CLASSIFICATION OF HANDWRITTEN CHARACTERS: DEFINITION, IDENTIFICATION AND UTILISATION OF REGIONAL CHARACTERISTICS FOR LOWER CASE ENGLISH CHARACTERS

Thesis submitted by

R. KRISHNAN THAMPI

for the award of the degree of

DOCTOR OF PHILOSOPHY

Under the

FACULTY OF TECHNOLOGY

DEPARTMENT OF ELECTRONICS

COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY

COCHIN - 682 022, KERALA

AUGUST 1994

DECLARATION

I hereby declare that the work presented in this thesis is based on the original work done by me under the supervision of Dr. C. S. Sridhar in the department of Electronics, Cochin University Of Science And Technology and that no part thereof has been presented for the award of any other degree.

Cochin 682 033

August 24, 1994.

R. Krishnan Tampi

<u>CERTIFICATE</u>

This is to certify that the thesis titled **CLASSIFICATION OF HANDWRITTEN CHARACTERS: DEFINITION, IDENTIFICATION AND UTILIZATION OF REGIONAL CHARACTERISTICS FOR LOWER CASE ENGLISH CHARACTERS** is a report of the original work carried out by R. Krishnan Tampi under my supervision and guidance in the department of Electronics, Cochin University Of Science And Technology and that no part thereof has been presented for the award of any other degree.

Dr. C. S. Sridhar
Professor
Department Of Electronics
Cochin University Of
Science And Technology

Cochin 682 022
August 24, 1994

ACKNOWLEDGEMENTS

It is with a deep sense of gratitude that I recall my association with Prof. C. S. Sridhar under whose supervision and guidance I began my forays into the field of character recognition. This thesis is the outcome of the education and guidance I received from Prof. C. S. Sridhar. I wish to express my hearty thanks to him for helping me in achieving this goal successfully.

I wish to express my thanks to Prof. K. G. Nair, Head, Department of Electronics, Cochin University of Science And Technology, Cochin for allowing me to work in the department for my Doctoral studies.

I wish to record my thanks to my employer, M/S FACT Ltd., Udyogamandal for permitting me to undertake this higher study.

I wish to thank my friends and colleagues for the support they had given me during the years when I was pursuing my studies.

ABSTRACT

A new procedure for the classification of lower case English language characters is presented in this work. The character image is binarised and the binary image is further grouped into sixteen smaller areas, called Cells. Each cell is assigned a name depending upon the contour present in the cell and occupancy of the image contour in the cell. A data reduction procedure called Filtering is adopted to eliminate undesirable redundant information for reducing complexity during further processing steps.

The filtered data is fed into a primitive extractor where extraction of primitives is done.

Syntactic methods are employed for the classification of the character. A decision tree is used for the interaction of the various components in the scheme, like the primitive extraction and character recognition. A character is recognized by the primitive by primitive construction of its description. Openended inventories are used for including variants of the characters and also adding new members to the general class. Computer implementation of the proposal is discussed at the end using handwritten character samples. Results are analyzed and suggestions for future studies are made. The advantages of the proposal are discussed in detail.

CONTENTS

LIST OF FIGURES

CHAPTER - I

CHAPTER - III

CHAPTER - IV

CHAPTER - V

CHAPTER - VII

LIST OF TABLES

CHAPTER - III

CHAPTER - V

CHAPTER - VI

CHAPTER - VII

....

INTRODUCTION

1.1 WHAT IS PATTERN RECOGNITION

The definition of "pattern" in English dictionary is, "an example or model", that is, some thing which can be copied. A pattern can be defined as any distinguishable interrelation of data, events or concepts. A pattern is also an imitation of a model. Examples of patterns are: The shape of a face, a table, the order of a musical note in a piece of music.....etc. Thus recognition of a face, a printed or handwritten word, the lyrics composed by a poet, the diagnosis of a disease from symptoms or data from clinical investigation,..etc, are all pattern recognition problems.

The recognition of patterns include visual and aural, recognition of spatial patterns (pictures, fingerprints, handwritten characters .....etc.) and temporal patterns (speech, wave forms....etc.) with the help of sensory aids. Conceptual or abstract items can be recognized without the help of sensory aids.

1.2 APPLICATIONS OF PATTERN RECOGNITION

The application areas of pattern recognition can be grouped into the following [GR1]:

1). Man machine communications like automatic speech recognition, speaker identification, OCR systems, cursive script recognition, image understanding.

2). Biomedical applications like ECG, EEG, EMG analysis, Cytological, Histological and other sterological, applications, X - Ray analysis and diagnosis.

3). Applications in Physics like Bubble chamber events, other forms of track analysis and high energy Physics

4). Crime and criminal detection - Finger print, Hand writing, Speech, sound and Photo identification.

5). Natural resources study and estimation - Agriculture, Hydrology, Forestry, Geology, Environment, Cloud pattern and Urban quality.

6). Sterological Applications - Metal and mineral processing and Biology.

7). Military Applications - Nuclear device explosion, Radar and Sonar detection, Missile guidance and detection, Target identification, Reconnaissance.

8). Industrial Applications - CAD, CAM, Computer assisted product assembly and testing, automated inspection and quality control, nondestructive testing.

9). Robotics and Artificial Intelligence - Intelligent sensor technology and natural language processing.

1.3 BASIC CONCEPTS OF PATTERN RECOGNITION

The subject of pattern recognition spans a number of scientific disciplines, uniting them in the search for the solution to the common problem of recognizing the members of a class in a set containing elements from many pattern classes. A pattern class is a category determined by some given common attributes. A pattern is the description of any member of a category representing a pattern class. When a set of patterns falling into a disjoint class is available, it is desirable to categorize these patterns into their respective classes through the use of some automatic device. The reading and processing of canceled checks is such a problem. Such tasks can easily be performed by human beings, but machines can achieve much greater speeds.

Hierarchical relations exist between patterns and pattern classes. Consider the character recognition problem. A specified letter or numeral, no matter how it is printed or written, retain some attributes which can be used as a means for identification. They are identified and classified according to the observed attributes [GR6]. Thus the basic functions of a pattern recognition system is to detect and extract common features from the patterns describing the objects that belong to the same pattern class and to recognize the pattern in a new environment and classify it as a member of the pattern class under consideration [GR2].

1.4 FUNDAMENTAL PROBLEMS IN PATTERN RECOGNITION SYSTEM DESIGN

The design of an automatic pattern recognition system involve some major problem areas [GR6]. The first one is the representation of input data which are measured from the objects to be recognized. This is a sensing problem. Each measured quantity describes the characteristics of a pattern and in turn the object. Suppose the pattern in question is a character. A grid measuring scheme as shown in Fig: 1.1, can be successfully used in the sensor [GR4]. If it is assumed that the grid has "n" elements, the measurements can be arranged in the form of a measurement vector:

$$\begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_n \end{bmatrix}$$

where each element x_i is assigned the value of 1 if the i^{th} cell contain a portion of the character and is

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fig:- 1.1

A GRID MEASURING SYSTEM

assigned a value of 0 otherwise.

The pattern vector contains all the measured information available about the pattern. The measurements performed on the objects of a pattern class may be regarded as a coding process which consists of assigning to each pattern characteristic, a symbol from the alphabet set $\{ x_i \}$.

The second problem in pattern recognition concerns the extraction of characteristic features or attributes from the input data and the reduction of dimensionality of pattern vectors. This is often referred to as the preprocessing and feature extraction problem. For example, in character recognition, strokes are often extracted as features.

The third problem involves the determination of the optimum decision procedures which are needed in the identification and classification process. After the observed data has been expressed in the form of pattern points or measurement vectors in the pattern space, the machine is required to decide to which class the data belong to.

With the above overview of the major problems, a functional block diagram, as shown in Fig:1.2, will provide a conceptual description of an adaptive pattern recognition system. The functional blocks are only for convenience in analysis and do not produce any isolation of interactive operations between blocks. The pattern to be recognized must posses a set of measurable characteristics and when these measurements are similar within a group of patterns the latter are considered to be members of the same class.

1.5 AN APPLICATION - CHARACTER RECOGNITION

Character recognition has been a subject of great interest to many computer scientists, engineers and people from other disciplines. Intensive research has been made in this field and this has made possible efficient means of entering data directly in to the computer and capturing information from data sheets, books and other machine printed or handwritten materials. Such capabilities greatly widen the application of computers in the areas like automatic reading of texts and data, man-machine communication, language processing and machine translation. Handling of bulk data generated by offices, banks and the like is made possible because of the capabilities of computers. While computers can process data very quickly, the input of data is very slow and this has been a major bottle neck in data processing.

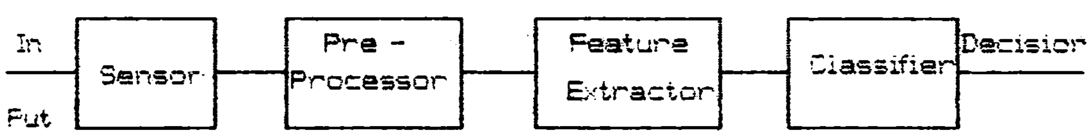


Fig :- 1.2

FUNCTIONAL BLOCK DIAGRAM OF A PATTERN RECOGNITION SYSTEM

The block diagram of a character recognition system is shown in Fig:1.3 [GR3]. At the input end, characters typed or written are scanned and digitized to produce a digitized image. The characters are typically scanned in a horizontal direction with a single-slit reading head which is narrower but taller than the character. As the head moves across the character, it produces a signal which is conditioned to be proportional to the rate of increase of the character area. At this stage the system will start to locate regions in which data was entered, type written or printed on the input document. Once these regions are located, the data blocks are segmented into character images. Instead of keeping the images in multi gray levels, it is common practice to convert them into binary matrices to save memory space and computational effort [SUE4]. Depending on the complexity of the character shapes and the vocabulary involved, the size of the matrix, which reflects the resolution of a digitized character, is varied to achieve speed and accuracy. Typically a character of size 8(wide) * 10 (high) pixels is sufficient for recognizing stylized type fonts, where as for handwritten English, Chinese and Indian characters, the dimension of the matrix size is comparatively high [SUE3].After digitization, location and

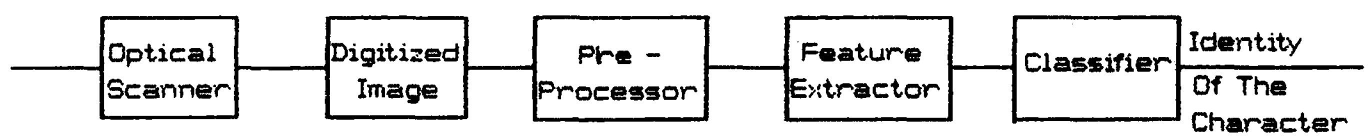


Fig 1.3

THE BLOCK DIAGRAM OF A CHARACTER RECOGNITION SYSTEM segmentation, characters in the form of binary matrices go through the preprocessor to eliminate random noise, voids, bumps and other spurious components which might still be contained in them. In some cases normalization in size and orientation in position are performed to facilitate the extraction of distinctive features in the subsequent stages [SUE2]. Once the characteristics of the cleaned character have been extracted, they are matched to a list of references. In many cases a knowledge base is built during the learning process to classify the characters. In addition, distance measures as well as shape derivation, shape matching and hierarchical feature matching in the form of decision trees are also used. The decision maker is influenced by the types of features that are detected. A successful recognition system is built on the joint operation and performance of the feature detector and classifier.

As a result of the great many styles and types of writings that can be seen in real life applications, recognition of handwritten characters is far from solved. The main problems are [GR8]:

1). Variations in character shapes, for example letters written as n and r.

2) Variations in size of the character.

3) Variations in pitch which correspond to proportional spacing. These variations affect the location and segmentation of characters.

4). Ornaments and serifs of the characters.

5). Variations in line thickness.

6). Touching of wide characters like m and w.

In order to recognize handwritten characters, the machine must be able to handle all the above problems.

Variations in handwritten characters are greater because, they can be written in innumerable ways. Since each person has his or her own ways and styles of writing, and character samples written by the same hand are not always identical in size or shape, there are an infinite number of possible character shapes. The problem of handwritten character recognition is of great interest to researchers, because even human beings are said to make about 4% error [SUE4]. Although a lot of research has been done and is going on, the following three problems still exist in the recognition of handwritten characters [SUE 5].

1) Fewer sub patterns are to be used to describe the complicated structure of handwritten characters.

2). Thinning distortion should be avoided when extracting features.

3). Seek a universal approach applicable to both printed and handwritten characters.

The first problem can be overcome by extracting peripheral features, which require fewer sub patterns. Thinning distortions can be overcome if features are extracted from the original image itself. Still the third problem of an universal approach exists. In this work this problem is addressed in a limited way.

1.6 THE PROBLEM

The problem on hand is the recognition of LOWER CASE ENGLISH LANGUAGE CHARACTERS. To enable this goal a new approach in image coding and data reduction is presented in this work. The classification techniques used are based on syntactic method. The character samples are binarized using a square grid of size 12*12. The physical dimensions of the grid depends on the size of the handwritten character and always a 12 * 12 grid is used for the digitization of the image. The binarized image is labeled. The labeling procedure adopted is a new formulation. A three digit labeling is adopted in this work. A set of features are defined for the class of images treated which help in the filtering. The modified image is subjected to FILTERING, a procedure used for data reduction. The reduced image is then subjected to primitive extraction and then a computationaly simple recognition technique is applied for classifying the characters into their respective classes. Syntactic methods are employed at the classification stage. Table look up operation is used to simplify the procedures at image coding, data reduction and classification stages. The primitives defined are of a new kind and an openended inventory is maintained for the addition of new primitives for different descriptions of the types of images handled here and for adding new members to the group.

1.7 A BRIEF REVIEW OF THE FOLLOWING CHAPTERS

In chapter 2, a literature survey is made. Here some of the works pertaining to pattern recognition, especially character recognition are discussed.

Chapter 3 is devoted to binarizing and labeling of the character image. A modified image labeling procedure suitable to represent and process the type of images treated in this work is presented. The properties exhibited by the images are analyzed and a most suitable labeling procedure is adopted.

Chapter 4 discusses the need of feature selection in character recognition. The selection of suitable features for lower case English language characters is presented. It is also shown that two types of features are required for the kind of data reduction technique used.

Chapter 5 deals with data reduction techniques employed in this work. The data reduction procedure called FILTERING, is performed in three stages. Necessary rules are generated and used for FILTERING. A PRESERVATION MEASURE is defined in this chapter. This measure ensure that the reduced version of the image retain sufficient characteristics of the original image.

Chapter 6 deals with the selection and extraction of Primitives for further processing. Two types of primitives, Main Primitives and Auxiliary Primitives, are defined. Their extraction is also discussed in this chapter.

In chapter 7, classification of the unknown character image and its computer implementation are discussed. Rules are developed for the purpose. Again table look up method is adopted for convenience.

In character 8, concluding remarks and suggestions for future works based on the new formulation are presented.

CHAPTER - II LITERATURE SURVEY

2.1. INTRODUCTION

Marc Berthod [BERT] states that "the number of different studies that can be seen in literature that deal with cursive script writing analysis is small, less than twenty. Although this statement was made twelve years ago, the situation has not changed much. Only very little work is being done in this field, eventhough cursive script writing is a natural way of communication. This lack of interest according to Berthod is because of the fact that, "the problem is as difficult as the recognition of continuous speech, while applications are closer to optical character recognition, which is a simpler problem to solve". Marc Berthod continues to state further that, "there nevertheless remain an extreme scientific interest in the quest for the solution of this problem, and there is no doubt that practical implications could be important in the future, since a large part of human communication is still, and will probably continue to be, by means of handwriting".

In this chapter a survey is made of the available works and also similar works like hand print recognition, Chinese character recognition..etc. To my knowledge no previous work was reported in computer recognition of handwritten LOWER CASE ENGLISH LANGUAGE CHARACTERS.

2.2. LITERATURE SURVEY:

The open literature on character recognition can be divided into three parts. The first deals with character description and their generation for different languages. Classifiers based on syntactic methods form the second part which has approaches like definition and extraction of primitives ...etc. in them. Syntax aided decision tree classifiers of various scripts form the third part of the reported work.

In one of the earliest works on character recognition [GRIM], Grimsdale et al says that, "a description of a character is produced in terms of the length and slope of straight line segments and the length and curvature of curved segments". Many authors like Narasimhan and Reddy [NAR1] describe systems with variations in the structural approach. In other scripts, structural character recognition have been employed by, Stalling [STAL] for Chinese characters, Seth and Chatterjee [SETH] for Devanagari, Yoshida ..et al for Japanese, and Chinnuswamy and Krishna moorthy [CHIN] for Tamil.

In [RAB1], Rabinow remarks that the terms "clearly written" has a loose interpretation and depends on the verdicts of the recognizer. He also makes the same comment on the term "unconstrained".

The structural description for character recognition can be found in the work of Grimsdale et al [GRIM]. This heuristically developed system does not employ any explicit syntactic technique. Narasimhan, an early proponent of syntactic description, later proposed a syntax-directed interpretation for a class of pictures in [NAR2]. In [NAR1], Narasimhan and Reddy provides a syntax aided approach to the recognition of hand printed English characters. They go on to say that "the syntax rules currently in use must be refined, modified and augmented continuously on the basis of experience, (ie, on the outcome of past performance) and other relevant knowledge acquired". They also mention in this work that the flexibility necessary in a recognition system should use the rules only as an aid and flexibility and openendedness shall be the basic features of a recognition system so that it can learn from and grow with experience. If the above condition is fulfilled, the system will imitate the performance of human beings. In this context it may noted that in the scheme developed in this thesis, an effort is made to incorporate flexibility and openendedness.

The shift from syntax guided to syntax aided recognition is mainly caused by the desire to evolve a recognition system which can imitate the human beings. A perusal of the relevant literature indicates that to equal human performance the computer must possess the sophistication of the eye-brain system which uses description as a tool. Description represents a higher level of intelligence. A perspective discussion on description is given in [KANE].

Uhr in [UHR], offers an "alphabet" of straight lines and curves from which patterns including characters can be generated.

Marc Berthod in [BERT], explains the process of cursive script writing. He explains the process of generation of handwriting and implies that this is caused by the following three types of forces.

1). Active forces due to muscular activity

2). Viscosity of muscles and articulation

3). Inertia of arms and muscles.

This work goes on to state that "excepting very few works, most reported systems relay on structural primitives". The consensus on structural approach although not a very common situation in pattern recognition, is mainly because of the geometrical shapes of these characters. Berthod goes on to explain both on-line and off-line processing in this work.

In [EDEN], Eden proposed a set of eighteen strokes as primitivs which can be deduced by symmetry about a horizontal or vertical axis and by vertical shift, from a set of four basic strokes called "hump", "bar", "hook" and "loop". Any isolated character can be defined by the concatenation of the Eden's primitives, where as it is not possible to completely represent a word using these. Eden along with Mermelstein used these primitives for the generation of a recognition system [MER1 & MER2].

In [BOZI], a method of estimating a correct string X from it's noisy version Y produced by cursive script writing is explained.

In [SUE1]. C. Y. Suen gives the major building blocks of the OCR system, including digitization, preprocessing, smoothing, standardization and feature extraction, and classification of characters. It also presents a brief survey of the challenging problems of recognition of handwritten characters. Suen in this analysis says that "there exists hundreds of type fonts and thousands of print fonts, each having its own style and peculiarities and as such machine recognition of multi font and hand-written characters is far from being solved". The main problems are variations in :-

1). Character shape

2). Size

3). Pitch

4). Line thickness

5). Ornaments and serifs.

It may be noted that hand printed characters do not exhibit these many variations.

Suen goes on to say that the hand written character recognition is a problem of great interest and challenge to researchers because of the complexity of the problem.

Q. R. Wong and C. Y. Suen, in [WONG], analyses a general decision tree classifier with overlap for large character set recognition. They say that the main advantage of a decision tree over a single stage classifier is that complex global decisions can be made via a series of simple and local decisions.

In [SUE2], C.Y.Suen discusses the distinctive features in automatic recognition of hand printed characters. In this work Suen describes the process of recognition as follows. Characters are recognized from the features extracted. The input character is smoothed and cleaned by the preprocessor before it reaches the feature extractor. Good preprocessors and feature extractors are the prerequisites to a successful character recognition system. The various features found in the literature are grouped in to the following two classes by Suen in this work. These are:

1) Global analysis, and

2) Structural analysis.

These two types of features are further subdivided into six categories. These categories are:

a) Distribution of points,

b) Transformations,

c) Physical measurements,

d) Line segments and edges,

e) Outline of character, and

f) Centre line of character.

Suen in this work discuss the performance and recognition rates of various systems employing these features.

C.Y.Suen and S.Mori discusses the need for standardization in [SUE3]. In this work they discuss in detail the necessity of standardization of character shape for the automatic recognition of hand printed characters. It is argued in this work that consistency in character shape is the key to any successful character recognition system. Since the written characters must be familiar to the human eye and readable by computers, special care must be taken in the design or adoption of the shapes of these characters which have similar geometrical and topological properties. Because of the rich contextual information available, there may not be any difficulty in the recognition even in the absence of discriminating features. But this is not the case with a computer and hence the need for the standardization in character shape.

In [SUE4], C. Y. Suen explains the need and process of feature extraction. He describe the hand print system in some detail. It is suggested in this work that instead of using multi-gray levels it is sufficient to convert the character image in to a binary image for further processing. By converting the image into a binary image, complexity in further processing can be avoided. It is mentioned here that a matrix size of 30 * 40 is appropriate in most character recognition problems.

On hand print recognition, Suen..et al presents a survey of recognition algorithms, data bases, character models and hand print standards in [SUE5]. Characteristics, problems and actual results on online recognition of hand printed characters for different applications are also discussed in this work. They attribute the possible causes for errors in hand print recognition to the infinite variations of shapes resulting from the writing habit, style, education, region of origin, social environment, mood, health and other conditions of writer, as well as other factors such as the writing instrument, writing surface, scanning methods and machine recognition algorithms. The paper presents the advances in hand print recognition according to the vocabulary studied and recognition techniques are examined and compared. They go on to emphasize the fact that the central issue in character recognition lies in the extraction of features. The paper classifies the recognition techniques in to three classes, namely, global features, distribution of points and geometrical and topological features. It is stated in this work that syntactic or logic methods are more frequently used in character recognition than in other fields of pattern recognition.

Online recognition of printed characters of any font and size is dealt with in [KAHA], by S.Kahan, T. Pavlidis and H. S. Baird. They describe a system that recognizes printed text of various fonts and size for the Roman alphabet. The system combines several techniques to improve the overall recognition rate and uses a binary image. The fact that feature based methods are less sensitive to font shape and size is stressed in this work.

Some work has been done in the recognition of indian language like Devanagari, Bengali, Tamil, Telugu and Kannada. The characters in these languages are large in number and are complex in their structure. Most of the researchers adopted the strategy of splitting the characters first into primitives (line-like elements) satisfying certain relational constraints. These are then used as features and classification is done by means of a decision tree approach or topological matching procedure. Sethi & Chatterjee, [SETH], considered loops and line-like primitives of constrained hand printed Devanagari (vowels and consonants) as features. They adopted a multistage decision process in which each stage of the decision narrows down the choice of the class membership. Sinha & Mahabala, [SINHA], used labeling as a local feature extraction operation for Devanagari characters. Every point in the pattern is assigned a label depending on the local property it exhibits with respect to the neighboring points. Primitives are searched from the labeled pattern and a syntactic method is used for classification. Som & Nath, [SOM], considered distances of the Bengali characters from the boundary of the frame and used non-parametric sequential method for their classification. Ray & Chatterjee, [RAY], considered thirteen different primitives for hand printed Bengali characters by a nearest neighbor classification scheme. Rajasekharan & Deekshatulu, [RAJA], used a directed curve tracing method and split the Telugu characters into primitives and basic letters. Classification was done by a decision tree. Siromony & Chandrasekaran, [SIRO], obtained a small string depending on the frequency of runs of 1's in both columns and rows of printed Tamil characters which was compared with the stored string pattern for recognition. Chinnuswamy & Krishnamoorthy, [CHIN], determined line-like primitives from hand printed Tamil characters. Labeled graphs are used to describe the structural composition of characters in terms of the primitives and the relational constraint satisfied by them which was then used for computing correlation coefficients and topological matching for classification. Rammohan &

Chatterji, [RAM], considered nine different primitives for distorted Kannada characters which were then recognized using the Viterbi algorithm.

Most of the earlier work on handwritten character recognition depended on the selection and use of primitives which are useful only to the particular procedure selected. A universal approach was not possible with these. Also the inventories of primitives were of a closed nature so that the addition of new members to the class was very difficult. In the present approach, an attempt is made to provide a universal set of primitives and the inventory of primitives is kept openended to facilitate addition of new members into the class.

Reported work on cursive script writing is very small, especially in the case of lower case English language characters.

CHAPTER III - IMAGES AND CODING

3.1 INTRODUCTION

The theme of the chapter is the development of a new vocabulary which when used for labeling the binary image helps in data reduction. A fixed number of symbols are aimed at in achieving this. This is done for standardizing the image shape as suggested by Suen in [SUE3].

