STRUCTURAL RECOGNITION OF Handwritten numeral strings.

AMIRA M. K. BADRELDIN

University of Windsor

Follow this and additional works at: https://scholar.uwindsor.ca/etd

Recommended Citation
https://scholar.uwindsor.ca/etd/3618

This online database contains the full-text of PhD dissertations and Masters’ theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000ext. 3208.
NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us a poor photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30. Please read the authorization forms which accompany this thesis.

THIS DISSERTATION HAS BEEN MICROFILMED EXACTLY AS RECEIVED
STRUCTURAL RECOGNITION OF
HANDWRITTEN NUMERAL STRINGS

by

© Amira M.K. Badreldin

A Dissertation

Submitted to the Faculty of Graduate Studies through the Department of Electrical Engineering in partial fulfillment of the requirements for the Degree of Doctor of Philosophy at the University of Windsor

Windsor, Ontario, Canada

1985
To my daughter, my husband and my parents

for all that they mean to me.
ABSTRACT

This thesis discusses the development of algorithms for the recognition of handwritten numeral strings in their various forms viz. isolated, broken and connected.

For isolated numerals, the use of a new class of Fourier shape descriptors derived from the contours of the numeral together with a new class of topological features is shown to yield high recognition accuracy (= 98%). For isolated and possibly broken numerals, a syntactic recognition algorithm that utilizes features derived from the left and right profiles of the numerals is shown to yield fast and accurate recognition. Finally, an algorithm for segmenting connected handwritten numeral strings has been developed and is shown to yield accurate segmentation. The segmented numerals are then identified by the syntactic recognition system.
ACKNOWLEDGMENTS

I would like to express my sincere thanks and appreciation to my supervisor, Dr. M. Shridhar, for his invaluable advice, guidance and constant encouragement throughout the progress of this thesis. No one could ask for a better advisor. Dr. Shridhar has been a constant source of ideas and inspiration during the course of this research. The simplest truth would be to say that this thesis could not have been written without him.

My sincerest appreciation goes to my thesis committee: Prof. P.H. Alexander, Dr. M. Ahmadi, and Dr. G. Rankin, for their encouragement, advice and critical appreciation during the course of this research.

The advice of the other faculty members of the Electrical Engineering Department is also gratefully acknowledged.

I extend my sincerest thanks and gratitude to my husband, Dr. M.A. Ismail, and to my parents for their moral support, help and constant encouragement at all times.

I thank the Natural Science and Engineering Research Council of Canada (NSERC) for its financial support during the academic years 1983 - 1985.

Thanks are also due to Mrs. Linda Kennedy for her diligence and excellence in typing this thesis.
### TABLE OF CONTENTS

**ABSTRACT** ................................................................. 1

**ACKNOWLEDGMENTS** ..................................................... ii

**TABLE OF CONTENTS** .................................................... iii

**LIST OF FIGURES** ........................................................ v

**LIST OF TABLES** ........................................................ ix

**LIST OF APPENDICES** ................................................... x

#### CHAPTER

**I. INTRODUCTION** ...................................................... 1

1.1 State-of-the-art - Literature Survey ............................... 2

1.1.1 Character Features ............................................. 2

1.1.2 Classifiers ..................................................... 5

1.2 Goals of the Thesis ................................................. 7

1.3 Outline of the Thesis .............................................. 8

**II. DATA COLLECTION AND PROCESSING** ............................. 9

2.1 Introduction ....................................................... 9

2.2 Digital Imaging and Processing .................................. 9

2.3 Image Processing and Contour Extraction ....................... 10

**III. NUMERAL RECOGNITION WITH FOURIER SHAPE DESCRIPTORS**
**AND TOPOLOGICAL FEATURES** ........................................ 15

3.1 Introduction ....................................................... 15

3.2 Derivation of Fourier Descriptors ................................ 17

3.3 Identification of Sub-classes .................................... 23

3.4 Recognition Algorithm ........................................... 28

3.5 Test Results ..................................................... 32

3.6 Border Transition Features ....................................... 33

3.7 Uniqueness of Border Transition Features ..................... 39

3.8 Summary .......................................................... 39
IV. STRUCTURAL RECOGNITION ALGORITHM FOR HANDWRITTEN NUMERALS

4.1 Introduction ........................................ 41
4.2 Character Description and Feature Extraction ............... 42
  4.2.1 Global Features .................................. 42
  4.2.2 Local Features .................................. 42
  4.2.3 Features for Broken Numerals ...................... 50
4.3 Relational Attributes ................................ 54
4.4 Syntactic Recognition Algorithm ........................ 54
  4.4.1 Determination of Groups .......................... 54
  4.4.2 Derivation of Chains ............................ 60
4.5 Production Rules ....................................... 64
4.6 Recognition Algorithm ................................ 67
4.7 Recognition Results ................................... 68
4.8 Summary ............................................... 68

V. CONTEXT-DIRECTED SEGMENTATION ALGORITHM FOR HANDWRITTEN NUMERAL STRINGS ....................................................... 73

5.1 Introduction ........................................ 73
5.2 Segmentation of Numeral Strings ........................ 76
  5.2.1 Segmentation Algorithm .......................... 77
  5.2.2 Procedure for Non-linear Segmentation .......... 79
  5.2.3 Determination of the Initial Scanning Point .... 90
5.3 Extension of the Algorithm ................................ 105
5.4 Recognition Algorithm ................................ 110
5.5 Summary ............................................... 110

VI. SUMMARY AND CONCLUSIONS .................................. 114

BIBLIOGRAPHY ............................................. 118

VITA AUCTORIS ............................................ 144
<table>
<thead>
<tr>
<th>Figure No.</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>11</td>
</tr>
<tr>
<td>2.2</td>
<td>13</td>
</tr>
<tr>
<td>2.3</td>
<td>13</td>
</tr>
<tr>
<td>2.4</td>
<td>14</td>
</tr>
<tr>
<td>3.1(a)</td>
<td>18</td>
</tr>
<tr>
<td>3.1(b)</td>
<td>18</td>
</tr>
<tr>
<td>3.2(a)</td>
<td>22</td>
</tr>
<tr>
<td>3.2(b)</td>
<td>22</td>
</tr>
<tr>
<td>3.2(c)</td>
<td>22</td>
</tr>
<tr>
<td>3.3</td>
<td>24</td>
</tr>
<tr>
<td>3.4</td>
<td>24</td>
</tr>
<tr>
<td>3.5(a)</td>
<td>26</td>
</tr>
<tr>
<td>3.5(b)</td>
<td>26</td>
</tr>
<tr>
<td>3.5(c)</td>
<td>27</td>
</tr>
<tr>
<td>3.6</td>
<td>29</td>
</tr>
<tr>
<td>3.7</td>
<td>29</td>
</tr>
<tr>
<td>3.8</td>
<td>34</td>
</tr>
<tr>
<td>3.9</td>
<td>38</td>
</tr>
<tr>
<td>4.1(a)</td>
<td>43</td>
</tr>
<tr>
<td>4.1(b)</td>
<td>43</td>
</tr>
<tr>
<td>4.2(a)</td>
<td>44</td>
</tr>
<tr>
<td>4.2(b)</td>
<td>44</td>
</tr>
</tbody>
</table>

2.1 Samples of the Numeric Data
   (a) Isolated Numerals
   (b) Numeral Strings
2.2 Digitized Image
2.3 Thresholded Image
2.4 Contours of the Numerals
3.1(a) Eight-neighbour Adjacent
3.1(b) Four-neighbour Adjacent
3.2(a) Original Numerals
3.2(b) Reconstructed Numerals Using 6 Fourier Descriptors
3.2(c) Reconstructed Numerals Using 15 Fourier Descriptors
3.3 Sub-classes of Numerals 6, 7, 8 and 9
3.4 Specimens of the Numeral '1'
3.5(a) Functional Mapping of the FDs of the Numeral '0'
3.5(b) Functional Mapping of the FDs of the Numeral '6'
3.5(c) Functional Mapping of the FDs of the Numeral '8'
3.6 Functional Mapping of the FDs of the Numeral '1'
3.7 Functional Mapping of the FDs of the Sub-classes of the Numeral '1'
3.8 Border Transitions
3.9 Flowchart of Two Stage Algorithm
4.1(a) Left and Right Profiles of the Numeral '5'
4.1(b) Left and Right Profiles of the Numeral '3'
4.2(a) Left and Right Profiles of the Numeral '9'
4.2(b) Left and Right Profiles of the Numeral '6'
4.3 Left and Right Profiles of the Numeral '1'

4.4(a) First Difference of the Left Profile of the Numeral '5' Shown in Fig. 4.1(a)

4.4(b) First Difference of the Right Profile of the Numeral '5' Shown in Fig. 4.1(a)

4.5 Specimens of Broken Numerals

4.6(a) Left and Right Profiles of the Broken Numeral '4'

4.6(b) Left and Right Profiles of the Broken Numeral '5' (With Vertical Gap)

4.6(c) Left and Right Profiles of the Broken Numeral '1' (With Horizontal Gap)

4.6(d) Left and Right Profiles of the Broken Numeral '5' (With Horizontal Gap)

4.7 Specimens of the Numeral '8'

4.8 Left and Right Profiles of the Numeral '8'

4.9 Production Rules of the Numeral '8'

4.10 Tree Representing the Numeral '8'

4.11 Portion of the Tree Used in the Recognition of the Numerals '0' to '2'

4.12 Sub-tree Used in the Recognition of the Numerals '0', '4', '6', and '8'

4.13 Misclassified Errors

5.1 Samples of Numeral Strings

5.2(a) Digitized Image

5.2(b) Thresholded Image

5.3 Linear Decision Boundary

5.4 Initial Scanning Point
5.5 Non-linear Decision Boundary
5.6(a) Scanning Direction for Algorithm (1)
5.6(b) Scanning Direction for Algorithm (2)
5.6(c) Scanning Direction for Algorithm (3)
5.6(d) Scanning Direction for Algorithm (4)
5.7 Segmentation Procedure
5.8 Horizontal Transitions
5.9 Numeral Strings With Three Horizontal Transitions
5.10(a) Transition Graph Recognizing the String $S_j^1(i)$
5.10(b) Transition Graph Recognizing the String $S_j^2(i)$
5.11 Non-connected Numeral Strings with Three Horizontal Transitions
5.12 Connected Numerals
5.13 Samples of Connected Numeral Strings with Three Horizontal Transitions
5.14 Two Numeral Strings Generating the Same String $S_j^1(i)$
5.15(a) Global Profiles Evaluated Over the Whole Frame
5.15(b) Global Profiles Evaluated Till the Middle of the Frame
5.16 Connected Strings
5.17 Numeral Strings With Three Digits
5.18 Left and Right Profiles of Segmented Numerals with $HT = 2$
5.19 Left and Right Profiles of Segmented Numerals with $HT = 3$
5.20 Left and Right Profiles of Segmented Numerals with HT = 4 113
A4.1 Profiles of Segmented String With HT = 2 136
A4.2 Profiles of Segmented String With HT = 2 136
A4.3 Profiles of Segmented String With HT = 4 137
A4.4 Profiles of Segmented String With HT = 3 137
A4.5 Profiles of Segmented String With HT = 3 138
A4.6 Profiles of Segmented String With HT = 3 138
A4.7 Profiles of Segmented String With HT = 3 139
A4.8 Profiles of Segmented String With HT = 3 139
A4.9 Profiles of Segmented String With HT = 3 140
A4.10 Profiles of Segmented String With HT = 3 140
A4.11 Profiles of Segmented String With HT = 3 141
A4.12 Profiles of Segmented String With HT = 3 141
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table No.</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Scanning Directions for BTFs</td>
<td>35</td>
</tr>
<tr>
<td>3.2</td>
<td>BTFs for the Numerals 2, 4, 5, 6, 7, and 9</td>
<td>36</td>
</tr>
<tr>
<td>3.3</td>
<td>Range of the Horizontal and Vertical Averages</td>
<td>40</td>
</tr>
<tr>
<td>4.1</td>
<td>Predicates Used in the Recognition of the Numerals '0', '1', ..., '9'</td>
<td>55</td>
</tr>
<tr>
<td>4.2</td>
<td>Groups of Numerals '0' to '9'</td>
<td>61</td>
</tr>
<tr>
<td>5.1</td>
<td>Scanning Directions</td>
<td>82</td>
</tr>
<tr>
<td>5.2</td>
<td>Determination of the Initial Scanning Point for Non-Connected Numeral Strings</td>
<td>101</td>
</tr>
<tr>
<td>5.3</td>
<td>Determination of the Initial Scanning Point for Connected Numeral Strings</td>
<td>107</td>
</tr>
<tr>
<td>A3</td>
<td>Sub-classes of the Numerals '0' through '9'</td>
<td>130</td>
</tr>
</tbody>
</table>
**LIST OF APPENDICES**

<table>
<thead>
<tr>
<th>APPENDIX NO.</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Properties of the Functional Mapping Technique</td>
<td>125</td>
</tr>
<tr>
<td>II</td>
<td>An Efficient Version of the K-MEANS Clustering Algorithm</td>
<td>128</td>
</tr>
<tr>
<td>III</td>
<td>Sub-classes of the Numerals '0' Through '9'</td>
<td>130</td>
</tr>
<tr>
<td>IV</td>
<td>Profiles of Segmented Numeral Strings</td>
<td>135</td>
</tr>
<tr>
<td>V</td>
<td>Data Base</td>
<td>142</td>
</tr>
</tbody>
</table>
Chapter I

INTRODUCTION

Machine recognition of handwritten characters has been the subject of extensive investigations for more than two decades. The potential benefits stemming from the research, as evidenced by the large number of patents on character recognition system, are the machine processing of large volumes of character data, such as, postal code reading, text to speech conversion systems, shape recognition and classification of chromosomal and cellular data.

