations [26,139,235,236] which considered only deterministic input/output. The overall recognition score is seen to be better than those obtained in [26,139,235,236].

The regions in a satellite imagery are ill-defined because of grayness and spatial ambiguities (including pixel resolution). For analyzing these regions, the multivalued recognition system has initially been applied to classify (based on the *green* and *infrared* band information of the image) the image pixels into six classes corresponding to six land cover types namely, *pond water*, *turbid water*, *concrete structure*, *habitation*, *vegetation* and *open space*. The clustered images are processed further for detecting various object regions namely, *roads*, *bridges*, *islands*, *sandbeds*, *city area*, *township/industrial areas*. In order to detect certain targets, some spatial knowledge about them and their interrelationships have been incorporated in the algorithms using some heuristic rules. The rules have been formulated keeping in view the Indian environment and the pixel resolution of IRS images which is $36.25m \times 36.25m$.

The concept of multiple choices of the recognition system is found to be very effective for detecting the *road* patterns from concrete structure pixels. Because of the low pixel resolution of the remotely sensed images, many portions of the *roads* were not classified as concrete structures. In order to identify them, a traversal algorithm through the detected *road* pixels has been developed where

some of the movements are governed by only the *second* and *combined choices* of the multivalued recognition system. Some morphological operations were used on the clustered image for locating the *city* and *township/industrial areas*. The results are found to agree well with the ground truths when the scenes of Bombay and Calcutta were considered as input.

In case the image frame consists of other classes like *deserts*, *hilly areas*, *snowy regions* etc., one needs only to include some training samples of these classes. To implement the proposed algorithms on other satellite imagery, like LAND-SAT, SPOT, Thematic Mapper (TM) etc., the heuristic rules adopted for the IRS image are to be modified according to their pixel resolutions. The investigation on IRS imagery can find several real life uses such as estimating the damage caused by natural disasters like flood, estimating the depleting forest area, planning the construction of a city from the *roadmap* and for defense purposes. ♠

7.2 Scope of Further Research

A suitable transformation of the feature axes may be sometimes very effective in determining the shape of a pattern class. For example, it will be convenient to determine the shape of a rectangular class if its axes of symmetry are parallel to the co-ordinate axes. So the proposed shape determining procedure can be extended by incorporating a suitable transformation of the feature axes, if necessary for a class. The value of the accuracy factor (δ_t) has been decided based on only the size (t) of the training samples. The distribution of the sampled points in the feature space may also be considered in this regard.

The shapes of the *space subdomains* (in the multivalued recognition systems) are all considered to be parallelepiped. If it is possible to consider various shapes (not restricting only to parallelepiped) relating the geometry of the classes, the performance may be improved in many cases. Again, the problem of developing a supervisory scheme for incorporating the *second* and *combined*

choices should be dealt with for further improvement of the performance. The concept of generalized guard zone algorithm [29] can be considered in this regard. There is also a scope to make the system adaptive. In this context, one can think of the neural network implementation [209,210] of the underlying concepts of the recognition systems. As the methodologies (described in chapters 2, 3 and 4) find difficulties from practical point of view in their implementation in higher dimensional feature spaces, there is a scope of future work to reduce such difficulties.

For IRS image analysis, the gray level distributions are not, in general, same for different image frames and so different sets of training samples are needed for processing them. The variation in gray value distribution is due to many factors like the temperature difference, the seasonal variation, the pollution etc. One may therefore find a model to overcome the effect of these factors in order to use the same set of training samples for different image frames. Moreover, one may attempt in developing a knowledge based system for efficient identification of target regions from satellite imagery.

Finally, the computational complexity of all the proposed methodologies needs to be analyzed. ♠

Bibliography

[1] L. A. Zadeh, "Fuzzy sets," *Information & Control*, vol. 8, no. 3, pp. 338–353, 1965.

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

[3] H. C. Andrews, *Mathematical Techniques in Pattern Recognition*. New York: Wiley Interscience, 1972,

[4] E. A. Patrick, *Fundamentals of Pattern Recognition*, Englewood Cliffs, NJ: Prentice-Hall, 1972.

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

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

[7] P. A. Devijver and J. Kittler, *Pattern Recognition : A Statistical Approach*, London: Prentice-Hall, 1982.

[8] E. S. Gelsema and L. N. Kanal (eds.), *Pattern Recognition in Practice II*, Amsterdam: North Holland, 1986.

[9] T. Pavlidis, *Structural Pattern Recognition*, New York: Springer-Verlag, 1977.

[10] R. C. Gonzalez and M. G. Thomason, *Syntactic Pattern Recognition: An Introduction*, Reading, MA: Addison-Wesley, 1978.

[11] K. S. Fu, *Syntactic Pattern Recognition and Applications*, London: Academic Press, 1982.

[12] L. Kanal, "Patterns in pattern recognition," *IEEE Trans. Information Theory*, vol. IT-20, no. 6, pp. 697–722, 1974.

[13] W. Pedrycz, "Fuzzy sets in pattern recognition : methodology and methods," *Pattern Recognition*, vol. 23, no. 1/2, pp. 121–146, 1990.

[14] S. K. Pal, "Fuzziness, image information and uncertainty management in pattern recognition," *J. Scientific & Industrial Research*, vol. 51, pp. 71–98, 1992.

[15] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms* New York: Plenum Press, 1981.

[16] A. Kandel, *Fuzzy Techniques in Pattern Recognition*, New York: Wiley Interscience, 1982.

[17] A. Kandel, *Fuzzy Mathematical Techniques with Applications*, New York: Addison-Wesley, 1986.

[18] S. K. Pal and D. Dutta Majumder, *Fuzzy Mathematical Approach to Pattern Recognition*, New York: John Wiley & Sons, 1986.

[19] J. C. Bezdek and S. K. Pal (eds.), *Fuzzy Models for Pattern Recognition: Methods that Search for Structures in Data*, New York: IEEE Press, 1992.

[20] C. H. Chen, "On information and distance measures, error bounds and feature selection," *Information Sciences*, vol. 10, pp. 159–173, 1976.

[21] C. B. Chittineni, "Efficient feature subset selection with probabilistic distance criteria," *Information Sciences*, vol. 22, pp. 19–35, 1980.

[22] J. C. Bezdek and P. F. Castelaz, "Prototype classification and feature selection with fuzzy sets," *IEEE Trans. Syst., Man & Cybern.*, vol. SMC-7, no. 2, pp. 87–92, 1977.

[23] V. Di Gesu and M. C. Maccarone, "Feature selection and 'possibility theory'," *Pattern Recognition*, vol. 19, pp. 63–72, 1986.

[24] K. S. Fu, *Sequential Methods in Pattern Recognition and Machine Learning*, London: Academic Press, 1968.

[25] J. M. Mendel and K. S. Fu (eds.), *Adaptive Learning and Pattern Recognition Systems - Theory and Applications*, New York: Academic Press, 1970.