3.2 PROPERTIES OF CHARACTERS

The images treated in this work are the Lower Case English Language Characters. These are finite in number, twenty six to be specific, and are geometric in shape [BERT], although many variations are possible for each character. These characters fall into the following five categories:

1). Characters which are symmetrically placed on the grid like the characters a, m, n, o, ...etc

2). Characters which show a predominance in the upper half of the grid, like p, q,...etc

3). Characters which show a predominance in the lower half of the grid, like b, d,...etc

4). Characters which show a predominance in the left hand side of the grid, and

5). Characters which show a predominance in the right hand side of the grid.

Here the term predominance is used in the sense of higher pixel occupancy. Even among these five classes there are sub classes. Table 3.1 gives a list of all the twenty six characters grouped into these five categories. Since punctuations and numerals (Arabic) do not exhibit variations like alphabets, they are not considered here.

The geometrical properties exhibited by these characters are unique and hence topological properties can be exploited successfully for further processing [BERT].

It is always advantageous to represent the character as a binary image [ROSE]. This form of digitization enables data reduction, thereby reducing the memory requirements. More over binary matrices can also represent a character image completely as effectively as any other multi-gray level scheme. Based on this argument characters are transformed into binary images.

A binary picture can be defined as a mapping of each grid point of the picture on an orthogonal co-ordinate system on to a set composed of 1's (image points) and 0's (blanks) [AGUI]. This is equivalent to saying that wherever a boundary of the image is present these points will

SET - I Characters which are symmetric about the grid

" a, c, e, f, k, l, m, n, o s, u, v, w, x, z "

SET - II Characters which show a predominance on the upper half of the grid.

" g, j, p, q, r, y "

SET - III Characters which show a predominance on the lower half of the grid.

" b, d, h, i, t "

SET - IV Characters which show a predominance on the left hand side of the grid.

" d, g, j, q, y "

SET - V Characters which show a predominance on the right hand side of the grid.

" b, h, k, p, t"

It may be noted that the sets IV and V are only combinations of characters belonging to sets II and III and hence they need not be considered as separate sets.

TABLE 3.1

CHARACTERS GROUPED IN TO THE FIVE CLASSES

be digitized as 1, and wherever no boundary is encountered those points will be digitized as "0".

The starting point of most image understanding schemes is the partitioning of the picture into pixels. The sub-sets of the pixel partition are all identical in size and shape and usually are geometrically simple, eg..rectangles, squares and hexagons. The partitioning adopted in this analysis is a rectangle/square (a special case of a rectangle) grid.

For analytical purposes, a picture is simplified when it is partitioned into pixels. Instead of keeping track of the values of the picture within each pixel subset, the picture can be assumed to have the same value at all points within the subset. The single value chosen is generally the average of the individual point values or whatever approximation thereof is produced by sampling.

The binary image is treated in this work is for the sake of simplicity as well as because this is sufficient at least for the class of images treated. The image understanding process is affected by the size of the subset chosen for the pixel partition. All practical pictures are finite in extent so that the pixel partition will have only a finite number of subsets. If the area of each subset is quite small, the number of pixels will be relatively large and processing burden will be severe. The number of pixels can be made small by increasing the area of the subsets, but the averaging process over each subset may then cause the pixel representation to differ significantly from the original image.

3.3 THE SIZE OF THE CHARACTER MATRIX

Pictures convey most of their information through edges. That is why edges are extracted in most image understanding systems. Contours can be represented by gray level differences in pictures. When information lies in boundaries and not in textures, as in the case of characters, the sampling of pictures is equal to the quantization of parametrized contour functions. In a picture, outlines can appear any where. Therefore a uniform quantizer is the most appropriate.

The resolution required for digitizing a character depends on the thickness of the lines making up the character. A 12 * 12 grid matrix has given satisfactory results.

The amount of information that can be extracted from a sample is restricted to the resolution of the digitizer or the size of the grid matrix. Higher resolution reduces the recognition error rate but does this at the cost of effort and speed. Different resolutions used by various researchers are given in Table 3.2.

The use of the 12 * 12 square grid is justified by the end results.

3.4 THE BINARY IMAGE

The analysis here starts with the image of a character superposed on a 12 * 12 grid. This is a square grid with 12 pixels in every row with 12 columns, thereby dividing the total image area into 144 pixels. The pixel size depends only on the character size and the total area could be either square or rectangle in shape. The binarization of the image is done as indicated earlier, that is, wherever a boundary is present the pixel value is recorded as 1 and in other cases the pixel value is recorded as 0. Fig: 3.1 shows the binary images of some characters.

The 144 pixels are grouped into 16 cells. A cell consists of 9 pixels in 3 rows and 3 columns. Hence each cell can be considered as a 3 * 3 matrix. A 3 * 3 cell was chosen to provide each pixel with 8 neighbors. This is one of the simplest and is related to the 8 connected chain code.

RESEARCH WORKER	YEAR	MATRIX SIZE
GRINSDALE.. et al	1959	40 * 64
HIGHLY MAN	1962	12 * 12
MUNSON	1968	24 * 24
TOU & GONZALEZ	1971	60 * 60
CASKEY & COSTS	1972	48 * 48
BEUN	1973	32 * 32
MORI... et al	1975	60 * 60
THE AUTHOR	1984	12 * 12

TABLE - 3.2

RESOLUTIONS USED IN DIFFERENT CHARACTER RECOGNITION SCHEMES

000011111000	000001100000	000000100000
000100001000	000010100000	000000100000
001000000100	000101000000	000000100000
010000000100	000101000000	000000100000
010000000010	000110000000	000000100000
100000000010	000100111111	000000100000
100000000100	001100000100	000000100000
100000000100	011100000100	011111100000
100000001000	110010000100	110000100000
110000011000	100010001100	111000110000
010000111000	000010001100	100001010000
001111100111	000001111100	100110011111
Character"a"	Character"b"	Character"d"

010010111100	000001111110	000000100000
010111100100	000001100001	000001000000
100110100100	000001000001	000001000000
000110100100	000011000001	000001000000
000110100100	000111000001	011110110000
000110100100	001101000001	000010000000
000100100100	110100000110	000100000000
000100100100	000111111000	000100000000
000100100100	000100000000	001100000000
000100100010	000100000000	110010000000
000100100010	000100000000	000100000011
000100100011	000100000000	000111111100
Character"m"	Character"p"	Character"t"

Fig: - 3.1

BINARY IMAGES OF SOME CHARACTERS

Depending upon the contour encountered in each cell, the cells have various pixel occupancy.

The pixel occupancy is defined as the existence of a character boundary segment in each cell, and thereby the number of 1's in the 3 * 3 binary matrix.

Each cell can have a pixel occupancy ranging from 0 to 9. The cell with 0 pixel occupancy is called a NULL CELL and the cell with a pixel occupancy equal to nine is called a FULL CELL. The existence of a Full Cells is rare, where as Null Cells are very common.

Each 3 * 3 binary matrix (a cell), can be transformed into a 9 bit linear word. The transformation is effected by writing the elements of the binary matrix in a particular order, starting with the element a_{11} at the LSB, moving along the periphery of the matrix from left to right and with a_{21} at the MSB. The central element of the matrix a_{22} is treated as the carry bit. Here a_{ij} are elements of the matrix:

$$[\ A\ ] = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

This transformation give rise to a binary word, and is equivalent to a polynomial which can be written as:

$$\sum_{i=1}^{7} \alpha_i * X_i,$$

where α_i = 0 or 1.

The cells are shown in Fig: 3.2. Each matrix can now be represented by an eight bit word and a carry bit (total 9 bits). The cell to byte transformation is unique and has a one - to - one correspondence.

Consider a binary word representing the contour in any cell. When this binary word is rotated, the central bit of the cell, that is the carry bit, remain stationary and all other bits take part in the rotate operation. Rotate left operation is chosen for coding the image in this work. Rotate right operation can also be chosen. However, a complementary set of rules are needed for further processing.

Since all but the central bit take part in the rotate operation, the basic characteristics like relative positions of bits do not change. Eight different words, including the original word, can be generated by this rotate operation from a particular binary word.

This transformation does not reduce or distort the information content in the cell. Also this transformation simplifies operations on the cells, to manipulation on

000 011 111 000	000 001 100 000	000 000 100 000
000 100 000 000	000 010 100 000	000 000 100 000
001 000 000 100	000 101 000 000	000 000 100 000
010 000 000 100	000 101 000 000	000 000 100 000
010 000 000 010	000 110 000 000	000 000 100 000
100 000 000 010	000 100 111 111	000 000 100 000
100 000 000 100	001 100 000 100	000 000 100 000
100 000 000 100	011 100 000 100	011 111 100 000
100 000 001 000	110 010 000 100	110 000 100 000
110 000 011 000	100 010 000 100	111 000 110 000
010 000 111 000	000 010 000 100	100 001 010 000
001 111 100 111	000 001 111 100	100 011 011 111
Character"a"	Character"b"	Character"d"

010 010 111 100	000 001 111 110	000 000 100 000
010 111 100 100	000 001 100 001	000 001 000 000
100 110 100 100	000 001 000 001	000 001 000 000
000 110 100 100	000 001 000 001	000 001 000 000
000 110 100 100	000 111 000 001	011 110 110 000
000 110 100 100	001 011 000 001	000 010 000 000
000 100 100 100	110 100 000 110	000 100 000 000
000 100 100 100	000 111 111 000	000 100 000 000
000 100 100 100	000 100 000 000	001 100 000 000
000 100 100 100	000 100 000 000	110 010 000 000
000 100 100 100	000 100 000 000	001 100 000 011
000 100 100 011	000 100 000 000	001 111 111 100
Character"m"	Character"p"	Character"t"

Fig: - 3.2

CELLS OF THE CHARACTERS OF FIG: - 3.1

BINARY WORD	ROTATIONAL FORMS OF THE BINARY WORD
00 000 110	00 001 100, 00 011 000, 00 110 000, 01 100 000, 11 000 000, 10 000 001, 00 000 110.
00 001 011	00 010 110, 00 101 100, 01 011 000, 10 110 000, 01 100 001, 11 000 010, 10 000 101.
00 011 001	00 110 010, 01 100 100, 11 001 000, 10 010 001, 00 100 011, 01 000 110, 10 001 100.
11 011 000	10 110 001, 01 100 011, 11 000 110, 10 001 101, 00 011 011, 00 110 110, 01 101 100.
10 001 011	00 010 111, 00 101 110, 01 011 100, 10 111 000, 01 110 001, 11 100 010, 11 000 101.
00 110 111	01 101 110, 11 011 100, 10 111 001, 01 110 011, 11 100 110, 11 001 101, 10 011 011.
11 010 101	10 101 011, 01 010 111, 10 101 110, 01 011 101, 10 111 010, 01 110 101, 11 101 010.
01 110 111	11 101 110, 11 011 101, 10 111 011.
11 111 111	All rotational forms are the same.

NOTE: 1) Carry bit is not included in the binary words as they do not take part in the rotate operation.
2) Some binary words have only 3 rotational forms.
3) Some others do not have any rotational forms.
4) Examples of these two types are the last two binary words in this figure.

Fig: 3.3

SOME BINARY WORDS AND THEIR ROTATIONAL FORMS

strings of binary words and no matrix manipulation is necessary.

Further it can be seen that the matrices are sparse matrices. Computer manipulation of sparse matrices is a difficult task and hence the cell-to-byte transformation.

3.5 VOCABULARY

Syntactic methods of recognition call for functional wordnames. To achieve this end, a new vocabulary is developed by ascribing labels to each binary word. If the word names include this feature and are amenable to arithmetic operations, procedures are simplified. It is difficult to choose labels which explicitly describe the contour encountered in every cell. A most suitable labeling procedure, which achieves the above goals is presented in the following.

With eight bits and a carry bit available, the total number of combinations is 512. This means that there is a space with 512 sample points. This space can be called a CIRCULAR VECTOR SPACE, similar to a linear vector space. All these 512 do not exist in real life applications due to the finite nature of the shapes encountered in character images.

Suppose, the number of 1 bits in a cell and consequently in a binary word is 3. The maximum number of variations possible with 3 entries of 1's in the binary word are: $^9C_3$ = 84 combinations. Hence with "n" varying from 0 to 9, the possible combinations in each category are as follows:

Number of combinations with no 1's	=	1
Number of combinations with one 1	=	9
Number of combinations with two 1's	=	36
Number of combinations with three 1,s	=	84
Number of combinations with four 1's	=	126
Number of combinations with five 1's	=	126
Number of combinations with six 1's	=	84
Number of combinations with seven 1's	=	36
Number of combinations with eight 1's	=	9
Number of combinations with nine 1's	=	1
TOTAL COMBINATIONS POSSIBLE	=	512

It can be seen that the distribution of combinations is symmetrical, with the null and full words contributing to one word each, progressively increasing to 126 combinations for binary words with 4 and 5 entries of 1's.

Assume the presence of a cell $C_1 = \begin{matrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{matrix}$

The binary word generated from C_1 is b_1 = 0 00 010 011

Now consider the existence of another cell $C_2 = \begin{matrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0. \end{matrix}$

The binary word generated from C_2 is b_2 = 0 11 000 100.

Now consider a third cell $C_3 = \begin{matrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0. \end{matrix}$

The binary word generated from C_3 is b_3 = 0 10 001 001.

Perform a rotate left operation on b_3.

Then CS_1 = 0 00 010 011 $= \begin{matrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{matrix} = C_1$

If b_1 is rotated left six times, we arrive at a binary word b_6 = 0 11 000 100, which is nothing but b_2 above and one more rotate left operation yields b_3. This means that eight different words are present in one group.

3.5.1 BASIC VECTORS/BASIC POLYNOMIALS

The circular vector space is of the type $\sum \alpha_i x_i$, where

α_i are 0 or 1. Hence the cell C_1 can be represented as a polynomial $P_1 = x^4 + x^1 + x^0$. Here $\alpha 4 = \alpha 1 = \alpha 0 = 1$, and $\alpha 7 = \alpha 6 = \alpha 5 = \alpha 3 = \alpha 2 = 0$. The cell C_2 can be written as polynomial $P_2 = x^7 + x^6 + x^2$, and the cell C_3 can be written as a polynomial $P_3 = x^7 + x^3 + x^0$. In all these transformations the carry bit is not considered.

Now consider the polynomial P_2. The binary word for this polynomial is 11 000 010. If this polynomial is rotated left once, we get the polynomial $X^6+X^5+X^1$. Rotate this polynomial one more time and the resultant polynomial is: $X^5+X^4+X^0$. If the rotate left operation is continued 4 more times we arrive at the polynomial $X^4+X^1+X^0$, which is nothing but the polynomial P_1. Since a modulo 8 operation is adopted P_1 is circularly congruent with P_2.

It was seen earlier that there can be 8 polynomials in each group. All of them are not unique and only one among these 8 is unique. This unique polynomial is termed as a BASIC POLYNOMIAL or BASIC VECTOR.

DEFINITION

A BASIC POLYNOMIAL is that polynomial which cannot be generated by the circular shift of a lower order polynomial.

COROLLARY

A BASIC POLYNOMIAL is that polynomial which has the least binary positional weight and is unique.

Least weight is binary positional value and so has the smallest HEX number.

Consider a polynomial of the form:

$$P_3 = X^7 + X^3 + X^0.$$

Divide this polynomial by X^7. Then,

$$P_1 = X^0 + X^{-4} + X^{-7} .$$

Circular congruent MOD 8 = $X^0 + X^4 + X^1 = P_1$

Thus P_2 is derived from P_1 and between P_1 and P_2. P_1 which has a smaller weight, is the BASIC POLYNOMIAL. It can be established that P_1 is the BASIC POLYNOMIAL in this group as this is the least weighted.

3.5.2 NUMBER OF BASIC VECTORS/BASIC POLYNOMIALS

The number of basic polynomials (words) that can be generated from a binary word of order "i" is given by the formula,

$$N\ B\ P = 2^i - [\ (i-4)\ (\ 2^{i-2} - 1)\ ] \quad \text{------3.1}$$

where N B P is the number of basic polynomials and "i" is the order of the polynomial.

For i = 4, this number is always 2^i.

Equation (3.1) is applicable only in the case of polynomials of order greater than "4", but is less than "7". In this work "i"=7, as the carry bit is considered separately and hence,

$N B P = 2^7 - [(7-4) (2^5-1)] = 128 - 93 = 35.$

A polynomial with all $a_i = 0$ does not exist in space and hence the NULL MATRIX is not accounted for in these 35 BASIC POLYNOMIALS. If the binary word representing the null matrix is also accounted for, then the total number of basic polynomials become 36.

These 36 basic polynomials can generate all the other (256-36) = 221 polynomials by the rotate left operation.

3.5.3 PROPERTIES OF CIRCULAR VECTORS/POLYNOMIALS

3.5.3a TRANSPOSE

If the polynomial is divided by X^8 and sign is neglected the resultant polynomial will be the transpose of the original polynomial.

Consider the polynomial $P_6 = X^6 + X^4 + X^1 + X^0$.

Divide this polynomial with X^8. The resultant polynomial

$P_6 = X^{-6} + X^4 + X^1 + X^0$

$P_6{}^T \bmod 8 = X^{-2} + X^{-4} + X^{-7} + X^{-8}$

$= X^4 + X^2 + X^7 + X^0$

Writing these two polynomials in a matrix form will yield the two matrices:

P_6 =

1	1	0
0	x	0
1	0	1

and

P_6^T =

1	0	1
1	x	0
0	0	1

The above two matrices show that by dividing the polynomial by X^3 generates the transpose of it's matrix.

3.5.3b REVERSE VIDEO

The complement of the original is a reverse video representation of the cell, which is obvious.

3.5.3c SYMMETRY

If a division by X^8 results in the same polynomial then it is symmetric.

Conversely a symmetric cell will have an even polynomial around 3.

Consider the polynomial $X_4 + X_0$. Let this polynomial be named as P_1. Divide this polynomial by X_8, and the resultant polynomial which is named as P_2 is, $X^4 + X^8$. These two polynomials can be represented in the matrix form as:

M_1 =

1	0	0
0	x	0
0	0	1

M_2 =

1	0	0
0	x	0
0	0	1

These two matrices are one and the same and M_2 is generated by rotating P_1 (M_1) four times right.

3.5.3d HEX REPRESENTATION

Let the polynomial $X^4 + X^1 + X^0$ be represented by the binary word 0 0001 0011. The HEX representation of this binary word is 13. A circular left rotation of $X^4 + X^1 + X^0$ generates $X^5 + X^2 + X^1$ and can be represented as a binary word 0 0010 0110 where the HEX representation of the last eight bits is 26. Three more rotations of the binary word yields the binary word 0 0011 0001.Hence HEX 13 = 31. Circular congruence of HEX representations are, 15 = 51, 19 = 91, 1B = B1, 1D = D1,....etc.

In general all HEX word pairs D1 D0 will exhibit this property if D0 is odd and arises out of the symmetry encountered in such cells.

3.5.3e REDUCTION OF HIGHER ORDER POLYNOMIALS TO LOWER ORDER POLYNOMIALS

A higher order polynomial can be reduced to a lower order polynomial by dividing it with X^{k+1} provided circular congruence is maintained, where k is the highest power in the polynomial.

3.5.3f APPROXIMATION OF POLYNOMIALS

An approximation $P_j = P_i$ can be done if and only if three terms match directly or circular congruence MOD 8 is satisfied.

There are other properties also, but only properties which are of interest to this work are listed. Next section uses the above to define "words" for syntactic processing at the classification level.

3.6 LABELING OF WORDS

It was shown earlier that for 8 bit words there are 36 BASIC WORDS. With the addition of the ninth bit in the central position, another set of 36 BASIC WORDS can be generated. Thus there are two sets of Basic Words, one set with carry bit equal to 1 and another set with carry bit equal to 0, making the total number of Basic Words 72. These 72 Basic binary words constitute the basic contours encountered in the type of images treated in this work. The basic contours/basic words are given a NAME or LABEL for ease of further processing. Each cell having a pixel occupancy of "n" is given a word name beginning with "n". To show the differences in contour a second digit is added. A third digit is also added to the word name to denote the number of rotations with respect to the chosen basic word. The scheme of naming the word can be arbitrary, but it is preferable to relate this with the sparseness of the contour.

The words have been written in such a way that they are amenable to octal division. This is done with a purpose. One of the binary representations chosen in Fig:3.3 is the binary word 00 000 111 with a carry bit equal to 0, representing the matrix

1 1 1
0 0 0
0 0 0.

This binary word can be represented in octal format as 007

The binary word has 7 rotational forms 00 001 110, 00 011 100, 00 111 000, 01 110 000, 11 100 000, 11 000 001 10 000 011, all words with carry bit equal to "0". The octal representations of these seven binary words are, 016 034, 070, 160, 340, 301 and 201 respectively. The binary words are rotated subject to the following conditions:

1). The carry bit remains unchanged.

2). The most significant bit flows into the least significant bit.

3) The most significant digit in the octal format can never be more than 3.

The third condition is so, because, the most significant digit has only two bits contributing to its value. Thus the number can be considered circular and modulo 128. One amongst the eight in the group can be considered as a Basic word.

It can be seen from the structure of the binary numbers, that only odd numbers can be BASIC WORDs.However only the least positional weighted binary numbers are considered as basic words. From the constraint that the overflow is form MSB to the LSB, the number of basic numbers is 36, and constitutes 512 different combinations of binary words by including the binary words with carry bit equal to 1, the number of basic numbers become 72 and account for the 512 various binary numbers.

Consider the procedure adopted for the cell to byte transformation. An octal representation is not for simplicity, alone, but for deriving the basic numbers. Consider the cell

0 1 1
0 0 1
0 0 1, whose binary representation is 00 011 110

with carry bit (C) = 0. The octal representation for the word 00 111 100, which corresponds to one left shift of bits with MSB overflowing into the position of the previous LSB. This new binary word has an octal representation of 074. The six other rotational forms of this binary word when the octal representation is used, are: 170, 360, 341, 303, 207 and 017. In this example during the fourth rotation, a 1 bit overflows from the MSD to LSD, and a 341 results when the previous word of 360 is rotated once. Similar is the case with the fifth sixth and seventh rotations. When 017 is rotated left once we arrive at the original word from which the operation started. The left rotate operation is merely doubling the previous octal word and means three things:

1). All octal numbers are not Basic Numbers. This is obvious since an even number is generated from an odd number and hence can not be a Basic Number.

2) All odd numbers are also not Basic Numbers. Examples are words like 303 and 207.

3). Only the first and hence the least (binary weighted) odd number in any group qualifies to become a Basic Number.

The above three conditions make the definition of the basic word a simple one. **A least weighted 8 bit combination in a circular group is the Basic Word.**

The two conditions that a circular number should satisfy to become a BASIC WORD are:

1). It should be ODD.

2). It should have remainder MOD 64 circular.

All these conditions are satisfied by the Basic Numbers mentioned in section 3.5.

Since other words are generated from the BASIC WORD, by the rotate operation, the MSB flows into the LSB, these words are called CIRCULAR WORDS. Table 3.2 lists basic numbers and the corresponding Basic Words.

For the image description these 72 Basic Words form the full complement of the vocabulary. However the following exceptions do exist in real handwriting. Due to the constraint that two 1's are separated by a 0 bit and hence a loss in connectivity, the binary words 01 010 101 with carry bit equal to 0 and 1 are not eligible to become Basic Words. Similarly the binary word 00 000 000 with carry bit equal to 1 is also considered illegal as this is only an isolated bit and its presence does not add to the information content already present. Thus the total number of basic words reduces to 69. Hence the required vocabulary contains only 69 basic words and their rotational forms totaling 499.

Depending on the pixels occupied by the contour in each cell, the binary words can be grouped into 10 classes. These 10 classes are:

1). Basic Word with no 1's in it

2). Basic Word with one 1 in them

3). Basic Words with two 1's in them

4). Basic Words with three 1's in them

5). Basic Words with four 1's in them

6). Basic Words with five 1's in them

7). Basic Words with six 1's in them

8). Basic Words with seven 1's in them

9). Basic Words with eight 1's in them

10). Basic Words with nine 1's in them.

BASIC NUMBER/ BASIC WORD	WORD NAME	BASIC NUMBER/ BASIC WORD	WORD NAME
00 000 000 C=0	000	00 101 111 C=0	510
00 000 001 C=0	100	00 110 111 C=0	520
00 000 011 C=0	200	00 111 011 C=0	530
00 000 101 C=0	210	00 111 101 C=0	540
00 001 001 C=0	220	01 010 111 C=0	550
00 010 001 C=0	230	01 011 011 C=0	560
00 000 001 C=1	240	00 011 111 C=1	570
00 000 111 C=0	300	00 010 111 C=1	580
00 001 011 C=0	310	00 011 011 C=1	590
00 001 101 C=0	320	00 011 111 C=1	5A0
00 010 011 C=0	330	00 100 111 C=1	5B0
00 010 101 C=0	340	00 101 011 C=1	5C0
00 011 001 C=0	350	00 101 101 C=1	5D0
00 100 101 C=0	360	00 110 011 C=1	5E0
00 000 011 C=1	370	00 110 101 C=1	5F0
00 000 101 C=1	380	01 010 101 C=1	5G0
00 001 001 C=1	390	00 111 111 C=0	600
00 010 001 C=1	3A0	01 011 111 C=0	610
00 001 111 C=0	400	01 101 111 C=0	620
00 010 111 C=0	410	01 110 111 C=0	630
00 011 011 C=0	420	00 011 111 C=1	640
00 011 101 C=0	430	00 101 111 C=1	650
00 100 111 C=0	440	00 110 111 C=1	660
00 101 011 C=0	450	00 111 011 C=1	670
00 101 101 C=0	460	00 111 101 C=1	680
00 110 011 C=0	470	01 010 111 C=1	690
00 110 101 C=0	480	01 011 011 C=1	6A0
00 000 111 C=1	490	01 111 111 C=0	700
00 001 011 C=1	4A0	00 111 111 C=1	710
00 001 101 C=1	4B0	01 011 111 C=1	720
00 010 011 C=1	4C0	01 101 111 C=1	730
00 010 101 C=1	4D0	01 110 111 C=1	740
00 011 010 C=1	4E0	11 111 111 C=0	800
00 100 101 C=1	4F0	01 111 111 C=1	810
00 011 111 C=0	500	11 111 111 C=1	900

TABLE - 3.3

BASIC NUMBERS/BASIC WORDS AND THEIR WORD NAMES

In applying this to handwritten character recognition three important aspects are to be considered. These are:

1) For reducing the processing complexity, the label should indicate the pixel occupancy.