Machine recognition of handwritten characters is a very challenging problem, mainly due to the large numbers of characters and symbols and the tremendous variability of the shapes resulting from the writing habits and styles of human beings.

In a study on human perception, it was found that an error rate of 4 per cent occurred when the participants were asked to recognize handwritten characters in the absence of context [1], [2], [3], and [4].

This thesis addresses the problem of handwritten numeral recognition in its most general formulation. The fact that the handwritten numerals could be broken, and possibly connected is taken into consideration in the development of recognition algorithms.

It is relevant to consider the state of the art in the field of character recognition and the different approaches taken by the researchers in tackling this difficult and challenging problem.
1.1 State of the Art - Literature Survey

Most recognition systems utilize two principal components viz. features derived from an analysis of the characters and a classifier that utilizes the features in arriving at a recognition decision.

1.1.1 Character Features

Features derived from an analysis of the character image are very crucial in the development of a feasible recognition system. The features used for representing the characters can be divided into the following classes:

(i) Thresholded Image of Character:

In this scheme, the binary image of the character framed in a rectangular grid is used as a global feature. Such a global feature is used in recognition systems utilizing template matching [5], [6], [7], and [8]. The main drawback of this scheme is the large number of reference templates needed to account for variations in size, shape and orientation. Also, the computational burden is high because of the dimensionality of the data.

(ii) Generalized Fourier Shape Descriptors:

In order to reduce the high dimensionality inherent in using the character image as a feature, several researchers have proposed the use of features derived from transformation of the original image. These features include the two dimensional Fourier Transform of the image [9], and [10], Fourier Transform
of the contours of the character [11], [12], and [13], Walsh, Haar and Hadamard Transforms [14], [15], and [16] and the Karhunen–Loeve Transforms [17]. In the transformation techniques, only transform components with significant values are retained and all other components are simply ignored. These features have been successfully used in many high accuracy recognition systems.

A further advantage of the transform components is that these can be modified to yield a set of features that are invariant to size, shift, translation, rotation and reflection. However, in the context of numeral recognition, the rotational invariance property is not desirable since the differences among certain numerals (2 and 5, 6 and 9, 4 and 7) can be attributed to rotation and/or reflection. Also, the derivation of Fourier coefficients could involve extensive arithmetic computation.

(iii) Features Derived From Contour Analysis:

Many researchers have proposed the use of features derived from an analysis of the contours of the characters. These include:

(a) Zonal Density Features:

In this scheme, the character image framed by a rectangular grid is divided into zones and the densities of the boundary points in the various zones (overlapping or non-overlapping) are used as features [18]. Such features are generally useful in the recognition of fonts and hand-printed characters. However,
these features are neither adequate nor unique for the general problem of handwritten character recognition.

(b) **Moment Descriptors:**

The moments of the character pixels about the centre of gravity are used as features. Features based on moments have been modified to yield a set of descriptors that are invariant to shift, translation and rotation [19], [20].

The main disadvantage of the moment descriptors is that they tend to give undue importance to the parts of the character lying furthest from the centre of gravity.

(iv) **Topological Features:**

In this technique, the characters are described in terms of a set of geometrical features such as strokes at different orientation. As an example, the numeral '5' may be represented by a horizontal stroke (−) at the top, a vertical stroke (|) on the left and a bay (≡) in the lower half of the frame. Such features and other variations have been used extensively in many commercial recognition systems [21-32].

The main advantage of geometrical and topological features is their insensitivity to distortion and style variations. In addition, these features can also tolerate a certain degree of translation and slant (rotation). The main drawback of these techniques is the difficulty and the complexity in the extraction of these features. However, when these features have been extracted, processing can be done at very high speed.
1.1.2 Classifiers

Classifiers utilize the feature derived from test characters and compare these with reference features on reference values to perform an identification. Classifiers can be divided into two classes: Decision-theoretic classifiers and syntactic or structural classifiers.

(i) Decision-Theoretic Classifiers:

The decision theoretic classifiers utilize the numerical values of the features (parameters) and compute a distance or a discriminant function. The recognition of the test character is made on the basis of the value of the distance or the discriminant function.

(a) Distance Classifiers:

In distance classifiers features are represented as vectors in the feature space. A weighted or non-weighted Euclidean distance is computed between the test feature vector and each of the reference feature vectors. The reference feature vector which is closest (in distance) to the test vector yields the identity. Most recognition systems utilize the Euclidean distance or the Mahalanobis distance classifier [33], and [34]. Mahalanobis distance classifiers use the inverse of the covariance matrix of the feature vectors as the weight. Mahalanobis distance classifiers yield more accurate recognition when an accurate estimate of the covariance matrix is available.

(b) Discriminant Functions:

In this type of classification, hyperplanes or hypersurfaces that separate one character class from another, are
derived in the feature space. The identity of a test character is established by the region in which the feature vector is located. Under certain conditions, distance classifiers can be shown to be equivalent to linear discriminant functions.

(c) Nearest-Neighbour Classifier:

While distance classifiers can be termed parametric (because of certain assumptions about the statistics of the feature vectors), the nearest neighbour classifiers are essentially non-parametric. In this technique, all feature vectors in the training set are used as reference vectors.

A distance is computed between the test vector and each of the reference vectors. The 'K' nearest neighbours of the test vector are polled for determining the identity of the test numeral.

The main drawback of this method is the need for a very large training set and the consequent increase in the computational burden. By contrast, distance classifiers use the mean values of the feature vectors in the training set as the reference vector.

(ii) Syntactic Classifiers:

In this approach, the patterns are viewed as sentences in a language defined by a formal grammar. The essence of this approach lies in the selection of pattern primitives, the assembling of the primitives and their relationships into pattern grammars, and the analysis and recognition in terms
of these grammars [35], and [36]. The decision as to whether or not the pattern is syntactically correct is made by the syntax analyzer or parser. The string of primitives representing a test pattern is matched against strings of primitives representing each reference pattern. The test pattern is classified in the same class as the reference pattern that provides the best match. The matching is done through look-up table, decision tree, logical operation or template matching.

Although classifications based on decision trees and look-up tables are fast and efficient, a major problem in the application of grammatical techniques is the inference (learning) of the grammars from a finite set of samples. The inference and clustering problems are very difficult for strings, due to the lack of well defined similarity measures for them. In practice, grammatical inference can be performed in an interactive way by modifying the grammars until the reference strings are separated.

1.2 Goals of This Thesis

A survey of the relevant literature in the area of handwritten numeral recognition revealed that no reliable algorithms have been derived for recognition of broken and/or connected numeral strings.

It was decided that this thesis would attempt to derive:

a) A reliable set of features in the topological and Fourier domains for uniquely representing the handwritten numerals;

b) Appropriate classifiers (distance and syntactic) for implementing the recognition algorithm;
c) Algorithms for segmenting connected numeral strings into their individual units.

1.3 Outline of the Thesis

Data collection, preprocessing and postprocessing steps are discussed in Chapter II.

The derivation of a new class of Fourier shape descriptors and their application to the handwritten numeral recognition is presented in Chapter III. In this chapter, a new set of topological features for resolving the ambiguities in the recognized character is also proposed.

Chapter IV discusses the derivation of a syntactic recognition algorithm utilizing a new set of topological features.

The segmentation of connected numeral strings into their individual components is presented in Chapter V.

The salient conclusions and contributions of this thesis are summarized in Chapter VI.
Chapter 11

DATA COLLECTION AND PROCESSING

2.1 Introduction

In this chapter the establishment of a representative data base and the various forms of pre- and post-processing of the collected data are described. Since extensive handwritten numeral specimens were needed for both the training set (to design the recognition system) and the test set (to verify the feasibility of the recognition system) a study was undertaken to investigate the writing habits of individuals under different real life settings.

To this end, 20 volunteers varying in age from 7 to 50 years and coming from 6 different countries (including Canada) were approached for the collection of numeral specimens.

A variety of real life situations were simulated and the volunteers were asked to participate by writing randomly selected numerals on a standard sheet of paper. The selected numerals consisted of 800 isolated digits and about 10,000 samples of 3-digit strings.

The data collected as above were then studied exhaustively and grouped together on the basis of the similarity in shapes of the various numerals. These groups were then reproduced on a white sheet of paper with a felt pen. A typical sheet (8.5" x 11") consisted of 3 rows of 5 digits each.

2.2 Digital Imaging and Processing

The sheets containing the numeral specimens were imaged by a Hamamatsu Vidicon Camera and then digitized into images of (128 x 128) pixel resolution.
with each pixel represented by one of 256 gray levels (8 bit words) ranging from dark (0) to bright (255). The digitized images were stored in Diablo disks of the NOVA 840 minicomputer system.

In this arrangement, each numeral has an approximate height of 35 pixels and an approximate width of 25 pixels. Any attempt to decrease the resolution below this level resulted in unacceptable distortion caused by the spatial quantization.

2.3 Image Processing and Contour Extraction

The images were processed by the technique of thresholding to yield binary valued images consisting of dark pixels representing the numeral and bright pixels representing the background.

The thresholded images were then processed by a contour following algorithm [37] to yield the contours of the numerals. These were then stored as data files in the disks. Of these, four hundred specimens of isolated numerals were used for training the classifier and four hundred for testing the recognition system. Figures 2.1 - 2.4 display the images obtained after various stages of processing. Figure 2.1 displays the photograph of a typical specimen. Figure 2.2 displays the digitized image, while Figures 2.3 and 2.4 display the thresholded image and their contours respectively.
12345 56324
67890 68923
60542 361279
12345 12346
67890 57789
05498 91433
12345 12345
67890 67890
42007 25861

(a)

Isolated Numerals

Fig. 2.1 Samples of the numeric data.
(b)

Numeral Strings

Fig. 2.1 (Cont?) Samples of the numeric data.
Fig. 2.2 Digitized image.

Fig. 2.3 Thresholded image.
Fig. 2.4 Contours of the numerals.
Chapter III

NUMERAL RECOGNITION WITH FOURIER SHAPE DESCRIPTORS
AND TOPOLOGICAL FEATURES

3.1 Introduction

Signals which are either periodic or are of finite duration can be effectively represented by the process of orthonormal expansion. The orthonormal functions which form the basis of such expansion are typically sine and cosine functions, complex exponentials, Walsh-Hadamard functions and even polynomials. In most cases an adequate characterization of a signal is obtained by using only a finite number of these basis functions. Of the many possible representations, the trigonometric Fourier series and the complex exponential Fourier series are the most widely used because of the many elegant properties which the basis functions possess. In a sense, such waveform descriptions are essentially shape representations and hence the Fourier series representation could be used for describing numerals.

The contours (coordinates of border elements) which are obtained from the thresholded images of the numerals can be viewed as a pair of discrete finite duration signals with respect to their arc lengths. Each of these two functions can be expanded in discrete Fourier series and a finite set of the resulting Fourier coefficients can be used to uniquely characterize a given numeral. However, the Fourier coefficients are sensitive to the location of the first border element (i.e., shift) and the position as well as the orientation of the character in the frame of reference. In many shape recognition problems, it is necessary to
derive a set of coefficients (features) that are invariant with respect to position, orientation and shift.

Granlund [11] proposed a set of Fourier descriptors that had the desirable invariant properties. In his approach the x - y coordinates of the border elements were viewed as the real and imaginary parts of a sequence of complex numbers and expanded into a complex exponential Fourier series. A further set of operations were then performed on these coefficients to yield a set of shape descriptors. Persoon and Fu [12] described the use of such descriptors for the recognition of handwritten numerals. A recognition accuracy of 90% was reported.

Another set of Fourier descriptors were defined by Zahn and Roskies [38] who utilized the arc length and the tangent angle in the Fourier expansion.

Badreldin [39] used a computationally simple technique to derive a set of Fourier shape descriptors that had all the desirable invariant properties and which was easily normalized to be size invariant as well. By the use of piecewise constant interpolation, the Fast Fourier Transform algorithm could be used to derive these descriptors in a very efficient manner.

In this chapter, the derivation of these descriptors and their use in a handwritten numeral recognition system will be presented. The problem of variability within each group representing a certain character is resolved using an interactive functional mapping approach.

However, the rotational invariance property of the Fourier descriptors does create difficulties in resolving characters that are similar in shape
and whose differences can be attributed to rotation and/or reflection. Thus, Fourier descriptors do not distinguish between '2', '5', '6' and '9', etc.

In order to resolve this dilemma, the classification obtained from Fourier descriptors is followed by a second stage algorithm that utilizes the topological differences between the characters that are otherwise similar in shape. A new set of features termed "Border Transition Features" is proposed for resolving the difficulties inherent to the first stage based on Fourier descriptors.

3.2 Derivation of Fourier Descriptors

The preprocessing operation on the character images yields the Cartesian coordinates \( \{x(l), y(l); l = 1, 2, ..., L\} \) of the boundary elements, where \( L \) is the total number of boundary elements. Since the boundaries are closed curves, it is observed that

\[
x(L) = x(1) \quad y(L) = y(1)
\]  \hspace{1cm} (3.1)

In order to retain computational simplicity in Fourier analysis, the boundary following algorithm used in the preprocessing stage is especially adapted to yield consecutive boundary elements that are "four-neighbour" adjacent (i.e., diagonal neighbours are not permitted). This arrangement makes it possible to achieve uniform perimeter variation from one border element to the next. Figures 3.1a and b show the contour of numerals using "eight-neighbour" adjacent and the corresponding "four-neighbour" adjacent respectively.