[26] S. K. Pal, "Optimum guard zone for self supervised learning," *IEE Proc.-E*, vol. 129, no. 1, pp. 9–14, 1982.

[27] A. G. Barton and P. Anandam, "Pattern recognizing stochastic learning automata," *IEEE Trans. Syst., Man & Cybern.*, vol. SMC-15, pp. 360–375, 1985.

[28] B. B. Devi and V. V. S. Sarma, "A fuzzy approximation scheme for sequential learning in pattern recognition," *IEEE Trans. Syst., Man & Cybern.*, vol. SMC-16, no. 5, pp. 668–679, 1986.

[29] A. Pal (Pathak) and S. K. Pal, "Generalized guard zone algorithm (GGA) for learning: automatic selection of threshold," *Pattern Recognition*, vol. 23, no. 3/4, pp. 325–335, 1990.

[30] A. Pal (Pathak), *On a Class of Stochastic Approximation-type Parameter-learning Algorithms for Pattern Recognition*, PhD thesis, Indian Statistical Institute, Calcutta, 1990.

[31] A. Pal (Pathak) and S. K. Pal, "Effect of wrong samples on the convergence of learning process-II: a remedy," *Information Sciences*, vol. 60, no. 1/2, pp. 77–105, 1992.

[32] R. C. Tryon, *Cluster Analysis*. Ann Arbor: Edwards Bros., 1939.

[33] E. Ruspini, "A new approach to clustering," *Information & Control*, vol. 15, no. 1, pp. 22–32, 1969.

[34] I. Gitman and M. D. Levine, "An algorithm for detection unimodal fuzzy sets and its application as a clustering technique," *IEEE Trans. Computer*, vol. C-19, no. 7, pp. 583–593, 1970.

[35] H. R. Anderberg, *Cluster Analysis for Applications*. New York: Academic Press, 1973.

[36] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *J. Cybern.*, vol. 3, no. 3, pp. 32–57, 1973.

[37] J. C. Bezdek, "Cluster validity with fuzzy sets," *J. Cybern.*, vol. 3, no. 3, pp. 58–72, 1974.

[38] E. Backer, *Cluster Analysis by Optimal Decomposition of Induced Fuzzy Sets*, PhD thesis, Delft University, Delft, The Netherlands, 1978.

[39] J. M. Keller, M. R. Gray, and J. A. Givens, "A fuzzy k-nearest neighbor algorithm," *IEEE Trans. Syst., Man & Cybern.*, vol. SMC-15, no. 4, pp. 580–585, 1985.

[40] T. W. Anderson, *An Introduction to Multivariate Statistical Analysis*, New York: Wiley, 1958.

[41] D. E. Rumelhart, J. McClelland, *et al.*, *Parallel Distributed Processing : Explorations in the Microstructure of Cognition*, Vol. 1, Cambridge, MA: MIT Press, 1986.

[42] Y. II. Pao, *Adaptive Pattern Recognition and Neural Networks*, New York: Addison-Wesley, 1989.

[43] *Proc. First Int. Conf. on Fuzzy Logic & Neural Networks.* Iizuka, Japan, 1990.

[44] *Proc. Second Int. Conf. on Fuzzy Logic & Neural Networks.* Iizuka, Japan, 1992.

[45] M. M. Gupta, "Fuzzy logic and neural networks," in *Proc. 2nd Int. Conf. on Fuzzy Logic & Neural Networks*, (Iizuka, Japan), pp. 157–160, 1992.

[46] *IEEE Trans. on Neural Networks.* (Spl. Issue on Fuzzy Systems), vol. NN-3, no. 5, 1992.

[47] R. P. Lippmann, "An introduction to computing with neural nets," *IEEE ASSP Magazine*, pp. 4–22, April 1987.

[48] B. Kosko, *Neural Networks and Fuzzy Systems: A Dynamical Approach to Machine Learning.* Englewood Cliffs, NJ: Prentice-Hall, 1991.

[49] *DARPA Neural Network Study.* Fairfax, VA: AFCEA Press, 1988.

[50] M. A. Arbib, *Brains, Machines and Mathematics*, 2nd ed., Berlin: Springer-Verlag, 1987.

[51] G. A. Carpenter and S. Grossberg, "ART2 : self-organization of stable category recognition codes for analog input patterns," *Applied Optics*, vol. 26, no. 23, pp. 4919–4930, 1987.

[52] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, New York: Academic Press, 1982.

[53] J. Serra, *Image Analysis and Mathematical Morphology*, London: Academic Press, 1982.

[54] R. C. Gonzalez and P. Wintz, *Digital Image Processing*, Reading, MA: Addison-Wesley, 1987.

[55] R. M. Harralick and G. Shapiro, *Computer and Robot Vision*, New York: Addison-Wesely, 1991.

[56] H. Edelsbrunner, D. G. Kirkpatick, and R. Seidel, "On the shape of a set of points in a plane," *IEEE Trans. Information Theory*, vol. IT-29, pp. 551–559, 1983.

[57] R. A. Jarvis, "Computing the shape hull of points in the plane," in *Proc. IEEE Comp. Soc. Conf. on Patt. Recog. and Image Processing*, pp. 231–241, 1977.

[58] S. K. Akl and G. T. Toussaint, "Efficient convex hull algorithm for pattern recognition applications," in *Proc. 4th Int. Jt. Conf. on Patt. Recog.*, Kyoto, pp. 483-487, 1978.

[59] J. Fairfield, "Contoured shape generation forms that people see in dot patterns," in *Proc. IEEE Conf. on Cybern. and Soc.*, pp. 60–64, 1979.

[60] G. T. Toussaint, "Pattern recognition and geometrical complexity," in *Proc. 5th Int. Conf. on Patt. Recog.*, (Miami Beach, Florida), pp. 1324–1347, 1980.

[61] C. A. Murthy, *On Consistent Estimation of Classes in $\mathbb{R}^2$ in the Context of Cluster Analysis*, PhD thesis, Indian Statistical Institute, Calcutta, India, 1988.

[62] A. Kaufmann, *Introduction to the Theory of Fuzzy Subsets - Fundamental Elements*, New York: Academic Press, 1975.

[63] D. Dubois and H. Prade, *Fuzzy Sets and Systems: Theory and Applications*, New York: Academic Press, 1980.

[64] P. P. Wang and S. K. Chang (eds.), *Fuzzy Sets: Theory and Applications to Policy Analysis and Information Systems*, New York: Plenum Press, 1980.

[65] G. J. Klir and T. Folger, *Fuzzy Sets, Uncertainty and Information*, NJ: Prentice-Hall, 1988.

[66] H. Zimmermann, *Fuzzy Set Theory and its Applications*, 2nd ed., Boston: Kluwer, 1990.

[67] L. A. Zadeh, "An outline of a new approach to the analysis of complex systems and decision processes," *IEEE Trans. Syst., Man & Cybern.*, vol. SMC-3, no. 1, pp. 28–44, 1973.

[68] L. A. Zadeh, K. S. Fu, K. Tanaka, and M. Shimura (eds.), *Fuzzy Sets and Their Application to Cognitive and Decision Process*, London: Academic Press, 1975.