2) The label should indicate the contour features.

3) The label should assist in reducing the vocabulary size.

All these aspects can be met using the three digit labeling scheme proposed. The MSD indicates the number of pixels occupied by the contour in the cell, the middle digit indicates the family of the curve and the LSD gives the number of rotations the basic word had to undergo to generate the present word. It may be noted that the LSD and the middle digit together indicate the feature in the contour. The word name starts with a digit 0 to 9. This is so because the MSD indicates the pixel occupancy in any cell. The LSD can be between 0 to 7, An LSD of 0 indicates that the particular word is a Basic Word. The middle digit indicates the family identity and is given according to the seniority of appearance of the Basic Word. The word family is used with the meaning of a group of 1/4/8 words belonging to the same class. Or in other words, a family of words is a particular binary word/basic word and the set of words generated from that particular basic word. For example, the word name given to the binary word whose octal representation is 133 is 560, where as that of 067 is 520, as 067 is a smaller octal number than 133. Word names of all Basic Words are given in Table 3.2.

Now suppose that there is a cell whose binary word representation is 01 110 010 with a carry bit equal to 1. The word name for this can be derived in two ways. One way is to use a table look up procedure where all the Basic Words and their rotational forms are stored. In this case a maximum of 124 comparisons will be necessary, where as the storage requirements are 499, three digit word names. In the second case, the word name is derived by rotating right the current word, checking if it is the least octal representation and finding out from the table of 69 Basic Words, the name of the corresponding Basic Word. Add the number of rotate right operations the present word had to undergo to arrive at the Basic Word. This procedure will require a maximum of 7 rotate right operations, 16 comparisons and one addition. Here rotate right operation is used for arriving at the number of rotations the basic word had to undergo to generate the present word and not for generating a word from a basic word. Also in this case the vocabulary storage requirements is only 69 three digit word names. As the memory requirements and other operations are considerably less in the second operation, this method is considered as the best suited.

In the example under consideration, the present word 01 110 010 with carry bit equal to 1 has an octal representation of 162 and had to be rotated right four times in order to arrive at the minimum octal number, (basic word), that is 047. The word name given to this Basic Word is 5B0 and hence the name of the present word is 5B4. This word name indicate the following:

1). The cell has a pixel occupancy of five.

2). The Basic Word had to be rotated left four times to arrive at the present word.

3). The word has a carry bit equal to 1 which means that the central element is occupied. This is so because of the presence of the family indicator B as the middle digit. In all cases where the family indicator is an alphabet it can be concluded that the carry bit is present in the particular word.

4). THe contour encountered in the cell is a line starting from top left, probably a continuation from NW or W neighbor, passing through the middle and branch off to both left and right at the bottom of the cell. That is, the line continues to both its SE and SW neighbors.

Fig: 3.4 shows a few examples in which various words, their Basic words, the number of rotations the present word had to undergo to arrive at the Basic Word and the word names of the Basic Word and the present word are given.

PRESENT WORD	BASIC WORD	NUMBER OF ROTATIONS	BASIC WORD NAME	PRESENT WORD NAME
00 010 000 C=0	00 000 001 C=0	4	100	104
10 000 010 C=0	00 000 101 C=0	7	210	217
11 100 000 C=0	00 000 111 C=0	5	300	305
01 000 100 C=1	00 010 001 C=1	2	3A0	3A2
00 001 111 C=0	00 001 111 C=0	0	400	400
01 100 011 C=0	00 011 011 C=0	5	420	425
01 001 110 C=0	00 110 011 C=0	1	470	471
00 101 100 C=1	00 001 011 C=1	2	4A0	4A2
11 111 000 C=0	00 011 111 C=0	3	500	503
01 101 110 C=0	00 110 111 C=0	1	520	521
11 101 000 C=1	00 011 101 C=1	3	5A0	5A3
10 001 100 C=1	00 011 101 C=1	7	5A0	5A7
11 011 110 C=0	01 110 111 C=0	2	630	632
11 100 011 C=1	00 011 111 C=1	5	640	645
11 001 110 C=1	00 111 011 C=1	6	670	676
01 111 101 C=1	01 011 111 C=1	2	720	722

Fig: 3.4

SOME BINARY WORDS, THEIR BASIC WORDS, NUMBER OF ROTATIONS IT HAD TO UNDERGO TO ARRIVE AT THE BASIC WORD AND THE WORD NAMES OF BOTH THE BINARY WORD AND THE BASIC WORD

CHAPTER IV - FEATURES AND THEIR SELECTION

4.1 INTRODUCTION

In pattern recognition, feature selection and extraction is performed for dimensionality reduction. Pattern descriptors constituting the lower dimensional representation are referred to as features because of their fundamental role in characterizing the distinguishing properties of pattern classes.

Though the purpose of feature selection and extraction is many fold, the primary justification stems from engineering considerations. The complexity of a classifier and the complexity of its hardware implementation grow rapidly with the number of dimensions of the pattern space. Therefore it is of importance to base decisions only on the most essential discriminatory information. This is conveyed by features.

4.2 PROBLEM FORMULATION

Dimensionality reduction can be achieved in two different ways. One approach is to identify measurements which do not contribute to class separability. The problem is then one of selecting a small sub-set x_j of "d" features,

$$X_j, \ j = 1, 2, 3.....d$$

out of the available D measurements (sensor outputs of Y),

$$Y_k = 1, 2, 3......D.$$

This dimensionality reduction process is referred to as the feature selection. No computation is required during pattern processing in this case. The redundant and irrelevant information are simply ignored.

In the second approach, all the sensor outputs are used to map the useful information into a lower dimensional feature space.

To solve a feature selection or feature extraction problem, three ingredients are to be specified. These are:

1). The feature evaluation criterion,

2). The dimensionality of the feature space and

3). The optimization procedure.

The factors governing the dimensionality of the feature space are common to both the dimensionality reduction approaches. They include, hardware or computational constraints, the peaking phenomenon or permissible information losses.

The problem of feature selection in this work refers to the selection of a set of features. which are a small sub set of the original binary words. These words should have the property of reducing the complexity of further processing.

As in any pattern recognition problem, the feature selection for character recognition is a difficult problem. More so when they are handwritten characters, written without constraints on the writing style. This is so because of the great many styles of writings exhibited by lower case English characters . The feature selection criteria should be so chosen, that variations in writing style do not affect the recognition rates considerably. Even distortions caused intentionally should not cause difficulties at the recognition stage. Fig 4.1 shows a set of normally written characters and Fig:4.2 shows a set of distorted characters.

Feature selection procedures usually depend on the capability for evaluating the effectiveness of any sub set of any given initial set of features, and on realizing an effective strategy for searching for a "best subset" among the subset of features.

In the design of automatic classifiers, the important problems are (1) invention and (2) selection of features.

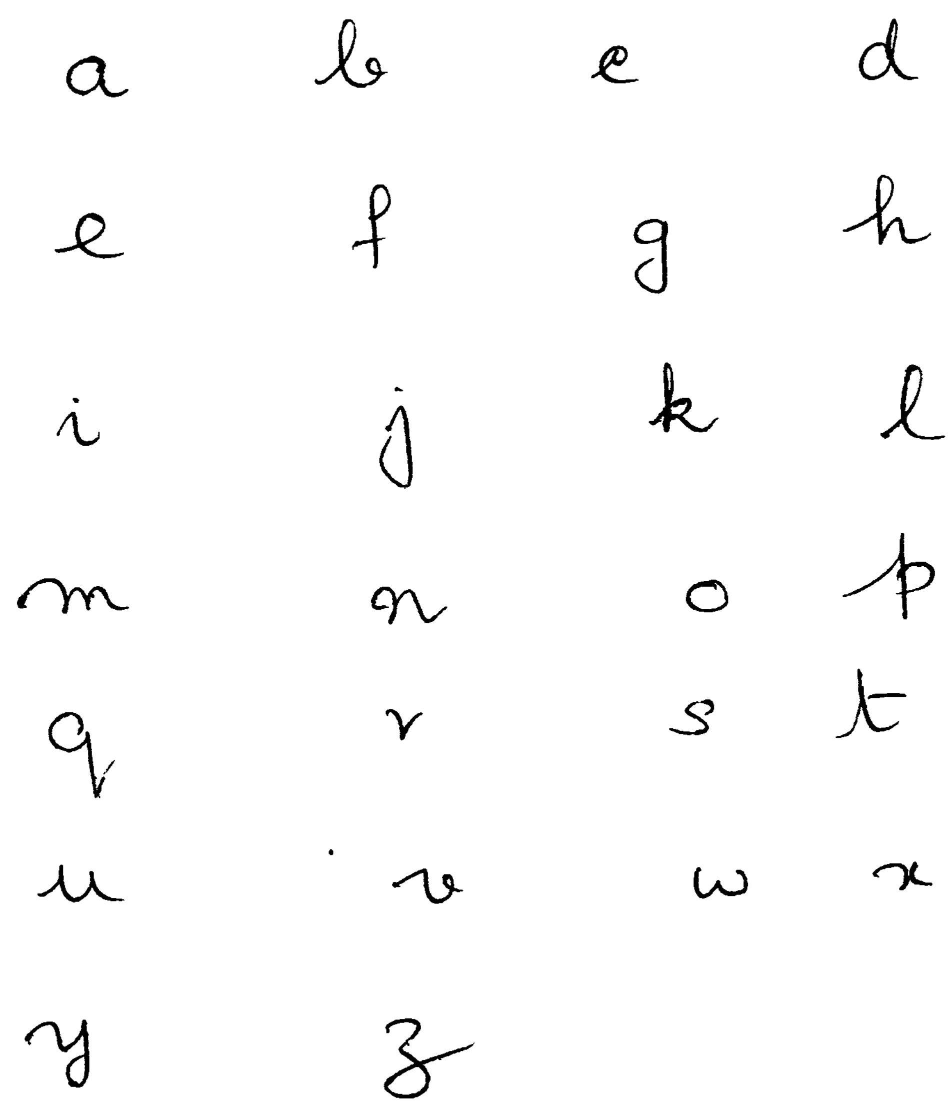


Fig: 4.1

A SET OF NORMALLY HAND WRITTEN CHARACTERS

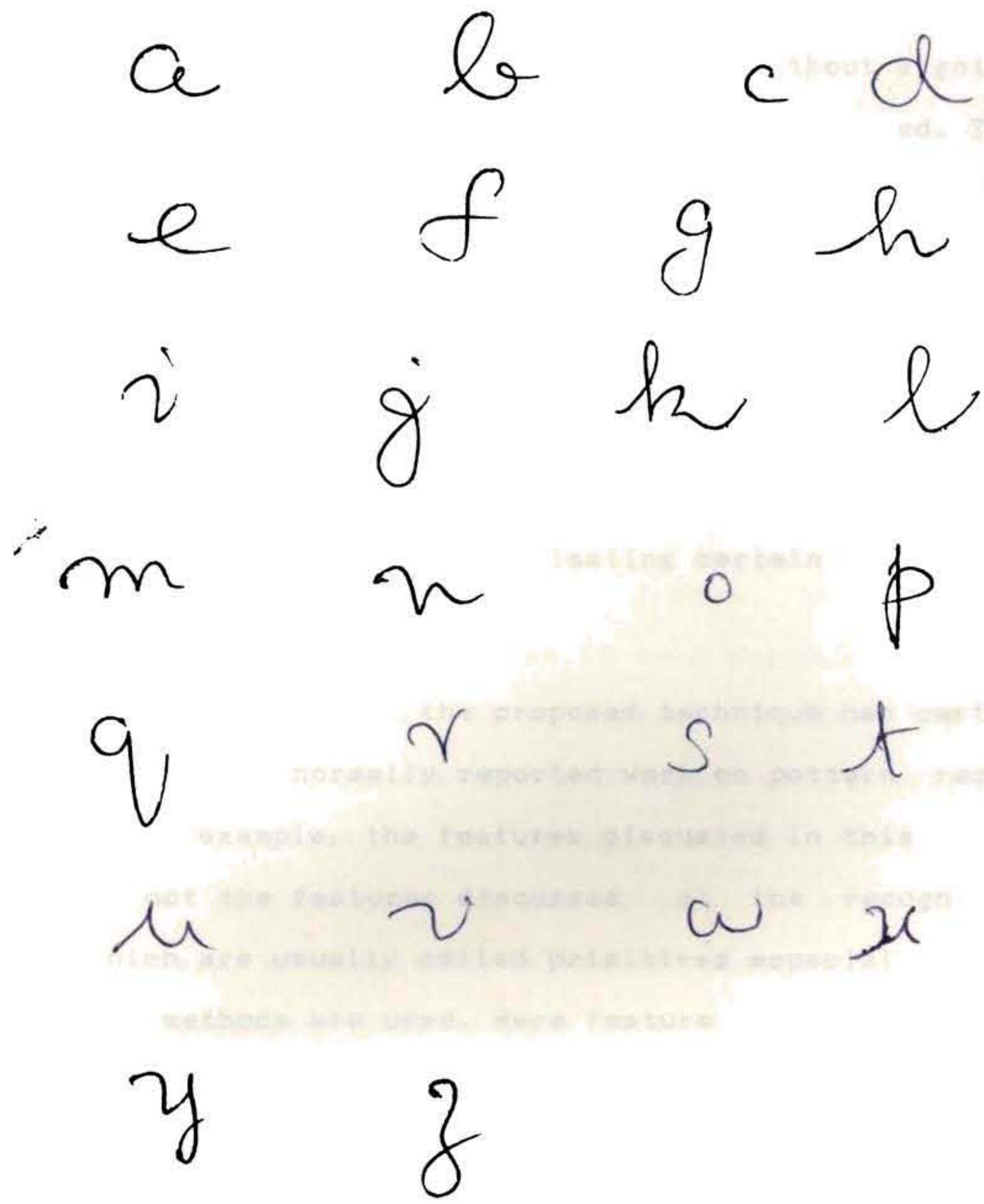


Fig: 4.2

A SET OF DISTORTED HAND WRITTEN CHARACTERS

In this chapter a procedure for selecting "n" features from a set of "d" features, ["n" << "d"], without significantly degrading the system performance is presented. This procedure is named as FEATURE SELECTION in this work. The problem is therefore:

1) To define a set of "Features" for the class of images which are made up of contours, and
2) For a given image select the applicable features at a low hardware and software cost. To realize this a small sacrifice in terms of approximating certain contours is worth a try.

The approach being new, the proposed technique has certain differences with normally reported work on pattern recognition. For example, the features discussed in this chapter are not the features discussed at the recognition stage which are usually called primitives especially when syntactic methods are used. Here feature selection is done at this stage to reduce data at the filter output.

In [KSFU], K. S. Fu states that only a small number of features are required to characterize each pattern. This also augments the selection of a small number of features which are used at the filtering and subsequent stages.

4.3 FEATURES FOR THE CLASS OF IMAGES TREATED

The cells defined in the previous chapter form the basic defining area for features in this work. With nine bits there can be as many as 512 different part contours. But because of the fact that the image is divided into 16 cells, only 16 from among the 512 part contours can exist in a character image. The central bits in all the cells play an important role in the feature definition and selection. Consider the rotate left operation on any typical cell. The rotation itself does not change the shape of the contour, where as it only changes the orientation of the contour. As an example consider the cell whose binary representation is 00001011 with a carry bit equal to 1 [Fig:4.3]. The contour represented by this word, when rotated does not get deformed or altered in its shape where as its orientation is changed with every rotate operation. This is very clear from Fig:4.3 where all the rotational forms of the word/cell are given. Eight rotational forms of some other words are also given in the figure. It should be remembered that the pixels and hence the bits in each binary word represent the presence or absence of the contour.

As each of the 512 binary words is generated from one of

111	011	001	000	000	000	100	110
000	001	001	001	000	100	100	100
000	000	001	011	111	110	100	000
300	301	302	303	304	305	306	307
110	011	001	000	100	110	011	101
001	000	001	101	100	000	100	101
001	011	110	101	011	110	100	000
420	421	422	423	424	425	426	427
110	011	001	000	000	100	010	101
011	010	011	011	110	010	110	110
000	001	010	101	011	110	100	000
4A0	4A1	4A2	4A3	4A4	4A5	4A6	4A7
111	011	001	100	010	101	110	111
001	001	101	001	100	100	101	100
010	101	011	111	111	110	100	001
510	511	512	513	514	515	516	517
111	011	001	100	010	101	110	111
011	011	111	011	110	110	111	110
010	101	011	111	111	110	100	001
650	651	652	653	654	655	656	657
111	011	101	110	011	101	110	111
001	101	101	001	100	101	101	100
110	101	011	111	111	110	101	011
620	621	622	623	624	625	626	627
111	011	101	110				
000	101	101	101				
111	110	101	011				
630	631	632	633				

NOTE:- Only three rotational forms for the Basic Word 630

Fig: 4.3

ROTATIONAL FORMS OF SOME WORDS IN THEIR CELL FORMAT

the 72 basic words, the basic words are considered as features, where as their rotational forms numbering 440 are not considered as features. The definition of the basic word is made to aid in reducing the complexity of the recognition algorithm. Hence the definition of features also become easier. During the rotate operation of the binary word, the smallest octal representation in the group is generated. This binary word represents the contour of the shape present in the cell. As the rotate operation only changes the orientation of the curve, the Basic Words qualify to become FEATUREs and hence they are defined as FEATURES. Only the 69 basic words are considered eligible to become FEATURES.

While basic words define features in a region of the image, the image itself is defined in terms of the inter-relations between 16 such basic words or their rotational forms, each representing a cell. Thus features by them-selves will not be able to define an image. Hence to define images the features defined earlier are selected in toto and a small set of relational features are defined. For each image, a cell and a relational auxiliary feature are selected. This necessitates the definition of another set of features, the auxiliary features, which define the connectivity between the cells. The last digit of each word, being the number of rotations the basic word had to undergo to generate the present word, represents the change in orientation exhibited by the contour from the basic word. In the character image this indicates the change in orientation of the part contour. As such there could be 8 possible tendencies. This serves as an auxiliary feature and is called a TENDENCY. This auxiliary feature TENDENCY, permits the system to determine the nature of the contour, like weather it goes to the next neighbor and in such a case to which neighbor ....etc. The last two digits taken together indicate the continuity of the contour. This auxiliary feature is defined as a TENSE. Theoretically a maximum of 64 TENSE features can exist. However only 4 of these are important in real life applications and are selected. These are:

1) The Past Tense

2) The Past Continuous Tense

3) The Future Tense and

4) The Future Continuous Tense.

These four auxiliary features determine whether the contour starts from a previous neighbor or from the cell itself or is a part coming from/going to a future cell ....etc.

The two features TENDENCY and TENSE are used in arriving at the appropriate substitute word at the FILTERING stage. The most important aspect of the definitions made above is the fact that these types of features can be extracted by mere arithmetic operations on the binary image. The simplest of grammars for the image have to necessarily use some sort of tests on continuity of contours. To enable this a subset of features is selected in conjunction with the four auxiliary features. Another situation may arise where selection is complicated by the presence of more than one feature. For example consider the binary word 01 110 111. This word is a basic word named 630. The cell configuration of this binary word is,

1 1 1
0 0 0
1 1 1. This cell shows that there are two possible candidates to be selected as features, the horizontal line segment at the top and segment of the horizontal line at the bottom of the cell. Such problems are resolved by splitting the word arithmetically as sum or difference of two or more words. Splitting into more than two words become necessary when the word contains more than six image points. By noting the context it can be seen that, the most appropriate word can be retained and the others can be discarded. This is where the

auxiliary features become useful. The decision can be made with reference to the auxiliary feature exhibited by the previous cell, or the decision can be postponed until the neighbors are resolved. This may prove in some cases to be non converging. To ensure convergence a hierarchy is adopted, west to east and north to south in that order. While adopting the procedure, it can be seen that the basic features and the auxiliary features are all preserved. The procedure for these replacements, called FILTERING is fairly simple and involve only simple substitutions. The substitution rules are presented in the next chapter.

The two auxiliary features, TENDENCY and TENSE, are defined for the explicit purpose of arriving at the suitable replacement words at the filtering stage.

CHAPTER V - FILTERING

5.1 INTRODUCTION

Many authors describe the term FILTER as a device in the form of a piece of physical hardware or computer software that is applied to a set of noisy or noise free data in order to extract information about the prescribed quantity of interest. A filter is to perform three basic functions of information processing [SIMO].

A FILTER in its most general sense is a device or system that alters in a prescribed way, the input that passes through it. That is any FILTER converts inputs into outputs in such a fashion that certain desirable features of the inputs are retained in the outputs while undesirable or unwanted features are eliminated. In this work the unwanted features are the 1's and 0's in any cell where redundant information is available. Analysis of this "FILTERING" is possible only on a heuristic basis.

5.2 FILTERING PRELIMINARIES

The objective of any pattern recognition system is to recognize and describe patterns correctly. The description can be in terms of two or three dimensional patterns and their relationships. But relevant information and information that is not needed at that particular time (noise) are also present in a picture description. Once the problem of noise and relevant information is understood clearly, the design of a filter for either of them is easy. In any picture description, the relevant information is that which is relevant to the observer at a given time and context in order to accomplish a given task. The task could be image enhancement, target finding or a complex scene description. Hence relevant information is time and context dependent.

Noise is anything in the signal which is not relevant information. Consequently any thing which is not of current interest is noise. This definition of noise is dynamic and hence any signal component which is noise at one moment or in a particular context need not necessarily be so in another context.

In the context of the present work the term FILTER is used in the following sense. FILTERING is used to achieve data reduction. The images in their binarized form contain a large amount of redundant information. These come in the form of ornaments, serifs...etc, attributable to the way and style of writing of the individuals and these are to be removed for ease of further processing. Removal of these do not alter the basic shape of the character. This redundant information in the character is treated as noise in this case and needs to be removed. Also the presence of such redundant information makes further processing complex and hence the data reduction technique called FILTERING is to be adopted. This data reduction is achieved by the suitable substitution of one word with another word and is a new form of filtering.

5.3 DEFINITIONS

The following terms which occur often in this work will have the following meaning.

ORDER OF THE WORD

This is defined as the pixel occupancy rate of the cell and is denoted by the M S D of the word name.

HIGHER ORDER WORD

A word name which starts with 4 and above. The Higher Order Words have a pixel occupancy rate of 4 or more in their cells and consequently in the binary word.

LOWER ORDER WORDS

Any word name which starts with 0,1,2 or 3 is defined as a Lower Order Word. These lower order words constitute the majority of cells in any given binary character image.

FILTERING

Filtering, the data reduction technique, refers to the process of replacing a particular word with a suitable word of the 3XX category. Filtering helps in reducing the complexity of the information content in the image.

The output set of words/cells of an image is a subset of the input set of words in this filtering, similar to Median Filtering, Stack Filtering ...etc. Both these sets are finite.

LEVEL OF FILTERING

Filtering is done in three levels or steps. Each step of Filtering is referred to as the Level of Filtering. Eg. First level of Filtering, Second level of Filtering and Third level of Filtering.

SPLITTING AND MERGING OF WORDS

Splitting is the process of separating a binary word into two parts and adding 0's to fill up the vacant places. Consider for example, the binary word 01 110 111 with carry bit equal to 0. This binary word can be split into two binary words 01 110 000 and 00 000 111. The carry bit does not have any significance in the splitting operation. These two binary words have the same kind of orientation, horizontal lines, one at the bottom and the other at the top respectively. Splitting is done only in the case of higher order words. This is done to assist in Filtering. Fig: 5.1(a) shows a few examples.

Merging is the process of combining two binary words to make them a single word. The process of merging can be explained as follows. Consider the two binary words 00 000 011 and 00 000 100, with the carry bit equal to 0 or 1. These two binary words can be combined to form the binary word 00 000 111, with carry bit equal to 0 or 1 as the case may be. That is, the binary words whose wordnames are 200 and 103 can be combined to form the binary word 300. Carry bit is not considered in this particular example. Here too there is no significance in the value of the carry bit as in the case of Splitting. Normally merging is done only in the case of words of order 2 or 1. Some examples of merging are shown in Fig: 5.1(b).