The Fourier analysis involves the derivation of the Fourier series for the two coordinate sequences \( x(l) \) and \( y(l) \). The Fourier series
Fig. 3.1 (a) Eight-neighbour adjacent
(b) Four-neighbour adjacent.
description of $x(\ell)$ and $y(\ell)$ are given by

$$x(\ell) = \sum_{n} a(n) e^{jn \omega_0 \ell}, \quad (3.2)$$

and

$$y(\ell) = \sum_{n} b(n) e^{jn \omega_0 \ell}, \quad (3.3)$$

where

$$\omega_0 = \frac{2 \pi}{l}. \quad \text{(3.4)}$$

$a(n)$ and $b(n)$ are the complex Fourier coefficients; they are derived as

$$a(n) = \sum_{\ell=1}^{L-1} x(\ell) e^{-jn \omega_0 \ell}/(L-1), \quad (3.4)$$

and

$$b(n) = \sum_{\ell=1}^{L-1} y(\ell) e^{-jn \omega_0 \ell}/(L-1). \quad (3.5)$$

where $n = 1, 2, ..., N$, and $N$ is the total number of coefficients.

The Fourier coefficients derived according to equations (3.4) and (3.5) are not rotation or shift invariant (to clarify, it is noted that a shift will occur if the starting point of boundary following is arbitrary). In order to derive a set of Fourier shape descriptors that have the invariant property with respect to rotation and shift the following operations are defined. For each $n$, compute a set of invariant descriptors $\tau(n)$ as

$$\tau(n) = \sqrt{|a(n)|^2 + |b(n)|^2}^{1/2}. \quad (3.6)$$

A further refinement in the derivation of the descriptors is realized if dependence of $\tau(n)$ on the size of the character is eliminated by
computing a new set of descriptors \( S(n) \) as

\[
S(n) = r(n)/r(1)
\]  
(3.7)

this set of descriptors \( \{S(n); n = 1, 2, \ldots, N\} \) is invariant to rotation, shift, size and translation [39].

The use of the Fast Fourier Transform algorithm in the derivation of \( a(n) \) and \( b(n) \) is feasible through piecewise constant interpolation. The procedure is as follows:

Let \( L \) be the number of border elements

(i) If \( L - 1 = 2^M \), then go to step (iv).

else set \( M = \lfloor \log_2 (L - 1) \rfloor + 1 \)

where \( NP = 2^M \) is the number of points for interpolation.

(ii) Evaluate new step size

\[
\Delta = (L - 2)/(NP - 1)
\]

(note: old step size is 1)

(iii) Set \( P = \Delta \)

Compute the new sequence \( Z(k) \)

For \( k = 2, \ldots, (NP - 1) \) do

\[
P = P + \Delta
\]

\[
I = \lfloor P + 1.5 \rfloor
\]

\[
Z(k) = x(I) + jy(I) \quad j = \sqrt{-1}
\]

Enddo

\[
Z(1) = x(1) + jy(1)
\]

\[
Z(NP) = x(L - 1) + jy(L - 1)
\]
(iv) Compute the FFT of the complex sequence $Z(k)$ to obtain the complex Fourier coefficients $C(n)$, $n = 1, 2, \ldots, N$.

(v) Compute FFT of the real sequences $x(k)$ and $y(k)$ as follows:

\[
a(n - 1) = \frac{[C(n) + C^*(NP + 2 - n)]}{2} \\
b(n - 1) = \frac{[C(n) - C^*(NP + 2 - n)]}{2j},
\]

\[n = 2, 3, \ldots, N\]

Let $S(1) = \left[ |a(1)|^2 + |b(1)|^2 \right]^{1/2}$.

(vi) Compute rotation, shift and size invariant Fourier descriptors from

\[
S(n) = \left[ |a(n)|^2 + |b(n)|^2 \right]^{1/2}/S(1),
\]

\[n = 2, 3, \ldots, N\]

An important question that arises is the number of Fourier descriptors that are needed to uniquely characterize a given numeral. The question of adequacy of a given number of descriptors is best answered by studying the distortion in the shapes of the numerals when they are reconstructed with only a finite number of Fourier series terms. Thus, if $N$ descriptors are used, then the reconstructed contours will be described by

\[X(\ell) = a(0) + \sum_{n=1}^{N} [a(n)e^{jn\omega\ell} + a(-n)e^{-jn\omega\ell}] \quad (3.8)\]

and

\[Y(\ell) = b(0) + \sum_{n=1}^{N} [b(n)e^{jn\omega\ell} + b(-n)e^{-jn\omega\ell}] \quad (3.9)\]

It is noted that $a(-n) = a^*(n)$ and $b(-n) = b^*(n)$. Figures 3.2 a, b and c illustrate the original and reconstructed characters using the first six and the first fifteen coefficients, respectively. It is clear that the reconstructed numerals are relatively distortion free especially the ones
Fig. 3.2(a) Original numerals

Fig. 3.2(b) Reconstructed numerals using 6 Fourier descriptors.

Fig. 3.2(c) Reconstructed numerals using 15 Fourier descriptors.
that used fifteen coefficients. In view of this observation, it was
decided to use fifteen Fourier descriptors in the recognition algorithm.

3.3 Identification of Sub-Classes

A study of the patterns corresponding to a given character outlines
the need to use an unsupervised learning technique to identify the cor-
rect number of existing prototypes within each character in the data set.
Figure 3.3 illustrates the existence of sub-classes within each character
for the numerals '6', '7', '8' and '9'. Although in many cases, the sub-
classes are easily identified, there are instances where the existence
of sub-classes is not evident. A typical set of the numeral '1' is shown
in Figure 3.4. Although the specimens appear to be very similar, the
Fourier descriptors of these numerals are significantly different. Thus
in these cases, separate sub-classes have to be generated based on the
values of the Fourier descriptors rather than the numeral shapes them-
selves. In order to derive the sub-classes, an interactive functional
mapping technique is proposed. In this technique, the N-dimensional
feature vector is mapped into a function of a single variable. This
function is then plotted against the variable. One such transformation
[40] is

\[ F(t) = \begin{bmatrix} S(1) & S(2) & \ldots & S(N) \end{bmatrix} \begin{bmatrix} q_1(t) \\ q_2(t) \\ \vdots \\ q_N(t) \end{bmatrix} \]  (3.10)

where \( F(t) \) is the one-dimensional projection of the feature vector
Fig. 3.3 Sub-classes of numerals 6, 7, 8 and 9.

Fig. 3.4 Specimens of the numeral '1'
\{S(n); n = 1, 2, \ldots, N\} in the 't' domain; and

\[
\begin{align*}
q_1(t) &= \sin t, & q_2(t) &= \cos t, \\
q_3(t) &= \sin 2t, & q_4(t) &= \cos 2t, \\
& \quad \vdots & \quad \vdots \\
q_N(t) &= \sin((N + 1) t/2), & N \text{ odd} \\
q_N(t) &= \cos (N t/2), & N \text{ even}
\end{align*}
\] (3.11)

for \(-\pi \leq t \leq \pi\).

The functional mapping technique preserves the statistical properties of the feature set as well as the Euclidean distance between two feature vectors [See Appendix I]. Figures 3.5 a, b and c illustrate the functional mapping of the Fourier descriptors of the numerals '0', '6' and '8' respectively. The existence of sub-classes within each numeral is clearly brought out in this functional mapping procedure. This cluster-detecting ability is due to the distance-preserving property of the mapping.

A practical problem with this approach is the simultaneous plotting of functions representing a large number of classes. In this case, detection of sub-classes becomes difficult. To overcome this difficulty, the following procedure is adopted:

1. The entire data is first examined by plotting all the feature vectors of a specified numeral. The functional mapping approach provides the proper number of clusters (sub-classes) existing in the data set of each character.

2. Separate plots of about ten functions are then used to assign an initial membership list of each cluster. This
Fig. 3.5a  Functional mapping of the FDs of the numeral '0'.

Fig. 3.5b  Functional mapping of the FDs of the numeral '6'.
Fig. 3.5c Functional mapping of the FDs of the numeral '8'.
provides the initial conditions used in an efficient version of the K-MEANS clustering algorithm [33]. The procedure for the modified K-MEANS algorithm is given in Appendix II.

3. Validation of the results is achieved through the application of the functional mapping technique.

Figure 3.6 displays the functional mapping of the Fourier descriptors of the numeral '1' shown in Figure 3.4. The above procedure, when applied, yielded three sub-classes of the numeral. Figure 3.7 displays the functional mapping of the mean vector of each sub-class.

This analysis is applied to the feature vectors of each numeral and a total of 26 distinct classes are obtained for the 400 numerals used for training the classifier.

3.4 Recognition Algorithm

A reference feature vector $F_m$ for each sub-class is obtained by averaging the feature vectors and is given by

$$F_m = \frac{1}{N_m} \sum_{i=1}^{N_m} F_{mi}$$ (3.12)

where $F_{mi}$ is the feature vector (whose elements are the $N$-Fourier descriptors) of the $i$th specimen of sub-class 'm' (where $m = 1, 2, \ldots, 26$) in the training set;

$F_m$ is the reference feature vector for sub-class 'm'; and

$N_m$ is the number of specimens of sub-class 'm' in the training set.

Two distance measures are used in this study:

(a) The Squared Euclidean Distance Measure:

The squared Euclidean distance between a test feature vector $F_x$ and
Fig. 3.6 Functional mapping of the FDs of the numeral '1'.

Fig. 3.7 Functional mapping of the FDs of the sub-classes of the numeral '1'.
a reference feature vector \( F_m \) is given by

\[
D_{xm}^2 = \| F_x - F_m \|^2 = (F_x - F_m)^T (F_x - F_m)
\]

\[
= F_x^T F_x - 2F_x^T F_m + F_m^T F_m
\]

\[
= F_x^T F_x - (2F_m^T F_m - F_m^T F_m)
\]

Since the term \( F_x^T F_m \) is independent of \( m \) in all \( D_{xm}^2 \), \( m = 1, 2, \ldots \).

26 sub-classes, therefore, choosing the minimum \( D_{xm}^2 \) is equivalent to choosing the maximum \( (F_x^T F_m - \frac{1}{2} F_m^T F_m) \). It is noted that \( F_m^T F_m \) may be precomputed and stored in a reference library.

Consequently, one may define the decision functions as

\[
d_m(x) = F_x^T F_m - d_{mM}
\]

where \( d_{mM} = \frac{1}{2} F_m^T F_m \)

\[ m = 1, 2, \ldots, 26 \]

A test feature vector \( F_X \) is identified as character 'i' if

\[
d_i(x) > d_j(x) \text{ for all } j \neq i,
\]

\[ j = 1, 2, \ldots, 26 \]

(b) The Mahalanobis Distance Measure

The Mahalanobis distance measure is defined as

\[
D_{xm}^M = (F_x - F_m)^T C^{-1} (F_x - F_m)
\]

where \( D_{xm}^M \) is the Mahalanobis distance between a test feature vector, \( F_x \), and a reference feature vector \( F_m \); and \( C^{-1} \) is the inverse of the covariance.
matrix of the feature vectors computed for the training set. Two cases were considered for C:

(i) The pooled covariance matrix:

A pooled covariance matrix $C$ is computed by using all specimens of all the characters in the training set. The elements of $C$ are defined as

$$C_{in} = \frac{1}{N_c N_m} \sum_{m=1}^{N_c} \sum_{k=1}^{N_m} \left( F_{m\ell k} - F_{m\ell} \right) \left( F_{mnk} - F_{mn} \right)$$

(3.16)

where $C_{in}$ is an element of $C$ occupying $i^{th}$ row and $n^{th}$ column.

$N_c =$ number of classes in training set

$N_m =$ number of specimens of the $m^{th}$ class in training set

$F_{m\ell k}$ is the $\ell^{th}$ element of the feature vector of the $k^{th}$ specimen of the $m^{th}$ class

$F_{m\ell}$ is the $\ell^{th}$ element of the mean feature vector of $m^{th}$ class

and $\ell = 1, 2, \ldots, N; \ n = 1, 2, \ldots, N.$

(ii) Distinct Covariance matrices

A covariance matrix is constructed for each distinct character in the training set. Thus an element of the $m^{th}$ class covariance matrix, $C_m$, is defined as

$$\left[ C_{in} \right]_m = \frac{1}{N_m} \sum_{k=1}^{N_m} \left( F_{m\ell k} - F_{m\ell} \right) \left( F_{mnk} - F_{mn} \right)$$

(3.17)
3.5 Test Results

The recognition algorithm, described in section 3.4, was applied to the 400 numerals representing the test set. The squared Euclidean distance when applied, yielded a recognition accuracy of 92%, while the Mahalanobis distance measure yielded 89.5% when the pooled covariance was used, and 97.3% when distinct covariance matrices were used. In order to keep the computational burden minimal, only the first six Fourier descriptors were used as the feature vector when the Mahalanobis distance measure was applied. In contrast fifteen Fourier descriptors were utilized in the Euclidean distance measure.