[69] R. R. Yager and L. A. Zadeh (eds.), *An Introduction to Fuzzy Logic Applications in Intelligent Systems*, Boston: Kluwer, 1992.

[70] L. A. Zadeh, "The role of fuzzy logic in the management of uncertainty in expert system," *Fuzzy Sets & Systems*, vol. 11, no. 3, pp. 199–223, 1983.

[71] L. A. Zadeh, "Fuzzy logic," *IEEE Computer*, pp. 83–93, April, 1988.

[72] L. A. Zadeh, "Fuzzy logic and approximate reasoning," *Synthese*, vol. 30, pp. 407–428, 1977.

[73] L. A. Zadeh, "The concept of linguistic variable and its applications to approximate reasoning-I," *Information Sciences*, vol. 8, pp. 199–249, 1975.

[74] L. A. Zadeh, "The concept of linguistic variable and its applications to approximate reasoning-II," *Information Sciences*, vol. 8, pp. 301–357, 1975.

[75] L. A. Zadeh, "The concept of linguistic variable and its applications to approximate reasoning-III," *Information Sciences*, vol. 9, pp. 43–80, 1975.

[76] R. R. Yager, "Validation of fuzzy linguistic models," *J. cybern.*, vol. 8, pp. 17–30, 1978.

[77] J. F. Baldwin, "A new approach to approximate reasoning using a fuzzy logic," *Fuzzy Sets & Systems*, vol. 2, no. 4, pp. 309–325, 1979.

[78] R. R. Yager, "Approximate reasoning and possibility model in classification," *Int. J. Computer & Information Science*, vol. 10, pp. 141–175, 1981.

[79] M. M. Gupta, R. K. Ragade, and R. R. Yager (eds.), *Advances in Fuzzy Set Theory and Applications*, New York: North Holland, 1979.

[80] M. M. Gupta, A. Kandel, W. Bandler, and J. B. Kiszka (eds.), *Approximate Reasoning in Expert Systems*, New York: North Holland, 1985.

[81] S. K. Pal and D. P. Mandal, "Fuzzy logic and approximate reasoning: an overview," *J. Inst. Elec. & Telecom. Engrs.*, vol. 37, no. 5/6, pp. 548–560, 1991.

[82] L. A. Zadeh, "Probability measures of fuzzy events," *J. Math. Anal. & Appl.*, vol. 23, pp. 421–427, 1968.

[83] A. Deluca and S. Termini, "A definition of a nonprobabilistic entropy in the setting of fuzzy set theory," *Information & Control*, vol. 20, no. 4, pp. 301–312, 1972.

[84] B. R. Gaines, "Fuzzy and probability uncertainty logics," *Information & Control*, vol. 38, no. 2, pp. 154–169, 1978.

[85] K. Hirota, "Concepts of probabilistic sets," *Fuzzy Sets & Systems*, vol. 5, no. 5, pp. 31–46, 1981.

[86] E. Hisdal, "Are grades of membership probabilities?," *Fuzzy Sets & Systems*, vol. 25, no. 3, pp. 325–348, 1988.

[87] B. Kosko, "Fuzziness versus probability," *Int. J. General Systems*, vol. 17, no. 2/3, pp. 211–240, 1990.

[88] I. B. Turksen, "Measurement of membership function and their acquisition," *Fuzzy Sets & Systems*, vol. 40, no. 1, pp. 5–38, 1991.

[89] C. A. Murthy and S. K. Pal, "Histogram thresholding by minimizing gray level fuzziness," *Information Sciences*, vol. 60, no. 1/2, pp. 107–135, 1992.

[90] S. K. Pal and A. Dasgupta, "Spectral fuzzy sets and soft thresholding," *Information Sciences*, 1992 (in press).

[91] M. Mizumoto and K. Tanaka, "Some properties of fuzzy sets of type 2," *Information & Control*, vol. 31, no. 4, pp. 312–340, 1976.

[92] L. A. Zadeh, "Making computer think like people," *IEEE Spectrum*, pp. 26–32, August, 1984.

[93] I. B. Turksen, "Interval valued fuzzy sets based on normal forms," *Fuzzy Sets & Systems*, vol. 20, pp. 191–210, 1986.

[94] C. A. Murthy, S. K. Pal, and D. D. Majumder, "Correlation between two fuzzy membership functions," *Fuzzy Sets & Systems*, vol. 7, no. 1, pp. 23–38, 1985.

[95] C. A. Murthy and S. K. Pal, "Fuzzy thresholding: Mathematical framework, bound functions and weighted moving average," *Pattern Recognition Letters*, vol. 11, pp. 197–206, 1990.

[96] A. Kandel and W. J. Byatt, "Fuzzy sets, fuzzy algebra and fuzzy statistics," *Proc. IEEE*, vol. 66, no. 12, pp. 1619–1639, 1978.

[97] L. A. Zadeh, "Fuzzy sets as a basis for a theory of possibility," *Fuzzy Sets & Systems*, vol. 1, no. 1, pp. 3–28, 1978.

[98] R. R. Yager, "A foundation for a theory of possibility," *J. Cybern.*, vol. 10, pp. 177–204, 1980.

[99] R. R. Yager, *Fuzzy Set and Possibility Theory*, New York: Pergamon Press, 1982.

[100] R. R. Yager, "Entropy and specificity in a mathematical theory of evidence," *Int. J. General Systems*, vol. 9, pp. 249–260, 1983.

[101] M. Schneider and A. Kandel, "Properties of the fuzzy expected value and the fuzzy expected interval," *Fuzzy Sets & Systems*, vol. 26, no. 3, pp. 373–385, 1988.

[102] M. Friedman, M. Schneider, and A. Kandel, "The use of weighted fuzzy expected value (WFEV) in fuzzy expert systems," *Fuzzy Sets & Systems*, vol. 31, no. 1, pp. 37–45, 1989.

[103] S. G. Loo, "Measures of fuzziness," *Cybernetics*, vol. 10, pp. 201–210, 1977.

[104] W. X. Xie and S. D. Bedrosian, "An information measure for fuzzy sets," *IEEE Trans. Syst., Man & Cybern.*, vol. SMC-14, no. 1, pp. 151–156, 1984.

[105] B. Kosko, "Fuzzy entropy and conditioning," *Information Sciences*, vol. 40, no. 2, pp. 165–174, 1986.

[106] S. K. Pal, "Fuzzy tools for the management of uncertainty in pattern recognition, vision and expert systems," *Int. J. System Sciences*, vol. 22, no. 3, pp. 511–549, 1991.

[107] N. R. Pal and S. K. Pal, "Higher order fuzzy entropy and hybrid entropy of a set," *Information Sciences*, vol. 61, no. 3, pp. 211–231, 1992.

[108] *Fuzzy Sets & Systems.* (Spl. Memorial Issue : Fuzzy Logic and Uncertainty Modelling), vol. 40, no. 3, 1991.