ORIGINAL WORD	SPLIT UP INTO WORD 1	WORD 2
00 011 111	00 011 100	00 000 011
01 110 100	01 110 000	00 000 100
01 101 100	01 101 000	00 000 100
11 001 110	00 001 110	11 000 000
01 110 011	01 110 000	00 000 011

Fig: - 5.1(a)
EXAMPLES OF SPLITTING OF BINARY WORDS

RESULTANT WORD	MERGED USING WORD 1	WORD 2
00 111 000	00 110 000	00 001 000
00 111 000	00 101 000	00 010 000
00 110 100	00 100 100	00 010 000
00 000 111	00 000 011	00 000 100
11 100 000	10 100 000	01 000 000

Fig: - 5.1(b)

EXAMPLES OF MERGING OF TWO BINARY WORDS

PRESERVATION MEASURE

This is an eight bit word computed for both the original and replacement words to ascertain the suitability of the replacement word.

5.4 FILTERING PROCEDURE

The encoded binary image is scanned. The scanning is done from left to right. All word names which start with a 3 or 0 are retained. Word names which start with a 1 are discarded and in their place a word 000 is substituted. This operation does not result in loss of information as a word of order 1 is an isolated bit or in the worst case a neighbor from which the next cell gets its contour started or may be the single bit in a cell where the contour gets terminated. In any case there is no loss of information as information in only one isolated pixel is lost when compared to its neighbors where more information is readily available for further processing. The condition when this is done ensures that a separation has occurred in data of a reasonably long contour due to instrumentation and not logic.

Words of order 2 are merged with a word of order 1 resulting in words of order 3. In this case the addition of information is to the extent of one bit in an eight bit word. This procedure does not distort the shape as this can easily be achieved by borrowing a single bit from a neighbor. An addition of a bit only increase the information content slightly and hence does not impair recognizability of the shape in the cell.

After the removal of isolated extensions and/or addition of tail where needed, and depending on the image description collected, all higher order words, that is, words having a pixel occupancy of 4 or more, are replaced with a word of order 3. This is done as follows. The higher order word is split into two or three binary words, one of which is of order 3. This order 3 word is compared with the original word, and the Preservation Measure is computed. If the Preservation Measure is high enough, then this word of order 3 is accepted as the substitute word. However a simplification is adopted in the implemenation here in this work. A large number of substitution words have already been generated. These words are acceptable as substitutes, as they readily agree with the thresholds of the preservation measure. Context also plays an important role and as explained in the preceding, in addition to table look up, ambiguity resolution call for context based choices if the preservation measure shows more than one choice for replacement.

In [SUE4], Suen comments as follows: " Even with a window size of 3 * 3, there already exists 512 possible configurations. To reduce this number, similar configurations can be grouped into one single category". He also mentions about 5 configurations which are similar and can be grouped into one class of a SLANT LINE. In this work also similar configurations are grouped into one category. This grouping has effectively reduced the number of words to be considered for further processing. The original words and their substitutions are given in TABLE 5.1.

At this stage the image is reduced into a description where words of order 3 or 0 exist. By adopting the above procedure, the character images are represented using the rotational forms of 12 basic words themselves. These are the null word and the 11 basic words of order 3, which are named as 300, 310, 320, 330, 340, 350, 360, 370, 380, 390, and 3A0. A statistical analysis on a sample set of 572 characters shows that the information loss by adopting the the above procedure is very small. It can be seen from Table - 5.2, that the 12 basic words and their rotational derivatives contribute to 67% of the total cells occupied.

TABLE - 5.1(a)

FILTERING RULES STAGE I

Replace the original word with the corresponding substitute given below.

ORIGINAL	REPLACEMENT	
000	000	
100	000	
20X	30X	if X is even
	30(X+1)	if X is odd
21X	30X	cyclic 8
22X	32X	cyclic 8
23X	3AX	cyclic 4
24X	3AX	cyclic 4
3XX	3XX	
40X	30X	if X is even
	30(x+1)	if X is odd
41X	30X	if X is even
	30(X+1)	if X is odd
42X	30(X+2)	if X is even
	30(X+3)	if X is odd
43X	30(X+2)	if X is even
	30(X+3)	if X is odd
44X	30X	if X is even
	30(X+1)	if X is odd
45X	30X	if X is even
	30(X+2)	if X is even
46X	30(X+1)	if X is odd
47X	30(X+2)	if X is even
	30(X+3)	if X is odd
48X	30(X+3)	if X is even
	30(X+4)	if X is odd
49X	30X	if X is even
	30(X-1)	if X is odd
4AX	3AX	cyclic in 4
4BX	3AX	cyclic in 4
4CX	3AX	cyclic in 4
4DX	3AX	cyclic in 4
4EX	3AX	cyclic in 4
4FX	3A(X+1)	cyclic in 4
4GX	3AX	cyclic in 4
50X	30(X+2)	if X is even
	30(X+1)	if X is odd
51X	30X	if X is even
52X	30X	if X is even
	30(X+1)	if X is odd

53X	30(X+4)	if X is even
	30(X+3)	if X is odd
54X	30(X+2)	if X is even
	30(X+3)	if X is odd
55X	3AX	cyclic in 4
56X	30(X+2)	cyclic in 8
57X	30X	if X is even
	30(X+1)	if X is odd
58X	3AX	cyclic in 4
59X	3AX	cyclic in 4
5AX	3AX	cyclic in 4
5BX	3AX	cyclic in 4
5CX	3A(X+1)	cyclic in 4
5DX	30(X+2)	if X is even
	30(X+1)	if X is odd
5EX	3AX	cyclic in 4
5FX	3AX	cyclic in 4
5GX	3AX	cyclic in 4
60X	30(X+2)	if X is even
	30(X+3)	if X is odd
61X	30X	if X is even
	30(X+1)	if X is odd
62X	30X	if X is even
	30(X+2)	if X is odd
63X	30X	if X is even
	30(X+1)	if X is odd
64X	3AX	cyclic in 4
65X	3A(X+1)	cyclic in 4
66X	3Ax	cyclic in 4
67X	3AX	cyclic in 4
68X	3AX	cyclic in 4
69X	3AX	cyclic in 4
6AX	3AX	cyclic in 4
70X	30X	if X is even
	30(X+1)	if X is odd
71X	3AX	cyclic in 4
72X	3AX	cyclic in 4
73X	30X	if X is even
	30(X+1)	if X is odd
74X	3AX	cyclic in 4
800	300	
900	300	

TABLE - 5.1(b)

FILTERING RULES STAGE II

ORIGINAL WORD	REPLACEMENT WORD	
30X	30X	if X is even
	30(X+1)	if X is odd
31X	30X	if X is even
	30(X-1)	if X is odd
32X	30X	if X is even
	30(X+1)	if X is odd
33X	3AX	cyclic in 4
34X	3AX	cyclic in 4
35X	3AX	cyclic in 4
36X	30X	if X is even
	30(X+1)	if X is odd
37X	3AX	cyclic in 4
38X	30X	if X is even
	30(X+1)	if X is odd
39X	3AX	cyclic in 4
3AX	3AX	

TABLE - 5.1(C)

FILTERING RULES STAGE III

ORIGINAL WORD	REPLACEMENT WORD
300	3A3
302	3A1
304	3A3
306	3A1
3AX	3AX

TABLE - 5.1

REPLACEMENT RULES FOR THE THREE STAGES OF FILTERING

CHARACTER	PRESENCE OF VARIOUS GROUPS IN THE SAMPLE SET									OCCUPANCY PERCENTAGE	
	1	2	3	4	5	6	7	8	9	CELL	PIXEL
a	23	21	148	52	20	2	-	-	-	75.6	26.2
b	22	33	70	46	38	14	2	-	-	63.9	19.2
c	24	32	99	33	30	12		-	-	66.1	23.4
d	17	13	106	43	37	3	-	-	-	62.2	23.3
e	22	32	113	37	25	5	2	-	-	67.1	23.4
f	32	36	66	34	30	23	9	-	-	65.3	24.9
g	16	35	103	52	34	7	7	2	-	72.7	27.8
h	24	19	90	29	45	9	2	1	-	62.2	23.4
i	21	21	97	25	16	2	-	-	-	51.7	17.3
j	25	31	83	32	29	10	3	-	-	60.3	21.6
k	28	32	104	46	31	15	2	-	-	73.3	26.7
l	20	26	79	37	16	10	6	-	-	55.4	20.4
m	10	24	173	31	38	18	9	1	1	86.6	33.4
n	21	33	143	18	31	8	4	1	-	73.6	24.7
o	19	27	143	30	19	12	3	-	-	71.9	25.6
p	25	21	114	34	28	3	4	-	-	65.0	23.1
q	25	33	99	41	30	8	-	-	-	67.0	23.7
r	29	30	103	32	18	7	1	-	-	62.5	21.0
s	24	25	139	47	29	14	6	-	-	80.7	29.9
t	38	39	91	27	25	11	-	-	-	65.6	21.7
u	25	35	140	36	21	9	-	-	-	76.6	25.9
v	37	40	111	35	25	5	-	1	-	71.6	23.5
w	31	30	137	28	36	13	4	-	-	79.3	28.4
x	27	20	116	46	22	9	1	-	-	68.5	24.3
y	29	25	77	52	36	17	1	-	-	67.3	25.5
z	36	43	75	46	51	8	6	-	-	75.3	27.7
TOTAL	653	756	2819	967	760	254	75	7	1		

TOTAL NUMBER OF CHARACTERS = 572
TOTAL NUMBER OF PARAGRAPHS = 572
TOTAL NUMBER OF WORDS = 9152
TOTAL NUMBER OF CELLS = 9152
TOTAL NUMBER OF PIXELS = 82386
TOTAL CELL OCCUPANCY = 6292
PERCENTAGE CELL OCCUPANCY = 68.75%
TOTAL PIXEL OCCUPANCY = 20404
PERCENTAGE PIXEL OCCUPANCY = 24.77%

TABLE - 5.2
STATISTICAL DATA ON A SAMPLE SET

The words of order 4 contribute to 19% and all the other combinations together contribute to only 14%. Pixel occupancy wise these percentages are:

Words of order '0' and '3' together 52%

Words of order '4' 19%

Words of order '1', '2', '5', '6', '7',
'8'& '9' together 29%

These two analysis show that the adoption of such a data reduction on any particular cell does not drastically reduce the information content in the whole image. The percentage loss of information content, cell occupancy ratio wise is only 14%, where as pixel occupancy wise this is again only 29% at the worst.

A further approximation in shape is done on the image described by the words of order 3. At this level of filtering, all basic words of order 3 and their rotational forms are further reduced to 8 combinations of 2 basic words and their four rotational forms . In this curved lines are approximated to straight lines. This approximation is effected after considering the tendency features exhibited by the word in question. Context can also be used in arriving at the substitute word. However after analyzing a large set of samples a substitution table is generated and this is used in selecting the appropriate word. Table 5.1 gives the correct substitution words.

In [HANA] Hanakata argues that straight line approximation will not destroy the basic recognizable shape of the character. This statement of Hanakata is very valid in the case of binary images of lower case English language characters also. That is approximating the character image using 16 straight line segments as is done in this work, will not destroy the recognizability of the character. Ambiguity and resolution of this is illustrated in the following example. Consider the following example. The matrix representing the binary word, 00 001 110 with carry bit = 0 is:

```
0 1 1
0 0 1
0 0 0.
```

The word name given to the binary word is 301. The two possible substitutions for this binary word are:

```
1 1 1          0 0 1
0 0 0    or    0 0 1
0 0 0          0 0 1.
```

The tendency feature exhibited by the curve in the original cell is to start from the middle of the top row and proceed down to the right half. Under this situation, sub-situation with the word 302 is more appropriate than substituting with the word 300.

Now assume another situation, in which the north neighbor of this cell is one whose word name is 307. The byte transformation of this cell is 11 100 000 with the carry bit equal to 0. It is obvious that the tendency of this curve is to start from the top, go down and continue to the next cell. But at the second level of filtration, these words will be given the word names 302 and 306 and hence these cells will be separated from each other by one column. That is, in this particular case, the line segments, two vertical line segments each in the cell will be in the right half in the case of the first cell and in the left half in the case of the second cell. This separation is however removed during the next level of filtering.

A further shape approximation is done at the third level of filtering. This is achieved by discarding the words in the group 30X and replacing them with the words in the group 3AX. The words discarded are the basic word 300 and its three rotational forms, 302, 304 and 306. The words 300 and 304 are two horizontal lines and the words 302 and 306 are two vertical lines. The words 300 and 304 lie at the top and bottom of the cell and the words 306 and 302 lie at left and right hand sides of the cell. The words 300 and 304 are replaced with the word 3A3 where as the words 302 and 306 are replaced with the word 3A1. This replacement only alters the position of the line segment but does not alter the information content. Therefore the shape resident in the cell after the second stage of filtering is retained. Moreover, this substitution provides continuity wherever broken line segments are caused by the use of the second level of filtering, and filling up takes place in places where there were possible gaps. Although at the third level of filtering only a substitution takes place and no data reduction is effected, the name filtering is retained for the sake of uniformity. At the end of the third level of filtering, straight line approximated image, which is slightly shifted to the centre of the cell is available. The shifting takes place generally in cells and not in the image as a whole. Only in very few cases total image shifting takes place. However such shiftings do not destroy the basic recognizability of the image and recognition rates are not adversely affected.

5.5 EVALUATION OF THE PRESERVATION MEASURE

Broadly speaking a comparison of the original cell and the replacement cell are made and if a correspondence exists between the two, then it can be said that the filtered image retains the shape features of the original image. The replacement words in Table 5.1 are chosen based on this principle. All replacement words are compared with the original words and preservation measure of each word is computed and only the best suited word is considered as the replacement word. In the case of multiple choice words preservation measure of all the choices are computed and again only the best choice is selected. A few examples on the computation of the Preservation Measure at the cell level is given in Fig: 5.2. Computation of paragraph level Preservation Measure is similar, in the sense that, a 16 bit PM word is generated from the cell correspondence and compared with a 16 bit binary word comprising of all 1's. As a result of this comparison if a 1 results then it can be ascertained that the paragraph retains all the shape features needed for the recognition of the character. How ever this may not be true in all practical cases. A threshold of 75% correspondence is considered sufficient to establish the character shape in the filtered image

COMPUTATION OF THE PRESERVATION MEASURE - CELL LEVEL

ORIGINAL/ REPLACEMENT WORD NAME	CARRY BIT	BINARY REP-RESENTATION	PRESERVATION BYTE BITS P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7	REMARKS
402	0	00 111 100	0	1	1	1	0	1	0	1	7 out of 8 bits
303	0	00 111 000									agree. Threshold
PM MASK			0	1	0	1	0	1	0	1	satisfied.
485	0	10 100 110	0	1	1	1	0	1	0	1	7 out of 8 bits
300	0	00 000 111									agree. Threshold
PM MASK			0	1	0	1	0	1	0	1	satisfied.
532	0	11 101 100	0	0	1	1	0	1	0	1	6 out of 8 bits
306	0	11 000 001									agree. Threshold
PM MASK			0	1	0	1	0	1	0	1	satisfied.
5B0	1	00 100 111	0	0	1	1	0	1	0	1	6 out of 8 bits
3A0	1	00 010 001									agree. Threshold
PM MASK			0	1	0	1	0	1	0	1	satisfied.
633	0	10 111 011	0	0	1	1	0	1	0	1	6 out of 8 bits
303	0	00 111 000									agree. Threshold
PM MASK			0	1	0	1	0	1	0	1	satisfied.
743	1	10 111 011	0	0	1	1	0	1	0	1	6 out of 8 bits
3A3	1	10 001 000									agree. Thresho
PM MASK			0	1	0	1	0	1	0	1	satisfied.
345	0	10 100 010	0	0	0	0	0	1	0	1	6 out of 8 bits
3A2	1	01 000 100									agree. Threshold
PM MASK			0	1	0	1	0	1	0	1	satisfied.
367	0	10 100 010	0	0	0	0	0	1	0	1	6 out of 8 bits
300	0	00 000 111									agree. Threshold
PM MASK			0	1	0	1	0	1	0	1	satisfied.
300	0	00 000 111	0	0	0	0	0	1	0	1	6 out of 8 bits
3A3	1	10 001 000									agree. Threshold
PM MASK			0	1	0	1	0	1	0	1	satisfied.

302	0	11 000 001	0	0	0	0	0	1	0	1	6 out of 8 bits
3A1	1	10 001 000									agree. Threshold
PM MASK			0	1	0	1	0	1	0	1	satisfied.
304	0	01 110 000	0	0	0	0	0	1	0	1	6 out of 8 bits
3A3	1	10 001 000									agree. Threshold
PM MASK			0	1	0	1	0	1	0	1	satisfied.
306	0	11 000 001	0	0	0	0	0	1	0	1	6 out of 8 bits
3A1	1	00 100 010									agree. Threshold
PM MASK			0	1	0	1	0	1	0	1	satisfied.

Fig: – 5.2

COMPUTATION OF PRESERVATION MEASURE AT CELL LEVEL

that is if 12 bit positions in the PM word and the 16 bit mask have correspondence, then it can be assumed that the filtered image retains sufficient characteristics for recognizability.

Once the cell level correspondence is ascertained, the paragraph level correspondence is also computed. If correspondence exists at this level also, it can be concluded that the filtered image retains shape features exhibited by the original image.

As was proved earlier, the sixteen cells represent an image completely. Hence the correspondence is a necessary and sufficient condition to conclude that the filtered image retains sufficient information for further processing. The threshold fixed for cell correspondence is 6 bit positions and 12 bit positions for the paragraph. These figures have been arrived at by trail and error after considering a number of examples.

The preservation measure defines various sub measures to determine the retention of shape features. This measure byte is made up of seven submeasures. The MSB is a dummy bit. The submeasures are defined severely or between words. The submeasures are listed below.

5.5.1 CELL OCCUPANCY MEASURE - P1

The cell occupancy measure is 1 if the pixel occupancy is 3 and 4 in the replacement and original words.

5.5.2 REDUNDANCY MEASURE - P2

This is 1 if the pixel occupancy is 4 or more in the original word and 0 in the case of the replacement word.

5.5.3 CONTOUR RETENTION MEASURE - P3

This is 1 if two or three occupied bit positions of the original and replacement words are the same.

5.5.4 TRANSPOSE PRESERVENT MEASURE - P4

This is 1 if the replacement words matrix is the transpose of the matrix of the original word.

5.5.5 SHAPE WEIGHT MEASURE - P5

This is 1 if the ranks of the original and the replacement matrices are the same.

5.5.6 CONTOUR SYMMETRY MEASURE - P6

This is 1 if the replacement word represents a matrix whose row or column is interchanged when compared with the original word's matrix.

5.5.7 CONTOUR WEIGHT MEASURE - P7

This is 1 if at least three bits in both the original and replacement words agree. This measure indicates that the dominant portion of the contour is retained by the replacement word.

A preservation measure byte (PMB) is made up of these seven bits with P1 at LSB, P2 at the second bit position and so on and P7 at the MSB-1 position. For the sake of completeness a '0' is added at the MSB position. This PMB is compared with a symmetric mask of 01010101. If the chosen words and hence the replaced paragraph retain sufficient information, that is if Mask - PM = 1, the computation will yield a 1. However, if correspondence is achieved in at least six bit positions, it can be concluded that the replacement words retain sufficient information.

The bit positions of the seven sub measures have a significance. They have to be necessarily in the order prescribed. Otherwise the comparison with the symmetric mask will not yield a 1 as the result.

5.6 FILTERING STEPS / PROCEDURE

1). If the word name starts with a 0 or 1, then name the word as 000

2). If the word name starts with a 2, then rename the word with a 3XX word chosen from the TABLE 5.1 (a) as the replacement word.

3). If the word name starts with a 3, then retain this word.

4). If the word name starts with a 4 or above, then choose an appropriate word name for replacing this word from the look up Table - 5.2(a).

5). See if all the sixteen words have been named properly Proceed to stage two of the filtering procedure

6). Retain all the words 000, 300 302, 304, 306, 3A0, 3A1 3A2 and 3A3.

7). Replace all the other words with suitable words using the look up Table - 5.2(b).

8). See if all the sixteen words are suitably named.

Once the step 8 is also completed successfully, then proceed to the third stage of filtration.

9). Retain all the words 000, 3A0, 3A1, 3A2, and 3A3.

10). Replace all the 300 and 304 words with the word 3A3.

11). Replace all the 302 and 306 words with the word 3A3.

12). Check and see that all the words are appropriately named.

Once all the above twelve steps are complete then the image is ready for further processing like recognition.

5.7 FILTERING AND RETENTION OF FEATURES AND IT'S RELATION WITH PRIMITIVES

The procedure of filtering produces invariants which will ultimately replace the contours in each cell of the image. Thus more than one type of cell may be replaced by a lower order cell, which is representative of the essential irreducible contour present in the cell. This many to one correspondence operation is equivalent to defining an invariant or a feature present in the cell with a difference in the fact that the orientation of the contour is also indicated by the replacement cell, particularly at the end of the second stage of filtering. Fig: 5.2 shows a set of original cells and their replacement cells to illustrate the point. It is shown here that the replacement cell invariably retains the feature present in the cell as long as a good preservation measure is used. The purpose is two fold. The first is the fact that generally features are defined and extracted to process images which contain a large number of different objects. For images which mainly are small in number and are made up of identifiable and repeatedly occurring contours or parts of contours of fixed shapes, processing is easier if primitives are defined, that is contours of fixed shape independent of the image handled. The character images treated in this work are basically only twenty six in number and variations arise due to recording differences only (that is writing individualities). Thus defining a set of primitives is better for processing, particularly for recognition. To retain the wide applicability of the cell level feature definition the correspondence that exist between the features as defined here and primitives as used by others is exploited. The next chapter treat primitives and their extraction and use in recognition of characters.

CHAPTER VI - PRIMITIVES AND THEIR EXTRACTION

6.1 INTRODUCTION

This chapter describes the techniques which are employed in the extraction of Primitives. Primitives are defined first and their usefulness is described. It may be noted that the Features discussed in Chapter IV are different from the primitives discussed in this chapter. Features are defined in chapter IV for reducing the information content in the image, where as Primitives discussed in this chapter are used for the classification of the unknown character. The primitives defined and extracted for character recognition in this work are a new set. The inventory of the primitives is kept openended with a view to add new primitives.

One of the important tasks in the design of any pattern recognition system is to develop methods to extract characteristics of the pattern. These characteristics are called the PRIMITIVES in syntactic pattern recognition. The primitive extraction techniques decide the recognition procedure.

A character is recognized by its description which is built out of primitives. Identifying these primitives from the image is a complex and crucial problem. In every character recognition scheme tolerances must be provided for commonly occurring variations in primitives. The information on inter-primitive context can be made use of in assigning primitive labels.

In this classification scheme openended inventories of primitives are updated through an interactive mode of operation with the computer. The primitive extraction rules are also introduced and modified through the same mode of operation.

6.2 PRIMITIVES:

As the description of an unknown character is built up from its characteristic primitives, the addition of each new primitive to the description reduces the members in the character class and with all the primitives specified the character is classified unambiguously.

In an ideal system the primitives should enable the correct discrimination of one class of character from another class [SUE4]. Also the primitives shall contain only a small set of features [KSFU], when syntactic methods are used in the character recognition scheme. It is important to note that most reported systems relay on structural primitives [BERT].

Pattern primitives are assumed to be easily recognizable. However this is not the case in most practical situations. As an example, strokes are considered as good primitives for script handwriting. But they are not easily extractable by machines. A compromise between its use as a basic part of the pattern and ease of recognition is often required in the process of selecting the pattern primitives [KSFU].

By the selection of the appropriate primitives the following advantages are realised:

1). Tedious manual involvement can be reduced.

2). It is applicable to a range of problems.

3). Selection is not restricted by lack of human insight.

4). Primitives can be confined to only those which are easily implementable.

5). complexity can be related to the required performance.

6). Primitives can be chosen according to a range of objective criteria.

In the scheme developed in this work, the parts or primitives are known as:

1). MAIN PRIMITIVES (M.P)

2). AUXILIARY PRIMITIVES (A.P).

The auxiliary primitives are the interconnection between the main primitives.

The following sections present the elements in each of the two classes of primitives. All the characters can be described by choosing the appropriate elements from these classes.

6.3. MAIN PRIMITIVES:

The Main Primitives are grouped into four. They are:

1). HORIZONTAL LINES

2). VERTICAL LINES

3). RIGHT SLANT LINES

4). LEFT SLANT LINES.

These are further divided into four categories as follows:

1). QUARTER LENGTH LINES

2). HALF LENGTH LINES

3). THREE QUARTER LENGTH LINES

4). FULL LENGTH LINES.

The assignment of a primitive into one of the four categories is decided by the number of components, Wordnames in this particular case. For example a Quarter Length Horizontal Line (QLHL) is a 3A3, where as a Half Length Horizontal Line (HLHL) is a 3A3 followed by a 3A3 or any combination of 3A3, 3A0 and 3A2. A Quarter Length Vertical Line (QLVL) is represented by a 3A1, and a Three Quarter Length Vertical Line (TLVL) is a combination of 3A1, 3A0, and 3A2. In the case of Right Slant and Left Slant lines, this is the combination of the Word names 3A0 and 3A2. Possible combinations of these four categories for all the four main primitives are given in Table 6.1. By the con catenation of these primitives, curves can be derived and character description can be generated. This is where the use of Auxiliary Primitives arise.