Due to the rotational invariance of the Fourier descriptors the numeral '6' was often identified as '9' and vice-versa. Similar observations were noted with respect to the numerals '2' and '5'. Such misclassifications were excluded in the determination of the overall accuracy. The other misclassifications that occurred (apart from the ones mentioned above) were generally caused by

(i) a class of '7's identified as '9'
(ii) a class of '7's identified as '4'

The above misclassification are also largely due to the rotational invariance of the Fourier descriptors. It was felt that these misclassifications can be eliminated only if the topological structure of the characters was also considered in the recognition algorithm. Such a technique which directly utilizes the thresholded image is described in the next section.
3.6 Border Transition Features (BTFs)

In order to avoid the misclassifications resulting from the rotational invariance, a new technique that effectively exploits the topological properties of the characters is proposed. In this technique it is assumed that all the characters are oriented vertically. The technique consists of:

1) Determination of the maximum height \( H \) and the width \( W \) of the character.

2) Setting up a square frame of size \( M = \lceil \text{MAX} (W, H) + 4 \rceil \) pixels enclosing the character.

3) Counting the number of transitions \( NT \) from bright to dark during row and column scans in each quadrant.

4) Computing the average transition for a given scan as \( 2 \ NT/M \).

The procedure is illustrated in Figure 3.8.

The determination of average transition will be illustrated by considering row scan in quadrant 1 (Figure 3.8). For quadrant 1, the horizontal transition is determined by counting all bright to dark gray level transitions as the rows are scanned left to right, sequentially. Given the gray level array \( \{IX(I, J), I = 1, M; J = 1, M\} \):

i) Set \( I = 2 \), \( NT = 0 \), \( J = 1 \)

ii) If \( (IX(I, J+1) - IX(I,J)) \geq 0 \) go to step (iv)

iii) Set \( NT = NT + 1 \)

iv) If \( (J \geq M/2) \) go to step (vi)

v) \( J = J + 1 \); go to (ii)

vi) \( I = I + 1 \)
Fig. 3.8  'Border Transitions.'
vii) if \( l > \frac{M}{2} \) go to (ix)

viii) Go to (ii).

ix) Average Horizontal Transition in quadrant 1 = \( \frac{\sum_{i} T_{i}}{(M/2)} \)

The average vertical transition is determined by modifying the above procedure to perform column scan instead of a row scan.

The direction of scan depends on the quadrant and is defined as per Table 3.1.

**Table 3.1**

<table>
<thead>
<tr>
<th>Quadrant</th>
<th>Transition</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Horizontal</td>
<td>Left to Right</td>
</tr>
<tr>
<td>3</td>
<td>&quot;</td>
<td>&quot;</td>
</tr>
<tr>
<td>2</td>
<td>&quot;</td>
<td>Right to Left</td>
</tr>
<tr>
<td>4</td>
<td>&quot;</td>
<td>&quot;</td>
</tr>
<tr>
<td>1</td>
<td>Vertical</td>
<td>Top Down</td>
</tr>
<tr>
<td>2</td>
<td>&quot;</td>
<td>&quot;</td>
</tr>
<tr>
<td>3</td>
<td>&quot;</td>
<td>Bottom Up</td>
</tr>
<tr>
<td>4</td>
<td>&quot;</td>
<td>&quot;</td>
</tr>
</tbody>
</table>

Table 3.2 lists the average horizontal and vertical transitions for the numerals, 2, 4, 5, 6, 7 and 9. It is noted that most of the misclassification errors with Fourier descriptors involved these digits.

It is clearly seen that the average horizontal transition for the numeral '2' in quadrant 3 is always greater than 0.5, while the same average is less than 0.5 for the numeral '5'. Proceeding in this way the following observations can be made:
Table 3.2

BTFs For the Numerals 2, 4, 5, 6, 7 and 9

<table>
<thead>
<tr>
<th>NUMERAL</th>
<th>HORIZONTAL AVERAGES</th>
<th>VERTICAL AVERAGES</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$H_1$</td>
<td>$H_2$</td>
</tr>
<tr>
<td>'2'</td>
<td>.311384</td>
<td>.737723</td>
</tr>
<tr>
<td>'4'</td>
<td>.657407</td>
<td>.115741</td>
</tr>
<tr>
<td>'5'</td>
<td>.506250</td>
<td>.587500</td>
</tr>
<tr>
<td>'6'</td>
<td>.546875</td>
<td>.480469</td>
</tr>
<tr>
<td>'7'</td>
<td>.226563</td>
<td>.656250</td>
</tr>
<tr>
<td>'9'</td>
<td>.57375</td>
<td>.688750</td>
</tr>
</tbody>
</table>
1) For '5'. \[ H_3 < 0.5 \]
   For '2'. \[ H_3 > 0.5 \]
2) For '4'. \[ H_1 > 0.5, V_2 < 0.5 \]
   For '7'. \[ H_1 < 0.5, V_2 > 0.5 \]
3) For '6'. \[ H_3 > 0.5, V_2 > 0.5 \]
   For '9' and '7'. \[ H_3 < 0.5, V_2 > 0.5 \]
4) For '9'. \[ H_1 > 0.5 \]
   For '7'. \[ H_1 < 0.5 \]

Thus a misclassification inherent in Fourier Descriptor method can be eliminated in two stages as follows:

1) In the first stage, Fourier descriptors are used to obtain a preliminary identification.

2) If the character identified is 2, 4, 5, 6, 7 or 9, then a second stage identification step is initiated by determining the average transitions and applying the conditions (3.18) to resolve ambiguities with regard to the character identified in the first stage. The entire algorithm is illustrated in the flow chart of Fig. 3.9.

The two stage algorithm (Figure 3.9) was implemented on the data set consisting of about 40 test specimens per numeral. The overall recognition accuracy over the two stages using squared Euclidean distance was 95%. When Mahalanobis distance measure was applied, it yielded 97% recognition accuracy using the pooled covariance matrix, and 99% accuracy using distinct covariance matrices.

The above distance measures were used with 26 classes in the recognition
Fig. 3.9 Flow chart of two stage algorithm.
algorithm. It is observed that although 26 classes were recognized, only
10 covariance matrices (one for each numeral in the training set) were
used in computed distances. It is also noted that 99% accuracy was
achieved with only six Fourier descriptors, as opposed to fifteen in
the Euclidean case.

3.7 Uniqueness of Border Transition Features

Although the Border Transition Features successfully resolved the
ambiguities in the first stage recognition, it must be emphasized that
these features do not provide a unique characterization of handwritten
numerals. An exhaustive study of the Border Transition features for
handwritten numerals was undertaken and the results are shown in Table
3.3. This table clearly illustrates the severe overlap of feature pro-
properties among the numerals with the sole exception of the numeral '1'.
However, these features are adequate for handprinted and machine printed
numerals. Thus for general numeral recognition one requires a more re-
liable set of topological descriptors. One such set is described in
the next chapter.

3.8 Summary

In conclusion, it is observed that Fourier descriptors can be used
for high accuracy character recognition provided that the contours of
the characters form an unbroken sequence. In numeral recognition, however,
a second stage classifier is needed to eliminate the ambiguities result-
ing from the rotation invariance of the descriptors. Difficulties will
also arise in the recognition of connected numeral strings, unless the
strings are properly segmented into their individual units.
Table 3.3

Range of the Horizontal and Vertical Averages

<table>
<thead>
<tr>
<th>Character</th>
<th>Vertical Averages</th>
<th>Horizontal Averages</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$V_1$</td>
<td>$V_2$</td>
</tr>
<tr>
<td>'0'</td>
<td>.6 - .85</td>
<td>.65 - .85</td>
</tr>
<tr>
<td>'1'</td>
<td>.1 - .2</td>
<td>.15 - .35</td>
</tr>
<tr>
<td>'2'</td>
<td>.2 - .5</td>
<td>.5 - .9</td>
</tr>
<tr>
<td>'3'</td>
<td>.5 - .85</td>
<td>.65 - 1.0</td>
</tr>
<tr>
<td>'4'</td>
<td>.2 - .75</td>
<td>.0 - .95</td>
</tr>
<tr>
<td>'5'</td>
<td>.3 - .8</td>
<td>.75 - 1.0</td>
</tr>
<tr>
<td>'6'</td>
<td>.4 - .85</td>
<td>.45 - .83</td>
</tr>
<tr>
<td>'7'</td>
<td>.45 - .65</td>
<td>.6 - .9</td>
</tr>
<tr>
<td>'8'</td>
<td>.4 - .95</td>
<td>.35 - .8</td>
</tr>
<tr>
<td>'9'</td>
<td>.25 - .47</td>
<td>.65 - .85</td>
</tr>
</tbody>
</table>
Chapter IV

STRUCTURAL RECOGNITION ALGORITHM

FOR HANDWRITTEN NUMERALS

4.1 Introduction

In Chapter III, the use of a new form of Fourier shape descriptors as features in a minimum distance classifier system was proposed. Although these descriptors are shown to be sensitive to variations in shape and high recognition accuracy (> 98%) is realized, they require a relatively high degree of computational effort. The same observation applies to the distance classifiers used for recognition. Also, due to the rotational invariance of the Fourier descriptors, ambiguities resulted from the similarity of shapes of numerals ('2', '5'), ('6', '9'), etc. Therefore, a supplementary technique using Border Transition Features was required to resolve such ambiguities. Difficulties also arise when one has to deal with broken numerals such as '1', '4' and '9', since more than one border is needed to represent these numerals.

In order to improve the speed of recognition while maintaining high accuracy and reliability, a new set of topological features (primitives) derived from the left and right profiles of the characters is proposed for a unique representation of the numerals. The numerals (patterns) are modeled as sentences of a formal language constructed from the set of terminals (primitives) and non-terminals. The recognition system consists of a syntactic classifier that analyzes the topological structures of the test patterns and is implemented as a binary tree classifier for identifying the test numeral.
4.2 Character Description and Feature Extraction

In this section, a global description of the character is first given. Features representing local properties of the character are then extracted and used as primitives in a syntactic pattern recognition mode.

4.2.1 Global Features (Left and Right Profiles)

A global description of the character is first derived from its contour \( \{ X_i, Y_i : i = 1,2,\ldots, l \} \). It consists of the profiles of the external contour of a character as seen from the left \( \{ LP(k); k = 1,2,\ldots,NH \} \), and from the right \( \{ RP(k); k = 1,2,\ldots,NH \} \), where \( NH \) is the height of the numeral. The two profiles are illustrated in Figures 4.1a and 4.1b for the numerals '5' and '3' respectively.

4.2.2 Local Features (Primitives)

These features are derived using the left and right profiles of the numeral. These are:

(a) **Width of the Character**

The width is defined as

\[
W(k) = RP(k) - LP(k)
\]

(4.1)

where \( k \) is a specified location.

Fig. 4.2a shows the width of the character '9' at locations \( k = 10 \) and \( k = 40 \).

The width is used in the description of some characters such as '0', '6', '8' and '9'. As an example, \( W(k_1) \) of the character '9' shown in Figure 4.2a is greater than \( W(k_2) \), while for a character '6', \( W(k_1) \) is less than \( W(k_2) \) as shown in Figure 4.2b.
Fig. 4.1a Left and right profiles of the numeral '5'.

Fig. 4.1b Left and right profiles of the numeral '3'.

Fig. 4.2a Left and right profiles of the numeral '9'.

Fig. 4.2b Left and right profiles of the numeral '6'. 
(b) **Ratio**

The ratio is given by

\[ \text{Ratio} = \frac{\text{Height}}{\text{Max}(W)} \]  

(4.2)

where

\[ \text{Max}(W) \triangleq \max_k \{ \text{RP}(k) - \text{LP}(k) \} \quad ; \quad k = 1, 2, \ldots, \text{NH} \]

and the height is equal to NH.

The ratio information is used to identify the character '1'. It has been found that for some classes of '1', the ratio is always greater than three, while it is not for the remaining characters. This is illustrated in Figure 4.3.

(c) **Locations of Maxima and Minima**

The location of a maximum on the left or right profiles is defined as

\[ \text{LMX} \triangleq \text{location of Max} \left\{ \text{LP}(k) \right\} \quad \frac{R_1}{R_1} \]  

(4.3)

and

\[ \text{RMX} \triangleq \text{location of Max} \left\{ \text{RP}(k) \right\} \quad \frac{R_2}{R_2} \]  

(4.4)

where \( R_1 \) and \( R_2 \) are defined ranges on LP(k) or RP(k) such that

\[ 1 \leq \alpha_1 < \frac{R_1}{R_1} < \alpha_2 \leq \text{NH}, \quad \text{and} \]

\[ 1 \leq \beta_1 < \frac{R_2}{R_2} < \beta_2 \leq \text{NH} \]

Similarly, the location of a minimum on the left or right profiles is defined as

\[ \text{LMIN} \triangleq \text{location of Min} \left\{ \text{LP}(k) \right\} \quad \frac{R_1}{R_1} \]  

(4.5)
Fig. 4.3 Left and right profiles of the numeral '1'.
and

\[ \text{RMIN} = \text{location of } \min_{R_2} \{ \text{RP}(k) \} \]  \hspace{1cm} (4.6)

Figure 4.1a shows the locations of maximum and minimum on the left profile
of character '5'. The locations of minimum and maximum are used in de-
scribing some characters such as 2, 3, 4, 6, 8 and 9. As an example, the
left profile of character '5', shown in Figure 4.1a will have one minimum,
with location LMIN and one maximum at location LMX, in the range 10 < R_1 < 40,
such that

\[ \text{LMX} > \text{LMIN} \]

On the other hand, the left profile of character '3', shown in Figure 4.1b
will have two maxima, with locations LMX_1 and LMX_2 and one minimum at loca-
tion LMIN, in the range 10 < R_1 < 40, such that

\[ \text{LMX}_2 > \text{LMIN} > \text{LMX}_1 \]

(d) **Discontinuities in Character Profiles**

These are quantified by using the first difference values of the
left and right profiles, LDIF(k) and RDIF(k) respectively, and are given
as

\[ \text{LDIF}(k) = \text{LP}(k) - \text{LP}(k-1) \]  \hspace{1cm} (4.7)

and

\[ \text{RDIF}(k) = \text{RP}(k) - \text{RP}(k-1) \]  \hspace{1cm} (4.8)

\[ k = 2, 3, \ldots, \text{NH} ; \]

where

LDIF(k) and RDIF(k) are the first difference of the left and right
profiles respectively.
Figures 4.4a and b illustrate the first difference of the left and right profiles shown in Figure 4.1a respectively.