[109] G. A. Shafer, *Mathematical Theory of Evidence*, Princeton, NJ: Princeton University Press, 1976.

[110] D. Dubois and H. Prade, "Properties of measures of information in evidence and possibility theories," *Fuzzy Sets & Systems*, vol. 24, pp. 161–182, 1987.

[111] U. Hohle, "Entropy with respect to plausibility measures," in *Proc. 12th Int. Symp. on Multiple Valued Logic*, (Paris), pp. 167–169, 1982.

[112] P. Smets, "Information content of an evidence," *Int. J. Man-Mach. Stud.*, vol. 19, pp. 33–43, 1983.

[113] M. Higashi and G. L. Klir, "Measures of uncertainty and information based on possibility distribution," *Int. J. General Systems*, vol. 9, pp. 43–58, 1983.

[114] D. Dubois and H. Prade, "A note on measures of specificity for fuzzy sets," *Int. J. General Systems*, vol. 10, pp. 279–283, 1985.

[115] G. J. Klir and A. Ramer, "Uncertainty in Dempster-Shafer theory: a critical re-examination," *Int. J. General Systems*, vol. 18, pp. 155–166, 1990.

[116] N. R. Pal, J. C. Bezdek, and R. Hemasinha, "Uncertainty measures for evidential reasoning I: a review," *Int. J. Approximate Reasoning*, vol. 7, no. 3/4, pp. 165–183, 1992.

[117] D. Bhandari and N. R. Pal, "Some new measures for fuzzy sets," *Information Sciences*, 1992 (in press).

[118] Z. Pawlak, "Rough sets," *Int. J. Inform. Comp. Sci.*, vol. 11, pp. 341–356, 1982.

[119] Z. Pawlak, "Rough sets and fuzzy sets," *Fuzzy Sets & Systems*, vol. 17, pp. 99–102, 1985.

[120] J. W. Grzymala-Busse, "Knowledge acquisition under uncertainty – a rough set approach," *J. Intell. & Robotic Systs.*, vol. 1, no. 1, pp. 3–16, 1988.

[121] J. C. Bezdek, "A physical interpretation of fuzzy ISODATA," *IEEE Trans. Syst., Man & Cybern.*, vol. SMC-6, no. 5, pp. 387–389, 1976.

[122] M. Roubens, "Pattern classification problems and fuzzy sets," *Fuzzy Sets & Systems*, vol. 1, no. 4, pp. 239–253, 1978.

[123] D. E. Gustafson and W. C. Kessel, "Fuzzy clustering with a fuzzy covariance matrix," in *Proc. IEEE CDC*, (San Diego, CA), pp. 761–766, 1979.

[124] M. P. Windham, "Geometric fuzzy clustering algorithms," *Fuzzy Sets & Systems*, vol. 10, no. 3, pp. 271–279, 1983.

[125] S. K. Pal and S. Mitra, "Fuzzy dynamic clustering algorithm," *Pattern Recognition Letters*, vol. 11, no. 8, pp. 525–535, 1990.

[126] J. Bezdek and J. C. Dunn, "Optimal fuzzy partitions: A heuristic for estimating the parameters in a mixture of normal distributions," *IEEE Trans. Computer*, vol. C-24, no. 8, pp. 835–838, 1975.

[127] M. J. Sabin, "Convergence and consistency of fuzzy c-means/ISODATA algorithms," *IEEE Trans. Patt. Anal. & Mach. Intell.*, vol. PAMI-9, no. 5, pp. 661–668, 1987.

[128] J. C. Bezdek, "A convergence theorem for the fuzzy ISODATA clustering algorithms," *IEEE Trans. Patt. Anal. & Mach. Intell.*, vol. PAMI-2, no. 1, pp. 1–8, 1980.

[129] J. C. Bezdek, R. J. Hathaway, M. J. Sabin, and W. T. Tucker, "Convergence theory for fuzzy c-means: counterexamples and repairs," *IEEE Trans. Syst., Man & Cybern.*, vol. SMC-17, no. 5, pp. 873–877, 1987.

[130] E. Backer and A. K. Jain, "A clustering performance measure based on fuzzy set decomposition," *IEEE Trans. Patt. Anal. & Mach. Intell.*, vol. PAMI-3, no. 1, pp. 66–75, 1981.

[131] M. Windham, "Cluster validity for fuzzy clustering algorithms," *Fuzzy Sets & Systems*, vol. 5, no. 4, pp. 177–185, 1981.

[132] M. Roubens, "Fuzzy clustering algorithms and their cluster validity," *Eur. J. Op. Res.*, vol. 10, pp. 294–301, 1982.

[133] I. Gath and A. B. Geva, "Unsupervised optimal fuzzy clustering," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. PAMI-11, no. 7, pp. 773–781, 1989.

[134] X. L. Xie and G. Beni, "A validity measure for fuzzy clustering," *IEEE Trans. Patt. Anal. & Mach. Intell.*, vol. PAMI-13, no. 8, pp. 841–847, 1991.

[135] R. Krishnapuram and J. Lee, "Fuzzy connective based hierarchical aggregation networks for decision making," *Fuzzy Sets and Systems*, vol. 46, pp. 11–27, 1992.

[136] R. Dave, "Fuzzy shell-clustering and applications to circle detection in digital images," *Int. J. General Systems*, vol. 16, no. 4, pp. 343–355, 1990.

[137] R. Bellman, R. Kalaba, and L. Zadeh, "Abstraction and pattern classification," *J. Math. Anal. Appl.*, vol. 13, pp. 1–7, 1966.

[138] L. A. Zadeh, "Fuzzy sets and their applications to pattern classification and cluster analysis," Tech. Rep. Memo UCB/ERL. M-607, University of California, Berkeley, 1976.

[139] S. K. Pal and D. Dutta Majumder, "Fuzzy sets and decision making approaches in vowel and speaker recognition," *IEEE Trans. Syst., Man & Cybern.*, vol. SMC-7, pp. 625–629, 1977.

[140] S. K. Pal and D. Dutta Majumder, "On automatic plosive identification using fuzziness in property sets," *IEEE Trans. Syst., Man & Cybern.*, vol. SMC-8, no. 4, pp. 302–308, 1978.

[141] S. K. Pal, A. K. Datta, and D. Dutta Majumder, "A self-supervised vowel recognition system," *Pattern Recognition*, vol. 12, no. 1, pp. 27–34, 1980.

[142] A. Pathak and S. K. Pal, "On the convergence of a self-supervised vowel recognition system," *Pattern Recognition*, vol. 20, no. 2, pp. 237–244, 1987.

[143] A. K. Nath and T. T. Lee, "On the design of a classifier with linguistic variables as input," *Fuzzy Sets & Systems*, vol. 11, pp. 265–286, 1983.

[144] A. K. Nath, S. W. Liu, and T. T. Lee, "On some properties of linguistic classifier," *Fuzzy Sets & Systems*, vol. 17, pp. 297–311, 1985.