6.4 AUXILIARY PRIMITIVES:

The type of Auxiliary Primitive indicate how many Main Primitives are intersecting and converging to a junction. The different types of Auxiliary Primitives are given in Table - 6.2 along with the comments on their role in character description.

The inventory of Auxiliary Primitives are given in Table - 6.3 (a) to Table - 6.3 (c).

6.5 PRIMITIVE EXTRACTION

Many an author speaks of primitive extraction as the most difficult and crucial stage in any character recognition

QUARTER LENGTH LINES

VERTICAL	3A1
HORIZONTAL	3A3
LEFT SLANT	3A0
RIGHT SLANT	3A2

HALF LENGTH LINES

VERTICAL	3A1+3A1, 3A1+3A0, 3A0+3A1, 3A1+3A2, 3A2+3A1
HORIZONTAL	3A3+3A3, 3A3+3A0, 3A0+3A3, 3A3+3A2, 3A2+3A3
LEFT SLANT	3A0+3A0 The first 3A0 shall be at NW
RIGHT SLANT	3A2+3A2 The second 3A2 shall be at SW

THREE QUARTER LENGTH LINES

VERTICAL	3A1+3A1+3A1, 3A1+3A1+3A0, 3A1+3A1+3A2, 3A1+3A0+3A1, 3A1+3A2+3A1, 3A0+3A1+3A1, 3A2+3A1+3A1, 3A1+3A0+3A2, 3A1+3A2+3A0, 3A0+3A1+3A2, 3A2+3A1+3A0, 3A0+3A2+3A1, 3A2+3A0+3A1.
HORIZONTAL	3A3+3A3+3A3, 3A3+3A3+3A0, 3A3+3A3+3A2, 3A3+3A0+3A3, 3A3+3A2+3A3, 3A0+3A3+3A3, 3A2+3A3+3A3, 3A3+3A0+3A2, 3A3+3A2+3A0, 3A0+3A3+3A2, 3A2+3A3+3A0, 3A0+3A2+3A3 3A2+3A0+3A3.
LEFT SLANT	3A0+3A0+3A0. The first shall be at NW and the third shall be at SE positions of the second.
RIGHT SLANT	3A2+3A2+3A2. The first shall be at SW and the third shall be at NE positions of the second

FULL LENGTH LINES

VERTICAL	3A1+3A1+3A1+3A1, 3A1+3A1+3A1+3A0, 3A1+3A1+3A1+3A2, 3A1+3A1+3A0+3A1, 3A1+3A1+3A2+3A1, 3A1+3A0+3A1+3A1, 3A1+3A2+3A1+3A1, 3A0+3A1+3A1+3A1, 3A2+3A1+3A1+3A1, 3A1+3A1+3A0+3A2, 3A1+3A1+3A2+3A0, 3A1+3A0+3A2+3A1, 3A1+3A2+3A0+3A1, 3A0+3A2+3A1+3A1, 3A2+3A0+3A1+3A1, 3A0+3A1+3A1+3A2, 3A2+3A1+3A1+3A0, 3A0+3A1+3A2+3A1, 3A2+3A1+3A0+3A1, 3A1+3A0+3A1+3A2, 3A1+3A2+3A1+3A0, 3A1+3A0+3A1+3A0,

	3A1+3A2+3A1+3A2,	3A0+3A1+3A0+3A1.
	3A2+3A1+3A2+3A1,	3A1+3A0+3A2+3A0,
	3A1+3A2+3A0+3A2,	3A0+3A2+3A0+3A1.
	3A2+3A0+3A2+3A1,	3A0+3A2+3A1+3A0,
	3A2+3A0+3A1+3A2,	3A0+3A1+3A0+3A2.
	3A2+3A1+3A2+3A0,	3A0+3A2+3A1+3A0,
	3A2+3A0+3A1+3A2,	3A0+3A2+3A1+3A2,
	3A2+3A0+3A1+3A0,	3A0+3A1+3A2+3A1.
	3A2+3A1+3A0+3A1,	3A0+3A2+3A0+3A2.
	3A2+3A0+3A2+3A0,	3A2+3A1+3A0+3A2.
HORIZONTAL	Replace all 3A1 by 3A3 in the previous to get the full complement of Full Length Horizontal Line	
LEFT SLANT	3A0+3A0+3A0+3A0.	
RIGHT SLANT	3A2+3A2+3A2+3A2.	

TABLE - 6.1
POSSIBLE COMBINATIONS OF MAIN PRIMITIVES

TYPE	COMMENT	No. OF LINES INTERSECTING AT THE JUNCTION
LINE END	INDICATES THE FREE END OF A LINE.	1
CORNER	INDICATES THE CORNER FORMED BY TWO LINES.	2
BRANCH JUNCTION	INDICATES THAT A LINE IS TERMINATING AT THE WAIST OF ANOTHER LINE.	3
FOUR LIMB JUNCTION	INDICATES THAT TWO LINES CROSS EACH OTHER OR TWO LINES TERMINATING AT THE WAIST OF ANOTHER LINE.	4

TABLE - 6.2

DIFFERENT TYPES OF AUXILIARY PRIMITIVES AND THEIR ROLE IN CHARACTER DESCRIPTION

TABLE 6.3(a) AUXILIARY PRIMITIVES- CORNERS (TWO LIMB JUNCTIONS)

1) L-CORNER (L C)	3A1+3A3, 3A3 shall be at EAST, SOUTH EAST or SOUTH neighbors of 3A1.
2) F-CORNER (F C)	3A1+3A3, 3A3 shall be at NORTH, EAST or NORTH EAST neighbors of 3A1.
3) G-CORNER (G C)	3A1+3A3, 3A3 shall be at WEST, NORTH or NORTH WEST neighbors of 3A1
4) J-CORNER (J C)	3A1+3A3, 3A3 shall be at WEST, SOUTH or SOUTH WEST neighbors of 3A1
5) W-CORNER (W C)	3A2+3A0, 3A0 shall be at the EAST neighbor of 3A2
6) V-ORNER (V C)	3A0+3A2, 3A2 shall be at the EAST neighbor of 3A0
7) Q-CORNER (Q C)	3A1+3A2, 3A2 shall be at the EAST neighbor of 3A1
8) B-CORNER (B C)	3A3+3A2, 3A2 shall be at the EAST neighbor of3A3
9) Z-CORNER (Z C)	3A2+3A3, 3A3 shall be at the SOUTH neighbor of 3A2
10) 7-CORNER (7 C)	3A3+3A2, 3A2 shall be at the SOUTH neighbor of 3A3
11) M-CORNER (M C)	3A3+3A2, 3A2 shall be at the WEST neighbor of 3A3
12) N-TOP (N T)	3A1+3A0, 3A0 shall be at the WEST neighbor of 3A1
13) N-BOTTOM (N B)	3A0+3A1, 3A1 shall be at the EAST neighbor of 3A0
14) 3-CORNER (3 C)	3A2+3A0, 3A0 shall be at the SOUTH neighbor of 3A2
15) O-CORNER (O C)	3A0+3A2, 3A2 shall be at the SOUTH neighbor of 3A2
16) 5-CORNER (5 C)	3A2+3A3, 3A3 shall be at the NORTH EAST neighbor of 3A2
17) LEFT SLANT (L S)	3A0+3A0, The second 3A0 shall be at the SOUTH EAST neighbor of the first 3A0
18) RIGHT SLANT (R S)	3A2+3A2, The second 3A2 shall be at the NORTH EAST neighbor of the first 3A2
19) U-RIGHT (TOP) (U R T)	3A1+3A2, 3A2 shall be at the NORTH EAST neighbor of 3A1
20) U-RIGHT (U R)	3A2+3A1, 3A1 shall be at the SOUTH WEST neighbor of 3A2

21)	U-LEFT (TOP) (U L T)	3A0+3A1, 3A1 shall be at the SOUTH EAST neighbor of 3A0
22)	U-LEFT (U L)	3A1+3A0, 3A0 shall be at the SOUTH EAST neighbor of 3A1
23)	S-CORNER (S C)	3A3+3A0, 3A0 shall be at the SOUTH neighbor of 3A3
24)	D-CORNER (D C)	3A0+3A3, 3A3 shall be at the SOUTH EAST neighbor of 3A0
25)	A-CORNER (A C)	3A3+3A0, 3A0 shall be at the NORTH neighbor of 3A3
26)	S-CORNER (S C)	3A3+3A0, 3A0 shall be at the SOUTH EAST neighbor of 3A3
27)	E-CORNER (E C)	3A0+3A2, 3A2 shall be at the SOUTH neighbor of 3A0

TABLE 6.3(b) AUXILIARY PRIMITIVES - JUNCTIONS (THREE LIMB JUNCTIONS)

1)	9-JUNCTION (9 J)	3A1+3A1+3A2, 3A2 shall be the SOUTH WEST neighbor of the first 3A1 and the second 3A1 shall be the SOUTH neighbor of the first 3A1
2)	6-JUNCTION (6 J)	3A1+3A1+3A2, 3A2 shall be the NORTH EAST neighbor of the first 3A1 and the second 3A1 shall be the SOUTH neighbor of the first 3A1
3)	RIGHT BRANCH (R B)	3A1+3A1+3A3, 3A3 shall be the EAST neighbor of the first 3A1 and the second 3A1 shall be the SOUTH neighbor of the first 3A1
4)	LEFT BRANCH (L B)	3A1+3A1+3A3, 3A3 shall be the WEST neighbor of the first 3A1 and the second 3A1 shall be the SOUTH neighbor of the first 3A1
5)	T JUNCTION (T J)	3A3+3A3+3A1, 3A1 shall be the SOUTH neighbor of the first 3A3 and the second 3A3 shall be the EAST neighbor of the first 3A3
6)	I JUNCTION (I J)	3A3+3A3+3A1, 3A1 shall be the NORTH neighbor of the 3A3 and the second 3A3 shall be the EAST neighbor of the first 3A3
7)	Y JUNCTION (Y J)	3A0+3A2+3A1, 3A1 shall be the SOUTH neighbor of either 3A0, or 3A0 and 3A2 shall be the EAST neighbor of 3A0 OR

	3A0+3A2+3A2, the second 3A2 shall be the SOUTH neighbor of 3A0 and the first 3A2 shall be the EAST neighbor of 3A0
8) B FORK (B F)	3A3+3A2+3A0, 3A2 shall be the EAST neighbor of 3A3 and 3A0 shall be SOUTH EAST neighbor of 3A3
9) B JUNCTION (B J)	3A3+3A3+3A2, 3A2 shall be the NORTH EAST neighbor of the first 3A3 and the second 3A3 shall be the EAST neighbor of the first 3A3
10) INVERTED Y JUNCTION (I Y J)	3A2+3A2+3A1, 3A1 shall be the EAST or SOUTH neighbor of the first 3A2 and the second 3A2 shall be the NORTH EAST neighbor of the first 3A2
11) INVERTED 9 JUNCTION (I 9 J)	3A0+3A1+3A1, 3A0 shall be the WEST neighbor of the first 3A1 and the second 3A1 shall be the SOUTH neighbor of the first 3A1
12) INVERTED B JUNCTION (I B J)	3A3+3A3+3A2, 3A2 shall be the SOUTH neighbor of the first 3A3 and the second 3A3 shall be the EAST neighbor of the first 3A3
13) INVERTED 6 JUNCTION (I 6 J)	3A1+3A1+3A0, 3A0 shall be the EAST neighbor of the first 3A1 and the 3A1's shall be the SOUTH neighbor of the first 3A1
14) R JUNCTION (R J)	3A3+3A2+3A1, 3A2 shall be the NORTH EAST neighbor of 3A3 and 3A1 shall be the SOUTH or SOUTH EAST neighbor of 3A3
15) Q J JUNCTION (Q J J)	3A3+3A1+3A2, 3A1 shall be the NORTH neighbor of 3A3 and 3A1 shall be the NORTH EAST neighbor of 3A3
16) G 7 JUNCTION (G 7 J)	3A3+3A1+3A2, 3A2 shall be the SOUTH neighbor of 3A3 and 3A1 shall be the SOUTH EAST neighbor of 3A3
17) L Z JUNCTION (L Z J)	3A1+3A2+3A3, 3A3 shall be the SOUTH neighbor of 3A1 and 3A2 shall be the EAST neighbor of 3A1
18) L 5 JUNCTION (L 5 J)	3A2+3A1+3A3, 3A3 shall be the EAST neighbor of 3A2 and 3A1 shall be the NORTH neighbor of 3A2
20) F M JUNCTION (F M J)	3A2+3A1+3A3, 3A3 shall be the NORTH EAST neighbor of 3A2 and 3A1 shall

	be the EAST neighbor of 3A2
21) F U JUNCTION (F U J)	3A2+3A3+3A1, 3A3 shall be the SOUTH neighbor of 3A2 and 3A1 shall be the SOUTH WEST neighbor of 3A2
22) J 7 JUNCTION (J 7 J)	3A3+3A1+3A2, 3A1 shall be the NORTH WEST neighbor of 3A3 and 3A2 shall be the EAST neighbor of 3A3
23) A S JUNCTION (A S J)	3A3+3A3+30, 3A0 shall be the SOUTH EAST neighbor of the first 3A3 and the second 3A3 shall be the EAST neighbor of the first 3A3
24) F Z JUNCTION (F Z J)	3A1+3A3+3A2, 3A2 shall be the NORTH EAST neighbor of 3A1 and 3A3 shall be the EAST neighbor of 3A1
25) L 5 JUNCTION (L 5 J)	3A1+3A2+3A3, 3A2 shall Ge the SOUTH neighbor of 3A1 and 3A3 shall be the SOUTH EAST neighbor of 3A1
26) W UR JUNCTION (W UR J)	3A2+3A0+3A1, 3A0 shall be the EAST neighbor of 3A2 and 3A1 shall be the north neighbor of 3A2
27) 5 F JUNCTION (5 F J)	3A2+3A1+3A3, 3A1 shall be the EAST neighbor of 3A2 and 3A3 shall be the NORTH EAST neighbor of 3A2

TABLE 6.3(C) - AUXILIARY PRIMITIVES - JUNCTIONS (FOUR LIMB JUNCTIONS)

1) K JUNCTION (K J)	3A1+3A1+3A2+3A0, 3A2 shall be the EAST neighbor of the first 3A1, 3A0 shall be the SOUTH EAST neighbor of the first 3A1 and the second 3A1 shall be the SOUTH neighbor of the first 3A1 OR 3A1+3A1+3A2+3A1, 3A2 shall be the EAST neighbor of the first 3A1, the second 3A1 shall be the SOUTH neighbor of the first 3A1 and the third 3A1 shall be the SOUTH EAST neighbor of the first 3A1
2) PLUS JUNCTION (P J)	3A1+3A1+3A3+3A3, the second 3A1 shall be the SOUTH neighbor of the first 3A1, the first 3A3 shall be the WEST or SOUTH WEST neighbor of the first 3A1 and the second 3A3

	shall be the EAST or SOUTH EAST neighbor of the first 3A1
3) X JUNCTION (X J)	3A0+3A2+3A2+3A0, the first 3A2 shall be the EAST neighbor of the first 3A0, the second 3A2 shall be the SOUTH neighbor of the first 3A0 and the second 3A0 shall be the SOUTH EAST neighbor of the first 3A0

TABLE - 6.3
LIST OF AUXILIARY PRIMITIVES

scheme. This is where human beings excel in their performance when compared to computers. A component of "intelligence" is being introduced into machines to achieve "human like" recognition capabilities as in Neural nets.

One of the basic difficulties encountered in primitive extraction is the unlimited variations or distortions in primitive shape in cursive script writing. Now the question to be answered is " what are the limits of accepted variations for a Primitive?". This is a very complex question to be answered. To achieve this goal, the tolerances are made flexible. The hierarchy among the auxiliary primitives, main primitives and the character description itself helps in the assignment of the primitive labels.

The inventory of standard primitives was generated from a sample character set. Tolerances were gradually added with the addition of new character sets. The primitive extractor is so designed that it will articulate a primitive if it has been added to the inventory, although it may not correspond to any character. The openended inventories for Auxiliary Primitives and Main Primitives provide facility for adding new primitives with the addition/modification of the extraction rules.

6.5.1 EXTRACTION OF AUXILIARY PRIMITIVES.

The Auxiliary Primitives are sorted out by the following information made available by the scanning of the modified image.

1). Number of junctions,

2). Number of limbs at a junction,

3). Interconnections of each of these limbs.

The Auxiliary Primitives can be divided into two categories:

1) Those which are extracted during the first scan, and

2) Those which are extracted during the subsequent scans.

The extraction of the first Auxiliary Primitive leads to the first sorting of the characters into different possible classes. However the labeling of this primitive is subject to confirmation by the relevant Main Primitive extracted later on.

The scan now proceeds along the vertical direction from this junction. The occurrence of the next junction or junctions are noted when :

1) More than one limb is intersected, or

2) A sharp change is recorded in the direction of the line.

An Auxiliary Primitive indicating the change in curvature is labeled only after the Main Primitive occurring ahead of it is extracted. Many characters contain such Auxiliary Primitives. The record of the intercepts made by the scan and the change in curvature/slant of a line pinpoint the presence of a junction.

The approach presented in this work is based on the primitive-by-primitive build up of the description of the character. From the complete description of the character it can be seen that some of the Auxiliary Primitives can occur only at given positions of the relevant Main Primitive. Thus for a given Main Primitive it's end or middle position is valid only for a few particular Auxiliary Primitives. Hence the Auxiliary Primitive labeling is established by using the context provided by the Auxiliary Primitives and the Main Primitives identified.

The parameters employed for the Auxiliary Primitive labeling are:

1) The position of the junction on the Main Primitive, that is, head, waist or tail.
2) The number of limbs intercepted by the scan at the junction.

3) The angles made by these limbs, which in this case are 0°, 45° and 90°.

It is evident that the two-limb junctions occur at the ends of primitives only. Four limb junctions occur either in the middle of a primitive with two other primitives diverging or may occur in the form of two primitives crossing each other. The three limb junctions indicate the middle of one primitive and the end of another primitive.

The Auxiliary Primitives are thus classified into different categories considering the above parameters. In the building up of a description in the recognition scheme, an Auxiliary Primitive is first labeled by an asterisk (*) but once the Main Primitive on which it is residing is extracted, the identity of the Auxiliary Primitive is picked up from the description of the character.

.5.2 EXTRACTION OF MAIN PRIMITIVES.

A Main Primitive is the realisation of the command for transcribing the part of a character. But even in a set of "normally" written characters, these primitives show distortions. For example the variations in a FLVL is so much that with all the sixty nine features defined earlier in Chapter 4, it will almost be impossible to realize a recognition system. However the reduction of these primitives into a small set have considerably reduced the complexity of the recognition algorithm. Even with this reduced number of primitives the complexity is not completely avoided. For example consider the FLVL. This Full Length Vertical Line can be defined by any one of the following combinations:

1).3A1 + 3A1 + 3A1 + 3A1

2).3A1 + 3AO + 3A1 + 3A1

3).3A1 + 3A1 + 3AO + 3A1

4).3A1 + 3A1 + 3A1 + 3AO

5).3AO + 3A1 + 3A1 + 3A1

6).3AO + 3AO + 3A1 + 3A1

7).3A1 + 3AO + 3AO + 3A1

8).3A1 + 3A1 + 3AO + 3AO

9).3AO + 3A1 + 3A1 + 3AO

Only nine different combinations are shown here. However a complete set will contain 42 members. Table 6.1 gives all the combinations that are commonly occurring in the character sample set. Another point to remember is variations in the size of these primitives. For example a vertical line can be any one among the four of its family.

The main primitives can be extracted using the following.

1). Direction of the line with respect to the datum line.

2). The length of the line.

3). The range of the line which is the straight line measure between the two end points.

4). The bearing of the line with respect to the datum line

5). The curvature at each major point.

6). The angle through which the line has turned.

The angle can be computed easily. This is the sum of all the angle changes along the line. For example consider the following combination for an FLVL , 3A1 + 3A0 + 3A2 + 3A1. In this case the angle change is zero, if all the components are in the same column. However assume the situation in which 3A1 + 3A0 are in the same column and 3A2 + 3A0 are in the next column. In this case there is an angle change of 45 . Assume another situation in which a FLVL is intersected by a QLHL, that is a three limb junction, and the angle change here is 90 .The angle changes can be computed very easily using the technique employed in this work, and is an advantage over other works.

The extraction of the Main Primitives can be done in two modes. In the first mode the measurements after each scan are "sensed" showing the gradual step by step identification of the primitive. In the second mode the primitive extraction is attempted when the next junction is reached.

6.6 TOLERANCES IN MEASUREMENTS

The versatility of a hand written character recognition system is indicated by the ability to recognize characters with large variations in their shape and size. Hence in extracting primitives it involves the introduction of tolerances in various primitives extracted. These are discussed below.

6.6.1 Tolerances in Auxiliary Primitives

The extraction of auxiliary primitive is accomplished by determining one or more of the three characteristics present in the modified character image;

1) Number of limbs at a junction,

2) Number of junctions in the image and their locations,

3) Interconnection of each of the limbs.

The hierarchy helps in classifying the auxiliary primitives with respect to the main primitives already extracted.

For example, when the primitives at a T-junction are extracted and with the presence of a main primitive at a particular position already extracted, sufficient information is available to proceed further to assign a class label for the image in question.

6.6.2 Tolerances in Main Primitives

From the sample set available, a peculiar phenomenon which arose is that the tolerance limit on various parameters with respect to the main primitives is not as large as in the case of the auxiliary primitives. To quote an example a main primitive, a full length vertical line, should be four units long where each unit is a quarter length line, either vertical or slant. The grid size does not alter this parameter since, whatever be the size of the character, it is expressed in terms of only sixteen word names. This is another advantage of this work.

6.7 MEASUREMENT INVENTORIES

Openended inventories are maintained for the extraction of auxiliary primitives and main primitives. The measurements made are compared with values available in the inventories for the purpose of labeling the primitives. Many variations are found in these and hence tolerances are applied and a measurement inventory was gradually built up by processing the character set. Each character set was studied in detail and new feature measurements were added whenever new ones were encountered. The addition of new members to the inventory was done by adding the new relevant information. Both these actions imply relevant changes in the decision logic for primitive extraction. The measurement inventories are linked to the description inventory in the domain of labeling.

CHAPTER VII - CLASSIFICATION

7.1 INTRODUCTION

The various aspects of Image Coding, Features used for Filtering and Primitive Extraction were discussed in detail in the last four chapters. This chapter discusses the classification procedure adopted in this work. Initially character description schemes are described, followed by classification methods. An interaction exists between the various stages of the scheme. The new ideas presented are:

1) Hierarchy maintained between the Main and Auxiliary Primitives and the character description, and
2) The provision of openended inventories which makes possible the addition of new members to the group and also handles new variations in the description of the existing characters.

A set of rules is used for the build up of the description of a character from an inventory of primitives. The rules are made as simple as possible and flexible, to add new descriptions of characters or even to add new groups of character sets.

Recognition is also a result of proper description of the image. This is very true for characters also. In the case of Kanji characters, characters represent a "picture" of the object it stands for. At the lower level where symbols which are seemingly unconnected with the object it represents, the character description should aim at classification or recognition. The character image is described in terms of "phrases" consisting of primitives. Classification is completed by applying rules or heuristically. The recognition process in this work consists of the comparison of the "description" of the character being investigated with a reference description of all the characters in the inventory. The description of the character is built up from its primitives extracted at the primitive extraction stage. With the addition of each new primitive to the description, the character class is reduced and with all the primitives specified the character is classified unambiguously.

Primitives are more purposeful in character description when compared to features. Some primitives may themselves be features while others could be combinations of features. The features proposed in this work, numbering 69, may be appropriately used for extracting primitives.

CHARACTER DESCRIPTION AND PRIMITIVES

Images are made up of structural parts like often occurring lines of definite shape or junctions of two or more lines ...etc.. These structural parts and the interconnections between them determine and describe a character. Some of the standard primitives used in image description, but not necessarily in character description, are symmetric, like a fork (⋖ ,⋋), a trident (⋔), bend (∠ , ⋗), cross (⋊) ...etc.. In this work it is proposed to use the above set of primitives along with a new set defined for the purpose of describing the characters. In the proposed scheme the parts or the primitives are called the MAIN PRIMITIVES and the interconnections between them as the AUXILIARY PRIMITIVES.

An example will show how description is built up employing the two types of primitives. Consider, for example the following two combinations:

000 3A1 000 3A2	3A0 000 000 3A2
000 3A1 3A2 000	000 3A0 3A2 000
000 3A1 000 000	000 3A2 3A0 000
000 3A1 000 000	3A2 000 000 3A0

The first is a Y junction where as the second combination stands for a X junction. It can be seen from these two examples that, in the case of a Y junction two lines, main primitives, join at a point, where as in the case of the X junction, two main primitives cross at a particular point. The elements of these kinds were discussed in detail in the previous chapter. All the characters can be described by choosing the appropriate elements from these classes.

The character images were limited to lower case English cursive script in order to achieve a coalescence in Features and Primitives. As mentioned in Chapter IV, features are defined and extracted at the cell level, where as primitives are identified and used at the paragraph level. It can be established that this process does not introduce any performance degradations, as long as the number of loops and crosses are small.