A positive peak of left or right profile of a character is defined as

\[ \text{LPEAK}^+ = \max_{R_3} \{ \text{LDIF}(k) \} \]  \hspace{1cm} (4.9)

and

\[ \text{RPEAK}^+ = \max_{R_4} \{ \text{RDIF}(k) \} \]  \hspace{1cm} (4.10)

where \( R_3 \) and \( R_4 \) are specified ranges on \( \text{LDIF}(k) \) or \( \text{RDIF}(k) \), so that

\[ 2 \leq \gamma_1 < R_3 < \gamma_2 \leq \text{NH}, \text{ and} \]

\[ 2 \leq \delta_1 < R_4 < \delta_2 \leq \text{NH} \]

Similarly, a negative peak of the left or right profile of a character is defined as

\[ \text{LPEAK}^- = \min_{R_3} \{ \text{LDIF}(k) \} \]  \hspace{1cm} (4.11)

and

\[ \text{RPEAK}^- = \min_{R_4} \{ \text{RDIF}(k) \} \]  \hspace{1cm} (4.12)

A single measure of discontinuity is defined as a left peak (LPEAK), as shown in Figure 4.4a, and is given by

\[ \text{LPEAK} = \left| \text{LPEAK}^+ \right| + \left| \text{LPEAK}^- \right| \]  \hspace{1cm} (4.13)

and a right peak (RPEAK), as shown in Figure 4.4b, is given by

\[ \text{RPEAK} = \left| \text{RPEAK}^+ \right| + \left| \text{RPEAK}^- \right| \]  \hspace{1cm} (4.14)
Fig. 4.4a First difference of the left profile of the numeral '5' shown in Fig. 4.1a.

Fig. 4.4b First difference of the right profile of the numeral '5' shown in Fig. 4.1a.
These features, LPEAK and RPEAK, can be used to describe the characters by assigning a threshold value $T$ to each. As an illustration, it has been found that RPEAK in the range $2 < R_4 < 50$ is always greater than 10 for the normalized numeral '5' to a height of 50 units, while for the numeral '3', normalized to the same height, RPEAK is less than 10.

4.2.3 Features for Broken Numerals

An analysis of the numeral specimens collected from the participating individuals revealed that the most commonly occurring broken numerals were '1', '4', and '5'. A typical collection is shown in Figure 4.5. It is assumed that the breaks in the numerals are due to the writing habits of the individuals and not to the use of faulty writing implements.

A study of the broken numerals such as the ones shown in Figure 4.5 clearly reveals the existence of a vertical or a horizontal gap separating the two segments of the broken numeral. The left and right profiles will not be affected by the presence of a vertical gap between the two segments. This is clearly the case for the broken specimens of the numeral '4' and some of the numeral '5'. Figures 4.6a and b display the left and right profiles for the broken numerals '4' and '5' with vertical gaps.

When horizontal gaps are encountered as in '1' and '5', left and right profiles will be undefined at the locations corresponding to the gap between the two segments. By simply skipping these locations and storing the profile information, only at locations where they are defined, the features can be properly evaluated. Figures 4.6c and d display the left and right profiles for the broken numerals '1' and '5' of the second type.
Fig. 4.5 Specimens of broken numerals.

Fig. 4.6a Left and right profiles of the broken numeral '4'.
Fig. 4.6b Left and right profiles of the broken numeral '5' (with vertical gap).

Fig. 4.6c Left and right profiles of the broken numeral '1' (with horizontal gap).
Fig. 4.6d Left and right profiles of the broken numeral '5' (with horizontal gap).
4.3 Relational Attributes

An attribute is now defined for each primitive. The attributes are based on an exhaustive qualitative evaluation of the local features of the numerals in the training set. The attributes take the form of logical predicates and play the role of semantic rules. Table 4.1 lists all the predicates that numerals normalized to a height of 50 units possess. The 48 entries in Table 4.1 take into account all the subclasses of all the numerals in the training set.

4.4 Syntactic Recognition Algorithm

In this section the basic identification scheme to be used will be presented using the concepts of syntactic pattern recognition. The first step in the recognition system development is the determination of "groups" consisting of topological descriptors. The next step in this process is the derivation of "chains" for each class in the training set.

4.4.1 Determination of Groups

A group, $G^k_i$, is defined as a set of primitives characterizing a specific subclass $k$ of numeral $i$. The subclasses are determined by the similarity of their left and right profiles. Features that show significant variability, are utilized to derive the different subclasses for each numeral. The procedure will be illustrated by considering different examples. An observation of Figures 4.1a, 4.4a and 4.4b reveals the following:

1) Continuity in the right profile near the top.
2) Continuity in the left profile near the bottom.
3) The left profile attains a minimum before it attains its maximum.
Table 4.1 Predicates Used in the Recognition of the Numerals '0', '1', ..., '9'

<table>
<thead>
<tr>
<th>No.</th>
<th>Primitive</th>
<th>Semantic Rule (Predicate)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>a₁</td>
<td>True if LPEAK &lt; 10; 2 ≤ R₃ ≤ 50</td>
</tr>
<tr>
<td>2</td>
<td>a₂</td>
<td>True if LPEAK &lt; 5; 2 ≤ R₃ ≤ 10</td>
</tr>
<tr>
<td>3</td>
<td>a₃</td>
<td>True if LPEAK &gt; 5; 2 ≤ R₃ ≤ 15</td>
</tr>
<tr>
<td>4</td>
<td>a₄</td>
<td>True if LPEAK &gt; 10; 2 ≤ R₃ ≤ 15</td>
</tr>
<tr>
<td>5</td>
<td>a₅</td>
<td>True if LPEAK &gt; 10; 2 ≤ R₃ ≤ 20</td>
</tr>
<tr>
<td>6</td>
<td>a₆</td>
<td>True if LPEAK &gt; 5; 2 ≤ R₃ ≤ 25</td>
</tr>
<tr>
<td>7</td>
<td>a₇</td>
<td>True if LPEAK &gt; 5; 5 ≤ R₃ ≤ 15</td>
</tr>
<tr>
<td>8</td>
<td>a₈</td>
<td>True if LPEAK &gt; 5; 5 ≤ R₃ ≤ 35</td>
</tr>
<tr>
<td>9</td>
<td>a₉</td>
<td>True if LPEAK &gt; 10; 5 ≤ R₃ ≤ 40</td>
</tr>
<tr>
<td>10</td>
<td>a₁₀</td>
<td>True if LPEAK &gt; 10; 10 ≤ R₃ ≤ 30</td>
</tr>
<tr>
<td>11</td>
<td>a₁₁</td>
<td>True if LPEAK &gt; 10; 15 ≤ R₃ ≤ 40</td>
</tr>
<tr>
<td>12</td>
<td>a₁₂</td>
<td>True if LPEAK &lt; 5; 25 ≤ R₃ ≤ 50</td>
</tr>
<tr>
<td>13</td>
<td>a₁₃</td>
<td>True if LPEAK &gt; 10; 30 ≤ R₃ ≤ 50</td>
</tr>
<tr>
<td>14</td>
<td>a₁₄</td>
<td>True if LPEAK &lt; 5; 30 ≤ R₃ ≤ 50</td>
</tr>
<tr>
<td>15</td>
<td>a₁₅</td>
<td>True if LPEAK &lt; 5; 35 ≤ R₃ ≤ 50</td>
</tr>
<tr>
<td>16</td>
<td>a₁₆</td>
<td>True if LPEAK &gt; 10; 35 ≤ R₃ ≤ 50</td>
</tr>
<tr>
<td>17</td>
<td>a₁₇</td>
<td>True if LPEAK &gt; 5; 40 ≤ R₃ ≤ 50</td>
</tr>
<tr>
<td>18</td>
<td>b₁</td>
<td>True if RPEAK &gt; 10; 2 ≤ R₄ ≤ 50</td>
</tr>
<tr>
<td>19</td>
<td>b₂</td>
<td>True if RPEAK &gt; 10; 2 ≤ R₄ ≤ 15</td>
</tr>
<tr>
<td>20</td>
<td>b₃</td>
<td>True if RPEAK &lt; 10; 2 ≤ R₄ ≤ 30</td>
</tr>
<tr>
<td>21</td>
<td>b₄</td>
<td>True if RPEAK &lt; 5; 2 ≤ R₄ ≤ 45</td>
</tr>
<tr>
<td>22</td>
<td>b₅</td>
<td>True if RPEAK &lt; 10; 25 ≤ R₄ ≤ 45</td>
</tr>
</tbody>
</table>
Table 4.1 (Continued)

<table>
<thead>
<tr>
<th>No.</th>
<th>Primitive</th>
<th>Semantic Rule (Predicate)</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>(b_6)</td>
<td>True if (R_{\text{PEAK}} &gt; 10); (25 &lt; R_4 &lt; 50)</td>
</tr>
<tr>
<td>24</td>
<td>(b_7)</td>
<td>True if (R_{\text{PEAK}} &lt; 5); (25 &lt; R_4 &lt; 50)</td>
</tr>
<tr>
<td>25</td>
<td>(b_8)</td>
<td>True if (R_{\text{PEAK}} &gt; 10); (30 &lt; R_4 &lt; 50)</td>
</tr>
<tr>
<td>26</td>
<td>(b_9)</td>
<td>True if (R_{\text{PEAK}} &gt; 5); (35 &lt; R_4 &lt; 50)</td>
</tr>
<tr>
<td>27</td>
<td>(b_{10})</td>
<td>True if (R_{\text{PEAK}} &gt; 10); (35 &lt; R_4 &lt; 50)</td>
</tr>
<tr>
<td>28</td>
<td>(b_{11})</td>
<td>True if (R_{\text{PEAK}} &gt; 5); (40 &lt; R_4 &lt; 50)</td>
</tr>
<tr>
<td>29</td>
<td>(c_1)</td>
<td>True if (R_{\text{MIN}} (1 \leq R_2 &lt; 30)) is less than (R_{\text{MIN}} (1 \leq R_2 &lt; R_{\text{MIN}})) and greater than (R_{\text{MIN}} (1 \leq R_2 &lt; R_{\text{MIN}}))</td>
</tr>
<tr>
<td>30</td>
<td>(c_2)</td>
<td>True if (R_{\text{MIN}} (10 &lt; R_2 &lt; 40)) is less than (R_{\text{MIN}} (10 &lt; R_2 &lt; 40)) and greater than (R_{\text{MIN}} (1 &lt; R_2 &lt; R_{\text{MIN}}))</td>
</tr>
<tr>
<td>31</td>
<td>(c_3)</td>
<td>True if (R_{\text{MIN}} (10 &lt; R_2 &lt; 45)) is less than (R_{\text{MIN}} (1 &lt; R_2 &lt; R_{\text{MIN}})) and greater than (R_{\text{MIN}} (1 &lt; R_2 &lt; R_{\text{MIN}}))</td>
</tr>
<tr>
<td>32</td>
<td>(d_1)</td>
<td>True if (R_{\text{MIN}} (1 &lt; R_2 &lt; 25)) is in the range (1 &lt; R_2 &lt; 25), and (R_{\text{MIN}}) in the range (1 &lt; R_2 &lt; R_{\text{MIN}})</td>
</tr>
<tr>
<td>33</td>
<td>(d_2)</td>
<td>True if (R_{\text{MIN}} (1 &lt; R_2 &lt; 25)) is in the range (1 &lt; R_2 &lt; 25), and (R_{\text{MIN}}) is in the range (R_{\text{MIN}} &lt; R_2 &lt; 40).</td>
</tr>
<tr>
<td>34</td>
<td>(e_1)</td>
<td>True if (L_{\text{MIN}} &lt; L_{\text{MIN}}); where (L_{\text{MIN}}) and (L_{\text{MIN}}) are in the range (1 &lt; R_1 &lt; 10)</td>
</tr>
<tr>
<td>35</td>
<td>(e_2)</td>
<td>True if (L_{\text{MIN}} &lt; L_{\text{MIN}}); where (L_{\text{MIN}}) in the range (1 &lt; R_1 &lt; 30), and (L_{\text{MIN}}) in the range (1 &lt; R_1 &lt; L_{\text{MIN}})</td>
</tr>
<tr>
<td>No.</td>
<td>Primitive</td>
<td>Semantic Rule (Predicate)</td>
</tr>
<tr>
<td>-----</td>
<td>-----------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>36</td>
<td>e₁</td>
<td>True if LMX &lt; LMIN; where LMX in the range 10 ≤ R₁ ≤ 30, and LMIN in the range 10 ≤ R₂ ≤ LMX</td>
</tr>
<tr>
<td>37</td>
<td>e₄</td>
<td>True if LMX &lt; LMIN; where LMX and LMIN are in the range 15 ≤ R₁ ≤ 45</td>
</tr>
<tr>
<td>38</td>
<td>e₅</td>
<td>True if LMX &lt; LMIN; where LMX and LMIN are in the range 20 ≤ R₁ ≤ 50</td>
</tr>
<tr>
<td>39</td>
<td>e₆</td>
<td>True if LMX &lt; LMIN; where LMX and LMIN are in the range 40 ≤ R₁ ≤ 50</td>
</tr>
<tr>
<td>40</td>
<td>f₁</td>
<td>True if RMIN &lt; RMX; where RMIN is in the range 1 ≤ R₂ ≤ 30, and RMX in the range 1 ≤ R₂ ≤ RMIN</td>
</tr>
<tr>
<td>41</td>
<td>f₂</td>
<td>True if RMIN &lt; RMX; where RMIN and RMX are in the range 20 ≤ R₂ ≤ 35</td>
</tr>
<tr>
<td>42</td>
<td>f₃</td>
<td>True if RMIN &lt; RMX; where RMIN and RMX are in the range 35 ≤ R₂ ≤ 50</td>
</tr>
<tr>
<td>43</td>
<td>f₄</td>
<td>True if RMIN &lt; RMX; where RMIN in the range 10 ≤ R₂ ≤ 45 and RMX in the range RMIN ≤ R₂ ≤ 45</td>
</tr>
<tr>
<td>44</td>
<td>g₁</td>
<td>True if W(20) ≥ W(40)</td>
</tr>
<tr>
<td>45</td>
<td>g₂</td>
<td>True if W(25) ≥ W(10)</td>
</tr>
<tr>
<td>46</td>
<td>g₃</td>
<td>True if W(25) ≥ W(40)</td>
</tr>
<tr>
<td>47</td>
<td>g₄</td>
<td>True if W(25) ≥ W(45)</td>
</tr>
<tr>
<td>48</td>
<td>h</td>
<td>True if Ratio ≥ 3</td>
</tr>
</tbody>
</table>
The above observations may be quantified as