[145] R. L. P. Chang and T. Pavlidis, "Fuzzy decision tree algorithms," *IEEE Trans. Syst., Man & Cybern.*, vol. SMC-7, no. 1, pp. 28–35, 1977.

[146] S. K. Pal and B. Chakraborty, "Fuzzy set theoretic measure for automatic feature evaluation," *IEEE Trans. Syst., Man & Cybern.*, vol. SMC-16, no. 5, pp. 754–760, 1986.

[147] J. C. Bezdek, S. Chuah, and D. Leep, "Generalized k-nearest neighbor rules," *Fuzzy Sets & Systems*, vol. 18, no. 3, pp. 237–256, 1986.

[148] E. T. Lee, "Fuzzy tree automata and syntactic pattern recognition," *IEEE Trans. Patt. Anal. & Mach. Intell.*, vol. PAMI-4, no. 4, pp. 445–449, 1982.

[149] M. G. Thomason, "Finite fuzzy automata, regular fuzzy language and pattern recognition," *Pattern Recognition*, vol. 5, pp. 383–390, 1973.

[150] G. F. DePalma and S. S. Yah, "Fractionally fuzzy grammars with applications to pattern recognition," in *Fuzzy Sets and Their Applications to Cognitive and Decision Processes*, (L. A. Zadeh, K. S. Fu, K. Tanaka, and M. Shimura, eds.), pp. 329–351, London: Academic Press, 1975.

[151] K. Peeva, "Fuzzy acceptors for syntactic pattern recognition," *Int. J. Approximate Reasoning*, vol. 5, no. 3, pp. 291–306, 1991.

[152] A. Pathak and S. K. Pal, "Fuzzy grammars in systactic recognition of skeletal from x-rays," *IEEE Trans. Syst., Man & Cybern.*, vol. SMC-16, no. 5, pp. 657–667, 1986.

[153] A. K. Majumder, A. K. Ray, and B. Chatterjee, "Inference of fuzzy regular language using formal power series representation," in *Proc. ISI Int. Conf. on Advances in Information Science and Technology*, (Indian Statistical Institute, Calcutta), pp. 155–165, 1982.

[154] R. Di Mori and P. Laface, "Use of fuzzy algorithms for phonetic and phonemic labeling of continuous speech," *IEEE Trans. Patt. Anal & Mach. Intell.*, vol. PAMI-2, no. 2, pp. 136–148, 1980.

[155] R. Di Mori, *Computerized Models of Speech Using Fuzzy Algorithms*, New York: Plenum Press, 1983.

[156] A. Pathak, S. K. Pal, and R. A. King, "Syntactic recognition of skeletal maturity," *Pattern Recognition Letters*, vol. 2, pp. 193–197, 1984.

[157] P. Siy and C. S. Chen, "Fuzzy logic for handwritten numeral character recognition," *IEEE Trans. Syst., Man & Cybern.*, vol. SMC-4, pp. 570–575, 1974.

[158] W. J. M. Kickert and H. Koppelaar, "Application of fuzzy set theory to syntactic pattern recognition of handwritten capitals," *IEEE Trans. Syst., Man & Cybern.*, vol. SMC-6, pp. 148–151, 1976.

[159] S. K. Pal, D. Dutta Majumder, and B. B. Chaudhuri, "Fuzzy sets in handwritten character recognition," in *Proc. All India Interdisciplinary Symp. on Digital Technology and Pattern Recognition*, (Indian Statistical Institute, Calcutta), pp. 63–71, 1977.

[160] P. Biswas and A. K. Majumder, "A multistage fuzzy classifier for recognition of hand printed characters," *IEEE Trans. Syst., Man & Cybern.*, vol. SMC-11, no. 12, pp. 834–838, 1981.

[161] A. Meisels, A. Kandel, and G. Gecht, "Entropy, and the recognition of fuzzy letters," *Fuzzy Sets & Systems*, vol. 31, pp. 297–309, 1989.

[162] J. R. Key, J. A. Maslanik, and R. G. Barry, "Cloud classification from satellite data using a fuzzy-sets algorithm - a polar example," *Int. J. Remote Sensing*, vol. 10, pp. 1823–1842, 1989.

[163] E. T. Lee, "Shape-oriented chromosome classification," *IEEE Trans, Syst., Man & Cybern.*, vol. SMC-5, pp. 629–632, 1975.

[164] A. Kumar, "A real-time system for pattern recognition of human sleep stages by fuzzy systems analysis," *Pattern Recognition*, vol. 9, no. 1, pp. 43–46, 1977.

[165] A. Kandel, "Fuzzy statistics and forecast evaluation," *IEEE Trans. Syst., Man & Cybern.*, vol. SMC-8, no. 5, pp. 396–401, 1978.

[166] A. Nafarieh and J. Keller, "A fuzzy logic rule-based automatic target recognition," *Int. J. Intell. Systs.*, vol. 6, pp. 295–312, 1991.

[167] A. de Korvin, R. Kleyle, and R. Lea, "The object recognition problem when features fail to be homogeneous," 1992 (communicated).

[168] A. de Korvin, R. Kleyle, and R. Lea, "An evidential approach to problem solving when a large number of knowledge systems is available," *Int. J. Intell. Systs.*, vol. 5, pp. 293–306, 1990.

[169] J. Yen, "Generalizing the Dempster-Shafer theory to fuzzy sets," *IEEE Trans. Syst., Man & Cybern.*, vol. SMC-20, no. 3, pp. 559–570, 1990.

[170] S. K. Pal and A. Dasgupta, "A way to handle subjective uncertainties and a quantitative measure of the same," in *Proc. Int. Conf. Fuzzy Logic & Neural Networks*, (Iizuka, Japan), pp. 299–302, 1990.

[171] A. Rosenfeld, "Fuzzy digital topology," *Information & Control*, vol. 40, no. 1, pp. 76–87, 1979.

[172] A. Rosenfeld, "The fuzzy geometry of image subsets," *Pattern Recognition Letters*, vol. 2, pp. 311–317, 1984.

[173] S. K. Pal and A. Ghosh, "Fuzzy geometry in image analysis," *Fuzzy Sets & Systems*, vol. 48, no. 2, pp. 23–40, 1992.

[174] C. R. Dyer and A. Rosenfeld, "Thinning algorithms for gray scale pictures," *IEEE Trans. Patt. Recog. & Mach. Intell.*, vol. PAMI-1, no. 1, pp. 88–89, 1979.

[175] S. K. Pal and R. A. King, "Image enhancement using smoothing with fuzzy sets," *IEEE Trans. Syst., Man & Cybern.*, vol. SMC-11, pp. 494–501, 1981.

[176] S. K. Pal and R. A. King, "On edge detection of x-ray images using fuzzy sets," *IEEE Trans. Patt. Anal. & Mach. Intell.*, vol. PAMI-5, no. 1, pp. 69–77, 1983.

[177] H. Li and H. S. Yang, "Fast and reliable image enhancement using fuzzy relaxation technique," *IEEE Trans. Syst., Man & Cybern.*, vol. SMC-19, no. 5, pp. 1276–1281, 1989.