7.3 HIERARCHY IN CHARACTER DESCRIPTION

Intuitively it can be felt that a hierarchy exists between the Main Primitives, Auxiliary Primitives and the description of the character in the classification procedure. The description of a character implies that the Auxiliary Primitives can occur only in particular combinations in a certain character and that a primitive can occur only at a particular position. To quote an example, the second auxiliary primitive GC at column 4 can occur only with the second main primitive QLHL at row 1 and in character "p". This fact can be made use of in the step - by - step recognition scheme implemented here.

The recognition of the auxiliary primitives concurrently indicates the host Main Primitive. Therefore the task of primitive extraction also directly extracts the main primitives. Normally these two decisions, that is Auxiliary Primitive recognition and Main Primitive location concur. In the event of a disagreement, the latter decision prevails. A disagreement, for instance, can occur if an auxiliary primitive is not recognized correctly or if an auxiliary primitive exists at a location where it should not exist. That is, an auxiliary primitive can occur only at designated places in a main primitive and cannot occur at any other place in a character. Conversely, the combinations of auxiliary primitive and the main primitive are fixed in any character.

7.4 RULES ADOPTED FOR CHARACTER RECOGNITION

A set of rules is generated for the structural description of the characters in terms of the primitives. These rules specify the order in which the primitives concatenate. The interconnections of the main primitives are indicated by the auxiliary primitives residing in them. An analysis of the characters reveals that:

1). One and two limb junctions may occur at line ends.

2). Three and four limb junctions reside at the waist of at least one main primitive.

3). A main primitive may have upto four auxiliary primitives residing in it.

Hence the standard description of a character takes the form:

MP1 (AP11,AP21,AP31,AP41) * MP2 (AP12,AP22,AP32,AP42) * -- --------- * MPn (AP1n,AP2n,AP3n,AP4n) ....(7.1),

where * stands for a relational operator and "n" may vary from 1 to 4.

7.4.1 GENERATION OF CHARACTER DESCRIPTION

The following examples show how these rules can be employed in generating the characters. The notations used in generating the descriptions are described as and when they are encountered.

1). A main primitive is accompanied by the auxiliary primitives residing in it.

2). The contact among primitives is denoted by a " + ", (plus). These can be seen in the following.

The description of the character "O" in which two main primitives are connected is:

O = TLVL (C1,LC,FC) + TLHL (R1,FC,JC) + TLVL (C4, GC,JC) + TLHL (R4,JC,FC)

where C1, C2, C3, C4 stands for columns 1 to 4 and R1, R2, R3, R4 stands for rows 1 to 4.

3). In the case of some of the characters where a main primitive is connected to more than one main primitive the descriptions will be in two forms. The main primitives may be mutually connected or disconnected. Typical examples for the two classes are given below. Consider the first type where the character is described by the following:

A = HLVL (C1,LC,FC) + TLHL (R1,FC,GC) + HLVL (C3, GC,IJ) + TLHL (R4,IJ,LC)

D = QLVL (C1,LC,FC) + QLHL (C3,FC,LB) + FLVL (C3, LB,QC,JC) + QLRS (R4,QC) . QLHL (R4,LC,JC)

The operator "." indicates a separator, indicates the fact that these two main primitives are non-touching.

7.5 ILLUSTRATIONS OF CHARACTER DESCRIPTIONS

A sample of lower case English language characters is given in Fig:4.1. A different set with distortions is given in Fig:4.2. The description is built up in the order in which the primitives are to be recognized. Fig:7.1 shows the labeling of the characters employing the primitives. In this figure the descriptions of the characters are also added. Table - 7.1 give the description of all the characters in Fig: - 4.1. The description of characters in Fig: 4.2 will also belong to one among this as normalization takes place at the filtering stage. This is an advantage of the new proposal made in this work.

7.6. OPEN - ENDED INVENTORIES

The inventories for descriptions are kept openended with a view to include more character descriptions. This facility permits the extension of the existing inventory to include more character definitions of the already present classes or adding new classes of characters to the inventory.

A powerful feature of the scheme is its versatility to adapt to situations in which more than one description is permitted for some of the characters. Whenever there is more than one description for a character in the inventory, the recognition becomes easier. More new characters can be included in the inventory by using suitable main primitives and auxiliary primitives from their inventories. By keeping the inventories of both main primitives

000 3A1 3A3 3A0

3A3 3A0 3A3 000

000 3A1 000 000

000 3A1 000 000

CHARACTER "f"

See the "+" junction at row 2 column 2 and the F Corner at row 1. The 3A1+3A0+3A1+3A1 in C2 is the FLVL, 3A3+3A0 in R1 is the H L at top and the 3A3+3A0+3A3 in R3 is the second H L.

3A1 3A3 3A1 000

3A1 000 3A1 000

3A1 000 3A1 000

3A1 3A3 3A3 3A3

CHARACTER "a"

See the L C at row 4, F C at row 3, G C at row 1, and the I J at row 4. The 3A1+3A1+3A1+3A1 in column 1 is the first vertical line in C1, 3A3 in row 1 is one of the H L's, 3A1+3A1+3A1 IN C3 is the second V L and 3A3+3A3+3A3 in row 4 is the second H L.

000 3A1 000 000

000 3A1 3A3 000

000 3A1 000 3A1

3A2 3A1 3A3 3A3

CHARACTER "b"

See 9 J at row 4, L C at row 4, R B at row 2, GC at row 2 and the J C at row 4 3A1+3A1+3A1+3A1 in C2 is the first V L, 3A3 in R3 is one of the two H L 's, 3A1 in C4, is the second V L and 3A3+3A3 in R4 is the second H L. The 3A2 is R4 is the tail at the bottom left of the char acter which is not necessary to be identified.

3A3 3A3 3A3 000

3A1 3A3 3A1 000

000 000 3A1 000

000 000 3A1 3A2

CHARACTER "q"

See the F C and G C in row 1, L C in row 2, the L in row 2 and the Q C in row 4. 3A1 in C1 is the connecting link between the two H L's in R1 and R2, 3A1+3A1+3A1 in C3 is the V L and the Q L R S, 3A2 in is the Q C.

Fig: 7.1

LABELING OF THE CHARACTER

DESCRIPTION OF THE FOUR CHARACTERS

CHARACTER " कि " FLVL [C2, PJ, FC] + HLHL [R1, FC].
QLHL [R2, PJ] + QLHL [R2, PJ].

CHARACTER " a " FLVL [C1, LC, FC] + QLHL [R1, FC, GC] +
TLVL [C3, GC, IJ] + FLHL [R4, LC, IJ].

CHARACTER " b " FLVL [C2, 9J, LC, FC] + HLHL [R4, LC,
JC] + QLVL [C4, JC, GC] + QLHL [R2,
GC, FC] . QLRS [R4, 9J].

CHARACTER " q " QLVL [C1, LC, FC] + TLHL [R1, FC. GC,
PR2] + TLVL [C3, GC, LB, QC] +
QLHL [R2, PR1, LC, LB] . QLRS [R4, QC].

NOTE: In the description of the character "q", the notation PR1 and PR2 means parallel lines in row1 and row2 respectively.

Fig: - 7.1.a

LABELING OF CHARACTERS

TABLE: 7.1

CHARACTER "a" - F/T/HLVL [C1, LC, FC] + HL [R1, FC, GC] + H/T/FLVL [C3, GC, IJ] + HL [R4, LC, IJ]

CHARACTER "b" - FLVL [C1/2, LC, RB] + T/HLHL [R2/3, RB, GC] + H/QLVL [C4, GC, JC] + T/HLHL [R4, LC, JC]

CHARACTER "c" - F/TLVL [C1, LC, FC] + F/T/HLHL [R1, FC] . F/TLHL [R4, LC]

CHARACTER "d" - QLVL [C1, LC, FC] + T/HLHL [R2/3, FC, LB] + F/TLVL [C3/4, LB, IJ/(JC&QC)] + F/T/HLHL [R4, LC, IJ/JC]/QLRS [R4, QC]

CHARACTER "e" - F/TLVL [C1, LG, RB, FC] + F/TLHL [R1, PR2 FC, GC/7C] + QLVL [C4, GC, JC]/QLRS[R2, 7C, BC] + H/QLHL [R2, PR1, JC/BC, RB]. F/TLHL [R4, LC]

CHARACTER "f" - F/TLVL [C2/3, FC, RJ] + T/HLHL [R1, FC]. Q/HLHL [R2/3, PJ] + Q/HLHL [R2/3, PJ]

CHARACTER "g" - QLVL [C1, LC, FC] + Q/HLHL [R1,PR2, FC, GC] + FLVL [C3, GC, LB, 9J, 6J, JC] + Q/HLHL [R2, PR1, JC, LC] . QLHL [R4, JC, ZC] + QLRS [R3, ZC, 9J] . QLRS [R2, 6J]

CHARACTER "h" - FLVL [C1, RB] + HLHL [R2/3, RB, GC] + H/TLVL [C4, GC]

CHARACTER "i" - FLVL [C2/3, IJ/(NB, QC) + Q/HLHL [R4, IJ]/ Q/HLLS [R3R4, NB] + H/QLHL [R4, IJ]/ Q/HLRS [R3R4, QC]

CHARACTER "j" - FLVL [C3, MC/GC, JC, 9J, 6J, LC] + QLHL [R4, JC, ZC] + HLRS [R4R3, ZC, 9J] + QLRS [R1, 6J] . QLRS [R1, MC]/QLHL [R1, GC]

CHARACTER "k" - FLVL [C2, KJ, MC/GC] + H/QLRS [R1R2, KJ] + H/QLLS [R3R4, KJ] . QLRS [R1, MC]/QLHL [R1, GC]

TABLE 7.1
DESCRIPTION OF THE CHARACTERS

CHARACTER "l" - F/TLVL [C2, PC3, UR/JC, LC] . H/QLVL [C3, PC2] . Q/HLHL [R4, LC] . QLHL [R4, JC]/ QLRS [R4, UR]

CHARACTER "m" - TLVL [C2, TJ, PC3] + H/T/FLHL [R1, TJ, TJ, GC] + TLVL [C3, PC2, GC/TJ, PC4] + TLVL [C4, PC3, GC].

CHARACTER "n" - F/TLVL [C2, TJ] + H/TLHL [R1, TJ, GC] + F/TLVL [C4, GC]

CHARACTER "o" - F/TLVL [C1, LC, FC] + F/T/HLHL [R1, FC, GC] + F/T/HLVL [C4, GC, JC] + F/T/HLHL [R4, JC, LC]

CHARACTER "p" - FLVL [C2, RB, FC, MC] + H/QLHL [R1, PR2, FC, GC] + QLVL [C4, GC, JC] + H/QLHL [R2, PR1, JC, RB] . QLRS [R1, MC]

CHARACTER "q" - QLVL [C1, FC, LC] + H/QLHL [R1, PR2, FC, GC] + FLVL [C3, GC, LB, QC] + H/QLHL [R2, PR1, LC, LB] . QLRS [R4, QC]

CHARACTER "r" - FLVL [C2, MC, FC] + QLRS [R1, MC] + T/HLHL [R1, FC]

CHARACTER "s" - QLVL [C1, FC, LC] + TLHL [R1, PR2, FC] . T/HLHL [R2, PR1, LC, GC] + H/QLVL [C4, GC, JC] + TLHL [R4, JC]

CHARACTER "t" - FLVL [C2/C3, LC, PJ] + HLHL [R2/R3, PJ] + Q/HLHL [R2/R3, PJ] . HLHL [R4, LC]

CHARACTER "u" - FLVL [C1, LC] + T/HLHL [R4, LC, TJ/JC] + TLVL [C3/C4, TJ/JC]

CHARACTER "v" - FLVL [C2, MC, QC] + QLRS [R1, MC] . Q/HLRS [R3/R2R3, QC, UR] + Q/HLVL [C4, UR]

CHARACTER "w" - FLVL [C1, PC2, LC] + T/HLHL [R4, LC, IJ, IJ/JC] + QLVL[C2,PC1, IJ] + T/HLVL [C3/C4, IJ/JC]

TABLE 7.1
DESCRIPTION OF CHARACTERS

CHARACTER "x" - FLVL [C2/C3, TJ, IJ] + H/QLHL [R1, TJ] + H/QLHL [R1, TJ] . H/QLHL [R4, IJ] + H/QLHL [R4, IJ]

CHARACTER "y" - QLVL [C1, LC] + H/QLHL [R2, LC, LB] + FLVL [C3, LB, JC, 9J, 6J] + QLHL [R4, JC, ZC] + QLRS [R3, ZC, 9J] . QLRS [R2, 6J]

CHARACTER "z" - HLVL [C3, GC, JC, 9J, 6J] + H/QLHL [R2, PR1, GC, ZC] + QLRS [R2, ZC, 7C] + HLHL [R1, PR2, 7C] . HLHL [R4, JC, ZC] + QLRS [R3, ZC, 9J] . QLRS [R2, 6J].

TABLE - 7.1

DESCRIPTION OF CHARACTERS

NOTE: 1) In the case of some characters more than one choice at some of the limbs is already given. However, this is not included in Fig: 7.2, The Decision Tree. However in the computer implementation these and other variants are also accounted for.

2) F/T/H/QL indicate Full Length / Three Quarter Length / Half Length / Quarter Length Lines

and auxiliary primitives openended new primitives can be added to their respective inventories.

7.7 CHARACTER CLASSIFICATION SCHEME

A decision tree is used for the classification scheme. This decision tree enables the interaction between the various elements in the classification procedure namely, the primitive extraction and character classification. The description of a character that is being classified, is built up with the location of each new primitive. Progress along any branch of the decision tree is made with the comparison of the built up description of the character with respect to a standard inventory of all the members of the lower case English language characters. With the addition of each new primitive, the comparison narrows down to the particular character in question and an unambiguous decision is reached finally. Because of the implementation of openended inventories the difficulty of many a problem like that of primitive separation, and extra primitives are avoided. This is achieved by dynamically adding more branches where ever necessary to the decision tree. The use of openended inventories permits this and helps avoid wrong classifications and minimize rejections. In a subsequent section of the chapter the computer implementation of the classification scheme is presented where a typical example is chosen to show the primitive by primitive extraction and the final unambiguous classification of the character. The variants in the decision tree is also indicated.

7.8 THE DECISION TREE

The classification scheme described in this work uses a decision tree for the interconnection of the various parts like modified image scanning, primitive extraction and the building up of the character description which in turn helps in the correct classification of the unknown character sample. J.R.Ullman [UMAN] in his famous book describes the use of decision tree in pattern recognition. In [NAR1] the use of decision tree is described in the context of hand printed English letters. Many authors have used the decision tree for the classification of particular group of character sets.

An introduction to multistage classification systems has been offered in [KURZ] and [SMAN]. It is stated in [SMAN] that the multistage classification system closely resembles the human recognition system.

The three main components that are involved in the design of a tree classifier as listed out in [KURZ] are:

1). The choice of the tree structure,

2). The choice of the primitives to be used and

3). The decision rules to be used at each nonterminal node for performing the classification.

The design of the decision tree used in this scheme which is a modification of the classical approach is presented below.

The structure of the decision tree employed in this classification scheme is shown in Fig:7.2, and is based on the character description scheme described earlier in Section 7.5. Any branch lying between the ROOT (designated as start in the figure) and any one of the terminal nodes (which is the identification of the character) is the description of the character. Each node indicates the presence of an auxiliary primitive. Except for the first row of auxiliary primitives, all other nodes show the discovery of a main primitive. It can be seen that every main primitive is represented by a branch in the decision tree. With the extraction of each primitive the membership in the character class decreases in number of members who satisfy the specified characteristics. The weakness of a conventional decision tree is the fact that a wrong branching taken at any node would affect the performance of the scheme adversely. The hierarchy of the character description scheme in this work avoids such situations. A new feature of this decision tree is the provision for introduction of new branches which lead to the original terminal node. The provision of additional branches leading to the same node allows simple techniques for handling variations in the same character.

Listing the primitives along a branch gives the description of a character which will be finally encountered at the terminal node of the branch. The build up of the description starts with the extraction of the first auxiliary primitive at the first node. The description of the character is enriched with the addition of every new main primitive and/or auxiliary primitive as shown by the successive nodes. The description of the character is thus built up in the form shown below:

AP1*MP1.AP2*MP2 ....APn-1* MPn ....(7.2)

In a few characters the auxiliary primitives may repeat. The above description is similar to the one shown earlier in section 7.4. The classic form is given in Eq:7.1, where as the actual form of generation of characters in real applications is of the form of Eq:7.2.

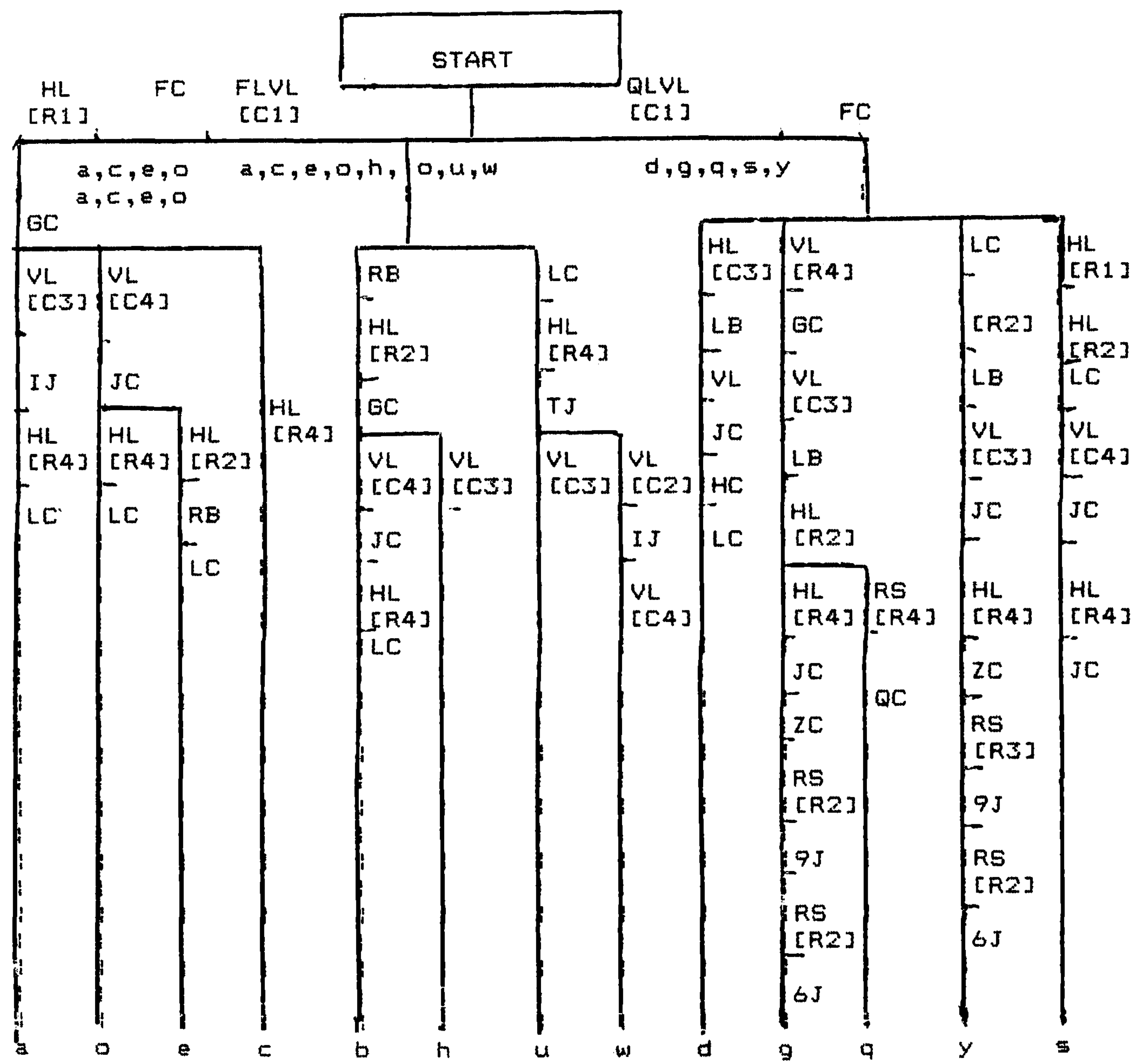

START
HL
[R1]
FC
FLVL
[C1]
QLVL
[C1]
FC
a,c,e,o
a,c,e,o
a,c,e,o,h,
o,u,w
d,g,q,s,y
GC
VL [C3] IJ HL [R4] LC
VL [C4] JC HL [R4] LC
HL [R2] RB LC
HL [R4]
RB HL [R2] GC VL [C4] JC HL [R4] LC
VL [C3]
LC HL [R4] TJ VL [C3]
VL [C2] IJ VL [C4]
HL [C3] LB VL JC HC LC
VL [R4] GC VL [C3] LB HL [R2] HL [R4] JC ZC RS [R2] 9J RS [R2] 6J
RS [R4] QC
LC [R2] LB VL [C3] JC HL [R4] ZC RS [R3] 9J RS [R2] 6J
HL [R1] HL [R2] LC VL [C4] JC HL [R4] JC
a
o
e
c
b
h
u
w
d
g
q
y
s

Fig:7.2

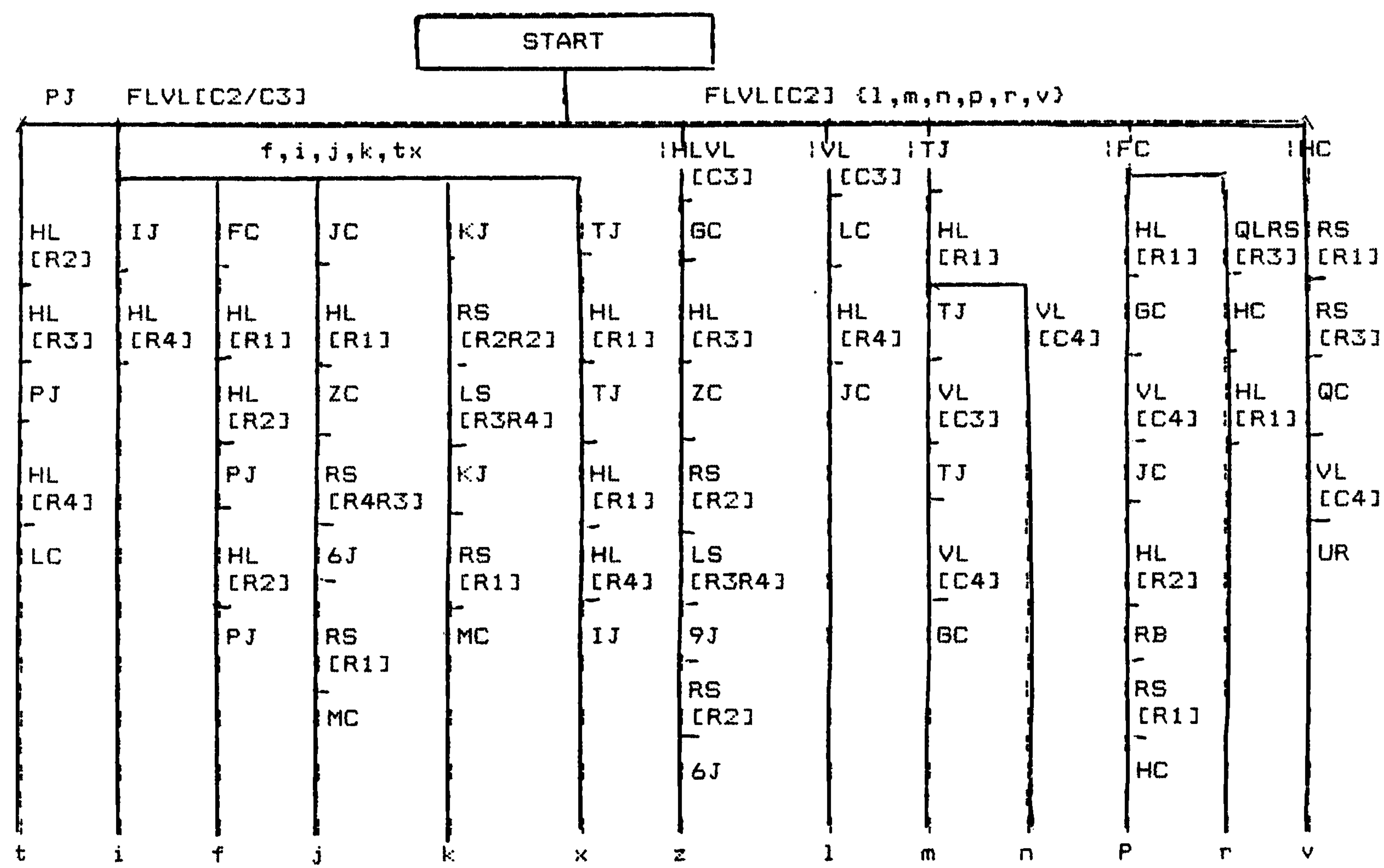


Fig:7.2

THE DECISION TREE

The decision tree shown in Fig:7.2 represents the standard form. The actual tree will contain additional branches to accommodate variants of the same character and new characters.

Recognition of characters by the decision tree is shown in Fig:7.3. Successive reduction in class with each Auxiliary Primitive and the ultimate arrival at the target character, are also shown in Fig: 7.3

7.9 INTERACTION AMONGST SCANNING, PRIMITIVE EXTRACTION AND CHARACTER CLASSIFICATION

The subsystems namely, scanning, primitive extraction and character classification interact through the decision tree. The branches of the tree describes all the haracters in the inventory. In the normal operation when a change is encountered the primitive is extracted and identified. This output is used to build the description of the character. During every such operation identification of a primitive leads to a decision at the node of the tree. The question of the character class is answered by referring to the hierarchy of the character description, Main Primitives and Auxiliary Primitives.