\[ \text{LPEAK} > 10 ; \quad 2 < R_3 < 50 \]
\[ \text{RPEAK} > 10 ; \quad 2 < R_4 < 30 \]
\[ \text{LMAX} > \text{LMIN}; \quad 15 < R_1 < 45 \]

Thus, the group of this subclass of numeral '5' consists of the above descriptors as its elements and is formally defined as

\[ G_{5} = \overline{a_1} \land \overline{b_3} \land \overline{c_4} \]

where \( a_1, b_1, c_4 \) are the predicates defined in Table 4.1. In a similar manner, numeral '3' whose profiles are shown in Figure 4.1b may be seen to possess:

\[ \text{LPEAK} > 10 ; \quad \text{(discontinuity in the left profile)} \]
\[ \text{RPEAK} < 10 ; \quad \text{(right profile is relatively smooth)} \]
\[ \text{RMXL} < \text{LMIN} < \text{RMX2} ; \quad \text{(there are two maxima and one minimum in between).} \]

The range, over which the observation is valid, is \( 1 < R_1 < 50 \). Therefore,

\[ G_{3} = \overline{a_1} \land \overline{b_1} \land c_2 \]

where \( a_1, b_1, c_2 \) are the above predicates.

Another illustration will consider the numeral '8'. Figure 4.7 displays a typical set of specimens. The numeral shown in Figure 4.7a may be described by the following feature properties:

\[ \text{LPEAK} < T_1 ; \quad \text{(left profile is relatively smooth)} \]
\[ \text{RPEAK} < T_2 ; \quad \text{(right profile is relatively smooth)} \]
\[ W(\text{RMXL}) < W(\text{RMIN}) < W(\text{RMX2}) ; \quad \text{(the width of the numeral is minimum in the middle region, as shown in Figure 4.8.)} \]
Fig. 4.7 Specimens of the numeral '8'.

Fig. 4.8 Left and right profiles of the numeral '8'. 
where $T_1$ and $T_2$ are specified threshold values. The range $R_1$ for the above features is $(1, 23)$, where $23$ is the height of the numeral. Therefore

$$Gr_8 = a_1 \land \bar{b}_1 \land c_2$$

While a majority of the specimens will have the above features, there are significant exceptions. The numeral specimen in Figure 4.7b, for example, has a break in the right profile (a relatively common occurrence in the numeral '8') and hence RPEAK for this type of '8' will exceed $T_2$. Thus a second subclass for this numeral must be defined. In a similar manner, the numeral specimen shown in Figure 4.7c will have a different left profile and hence a third subclass must be recognized. Therefore, the numeral '8' is characterized by three subclasses.

This procedure is used for each numeral and appropriate groups are then determined.

A group table is constructed for each subclass and stored for use in recognition tests. Table 4.2 lists the "Boolean" expressions for some typical classes of numerals in terms of the logical predicates. More exhaustive listing is presented in Table A3 in Appendix III.

4.4.2 Derivation of Chains

A chain, $Ch(i)$, is either a single group or a union of groups all identifying a specific numeral 'i'. Thus, the chain for a numeral 'i' is defined as

$$Ch(i) \Delta \bigcup_{k=1}^{K} Gr_{i}^{k}$$ (4.15)
<table>
<thead>
<tr>
<th></th>
<th>Gr&lt;sub&gt;1&lt;/sub&gt;</th>
<th>Gr&lt;sub&gt;2&lt;/sub&gt;</th>
<th>Gr&lt;sub&gt;3&lt;/sub&gt;</th>
<th>Gr&lt;sub&gt;4&lt;/sub&gt;</th>
<th>Gr&lt;sub&gt;5&lt;/sub&gt;</th>
<th>Gr&lt;sub&gt;6&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$\tilde{a}_1 \Lambda b_1 \Lambda b_4 \Lambda (\tilde{a}<em>7 \Lambda \tilde{a}</em>{12}) \Lambda a_8$</td>
<td>$\Lambda (d_1 \Lambda d_2) \Lambda \tilde{d}<em>4 \Lambda \tilde{a}</em>{16} \Lambda \tilde{a}_{11}$</td>
<td>$\Lambda f_1 \Lambda$</td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
</tr>
<tr>
<td>0</td>
<td>$\tilde{a}<em>1 \Lambda b_1 \Lambda b_3 \Lambda (\tilde{a}<em>7 \Lambda \tilde{a}</em>{12}) \Lambda a_8 \Lambda (d_1 \Lambda d_2) \Lambda \tilde{a}<em>4 \Lambda \tilde{a}</em>{16} \Lambda \tilde{a}</em>{11}$</td>
<td></td>
<td>$\Lambda f_1 \Lambda$</td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
</tr>
<tr>
<td>1</td>
<td>$\tilde{a}<em>1 \Lambda b_1 \Lambda b_3 \Lambda (\tilde{a}<em>7 \Lambda \tilde{a}</em>{12}) \Lambda a_8 \Lambda (d_1 \Lambda d_2) \Lambda \tilde{a}<em>4 \Lambda \tilde{a}</em>{16} \Lambda \tilde{a}</em>{11}$</td>
<td></td>
<td>$\Lambda e_3 \Lambda f_1$</td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
</tr>
<tr>
<td>1</td>
<td>$a_1 \Lambda b_1 \Lambda b_3 \Lambda (\tilde{a}<em>7 \Lambda \tilde{a}</em>{12}) \Lambda a_8 \Lambda (d_1 \Lambda d_2) \Lambda \tilde{a}<em>4 \Lambda \tilde{a}</em>{16} \Lambda \tilde{a}_{11}$</td>
<td></td>
<td>$\Lambda e_3 \Lambda f_1$</td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
</tr>
<tr>
<td>1</td>
<td>$a_1 \Lambda b_1 \Lambda e_3 \Lambda (a_{13} \Lambda b_8) \Lambda e_3 \Lambda f_1$</td>
<td></td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
</tr>
<tr>
<td>2</td>
<td>$\tilde{a}<em>1 \Lambda b_1 \Lambda b_3 \Lambda a_5 \Lambda (b_9 \Lambda \tilde{a}</em>{15}) \Lambda f_3$</td>
<td></td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
</tr>
<tr>
<td>2</td>
<td>$a_1 \Lambda b_1 \Lambda b_3 \Lambda a_5 \Lambda (a_{13} \Lambda b_8) \Lambda a_{14} \Lambda b_{10} \Lambda f_2$</td>
<td></td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
</tr>
<tr>
<td>2</td>
<td>$\tilde{a}<em>1 \Lambda b_1 \Lambda b_3 \Lambda a_5 \Lambda \tilde{a}</em>{10} \Lambda (a_{13} \Lambda b_8) \Lambda f_2$</td>
<td></td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
</tr>
<tr>
<td>2</td>
<td>$a_1 \Lambda b_1 \Lambda b_3 \Lambda \tilde{a}<em>{10} \Lambda (a</em>{13} \Lambda b_8) \Lambda f_2$</td>
<td></td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
</tr>
<tr>
<td>3</td>
<td>$\tilde{a}_1 \Lambda b_1 \Lambda b_3 \Lambda (\tilde{a}<em>7 \Lambda \tilde{a}</em>{12}) \Lambda a_8 \Lambda (d_1 \Lambda d_2) \Lambda \tilde{a}<em>4 \Lambda a</em>{16} \Lambda a_2 \Lambda f_3$</td>
<td></td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
</tr>
<tr>
<td>3</td>
<td>$\tilde{a}_1 \Lambda b_1 \Lambda b_3 \Lambda (\tilde{a}<em>7 \Lambda \tilde{a}</em>{12}) \Lambda a_8 \Lambda (d_1 \Lambda d_2) \Lambda \tilde{a}<em>4 \Lambda a</em>{16} \Lambda a_2 \Lambda f_3$</td>
<td></td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
</tr>
<tr>
<td>3</td>
<td>$a_1 \Lambda b_1 \Lambda b_3 \Lambda (\tilde{a}<em>7 \Lambda \tilde{a}</em>{12}) \Lambda a_8 \Lambda (d_1 \Lambda d_2) \Lambda a_{16} \Lambda a_2 \Lambda f_3$</td>
<td></td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
</tr>
<tr>
<td>4</td>
<td>$\tilde{a}<em>1 \Lambda b_1 \Lambda b_3 \Lambda (\tilde{a}</em>{12} \Lambda b_8) \Lambda \tilde{a}_8$</td>
<td></td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
</tr>
<tr>
<td>4</td>
<td>$a_1 \Lambda b_1 \Lambda b_3 \Lambda (b_9 \Lambda \tilde{a}_{15}) \Lambda \tilde{a}_7 \Lambda \tilde{a}_5 \Lambda b_5$</td>
<td></td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
</tr>
<tr>
<td>5</td>
<td>$\tilde{a}<em>1 \Lambda b_1 \Lambda b_3 \Lambda (b_9 \Lambda \tilde{a}</em>{15}) \Lambda e_3 \Lambda b_7$</td>
<td></td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
</tr>
<tr>
<td>5</td>
<td>$\tilde{a}<em>1 \Lambda b_1 \Lambda b_3 \Lambda (b_9 \Lambda \tilde{a}</em>{15}) \Lambda a_7 \Lambda a_{13} \Lambda e_1 \Lambda e_6$</td>
<td></td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
</tr>
<tr>
<td>5</td>
<td>$\tilde{a}_1 \Lambda b_1 \Lambda b_4 \Lambda e_4 \Lambda b_9 \Lambda f_3 \Lambda \tilde{b}_6 \Lambda c_1 \Lambda$</td>
<td></td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
</tr>
<tr>
<td>6</td>
<td>$\tilde{a}<em>1 \Lambda b_1 \Lambda b_3 \Lambda (a</em>{13} \Lambda b_8) \Lambda a_{14} \Lambda b_{10}$</td>
<td></td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
<td>$\Lambda$</td>
</tr>
<tr>
<td>Gr\textsuperscript{2} \text{Gr}_6</td>
<td>\tilde{a}_1 \land b_1 \land \bar{b}<em>3 \land (\tilde{a}</em>{15} \lor b_9) \land \bar{a}_7 \land \bar{b}_7 \land \bar{a}_5 \land b_5 \land \bar{b}_2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------</td>
<td>--------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\text{Gr}_6</td>
<td>a_1 \land \bar{b}<em>1 \land \bar{b}<em>3 \land (\tilde{a}</em>{15} \lor \tilde{a}</em>{20}) \land \bar{a}_2 \land b_3 \land \bar{a}_5 \land \bar{b}_7 \land \bar{a}_9 \land \bar{b}_1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\text{Gr}_7</td>
<td>\tilde{a}<em>1 \land b_1 \land b_4 \land (a</em>{12} \land b_7) \land a_7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gr\textsuperscript{2} \text{Gr}_7</td>
<td>\tilde{a}_1 \land b_1 \land \bar{b}<em>3 \land (a</em>{15} \land \bar{b}_9) \land a_6 \land \bar{c}_2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\text{Gr}_8</td>
<td>\tilde{a}_1 \land b_1 \land \bar{b}<em>3 \land (a</em>{15} \land \bar{b}_9) \land a_6 \land c_2 \land \bar{e}_5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\text{Gr}_8</td>
<td>a_1 \land \bar{b}<em>1 \land \bar{b}<em>3 \land (\tilde{a}</em>{20} \lor \tilde{a}</em>{35}) \land \bar{e}_2 \land \bar{e}_3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\text{Gr}_9</td>
<td>a_1 \land b_1 \land \bar{a}<em>3 \land (a</em>{13} \land \bar{b}_1) \land \bar{e}_3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\text{Gr}_9</td>
<td>\tilde{a}_1 \land \bar{b}<em>1 \land b_4 \land (a</em>{12} \land b_7) \land \bar{z}_7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\text{Gr}_9</td>
<td>\tilde{a}_1 \land \bar{b}<em>1 \land b_4 \land (\tilde{a}</em>{12} \lor \bar{b}_7) \land a_8 \land (d_1 \lor d_2) \land \bar{a}<em>4 \land a</em>{16} \land \bar{c}_2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
where $K$ is the number of subclasses for numeral 'i', and $Gr^k_1$ is the group defined for the $k^{th}$ subclass of numeral 'i'.