[178] V. Goetcherian, "From binary to grey tone image processing using fuzzy logic concepts," *Pattern Recognition*, vol. 12, no. 1, pp. 7–15, 1980.

[179] J. Keller, G. Hobson, J. Wootton, A. Nafarieh, and K. Luetkemeyer, "Fuzzy confidence measures in midlevel vision," *IEEE Trans. Syst., Man & Cybern.*, vol. SMC-17, no. 4, pp. 676–683, 1987.

[180] W. X. Xie and S. D. Bedrosian, "Experimentally derived fuzzy membership function for gray level images," *J. Franklin Inst.*, vol. 325, pp. 154–164, 1988.

[181] N. R. Pal and S. K. Pal, "Entropy: a new definition and its applications," *IEEE Trans. Syst., Man & Cybern.*, vol. SMC-21, no. 5, pp. 1260–1270, 1991.

[182] M. K. Kundu and S. K. Pal, "Automatic selection of object enhancement operator with quantitative justification based on fuzzy set theoretic measure," *Pattern Recognition Letters*, vol. 11, pp. 811–829, 1990.

[183] H. Tahani and J. Keller, "Information fusion in computer vision using the fuzzy integral," *IEEE Trans. Syst., Man & Cybern.*, vol. SMC-20, no. 3, pp. 733–741, 1990.

[184] S. K. Pal, "A note on the quantitative measure of image enhancement through fuzziness," *IEEE Trans. Patt. Anal. & Mach. Intell.*, vol. PAMI-4, pp. 104–208, 1982.

[185] S. K. Pal, "A measure of edge ambiguity using fuzzy sets," *Pattern Recognition Letters*, vol. 4, no. 1, pp. 51–56, 1986.

[186] W. X. Xie, "An information measure for a color space," *Fuzzy Sets & Systems*, vol. 36, pp. 157–165, 1990.

[187] K. Tanaka and M. Sugeno, "A study on subjective evaluations of printed color images," *Int. J. Approximate Reasoning*, vol. 5, no. 3, pp. 213–222, 1991.

[188] J. M. B. Prewitt, "Object enhancement and extraction," in *Picture Processing and Psychopictorics*, (B.Lipkin and A.Rosenfeld, eds.), pp. 75–149, New York: Academic Press, 1970.

[189] S. K. Pal and A. Rosenfeld, "Image enhancement and thresholding by optimization of fuzzy compactness," *Pattern Recognition Letters*, vol. 7, no. 2, pp. 77–86, 1988.

[190] S. K. Pal, R. A. King, and A. A. Hashim, "Automatic gray level thresholding through index of fuzziness and entropy," *Pattern Recognition Letters*, vol. 1, no. 3, pp. 141–146, 1983.

[191] N. R. Pal, *On Image Information Measures and Object Extraction*, PhD thesis, Indian Statistical Institute, Calcutta, 1990.

[192] Y. W. Lim and S. U. Lee, "On the color image segmentation algorithm based on the thresholding and the fuzzy c-means techniques," *Pattern Recognition*, vol. 23, no. 9, pp. 935–952, 1990.

[193] T. L. Huntsberger, C. L. Jacobs, and R. L. Cannon, "Iterative fuzy image segmentation," *Pattern Recognition*, vol. 18, no. 2, pp. 131–138, 1985.

[194] R. L. Cannon, J. V. Dave, J. C. Bezdek, and M. M. Trivedi, "Segmentation of a thematic mapper image using the fuzzy c-means clustering

aalgorithm," *IEEE Trans. Geoscience & Remote Sensing*, vol. GE-24, no. 3, pp. 400–408, 1986.

[195] M. Trivedi and J. C. Bezdek, "Low-level segmentation of areal images with fuzzy clustering," *IEEE Trans. Syst., Man & Cybern.*, vol. SMC-16, no. 4, pp. 589–598, 1986.

[196] T. L. Huntsberger, C. Rangarajan, and S. N. Jayaramamurthy, "Representation of uncertainty in computer vision using fuzzy sets," *IEEE Trans. Computer*, vol. C-35, no. 2, pp. 145–156, 1986.

[197] S. K. Pal, "Fuzzy skeletonization of an image," *Pattern Recognition Letters*, vol. 10, pp. 17–23, 1989.

[198] S. K. Pal and A. Rosenfeld, "A fuzzy medial axis transformation based on fuzzy disks," *Pattern Recognition Letters*, vol. 12, no. 10, pp. 585–590, 1991.

[199] S. K. Pal and L. Wang, "Fuzzy medial axis transformation (FMAT): Practical feasibility," *Fuzzy Sets & Systems*, vol. 50, no. 1, pp. 15–34, 1992.

[200] S. Peleg and A. Rosenfeld, "A min-max medial axis transform," *IEEE Trans. Patt. Anal. & Mach. Intell.*, vol. 3, pp. 208–210, 1981.

[201] E. L. Hall, *Computer Image Processing and Recognition*, New York: Academic Press, 1978.

[202] J. J. Hopfield, "Neural network and physical systems with emergent collective computational abilities," in *Proc. National Academy of Science*, (USA), pp. 2554–2558, 1982.

[203] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two state neurons," in *Proc. National Academy of Science*, (USA), pp. 3088–3092, 1984.

[204] T. Kohonen, *Self-organization and Associative Memory*, 3rd ed., Berlin: Springer-Verlag, 1989.

[205] G. A. Carpenter, S. Grossberg, and D. Rosen, "Fuzzy ART : fast stable learning and categorization of analog patterns by an adaptive resonance system," *Neural Networks*, vol. 4, pp. 759–771, 1992.

[206] S. Grossberg (ed.), *Neural Networks and Natural Intelligence*, Reading, MA: MIT Press, 1988.

[207] K. Fukushima, "A hierarchical neural network capable of visual pattern recognition," *Neural Networks*, vol. 1, pp. 119–130, 1988.

[208] J. C. Bezdek, "On the relationship between neural networks, pattern recognition, and intelligence," *Int. J. Approximate Reasoning*, vol. 6, no. 2, pp. 85–107, 1992.

[209] S. K. Pal and S. Mitra, "Multilayer perceptron, fuzzy sets, and classification," *IEEE Trans. Neural Networks*, vol. NN-3, no. 5, pp. 683–697, 1992.

[210] S. K. Pal and S. Mitra, "Fuzzy version of Kohonen's net and MLP based classification : performance evaluation for certain nonconvex decision regions," *Information Sciences* (to appear).

[211] A. Ghosh, N. R. Pal, and S. K. Pal, "Self-organization for object extraction using multilayer neural network and fuzziness measures," *IEEE Trans. Fuzzy Systems*, vol. FS-1, no. 1, 1993 (in press).

[212] K. J. Cios and L. M. Sztandera, "Continuous ID3 algorithm with fuzzy entropy measures," in *Proc. IEEE Int. Conf. on Fuzzy Systems*, (San Diego), pp. 469–476, 1992.