Hence the description dictionary is used for primitive

A CHARACTER	**IT'S EXTRACTED PRIMITIVES**
000 3A1 3A3 3A1	FIRST M.P - 3A1+3A1+3A1+3A1(FLVL) at C2
	FIRST A.P - FC at R1
3A2 3A1 000 3A1	SECOND M.P - 3A3 (QLHL) at R1
	SECOND A.P - GC at R1
000 3A1 3A3 3A3	THIRD M.P - 3A1+3A1 (HLHL) at C4
	THIRD A.P - JC at R3
000 3A1 000 000	FOURTH M.P - 3A3+3A3 (HLHL) at R3
	FOURTH A.P - RB at R3
	FIFTH M.P - 3A2 (QLRS) at R2
	FIFTH A.P - 9J at R2

CHARACTER DESCRIPTION :
FLVL [C2, RB, MC, FC] + QLHL [R1, FC, GC] + HLVL [C4, GC, JC] + HLHL [R3, JC, RB] . QLRS [R2, 9J]

PRIMITIVE		POSSIBLE CHARACTERS
FIRST MP	FLVL at C2	f, i, j, k, l, m, n, p r, t, x, v
FIRST AP	FC at R1	f, p, r
SECOND MP	QLHL at R1	p
SECOND AP	GC at C4	p
THIRD MP	HLVL at C4	p
THIRD AP	JC at R3	p
FOURTH MP	HLHL at R2	p
FORTH AP	RB at R3	p
FIFTH MP	QLRS at R1	p
FIFTH AP	MC at R1	P

NOTE: Although from the extraction of the second main primitive onwards the character was unambiguously classified as "p", the procedure was continued till the last step for the sake of completeness.

Fig: 7.3

RECOGNITION EMPLOYING THE DECISION TREE

by primitive recognition of a character. By adopting this type of interaction the computer emulates human recognition process. As can be seen form Fig:7.2 at any node the class of later possible primitives is reduced in number. This fact itself is utilised in the primitive extraction procedure. It can be said with certainty that for a few characters, if the characters are written normally, their classification can be done with a partial scan of the character image. However even in cases where a partial scan yields correct classification, the image is scanned completely to corroborate the earlier classification decision. For example consider the character recognition scheme in Fig: 7.3. Eventhough at the third stage, that is extraction of the second Main Primitive "QLHL" at R1, the character can be classified as "p", the routine continues until all the primitives are extracted and the character classification is unambiguously reiterated.

7.10. COMPUTER IMPLEMENTATION OF THE PROPOSAL

In this section the computer implementation of the new proposal is discussed. The section starts with the preparation of the data, digitization of the character, followed by a flow diagram. The primitive by primitive articulated classification of the character, through a decision tree,is traced to show the working of the scheme.

A manual sample collection formed the first stage of the experimental work. A number of subjects were asked to write lower case English language characters on a fine graph sheet. By scaling, the size of each character was brought to a 12 * 12 modified grid size. No constraint was placed on the use of the writing equipment. The choice of the 12 * 12 grid ensure that line continuity will be maintained, whatever be the writing aid used. For binarizing the following rule is used:
"Assign a value of 1 to a cell if the character outline passes through it. Otherwise assign the value of 0."

The binarized image data is entered through a key board (this work was started in the year 1987), and is scanned for further processing. Currently all this can be realized using an image scanner / camera combination.

The different units of the programme are shown in the flow chart, depicted in Fig: 7.4. In the scheme envisaged there are three basic operations:

1). Scanning of the modified image
2). Primitive extraction and unifying feature and primitive definitions.

3). Classification of the character through the build up of the primitives.

From the implementation point of view these three operations are further subdivided into many smaller units.

The search for the character is made as follows. The data is fed into the computer. The search subroutine sweeps the character image by checking the value of each pixel. The search begins with the left top pixel, proceeds along the row and when finally the top right pixel is reached the scan moves over to the next row. The procedure is repeated until the values of all the pixels are noted.

The program now moves over to the next subroutine which groups the pixels into cells. There are nine pixels in each cell. A naming subroutine assigns word names to all the cells.

The image is now ready for data reduction, for which a scheme called FILTERING is adopted. A modified image is available at the out put of the Filter. All further processing is done on this modified image.

Another subroutine which extracts primitives scans the modified image starting with the left bottom cell (cell 40). Once this subroutine encounters a cell where a non-

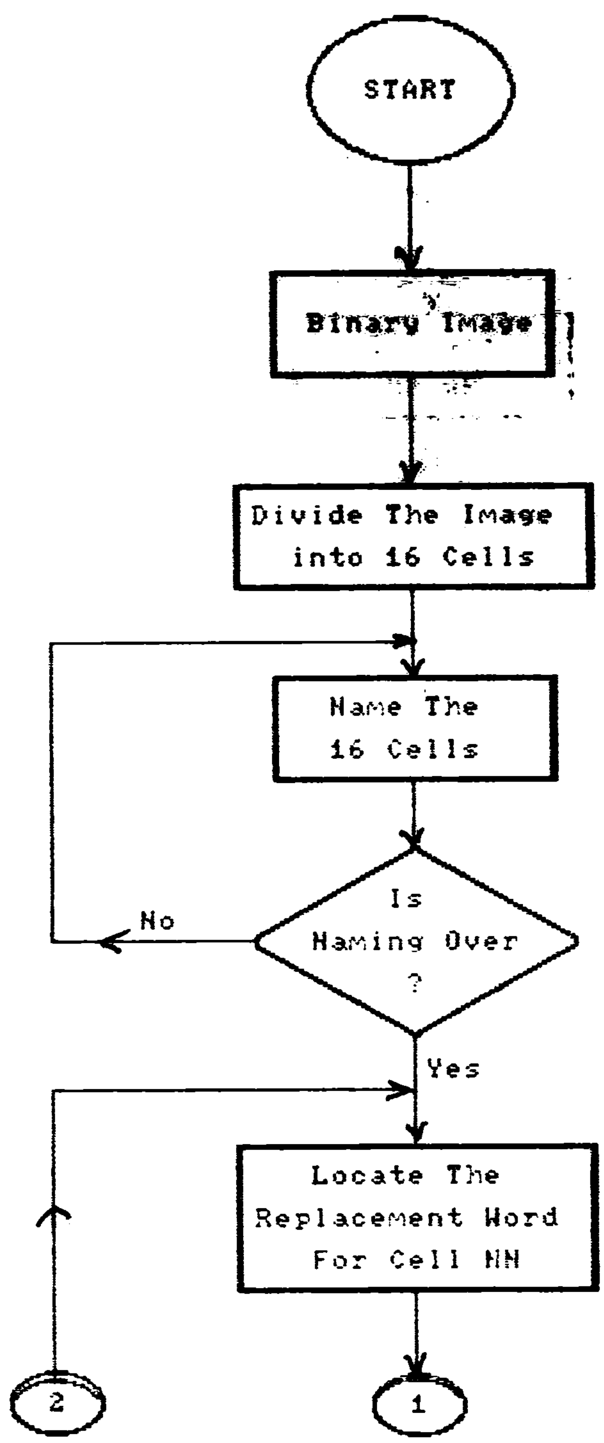

START
Binary Image
Divide The Image
into 16 Cells
Name The
16 Cells
Is
Naming Over
?
No
Yes
Locate The
Replacement Word
For Cell NN
2
1

Fig:- 7.4

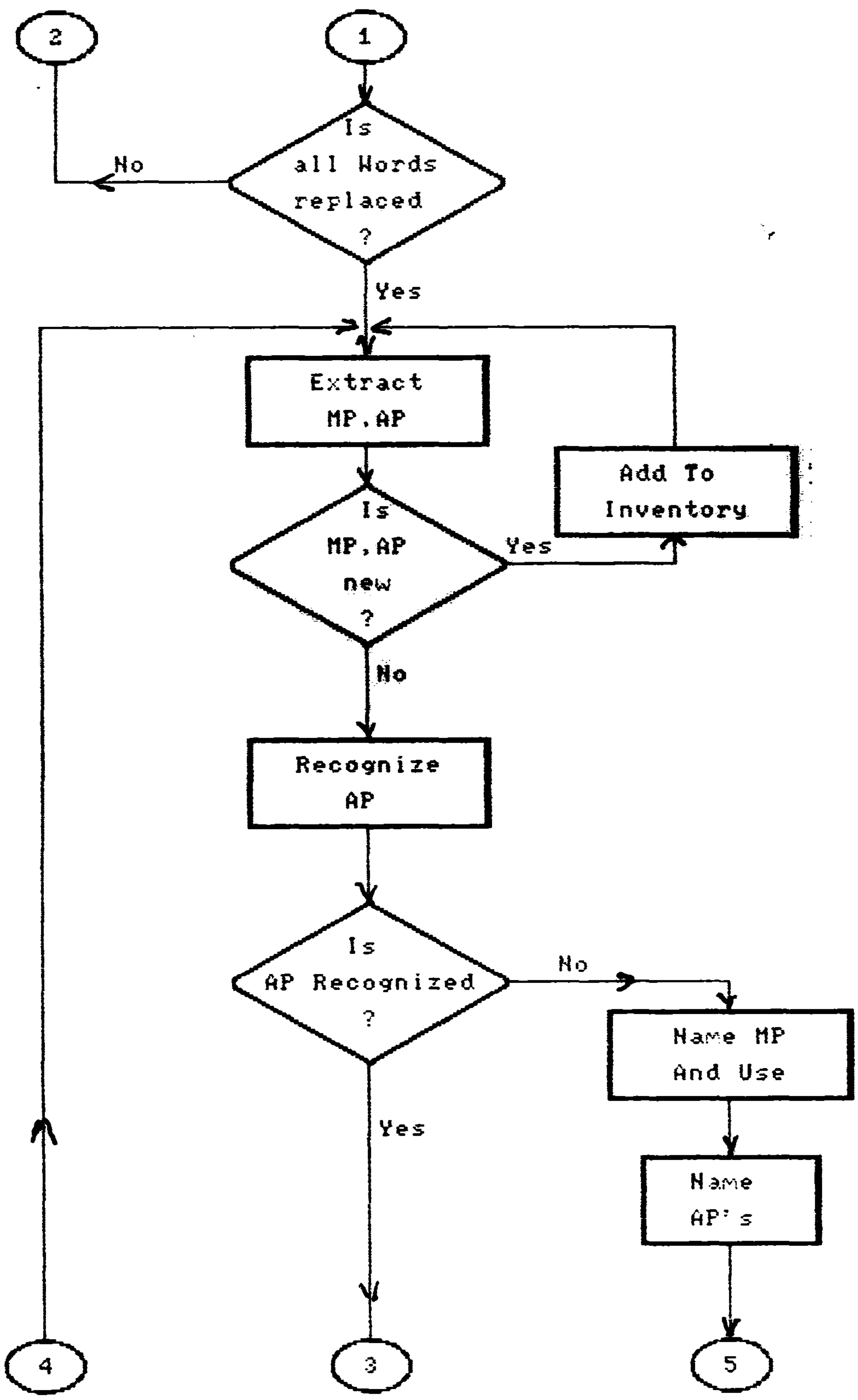

2
1
Is
all Words
replaced
?
No
Yes
Extract
MP,AP
Add To
Inventory
Is
MP,AP
new
?
Yes
No
Recognize
AP
Is
AP Recognized
?
No
Yes
Name MP
And Use
Name
AP's
4
3
5

Fig:- 7.4

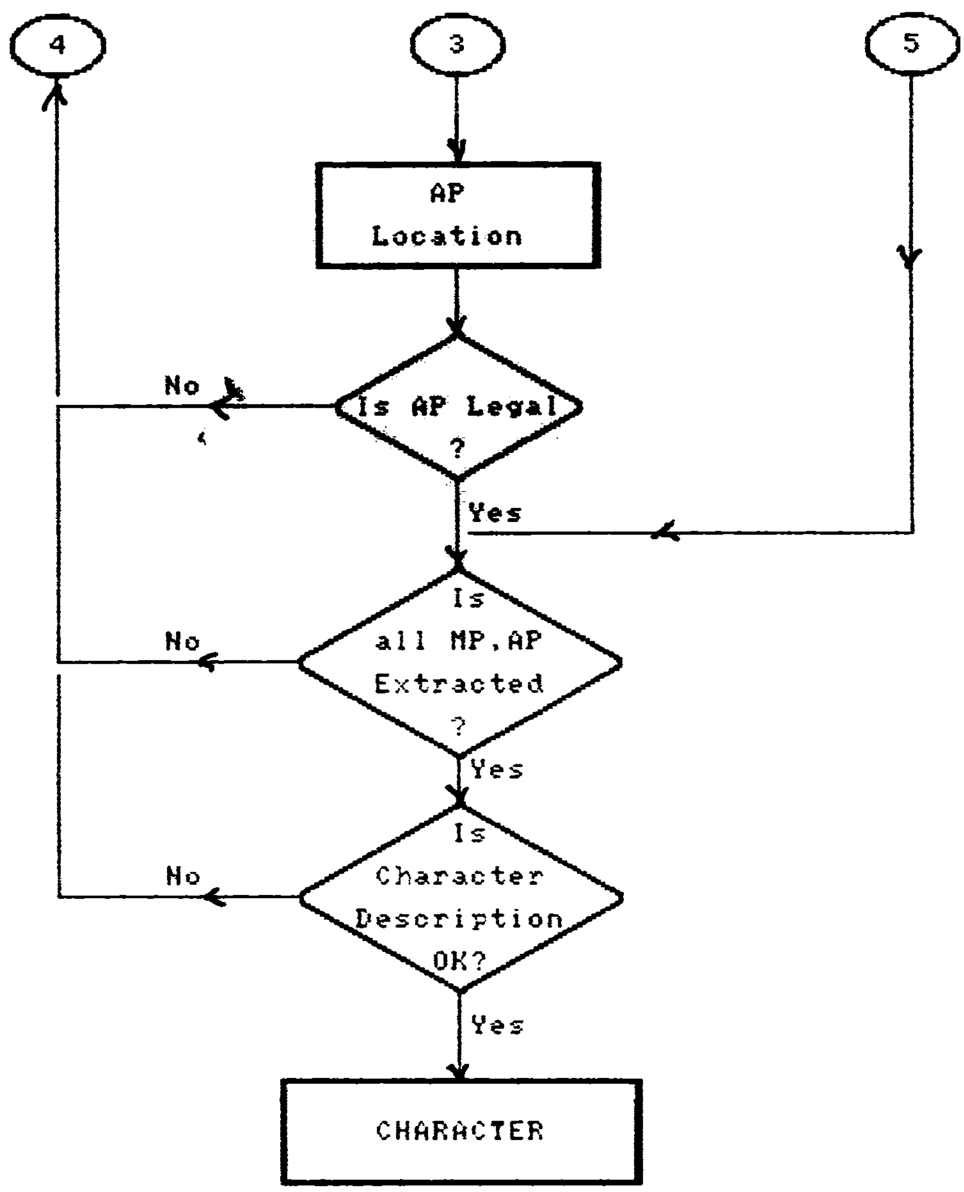


Fig: 7.4

FLOW CHART FOR COMPUTER IMPLEMENTATION

zero word name is detected the primitive extractor is brought into operation. The primitive is extracted and recorded. If no wordname is detected then the program goes to the north neighbor of the first cell. The search continues until a cell with a non zero word name is detected. The scanning of the image continues until all the cells are scanned, and all the primitives, both main and auxiliary, are extracted. The classification module now takes over and character description starts forming and finally the character classification is effected.

The description of the output is shown below. The character was classified primitive by primitive. The classification procedure is also presented in Fig: 7.3. Fig: 7.5 illustrates the fact that the characters do not loose their recognizability.

7.11 REALISATION OF THE DECISION TREE

The decision tree of this scheme can be realised by scanning the standard description inventory in a particular mode. The character description is written as a string from left to right. In the scan operation also, the build up of the description of the unknown character is made in the same fashion, that is from left to right. With the addition of each new main or auxiliary primitive the new description is compared with the standard inventory. The members with which this new description does not agree are ignored. With the growth in the addition of each new primitive resulting in the development of the description, the number of members with which the description agrees decrease and finally the agreement comes down to just one character. Even in such cases, the character search is continued along the decision tree until the final node is reached.

7.12 THE GRAPHICAL REPRESENTATION OF THE DECISION TREE

A part of the decision tree is presented in Fig: 7.6. Here the recognition of the characters " a " and " o " is presented. The decision process is explained below for the purpose of explaining the graphical representation of the decision tree.

On spotting the first primitive, a QLVL at the cell 41, the scan moves in a vertical direction moving on to the cell 31 and it spots the first vertical line, a FLVL or a TLVL. By the identification of this Main Primitive the character class reduces to a, b, c, e, h, o, u and w. The HL [R1] does not eliminate any member from the group.

However, the extraction of 'GC' eliminates the character "c" from the class. Discovery of VL [C3] eliminate both "o" and "e" which leaves behind only the character "a". Although only one or two more auxiliary features are re quired for the final classification of the character, all the remaining auxiliary primitives are extracted.

7.13 CONVENTIONS ADOPTED FOR CHARACTER SCANNING

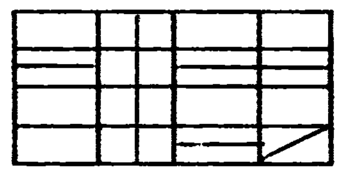
A set of conventions are formulated for the scanning of the modified character image. These are not invariant. Other conventions are also possible. The details given below cover all the characters. The scan procedure starts from the left bottom cell.

Generally scanning conventions are needed at junctions. At the first junction the system may encounter 1, 2, or 3 limbs. In the case of 1 limb junctions there is no difficulty. In the case of 2 or 3 limb junctions a helical scan will be followed. This means that the limb encountered first by the scan in its operation will be followed only at the end. The scan will proceed to the top of the vertical limb and continue from there to the next primitive moving in a helical path. The scan will come down to the second top primitive if the helical scan is not possible

FILTERED IMAGE OF CHARACTER	IT'S GRAPHICAL REPRESENTATION

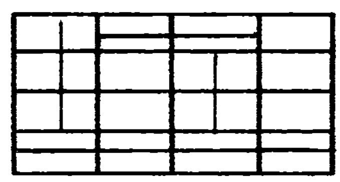
000 3A1 000 000
3A3 3A1 3A3 3A3
000 3A1 000 000
000 3A1 3A3 3A2

CHARACTER "[illegible]"



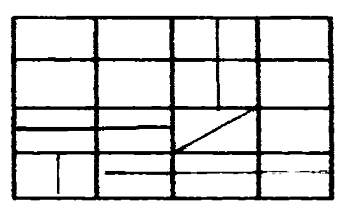
3A1 3A3 3A3 000
3A1 000 3A1 000
3A1 000 3A1 000
3A3 3A3 3A3 3A3

CHARACTER "a"



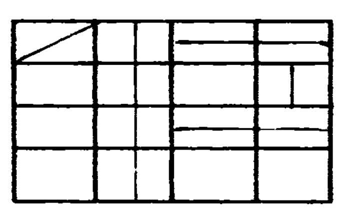
000 000 3A1 000
000 000 3A1 000
3A3 3A3 3A2 000
3A1 3A3 3A3 3A3

CHARACTER "d"



3A2 3A1 3A3 3A3
000 3A1 000 3A1
000 3A1 3A3 3A3
000 3A1 000 000

CHARACTER "p"



3A1 000 000 000
3A1 000 000 000
3A1 3A3 3A3 3A1
3A1 000 000 3A1

CHARACTER "h"

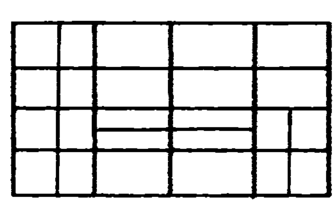


Fig :- 7.5

IMAGES OF A FEW CHARACTERS TO SHOW THAT THE FILTERING DOES NOT AFFECT RECOGNIZABILITY

It may be noted that the scanning of the modified image can be done in more than one way. Description of the character is scan mode dependent, that is, if scanned differently, even if the basis, that is the primitives, are kept the same, the description of the character will change.

7.14 ADDITIONAL BRANCHES IN THE DECISION TREE

Fig: 7.2 shows a decision tree which can be used for the classification of lower case English language characters. If a different set of data is to be classified using the decision tree, the decision tree should be modified to accommodate the members of this new class. Because of the openended inventories used in this scheme the decision tree can be made to accept more members in it, which include both new characters and variants of the existing characters. The extension of the decision tree to include both these types is shown in Fig: 7.7. A comparison of Fig: 7.7 with Fig: 7.2 show that the variants shown are new parallel branches added to the original decision tree.

7.15 RECOGNITION OF CHARACTER VARIANTS WITHOUT ALTERING THE DECISION TREE

This approach provides for the recognition of character

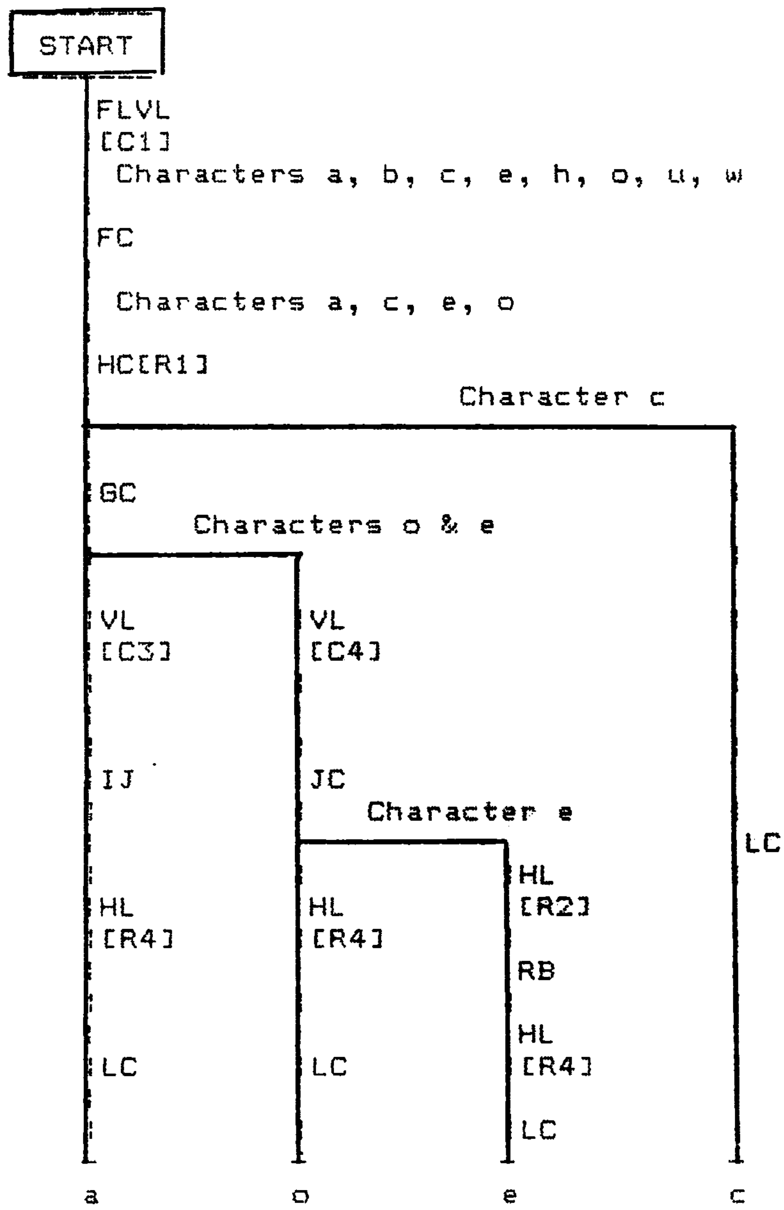


Fig: 7.6

REPRESENTATION OF THE DECISION TREE

variants without altering the decision tree. The two types of variants generally seen in character recognition problem are:

1) Separation of primitives in characters.

2) Extra primitives in characters.

In the case of characters with separated primitives a rule is formulated for an acceptable limit in the distance of separation. in the case of extra primitives another rule is laid down for the number of extra primitives in one character. The logic of classification can be altered slightly to accommodate these variants also.