For example, the chains for the numerals '0', '6' and '8' will be represented as a concatenation of groups as follows: (See Table 4.2)

(i) $\text{Ch}(0) \triangleq \bigcup_{k=1}^{3} Gr^k_0 \quad (4.16)$

where

$Gr^1_0 = a_1 \Lambda b_5 \Lambda a_4 \Lambda (a_{12} V b_7) \Lambda a_8 \Lambda (d_1 V d_2) \Lambda \bar{a}_4 \Lambda \bar{a}_{16} \Lambda \bar{a}_{11} \Lambda g_4$

$Gr^2_0 = a_1 \Lambda \bar{b}_1 \Lambda h \Lambda (g_2 \Lambda g_3) \Lambda e_3 \Lambda f_1$

$Gr^3_0 = a_1 \Lambda b_3 \Lambda \bar{a}_3 \Lambda (\bar{a}_{13} \Lambda \bar{b}_8) \Lambda e_3 \Lambda f_1$

(ii) $\text{Ch}(6) \triangleq \bigcup_{k=1}^{3} Gr^k_6 \quad (4.17)$

where

$Gr^1_6 = a_1 \Lambda b_1 \Lambda a_3 \Lambda (a_{13} V b_8) \Lambda a_{14} \Lambda \bar{b}_{10}$

$Gr^2_6 = \bar{a}_1 \Lambda b_1 \Lambda \bar{a}_5 \Lambda (\bar{a}_{15} V b_9) \Lambda \bar{a}_7 \Lambda \bar{b}_7 \Lambda \bar{a}_5 \Lambda b_5 \Lambda \bar{b}_2$

$Gr^3_6 = a_1 \Lambda \bar{b}_1 \Lambda \bar{a}_3 \Lambda (\bar{g}_2 V \bar{g}_3) \Lambda f_2 \Lambda e_3 \Lambda \bar{f}_1 \Lambda \bar{b}_4 \Lambda \bar{a}_9 \Lambda \bar{g}_1$

(iii) $\text{Ch}(8) \triangleq \bigcup_{k=1}^{3} Gr^k_8 \quad (4.18)$
where

\[
\text{Gr}_8^1 = \overline{a_1} \land \overline{b_1} \land \overline{b_3} \land (\overline{a_{15}} \land \overline{b_9}) \land a_6 \land c_2 \land e_5
\]

\[
\text{Gr}_8^2 = \overline{a_1} \land \overline{b_1} \land \overline{h} \land (\overline{g_2} \lor \overline{g_3}) \land e_2 \land \overline{e_3}
\]

\[
\text{Gr}_8^3 = \overline{a_1} \land b_1 \land \overline{a_3} \land (\overline{a_{13}} \land \overline{b_8}) \land \overline{e_3}
\]

where '⊕', '∧' and '¬' are the logical OR, logical AND and negation respectively, and where \(a, b, \ldots\) are the primitives defined in Table 4.1.

A verbal description of the group \(\text{Gr}_8^2\), as an example, is as follows:

(i) The left and right profiles are relatively smooth, i.e., LPEAK and RPEAK are less than 10 in the ranges \(2 < R_3 < 50\), \(2 < R_4 < 50\).

(ii) The ratio of height to maximum width is less than three.

(iii) The widths at the upper and lower halves are larger than in the middle.

(iv) The left profile attains its minimum value before it attains its maximum value in the range 10 to 30, i.e.,

\[
\text{LMX} > \text{LMIN}
\]

for \(10 < R_1 < 30\).

4.5 Production Rules

Since the structure of a character is defined by the composition of groups in a chain, a set of production rules is formulated using tree grammar [35]. Such a grammar is defined as

\[
G_c : \Delta (V, r, P, S)
\]
\[ V = N \cup \Sigma \] is the grammar alphabet.

\[ N = \text{non-terminal symbols} \ (X_1, X_2, \ldots) \]

\[ \Sigma = \text{terminal symbols (primitives } a, b, c, \ldots) \]

\[(V, r) = \text{is a ranked alphabet, and}\]

\[ P = \text{defines the productions of the form} \]

\[ T_i \rightarrow T_j, \text{ where} \]

\[ T_i \text{ and } T_j \text{ are sub-trees} \]

\[ S = \text{is the starting tree.} \]

The language generated by \( G_\Sigma \) is the set of trees

\[ L(G_\Sigma) = \{ T \mid T \text{ in } T_\Sigma, T_i \xrightarrow{G_\Sigma} T \text{ for some } T_i \text{ in } S \} \quad (4.20) \]

where \( T_\Sigma \) is the set of trees with nodes in \( \Sigma \).

Figure 4.9 illustrates the production rules for character '8' which was previously described as a chain of 3 groups, where

\[ N = \{X_1, X_2, \ldots, X_{19}\} \]

\[ \Sigma = \{a_1, \overline{a_1}, b_1, \overline{b_1}, \overline{h}, \overline{b_3}, \overline{a_3}, a_{15}, \overline{b_9}, \overline{b_2}, \overline{b_3}, \overline{a_{13}}, \overline{b_8}, a_6, c_2, e_5, f_2, \overline{e_3}\} \]

and \( r(\cdot) = \{0, 1\} \cup \{2\} \)

A typical generation of \( G_{\Sigma} \) would be
Fig. 4.9 Production rules of the numeral '8'.
\[(1) \quad S = X_1 \quad (8) \quad a_1 b_1 X_{10} \quad (9) \quad a_1 b_1 a_3 \quad (10) \quad a_1 b_1 a_3 a_{13} X_{12} \\
(11) \quad a_1 b_1 a_3 a_{13} b_8 X_9 \quad (7) \quad a_1 b_1 a_3 a_{13} b_8 e_3 \]

where the parenthetical numbers indicate the production used. Figure 4.10 illustrates the generated tree, representing the numeral '8', after applying productions (1) to (19). The production rules thus derived for the different numerals are used in the parsing process at which stage an identification of the test character is made.

4.6 Recognition Algorithm

The recognition algorithm utilizes the topological features defined earlier, to derive a binary tree structure for eventual numeral identification. Starting with the first difference arrays of the left and right profiles, the tree groups together all the numerals whose LPEAK is less than 10 from those that have LPEAK > 10. Further subclassification is carried out depending on the value of RPEAK and the tree proceeds in this manner to test for the presence or absence of the various features associated with the subgroups of numerals. The tree eventually yields a decision on the identity of the test character. In this study, no provision was made for rejecting invalid characters. An identification (even if wrong) always results.

Figure 4.11 illustrates a portion of the overall tree that leads to recognition of the numerals '0', '1', '2', '3', ..., '8', '9' (subclasses), and Figure 4.12 illustrates a subtree used in the recognition of the numerals '0', '4', '6' and '8'. Since the algorithm only checks for the presence of a specified feature (primitive) at any stage the computational effort is quite small and thus high speed implementation
is made possible.

4.7 Recognition Results

Four hundred and fifty test specimens including broken and continuous numerals were applied to the above recognition process. There were only three misclassifications and none of the errors involved the broken numerals. The misclassified numerals are shown in Figure 4.13.

The overall accuracy worked out to be 99%. In the above experiment all the test specimens had a normal orientation, i.e., there was no deliberate slanting (other than what occurs in writing styles) or rotation of the numeral specimens.

In a limited study to test the robustness of the algorithm against rotation, it was observed that the recognition was affected significantly only if the angle of slant exceeded 15°.

4.8 Summary

In this chapter, a new set of topological features is described. It is assumed that the numerals may be continuous or broken. The topological features are derived from the left and right profiles of the thresholded image of the numerals. Using a structural scheme, features are defined and these are combined to yield a logical description of the different numerals and their subclasses. The recognition algorithm utilizes a tree classifier to determine the identity of the test numeral. Tests with 450 numeral specimens yielded an overall accuracy of 99%.
Fig. 4.10 'Tree representing the numeral '8'.

For $k = 1, 50$
Compute $LP(k), RP(k)$

Fig. 4.11  Portion of the tree used in the recognition of the numerals '0' to '9'.

Fig. 4.12 Sub-tree used in the recognition of the numerals '0', '4', '6' and '8'.
Fig. 4.13 Misclassified errors.
5.1 Introduction

In the two previous chapters, techniques for the recognition of handwritten numerals were presented. It was shown that recognition accuracies in the range of 98-99% were realized with both the decision-theoretic technique using Fourier descriptors and the syntactic technique utilizing topological features. However, these algorithms assume that the numerals are completely isolated, i.e., the characters must be drawn in boxes, and each character is assumed to be isolated from its neighbors. Such assumptions are restrictive since some of the handwritten numerals encountered in the real world are connected, overlapping, and/or touching. Figure 5.1 illustrates some typical specimens of such numeral strings. Therefore, the problem of segmentation of handwritten numerals becomes a central issue in realizing practical recognition systems.

While the features of isolated numerals are easily defined and evaluated, the same is not true of connected strings of numerals. In these cases, measurement of the features of the individual numerals in the string becomes difficult, if not impossible unless the strings are segmented into their components. Moreover, where numerals are connected and/or touching, pseudo-features will be generated and these will have to be filtered out or incorporated in the recognition algorithm.

A literature survey reveals a lack of any relevant work in the field. There is, however, some literature [41, 42, 43] in the area of cursive script recognition. The approaches taken in cursive script reading range
<table>
<thead>
<tr>
<th>02</th>
<th>55</th>
<th>05</th>
<th>07</th>
<th>02</th>
</tr>
</thead>
<tbody>
<tr>
<td>70</td>
<td>76</td>
<td>10</td>
<td>02</td>
<td>54</td>
</tr>
<tr>
<td>75</td>
<td>00</td>
<td>57</td>
<td>05</td>
<td>04</td>
</tr>
<tr>
<td>03</td>
<td>55</td>
<td>53</td>
<td>52</td>
<td>86</td>
</tr>
<tr>
<td>550</td>
<td>753</td>
<td>755</td>
<td>007</td>
<td>954</td>
</tr>
<tr>
<td>000</td>
<td>100</td>
<td>675</td>
<td>760</td>
<td>576</td>
</tr>
</tbody>
</table>

Fig. 5.1 Samples of numeral strings.
from the recognition of whole words to the recognition of words on a letter-by-letter basis. The latter approach [42, 43] that attempts to segment a word into its letter units had the most relevance to the recognition of handwritten numeral strings. One approach to segmentation of words [43] utilizes temporal data on the instantaneous x - y position of the writing stylus to obtain estimates of letter-widths and letter-counts in a word. Additional features such as vertical extent, cusps, closures, strokes, etc., are used to identify the individual letters. An accuracy of 90% was reported. Another approach for segmentation of words[42] detects major changes along the upper and lower envelopes of the word in the course of tracing its top and bottom boundaries from left to right. Segmentation of the cursive line is based on these changes and according to information gathered by an edge tracing routine. The recognition algorithm uses context-information to further refine the accuracy of letter identification. An accuracy of 79% was obtained in a limited study.

The above approaches, however, could not be extended to the recognition of numeral strings due to the following reasons:

1. The cursive scripts are assumed to have a non-overlapping string of connected letters, a condition that is not valid for handwritten numeral strings.

2. The recognition accuracies of 79% is too low to be of practical significance.

3. The techniques are, in general, very writer dependent.
In view of these difficulties a totally different approach that utilizes the unique characteristics of handwritten numeral strings is proposed. The technique utilizes semantic knowledge together with sequential reasoning to determine the decision boundary between the numerals in the string. The decision boundary is evaluated by testing various hypotheses concerning the numerals in the string and is essentially hierarchical.

5.2 Segmentation of Numeral Strings

The recognition of connected numeral strings is best achieved by segmenting the numeral string into its individual units and applying the recognition algorithm to the isolated numerals. Since the main focus in this approach is the recognition of the numerals in a string, segmentation is defined, in a broad sense as the process by which the features of the individual numerals in a connected string are obtained. The segmentation process is very difficult, in general, due to the following reasons:

(a) the number of numerals in a string is variable;
(b) the numerals encountered in practice are isolated, broken, touching, connected or appear with a combination of these attributes;
(c) the boundary between two numerals, connected or not, may in general be nonlinear.

The segmentation algorithm described in this section is derived on the following assumptions:

(i) The number of numerals in a string is known a priori.
(ii) The heights of the numerals are nearly the same.