[213] S. C. Lee and E. T. Lee, "Fuzzy neural networks," *Mathematical Biosciences*, vol. 23, pp. 151–177, 1975.

[214] T. Yamakawa and S. Tomada, "A fuzzy neuron and its applications to pattern recognition," in *Proc. 3rd IFSA Cong.*, (Seattle), pp. 30–38, 1989.

[215] H. Takagi and I. Hayashi, "Artificial neural network driven fuzzy reasoning," *Int. J. Approximate Reasoning*, vol. 5, pp. 191–212, 1991.

[216] J. M. Keller, R. R. Yager, and H. Tahani, "Neural network implementation of fuzzy logic," *Fuzzy sets & systems*, vol. 45, no. 1, pp. 1–12, 1992.

[217] J. M. Keller and H. Tahani, "Implementation of conjunctive and disjunctive fuzzy logic rules with neural networks," *Int. J. Approximate Reasoning*, vol. 6, pp. 221–240, 1991.

[218] D. P. Mandal, C. A. Murthy, and S. K. Pal, "Determining the shape of a pattern class from sampled points in $I\!R^2$," *Int. J. General Systems*, vol. 20, no. 4, pp. 307–339, 1992.

[219] D. P. Mandal, S. K. Pal, and C. A. Murthy, "A multivalued pattern class determining procedure," in *Proc. Int. Conf. on Fuzzy Logic & Neural Networks*, (Iizuka, Japan), pp. 343-346, 1990.

[220] D. P. Mandal, "Determining a pattern class from a set of training samples," *Presented in 78th Session of Indian Science Congress Association*, Indore, India, 1991.

[221] K. Kuratowski, *Topology, vol. I.* New York: Academic Press, 1966.

[222] S. K. Pal and D. P. Mandal, "Linguistic recognition system based on approximate reasoning," *Information Sciences*, vol. 61, no. 1/2, pp. 135–161, 1992.

[223] S. K. Pal and D. P. Mandal, "Design of a recognition system based on approximate reasoning," in *Proc. National Conf. on Ecectronics Circuit & Systems (NACONECS-89)*, (Roorkee), pp. 524–526, 1989.

[224] D. P. Mandal, C. A. Murthy, and S. K. Pal, "Formulation of a multivalued recognition system," *IEEE Trans. System, Man & Cybern.*, vol. SMC-22, no. 4, pp. 607–620, 1992.

[225] D. P. Mandal, S. K. Pal, and C. A. Murthy, "Formulation of a multivalued linguistic recognition system," in *Proc. Int. Conf. on Fuzzy Logic & Neural Networks*, (Iizuka, Japan), pp. 845-848, 1990.

[226] D. P. Mandal, "Performance of a multivalued recognition system," *Presented in 79th Session of Indian Science Congress Association*, Baroda, India, 1992.

[227] D. P. Mandal, C. A. Murthy, and S. K. Pal, "Theoretical performance of a multivalued recognition system," *IEEE Trans. Syst., Man & Cybern.* vol. SMC-24, no. 7, 1994 (To appear).

[228] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Comm. ACM*, vol. 27, no. 3, pp. 236–239, 1984.

[229] F. P. Preparata and M. I. Shamos, *Computational Geometry : An Introduction*, New York: Springer-Verlag, 1985.

[230] U. Grenander, *Abstract Inference*, New York: John Wiley, 1981.

[231] S. K. Parui, *Some Studies in Analysis and Recognition of 2-Dimensional Shapes*, PhD thesis, Indian Statistical Institute, Calcutta, 1984.

[232] R. Martin-Clouaire and H. Prade, "On the problems of representation and propagation of uncertainty in expert systems," *Int. J. Man-Mach. Studies*, vol. 22, pp. 251–264, 1985.

[233] E. Sanchez, "Medical diagnosis and composite fuzzy relations," in *Advances in Fuzzy Set Theory and Applications*, (R. K. R. M. M. Gupta and R. R. Yager, eds.), pp. 437–444, Amsterdam: North Holland, 1979.

[234] R. R. Yager, "Multiple objective decision using fuzzy subsets," *Int. J. Man-Mach. Studies*, vol. 9, pp. 375–382, 1977.

[235] A. K. Datta, N. R. Gangully, and S. Ray, "Maximum likelihood methods in vowel recognition: a comparative study," *IEEE Trans. Patt. Anal. & Mach. Intell.*, vol. PAMI-4, pp. 683–689, 1982.

[236] S. K. Pal, A. Pathak, and C. Basu, "Dynamic guard zone for self supervised learning," *Pattern Recognition Letters*, vol. 7, pp. 135 144, 1988.

[237] D. G. Schwartz, "The case for an interval based representation of linguistic truth," *Fuzzy Sets & Systems*, vol. 17, pp. 153–165, 1985.

[238] P. Billingsley, *Probability and Measure*, New York: John Wiley & Sons, 1979.

[239] R. B. Ash, *Real analysis and Probability*, New York: Academic Press, 1972.

[240] J. R. Jensen, *Introductory Digital Image Processing: A Remote Sensing Perspective*, Englewood Cliffs, NJ: Prentice Hall, 1986.

[241] P. M. Mather, *Computer Processing of Remotely Sensed Images: An Introduction*, New York: John Wiley & Sons, 1987.

[242] J. Ton, *A Knowledge Based Approach for LANDSAT Image Interpretation*, PhD thesis, Michigan State University, 1988.

[243] M. A. Fischer, J. M. Tanenbaum, and H. C. Wolf, "Detection of roads and linear structures in low-resolution aerial imagery using a multiple knowledge integration technique," *Computer Graphics & Image Processing*, vol. 15, pp. 201–223, 1981.

[244] R. Bajcsy and M. Tavakoli, "Computer recognition of roads from satellite pictures," *IEEE Trans. Syst., Man & Cybern.*, vol. SMC-6, pp. 623–637, 1976.

[245] W. D. Groch, "Extraction of line shaped objects from aerial images using a special operator to analyze the profiles of functions," *Computer Graphics & Image Processing*, vol. 18, pp. 347–358, 1982.

[246] D. M. Mckeown and J. F. Pane, "Alignment and connection of fragmented linear features in aerial imagery," Tech. Rep. CUU-CS-85-122, Department of Computer Science, Carnegie-Mellon University, 1985.

[247] M. L. Zhu and P. S. Yeh, "Automatic road network detection on aerial photographs," in *Proc. IEEE Conf. on Computer Vision & Patt. Recog.*, 1986.

[248] J. C. Bezdek, R. L. Cannon, and M. L. Pai, "An experimental system for road detection from multiple radar images," Tech. Rep., Department of Computer Science, University of South Carolina, Columbia, 1985.

[249] S. Vasudevan, R. L. Cannon, J. C. Bezdek, and W. L. Cameron, "Heuristics for intermediate level road finding algorithm," *Computer Vision, Graphics & Image Processing*, vol. 44, pp. 175–190, 1988.