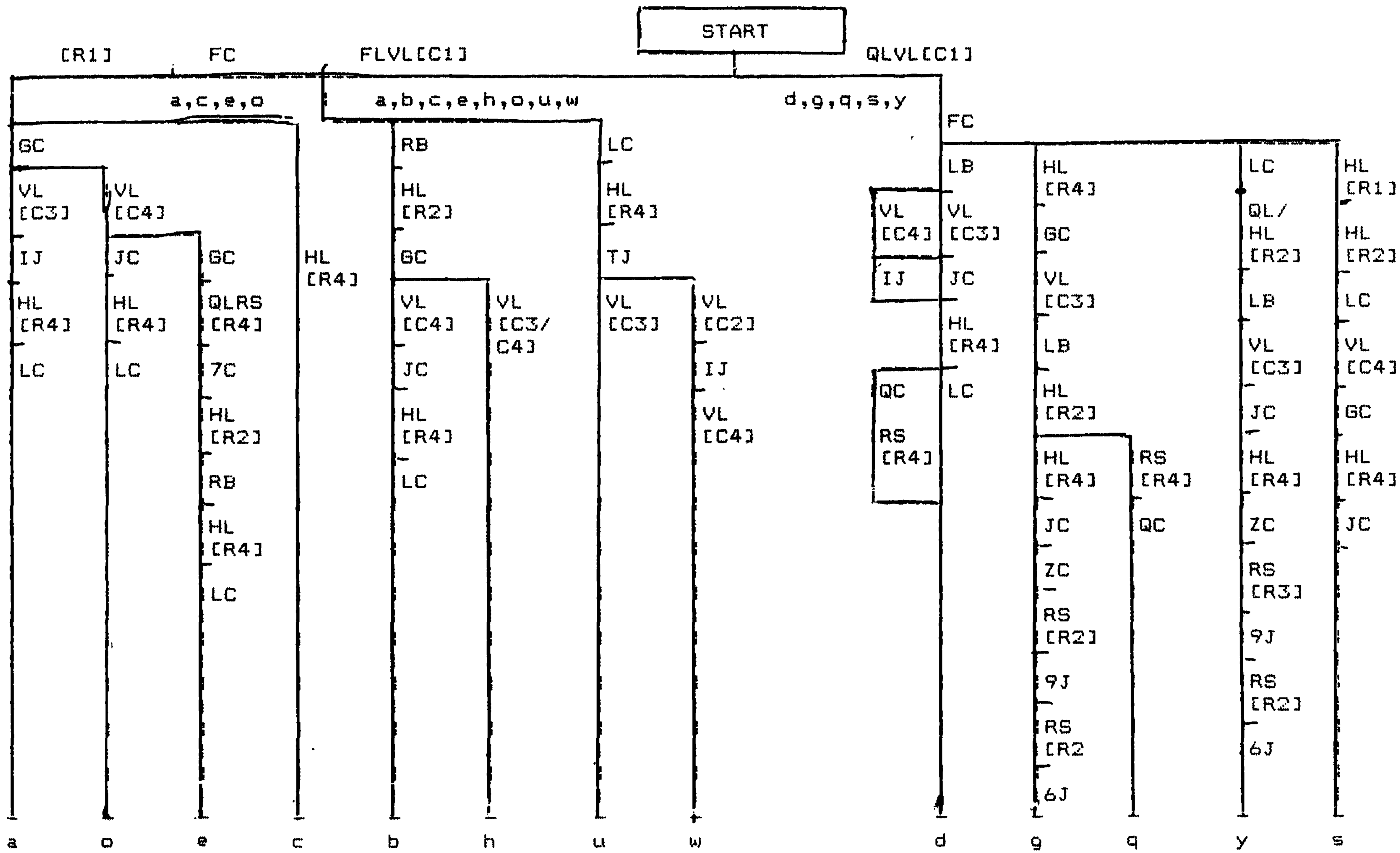


Fig: 7.7

VARIANTS IN THE DECISION TREE

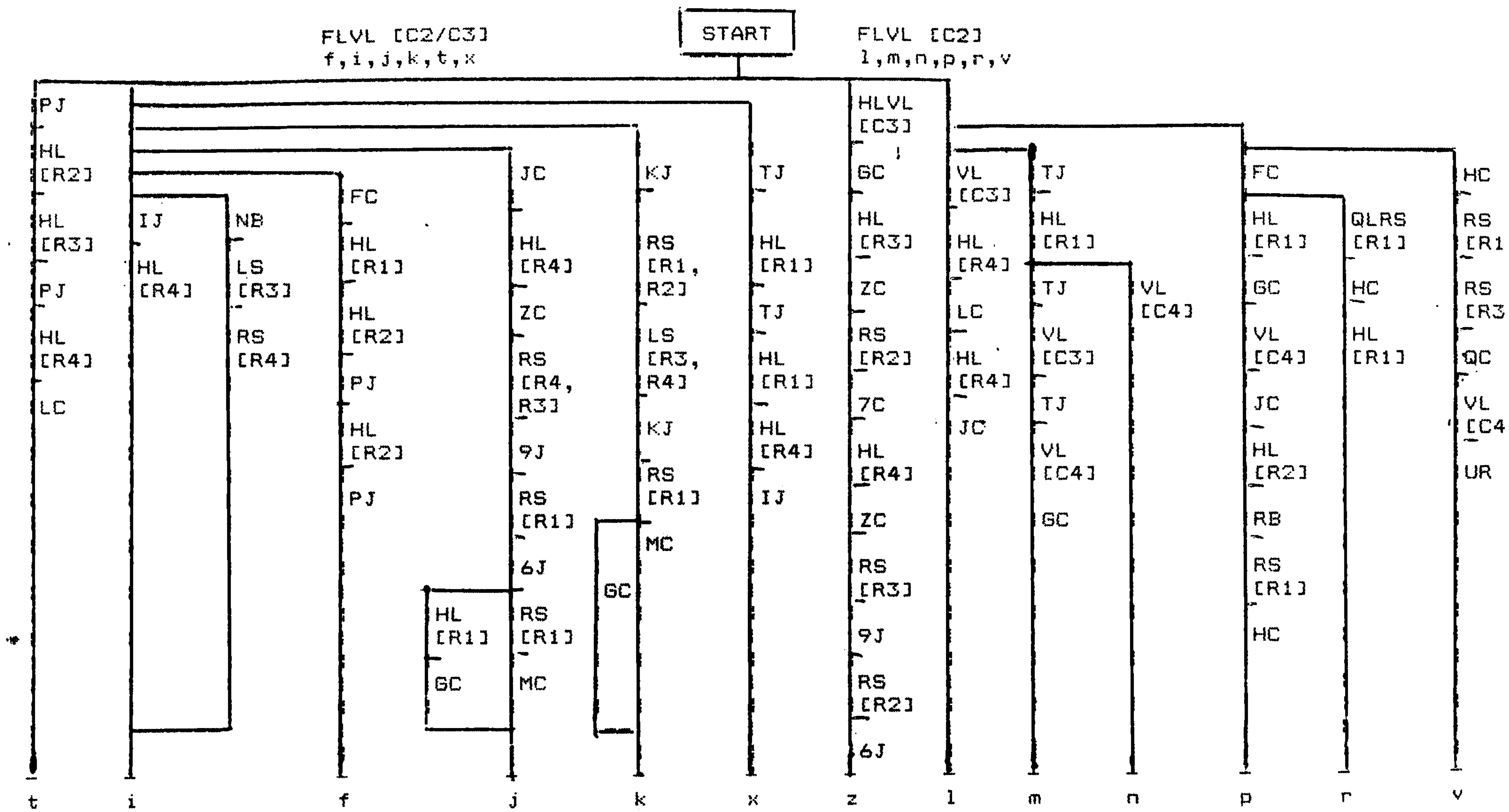


Fig: 7.7

VARIANTS IN THE DECISION TREE

CHAPTER - VIII CONCLUDING REMARKS

8.1. SUMMARY

This thesis has presented the following aspects for the recognition of LOWER CASE ENGLISH LANGUAGE HANDWRITTEN CHARACTERS.

1) A new uniform division of the character image into 144 pixels grouped as 3 * 3 pixel cells in four rows and four columns is shown possible and the 144 pixel picture description is sufficient for the description and analysis of the types of images handled. Single character images are captured and from the true size, the division into regions is carried out. Scaling is adopted for recording the binary image.

The advantage of this new proposal is that, once the binary image is available, true size of the character is immaterial, since each character image occupies only 144 bits of memory space. A seven character group occupies only 1K memory which is much less than most of the previous methods.

The disadvantage is that, dynamic sensing of cell boundaries may pose hardware realization problems. However

The advantage is the following. Since only 69 basic features are defined, syntactic processing is simpler. Also the rules are simple because of the small number of features. Complex rules can be eliminated in this approach. Processing required is only at the algebraic / logical level of words. There is no need for semantic processing or complex matrix operations.

One likely disadvantage is that minor ambiguities should be resolved at the word formation stage. Else it may lead to wrong definition of features.

4) By defining a new three stage filtering procedure, reduction in the dimensionality of the problem is achieved. Filtering defined in this thesis is a new concept related to order filtering. Each pixel in a cell is treated as a sample point and its distance is measured from the cell boundary. If the distance exceeds a threshold, that pixel value is either changed or nulled. Repeated filtering lead to uniform contours, which are lines symmetric with reference to the boundaries of the cell. Further, this new concept automatically takes care of the line width problems and hence line thinning algorithms are not needed for further processing.

5) Primitives and features defined in this work are closely related to the features extracted and defined earlier, is sufficient for the formulation of a decision tree classifier. This is smaller than the sets used by other researchers.

The advantage of the scheme is the following. The decision tree for classification can be reduced to a table look up operation in a large number of cases. This is particularly true for geometrically symmetric characters and shapes like, lower case English language handwritten characters. By virtue of the fact that the auxiliary primitives are a closed subset of the main primitives which in turn is a connected subset of the description rules, the classification scheme is rendered very flexible. Further, the scheme provides for open-ended inventory like the addition of new primitives, alternate routing in the decision paths and reduced search space at the end of each limb traverse.

6) The performance of the system is comparable to other existing cursive script recognition systems. Some portions of the work, like data input, can be done manually. This leads to slow speed of the system. Once automatic data input is used, the system can work quite fast.

The main advantage of the system is: As against the simpler problem of hand printed characters, the more complex cursive script recognition is tackled in this work. Similarly signature analyzers for recognition handle only a finite set of character types, for example, the signature of a finite number of account holders in a bank. There is no such constraint or limitation in the proposed scheme.

7) An important aspect of the thesis is the open-ended inventories for primitives and character description. The use of open-ended inventories helps in the addition of new primitives as well as new characters and variants of the characters itself. This is of great use while handling large data sets.

8.2 RESULTS OF THE THESIS

The scheme was mainly intended for implementation in small machines. However, the scheme can also be used in more powerful and large machines. The proposal was implemented in a PC environment using IBM compatible machines. The classification routine was developed using a small set of lower case English language characters comprising characters of varying complexity. Subsequently a few more sets of characters were added to the original set. The following conclusions are made from the tests conducted on these sets.

1) The grid size of 12 *12 provides sufficient information for further processing.

2) The new vocabulary developed for image coding is very useful for further processing like FILTERING, PRIMITIVE EXTRACTION and CLASSIFICATION. This new vocabulary reduces the computational efforts needed. Also the use of this new vocabulary makes the use of small machines possible.

3). The introduction of the hierarchy amongst the auxiliary primitives, main primitives and the character description itself is very useful in the extraction of primitives.The auxiliary primitives can be deduced from the presence of main primitives on which these auxiliary primitives reside.

4). The openended inventories make it possible to include more members in the character set, more variants for the same character and more primitives for the same character.

5). The decision tree has provided an interaction between the various units of the scheme. The structure of the decision tree selected is most suited for the purpose of this classification scheme. Also the structure of the decision tree is based on the description of the character. Samples with separated primitives and extra primitives were recognized by the scheme without much difficulty.

6). An advantage of the scheme is that it automatically provides for the variations in the size, shape and orientation of the character.

8.3 SUGGESTIONS FOR FUTURE WORK AND THE RANGE OF THE CURRENT WORK

This new classification scheme was tested on a set of 1352 character samples. The data set was generated in two steps and using two groups of people, school/college going children whose handwriting is in the formative stage and elderly people in business/employment ..etc. Each person was requested to write the set of 26 characters, in a previously prepared graph sheet to enable making a 12*12 grid. These grids were of varying sizes and shapes and fifty two sample sets were collected for the analysis. The only constraint proposed was to request each individual to write the character close in size to a 12*12 grid. The use of a very large sample set will improve the various decision mechanisms in the scheme. The openended inventories used for Auxiliary Primitives, Main Primitives and the character description and also the existence of the hierarchy among these three allows for the increase of the data set. The larger data sets may need refining of the primitive extraction routines. New characters and characters composed of new primitives can be added to test the performance of the scheme.

Certain areas of work related to the new concepts introduced in this thesis clearly fall beyond the scope of the thesis. Further research areas are therefore seen possible, and are listed below.

1) Segmentation algorithms for cursive script or images where the character size may show variations of more than 1:3.

2) Corrupted character image recognition like, missing or obliterated parts of a character.

3) Labeling and feature extraction for three dimensional object images. It is likely that a pixel cell in a 3*3*3 format is to be defined. The number of basic words or features is estimated at $[36^3 - 3 * 36^2]$. The proof and accuracy of this is to be ascertained.

4) The possible logical relationship between the features in cells and primitives in paragraphs is to be established.

REFERENCES

1) [AGUI] AGUI.T, NAKAJIMA.M, & ARAI.Y;
"AN ALGEBRAIC APPROACH TO THE GENERATION AND DESCRIPTION OF BINARY PICTURES" ;IEEE TRANS. PAMI, VOL PAMI 4, NO. 6, PP 635 to 641, 1982.

2) [BERT] BERTHOD, MARC;
"ON-LINE ANALYSIS OF CURSIVE WRITING" in COMPUTER ANALYSIS AND PERCPTION; VOL.I VISUAL SIGNALS, CHAPTER 4, PP 55 to 81, Eds. MARC BERTHOD

3) [BOZI] BOZIMOVIC R., AND SRI HARI S. N.;
"A STRING CORRECTION ALGORITHM FOR CURSIVE SCRIPT WRITING", TRAN. IEEE PAMI, VOL. PAMI-4, NO. 6, NOV. 1984, PP 655 to 663.

4) [CHIN] CHINNUSWAMY. P, & KRISHNA MOORTHY S.G;
"RECOGNITION OF HAND PRINTED TAMIL CHARACTERS", PATTERN RECOGNITION, VOL.12, PP 141 to 152, 1980.

5) [KSFU1] FU. K. S;
"A STEP TOWARDS THE UNIFICATION OF SYNTACTIC AND STATISTICAL PATTERN RECOGNITION", IEEE PAMI, VOL. PAMI - 5, PP 200 to 205, MARCH 1983.

6) [KSFU2] FU. K. S;
"SYNTACTIC PATTERN RECOGNITION",in HAND BOOK OF PATTERN RECOGNITION AND IMAGE PROCESSING, Eds. T. Y. YOUNG AND K. S. FU, ACADEMIC PRESS, NEW YORK, 1986.

7) [GRIM] GRIMSDALE R. L., SUMNER F.H., TUNIS C.J, and KILBURN T.;
"A SYSTEM FOR THE AUTOMATIC RECOGNITION OF PATTERNS", Proc. OF IEE, VOL. 108 (Part-B), PP 210 to 221, 1959.

8) [HANA] HANAKATA K.;
"FEATURE SELECTION AND EXTRACTION FOR DECISION THEORETIC AND STRUCTURAL APPROACH"., in PATTERN RECOGNITION - THEORY AND APPLICATIONS, Eds. FU. K.S & WHINSTON A.B, NOORDHOFF - LYEDEN -1977.

9) [KAHA] KAHAN S., PAVLIDIS T., BAIRD H. S.;
" ON THE RECOGNITION OF PRINTED CHARACTERS OF ANY FONT AND SIZE ", Trans. OF IEEE PAMI, VOL.PAMI-9, MARCH 1987

10) [KANE] KANEFF S.;
"PATTERN COGNITION AND ORGANIZATION OF INFORMATION", in FRONTIERS OF PATTERN RECOGNITION, Ed. WATANABE S., PP 193 to 223, ACADEMIC PRESS, NEW YORK 1972.

11) [KURZ] KURZYNSKI M. W.;
"THE OPTIMAL STRATEGY OF TREE CLASSIFIER", PATTERN RECOGNITION , VOL. 16, NO. 1, PP 81 to 87, 1983.

12) [MER1] MERMELSTEIN P., AND EDEN M.;
" A SYSTEM FOR AUTOMATIC RECOGNITION OF CONNECTED HAND WRITTEN WORDS", PROC. F. J. C. C., SAN FRANCISCO, SPARTAN, 1964.

13) [MER2] MERMELSTEIN P., AND EDEN M.;
"EXPERIMENTS ON COMPUTER RECOGNITION OF CONNECTED HAND WRITTEN WORDS", INFORMATION AND CONTROL, 1964.

14) [NAR1] NARASIMHAN R., AND REDDY V.S.N.;
" A SYNTAX AIDED RECOGNITION SCHEME FOR HAND PRINTED ENGLISH LETTERS", PATTERN RECOGNITION, VOL.3, PP 345 to 361, 1971.

15) [NAR2] NARASIMHAN R.;
"ON THE DESCRIPTION, GENERATION AND RECOGNITION OF CLASSES OF PICTURES", in AUTOMATIC INTERPRETATION AND CLASSIFICATION OF IMAGES, Ed. GRASSELI A., ACADEMIC PRESS, NEW YORK, 1969, PP 1 to 42.

16) [RABI] RABINOW J.;
"THE PRESENT STATE OF THE ART OF READING MACHINES", in PATTERN RECOGNITION, Ed. KANAL L. N., THOMPSON BOOK COMPANY, WASHINGTON, D, C., 1968.

17) [RAJA] RAJASEKARAN S. N. S. AND DEEKSHATULU B. N.;
"RECOGNITION OF PRINTED TELUGU CHARACTERS", COMPUT. GRAPHIC AND IMAGE PROCESSING, VOL.6, PP 335-360, 1977

18) [RAM] RAMMOHAN T. R., AND CHATTERJI B. N.;
"RECOGNITION OF DISTORTED KANNADA CHARACTERS", JOURNAL OF IETE, VOL.30, PP 223-225, NOV. 1984.

19) [RAY] RAY A. K., AND CHATTERJEE B.;
"DESIGN OF A NEAREST NEIGHBOR CLASSIFIER SYSTEM FOR BENGALI CHARACTER RECOGNITION", JOURNAL OF IETE, VOL.30, PP 226-228, NOV. 1984.

20) [ROSE] ROSENFELD A., KAK A. C.;
DIGITAL PICTURE PROCESSING, ACADEMIC PRESS, NEW YORK, 1982.

21) [SETH] SETHI I. K., AND CHATTERJEE B.;
"MACHINE RECOGNITION OF CONSTRAINED HAND PRINTED DEVANAGARI", PATTERN RECOGNITION, VOL. 9, 1977, PP 69 to 75.

22) [SIMO] SIMON HAYKIN.;
in ADAPTIVE FILTER THEORY, PRENTICE HALL, ENGLE WOOD CLIFFS, NEW JERSEY.

23) [SINHA] SINHA R. M. K., AND MAHABALA H. N.;
"MACHINE RECOGNITION OF DEVANAGARI SCRIPT", IEEE TRANS., VOL. SMC-9, PP 435-441, AUG. 1979.

24) [SIRO] G. SIROMONY, R. CHANDRASEKARAN AND M. CHANDRASEKARAN.;
"COMPUTER RECOGNITION OF PRINTED TAMIL CHARACTERS", PATTERN RECOGNITION, VOL. 10, PP 243-248, APR. 1978.

25) [SOM] SOM A., AND NATH A. K..;
"ON SOME METHODS OF SEQUENTIAL PATTERN RECOGNITION", ISI SYMP DIGITAL TECH AND PATTERN RECOGNITION, PP 246-265, 1977.

26) [SMAN] SCHUERMANN J., AND DOSTER W.;
"DECISION THEORETIC APPROACH TO HEIRARCIAL CLASSIFIER DESIGN", PATTERN RECOGNITION VOL. 17, NO. 3, PP 359 to 369, 1984.

27) [STAL] STALLING W. W.;
"CHINESE CHARACTER RECOGNITION", in SYNTACTIC PATTERN RECOGNITION, APPLICATIONS, Ed. FU K. S., SPRINGER-VERLAG, 1977, PP 95 to 123.

28) [SUE1] SUEN C. Y.;
"CHARACTER RECOGNITION BY COMPUTERS AND APPLICATIONS", CHAPTER 23, in HAND BOOK OF PATTERN RECOGNITION AND IMAGE PROCESSING, Ed YOUNG T. Y., AND FU K. S., ACADEMIC PRESS, NEW YORK, 1984.

29) [SUE2] SUEN C. Y.;
"DISTINCTIVE FEATURES IN AUTOMATIC RECOGNITION OF HAND PRINTED CHARACTERS", in SIGNAL PROCESSING 4, 1982, PP 193 to 207.

30) [SUE3] SUEN C. Y., MORI S.;
"STANDARDIZATION AND AUTOMATIC RECOGNITION OF HAND PRINTED CHARACTERS",in COMPUTER ANALYSIS AND PERCEPTION, VOL. I, VISUAL SIGNALS, CHAPTER 3, Ed, BERTHOD M., ACADEMIC PRESS, NEW YORK.

31) [SUE4] SUEN C. Y.;
"FEATURE EXTRACTION IN AUTOMATIC RECOGNITION OF HAND PRINTED CHARACTERS", in SIGNAL PROCESSING: THEORIES AND APPLICATIONS, Eds. KUNT M., AND F. de COULON, NORTH HOLLAND PUBLISHING CO., EURASIP, 1980.

32) [SUE5] SUEN C. Y., BERTHOD M., AND MORI S.;
"AUTOMATIC RECOGNITION OF HAND PRINTED CHARACTERS-THE STATE OF THE ART", PROC. OF IEEE, VOL.68, NO.4, PP 469 to 487, 1980.

33) [UHR] LEONARD UHR.; PATTERN RECOGNITION , LEARNING AND THOUGHT, PRENTICE - HALL, NEW JERSEY, 1973.

34) [UMAN] ULLMAN J. R.;
PATTERN RECOGNITION, BUTTER WORTH, 1973.

35) [WANG] WANG Q.R., AND SUEN C. Y.;
"ANALYSIS AND DESIGN OF A DECISION TREE BASED ON ENTROPY REDUCTION AND IT'S APPLICATION TO A LARGE CHARACTER SET RECOGNITION", TRANS. IEEE PAMI, VOL. PAMI-6, NO.4, 1984, PP 406 to 417.

36) [YOSH] YOSHIDA M., IWATA K., YAMAMOTO E., MASUI T., AND KABUYAMA Y.;
"THE RECOGNITION OF HAND PRINTED CHARACTERS INCLUDING 'KATAKANA', 'ALPHABETS' AND 'NUMERALS'", PROC. OF THE THIRD INT. CONF. ON PATTERN RECOGNITION, NOV. 1976, CALIFORNIA, PP 645 to 649.

GENERAL REFERENCES

GR1) MACHINE RECOGNITION OF PATTERNS, Ed. AGRAWALA A.K., ACADEMIC PRESS, NEW YORK, 1977.

GR2) PATTERN RECOGNITION THEORY AND APPLICATIONS, Eds. WHINSTON A. B., AND FU K. S., NOORDHOFF-LEYDEN,1977.

GR3) HAND BOOK OF PATTERN RECOGNITION AND IMAGE PROCESSING, Eds. YOUNG T. Y., AND FU K.S., ACADEMIC PRESS, NEW YORK, 1984.

GR4) PATTERN RECOGNITION, LEARNING AND THOUGHT, LEONARD UHR. PRENTICE-HALL, NEW JERSEY, 1977.

GR5) DIGITAL PICTURE PROCESSING, ROSENFELD A. & KAK A. C. ACADEMIC PRESS, NEW YORK, 1982.

GR6) PATTERN CLASSIFICATION AND SCENE ANALYSIS, DUDA R. O & HART P. E., WILEY-INTERSCIENCE, 1978, NEW YORK.

GR7) SYNTACTIC PATTERN RECOGNITION AN INTRODUCTION, GONZALEZ R. C., & THOMASON M. G., ADDISON-WESLEY, READING, MASSACHUSETTS, 1978.

GR8) PATTERN RECOGNITION PRINCIPLES, TOU J. T. & GONZALEZ R. C., ADDISON-WESLEY, READING, MASSACHUSETTS, 1974.

GR9) SEQUENTIAL METHODS IN PATTERN RECOGNITION AND MA CHINE LEARNING, FU K. S., ACADEMIC PRESS, NEW YORK, 1968.

....

- G5451 -

PUBLICATIONS BASED ON THE THESIS

1) R. K. TAMPI & C. S. SRIDHAR, " A NEW CODING AND FILTERING APPROACH TO AID IN THE RECOGNITION OF NEAR GEOMETRIC SHAPES", Proc. of the 7th International Conference on Pattern Recognition, July-August, 1984, Montreal, Canada.

2) --------, "FEATURES, DEFINITION AND EXTRACTION FOR IMAGES WITH CONTOURS", Proc. of Con.786, SPIE, April'87 NEWPORT BEACH, CALIFORNIA, U. S. A

3) --------,"SEGMENTATION OF HANDWRITTEN CHARACTERS",Proc. of the 8th International Conference on Pattern Recognition, September, 1986, Paris, France.

4) --------, "A NEW CODING AND FILTERING APPROACH TO AID IN THE RECOGNITION OF NEAR GEOMETRIC SHAPES", Proc. of the International Symposium on Electronic Devices Circuits and Systems", 1987, Kharagpur, India.

5) --------, "NUMBER OF FEATURES FOR HANDWRITING RECOGNITION", 12th GRETSI Symposium, 1989, Juan Les Pins, France.

6) -------, "FIXED REGION LABELING AND FEATURE EXTRACTION" Communicated to ELECTRONIC LETTERS for possible publication.

....