[250] D. M. Mckeown, W. A. Harvey, and J. Mcdermatt, "Rule based interpretation of aerial imagery," *IEEE Tran. Patt. Anal. & Mach. Intell.*, vol. PAMI-7, pp. 570–585, 1985.

[251] S. W. Wharton, "A spectral knowledge based approach for urban land cover discrimination," *IEEE Trans. Geoscience & Remote Sensing*, vol. GE-25, pp. 272–282, 1987.

[252] D. M. Mckeown, "The role of artificial intelligence in integration of remotely sensed data with geographic information systems," *IEEE Trans. Geoscience & Remote Sensing*, vol. GE-25, pp. 330–348, 1987.

[253] D. G. Goodenough, M. Goldberg, G. Plunkell, and J. Zelek, "An expert system for remote sensing," *IEEE Trans. Geoscience & Remote Sensing*, vol. GE-25, pp. 349–359, 1987.

[254] A. M. Nazif and M. D. Levine, "Low level image segmentation : an expert system," *IEEE Trans. Patt. Anal. & Mach. Intell.*, vol. PAMI-6, pp. 555–577, 1984.

[255] T. M. Silberberg, "Multiresolution aerial image interpretation," in *Proc. DARPA Image Understanding Workshop*, (Cambridge, MA), pp. 505–511, 1988.

[256] R. Mohan and R. Nevatia, "Percertual grouping for the detection and description of structures in aerial images," in *Proc. DARPA Image Understanding Workshop (Cambridge, MA)*, pp. 512–526, 1988.

[257] C. A. Murthy, N. Chatterjee, D. Dasgupta, S. Ghosh, B. Uma Shankar, and D. Dutta Majumder, "Development of algorithm and software for shape analysis and pattern recognition," *Tech. Rep. ECSU/Def/1*, Electronics and Communication Sciences Unit, Indian Statistical Institute, Calcutta, 1991.

[258] C. A. Murthy, N. Chatterjee, B. Uma Shankar, and D. Dutta Majumder, "IRS image segmentation: minimum distance classifier approach," in *Proc. Int. Conf. on Patt. Recog. (ICPR)*, 1992.

[259] S. K. Parui, B. UmaShankar, A. Mukherjee, and D. Dutta Majumder, "A parallel algorithm for detection of linear structures in satellite images," *Pattern Recognition Letters*, vol. 12, no. 12, pp. 665–670, 1991.

[260] R. Jeansoulin, Y. Fonyaine, and W. Frei, "Multitemporal segmentation by means of fuzzy sets," in *Proc. 1981 Machine Process. of Remotely Sensed Data Symp.*, (West Lafayetts, IN), pp. 336–339, 1981.

[261] S. D. Zenzo, R. Bernstein, S. D. Degloria, and H. G. Kolsky, "Gaussian maximum likelihood and contextual classification algorithms for multicrop classification," *IEEE Trans. Geoscience & Remote Sensing*, vol. GE-25, no. 6, pp. 805–814, 1987.

[262] J. T. Kent and K. V. Mardia, "Spatial classification using fuzzy membership models," *IEEE Trans. Patt. Anal. & Mach. Intell.*, vol. PAMI-10, no. 5, pp. 659–671, 1988.

[263] F. Wang, "Fuzzy supervised classification of remote sensing images," *IEEE Trans. Geoscience & Remote Sensing*, vol. GE-28, no. 2, pp. 194–201, 1990.

[264] F. Wang, *Integrating Expert Vision Systems and Spatial Database by Unifying Knowledge Representation Schemes: Development in Remote Sensing Image Analysis Systems and Geographical Information System*, PhD thesis, University of Waterloo, Waterloo, Canada, 1989.

[265] "IRS data users hand book," *Document No.- IRS/NRSA/NDC/HB-01/86*, NRSA, Hyderabad, Sept., 1986.

[266] B. L. S. Prakasa Rao, *Nonparametric functional estimation*, Academic Press, New York, 1983.

[267] E. Parzen, "On the estimation of a probability density function and the mode," *Annals of Mathematical Statistics*, vol. 33, pp. 1065–1076, 1962.

[268] E. Cacoullos, "Estimation of a multivariate density," *Annals of the Institute of Statistical Mathematics*, vol. 18, pp. 178–189, 1966.

[269] A. K. Jain and M. D. Ramaswami, "Classifier design with Parzen windows," *Int. J. of Pattern Recognition & Artificial Intelligence*, vol. 2, pp. 211–228, 1988.

[270] R. P. W. Duin, "On the choice of smoothing j parameters for Parzen estimation of probability density functions," *IEEE Trans. Computer*, vol. C-25, pp. 1175–1179, 1976.

[271] D. D. Loftsgaarden and C. P. Quensenberry, "A nonparametric estimate of a multivariate density function," *Annals of Mathematical Statistics*, vol. 36, pp. 1049–1051, 1965.

LIST OF PUBLICATIONS OF THE AUTHOR

1. "Fuzzy logic and approximate reasoning: an overview," *J. Inst. Elec. & Telecom. Engrs.*, vol. 37, no. 5/6, pp. 548–560, 1991
   [Co-author: S. K. Pal].

2. "Determining the shape of a pattern class from sampled points in $I\!R^2$," *Int. J. General Systems*, vol. 20, no. 4, pp. 307–339, 1992
   [Co-authors: C. A. Murthy and S. K. Pal].

3. "Linguistic recognition system based on approximate reasoning," *Information Sciences*, vol. 61, no. 1/2, pp. 135–161, 1992
   [Co-author: S. K. Pal].

4. "Formulation of a multivalued recognition system," *IEEE Trans. System, Man & Cybern.*, vol. SMC-22, no. 4, pp. 607–620, 1992
   [Co-authors: C. A. Murthy and S. K. Pal].

5. "Theoretical performance of a multivalued recognition system," *IEEE Trans. Syst., Man & Cybern.* vol. SMC-24, no. 7, 1994 (To appear).
   [Co-authors: C. A. Murthy and S. K. Pal].

6. "Design of a recognition system based on approximate reasoning," in *Proc. National Conf. on Ecectronics Circuit & Systems (NACONECS-89)*, (Roorkee), pp. 524–526, 1989
   [Co-author: S. K. Pal].

7. "A multivalued pattern class determining procedure," in *Proc. Int. Conf. on Fuzzy Logic & Neural Networks*, (Iizuka, Japan), pp. 343-346, 1990
   [Co-authors: S. K. Pal and C. A. Murthy].

8. "Formulation of a multivalued linguistic recognition system," in *Proc. Int. Conf. on Fuzzy Logic & Neural Networks*, (Iizuka, Japan), pp. 845-848, 1990
   [Co-authors: S. K. Pal and C. A. Murthy].

9. "Determining a pattern class from a set of training samples," *Presented in 78th Session of Indian Science Congress Association*, Indore, India, 1991.

10. "Performance of a multivalued recognition system," *Presented in 79th Session of Indian Science Congress Association*, Baroda, India, 1992.