The segmentation algorithm proposed is hierarchical in nature. It tests various hypotheses ranging from the case where the numerals are completely isolated and well separated to that where the numerals may be connected, touching and possibly broken as well. Each numeral string is first imaged thresholded and then enclosed with a rectangular frame of height $H$ and width $W$. Fig. 5.2 displays a digitized image and the corresponding thresholding image.

5.2.1 Segmentation Algorithm

The algorithm is developed by testing the validity of certain assumptions about the numerals in a given string. It is assumed without loss of generality, that there are only two numerals in a string. The following are the various tests presented in an ascending order of complexity.

(a) Test for isolated and well separated numerals

Two numerals are considered to be isolated if a vertical line separating the two numerals can be found.

The assumption that the two numerals are isolated is tested first by initiating a vertical scan starting in the middle of the frame, and counting the number of transitions $VT$ from bright (boundary) to dark (character).

(i) If $VT = 0$, then the two numerals are isolated and hence are already in a segmented form.

(ii) In case $VT > 1$, initiate a vertical scan at a point one pixel to the left of the center of the frame and count the number of transitions. If $VT$ is still greater than or equal
<table>
<thead>
<tr>
<th>(a)</th>
<th>(b)</th>
</tr>
</thead>
<tbody>
<tr>
<td>785 02</td>
<td></td>
</tr>
<tr>
<td>S7 007</td>
<td></td>
</tr>
<tr>
<td>55 70 00</td>
<td></td>
</tr>
<tr>
<td>03 57 76</td>
<td></td>
</tr>
<tr>
<td>785 02</td>
<td></td>
</tr>
<tr>
<td>S7 007</td>
<td></td>
</tr>
<tr>
<td>55 70 00</td>
<td></td>
</tr>
<tr>
<td>03 57 76</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 5.2 (a) Digitized image
(b) Thresholded image.
to 1, start the vertical scan at a point one pixel to the
right of the center of the frame. This procedure is repeated
by moving further to the left or to the right and initiating
a new scan until VT = 0. Figure 5.3 displays a string with
two isolated numerals and the corresponding decision boundary.
If the procedure fails, that is, VT is still \( \geq 1 \), after a
prescribed number of vertical scans, \( \epsilon_v \), then a test for non-
isolated numerals is implemented.

(b) **Test for non-isolated numerals**

If the numerals in a string are connected or touching or if they
exist in overlapping fields, the above test will fail. In such cases
segmentation may still be achieved if at least one point on the decision
boundary is either known a priori or determined from topological analysis.
In this procedure, the scan is initiated at a point \( P \) inside the frame
with coordinates \( (L,K) \), as shown in Figure 5.4. The scanning procedure
is described in the next section.

5.2.2 **Procedure for Non-linear Segmentation**

In order to find the non-linear decision boundary between the
numerals in a string, the scan starts at a point \( P \) and proceeds up by
a path that is midway between the left and right numerals till the top
of the frame, and down by following the border of the left numeral till
the end of the frame as shown in Figure 5.5. In both cases (up and down);
sixteen possible situations are encountered, and these determine the
scanning direction and subsequent operations. These conditions are
Fig. 5.3 Linear decision boundary.

Fig. 5.4 Initial scanning point.

Fig. 5.5 Non-linear decision boundary.
illustrated in Table 5.1 where

'1' the background of the image,

'0' the character,

'b' the label of the border element,

'x' the current point \( P_1 \) with coordinates \((L_1,K_1)\) on the decision boundary,

\( \uparrow \) moving up,

\( \uparrow \) moving up and to the right,

\( \downarrow \) moving up and to the left,

\( \downarrow \) moving down,

\( \downarrow \) moving down and to the right,

\( \downarrow \) moving down and to the left,

\( (J) \) the move is followed by algorithm \( (J) \), \( J = 1 \) or 2 or 3 or 4 as indicated in Table 5.1.

At any stage the scanning direction for finding the next point on the decision boundary is determined as per Table 5.1 and this move is then followed by the implementation of one of four algorithms as needed.

Given the thresholded image \((IX(i,j), \ i = 1,m; \ j = 1,n)\), where \( m \) and \( n \) are the dimensions of the frame, and defining \((L_1,K_1)\) as the coordinates of the current point on the decision boundary, and \((L,K)\) as the coordinates of the next point on the decision boundary, the descriptions of algorithms (1), (2), (3) and (4) are given below.

**Algorithm (1)**

The scanning direction is up and to the left (Figure 5.6a).
### Table 5.1 Scanning Directions

<table>
<thead>
<tr>
<th>Case No.</th>
<th>Scanning Up</th>
<th>Scanning Direction and Algorithm (j) if needed</th>
<th>Scanning Down</th>
<th>Scanning Direction and Algorithm (j) if needed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 1 1</td>
<td><img src="image" alt="Arrow Up" /> (3)</td>
<td>x</td>
<td><img src="image" alt="Arrow Down" /> (4)</td>
</tr>
<tr>
<td></td>
<td>x</td>
<td></td>
<td>1 1 1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0 0 0</td>
<td><img src="image" alt="Arrow Down" /></td>
<td>x</td>
<td><img src="image" alt="Arrow Down" /></td>
</tr>
<tr>
<td></td>
<td>x</td>
<td></td>
<td>0 0 0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0 0 b</td>
<td><img src="image" alt="Arrow Right" /> (2)</td>
<td>x</td>
<td><img src="image" alt="Arrow Right" /></td>
</tr>
<tr>
<td></td>
<td>x</td>
<td></td>
<td>0 0 b</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0 b 0</td>
<td><img src="image" alt="Arrow Down" /></td>
<td>x</td>
<td><img src="image" alt="Arrow Down" /></td>
</tr>
<tr>
<td></td>
<td>x</td>
<td></td>
<td>0 b 0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>b 0 0</td>
<td><img src="image" alt="Arrow Right" /> (1)</td>
<td>x</td>
<td><img src="image" alt="Arrow Right" /></td>
</tr>
<tr>
<td></td>
<td>x</td>
<td></td>
<td>b 0 0</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1 1 b</td>
<td><img src="image" alt="Arrow Right" /> (1)</td>
<td>x</td>
<td><img src="image" alt="Arrow Right" /> (4)</td>
</tr>
<tr>
<td></td>
<td>x</td>
<td></td>
<td>1 1 b</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>b 1 1</td>
<td><img src="image" alt="Arrow Right" /> (2)</td>
<td>x</td>
<td><img src="image" alt="Arrow Right" /></td>
</tr>
<tr>
<td></td>
<td>x</td>
<td></td>
<td>b 1 1</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>b 1 1</td>
<td><img src="image" alt="Arrow Down" /></td>
<td>x</td>
<td><img src="image" alt="Arrow Down" /></td>
</tr>
<tr>
<td></td>
<td>x</td>
<td></td>
<td>b 1 1</td>
<td></td>
</tr>
<tr>
<td>Case No.</td>
<td>Scanning Up</td>
<td>Scanning Direction and Algorithm (j) if needed</td>
<td>Scanning Down</td>
<td>Scanning Direction and Algorithm (j) if needed</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td>-----------------------------------------------</td>
<td>--------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>9</td>
<td>0 b b</td>
<td>x</td>
<td>x</td>
<td>0 b b</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>b b 0</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>b b 1</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1 1 1</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>b 1 b b b b</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>0 b 1</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>b 1 b b b 1</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>1 1 1 0</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
IF (IX(L₁−1,K₁−1) is a border element 'b'
and IX(L₁−1,K₁−2) is a background '1'
THEN
scan right to left until a border element 'b'
at IX(L₁−1,K₂) is found
IF [(K₁−1)−K₂] > 2
THEN
  K = (K₁−1) + K₂/2
  L = L₁−1
ELSE
  K = K₁−1
  L = L₁−1
ENDIF
ELSE
  K = K₁−1
  L = L₁−1
ENDIF

Algorithm (2)
The scanning direction is up and to the right (Figure 5.6b).
IF IX(L₁−1,K₁+1) is a border element 'b'
and IX(L₁−1,K₁+2) is a background '1'
THEN
scan left to right until a border element 'b'
at IX(L₁−1,K₂) is found
Fig. 5.6a Scanning direction for algorithm (1).

<table>
<thead>
<tr>
<th>L_{i-1}, K_2</th>
<th>---</th>
<th>L_{i-1}, K_{i-2}</th>
<th>L_{i-1}, K_{i-1}</th>
<th>L_{i-1}, K_{i}</th>
<th>L_{i-1}, K_{i+1}</th>
<th>---</th>
</tr>
</thead>
<tbody>
<tr>
<td>---</td>
<td>?</td>
<td>X</td>
<td>L_{i}, K_{i}</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
</tbody>
</table>

Fig. 5.6b Scanning direction for algorithm (2).

<table>
<thead>
<tr>
<th>---</th>
<th>L_{i-1}, K_{i-1}</th>
<th>L_{i-1}, K_{i}</th>
<th>L_{i-1}, K_{i+1}</th>
<th>L_{i-1}, K_{i+2}</th>
<th>---</th>
<th>L_{i-1}, K_{2}</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>L_{i}, K_{i}</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
</tbody>
</table>
IF \[K_2 - (K_1 + 1)\] > 2

THEN

\[K = [(K_1 + 1) + K_2]/2\]

\[L = L_1 - 1\]

ELSE

\[K = K_1 + 1\]

\[L = L_1 - 1\]

ENDIF

ELSE

\[K = K_1 + 1\]

\[L = L_1 - 1\]

ENDIF

Algorithm (3)

The scanning direction is up (Figure 5.6c).
Scan left to right until a border element 'b'

at IX(L_1 -1,K_2) is found, and scan right to left

until a border element 'b' at IX(L_1 -1,K_3) is found.

IF \((K_2 - K_3) > 2\)

THEN

\[K = (K_2 + K_3)/2\]

\[L = L_1 - 1\]

ELSE

\[K = K_1\]

\[L = L_1 - 1\]

ENDIF
Fig. 5.6c Scanning direction for algorithm (3).

<table>
<thead>
<tr>
<th>L_{1-1}, K_2</th>
<th>L_{1-1}, K_{1-1}</th>
<th>L_{1-1}, K_1</th>
<th>L_{1-1}, K_{1+1}</th>
<th>---</th>
<th>L_{1-1}, K_2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 5.6d Scanning direction for algorithm (4).
Algorithm (4)

The scanning direction is down, or down and to the left (Figure 5.6d).
Scan right to left.

If a border element 'b' at IX(L_1+1,K_2) is found

Then

\[ K = K_2 \]
\[ L = L_1 + 1 \]

Else

\[ K = K_1 \text{ (or } K_1 - 1 \) \]
\[ L = L_1 + 1 \]

Endif

The procedure of segmentation will be illustrated by considering the string shown in Figure 5.7a. In this case, the starting point P for determining the decision boundary is located on the right portion of the contour of the numeral '5'. This point P_1 corresponds to the situation in entry 14 in Table 5.1. From Table 5.1, the scanning direction is found to be vertical (↑) and the use of algorithm (2) is required. In algorithm (2), a 'left to right' scan is initiated until a border element is encountered. The midpoint of the two border elements is evaluated and the next point on the decision boundary is located at this midpoint (P_2) as shown in Figure 5.7a. The new boundary point P_3 corresponds to entry 1 in Table 5.1. The scanning direction is vertical and the implementation of algorithm (3) is required. In algorithm (3), starting at P_3, horizontal scans on either sides of P_3 are carried out until two border elements on either sides of P_3 are determined. The
Fig. 5.7 Segmentation procedure.
mid point of these two border elements is the next point on the decision boundary as shown in Figure 5.7b. This procedure is repeated until the top of the frame is reached.

In a similar manner, starting at $P_1$, a downward search is initiated to determine the lower portion of the decision boundary as shown in Figure 5.7c. The results of this procedure are shown in Figure 5.7d.

The success of this scheme is critically dependent on the knowledge or the evaluation of the initial point $P$ on the decision boundary. The choice of this initial point is discussed next.

5.2.3 Determination of the Initial Scanning Point

The determination of the initial scanning point, a key aspect of the segmentation scheme, is in general quite complicated. This difficulty is caused by the diversity and the manner in which non-isolated numerals occur in the real world. The procedure for choosing the initial scanning point utilizes the known topological features of the individual numerals. The topological features used in this procedure are:

(i) The number of horizontal border-to-background, or border-to-border transitions at certain locations in the frame.

The above features provide clues about the numerals in a string. Thus if the number of transitions is only two, then the two numerals in a string are any two of '1', '2', '3', '5' and '7'. If the number of transitions at a location is four, then the numerals are '0', '4', '6', '8' or '9'.

(ii) The smoothness of the left and right profiles of the string.

The smoothness measure gives important information about the type of numeral in the left or right most positions. Thus
numerals such as '0', '1', '6' and '8' have relatively smooth left profiles, while numerals such as '0', '1', '3', '7', '8' and '9' have relatively smooth right profiles.

The following steps utilize the above features to determine the initial scanning point.

**Step 1: (HT = 2 or 4)**

Starting at the middle of the frame, evaluate the number of horizontal border-to-background or border-to-border transitions HT. If HT is not equal to two or four, then initiate a horizontal scan at a point one pixel up from the middle of the frame and count the number of transitions HT. If HT is still not equal to two or four, then start the horizontal scan one pixel down from the middle of the frame. Repeat this procedure by moving further up or down until HT is equal to two or four.

If HT = 2, then L corresponds to the row where the number of horizontal transitions is equal to two, and K is chosen after the first transition as shown in Figures 5.8a and b.

If HT = 4, then L corresponds to the row where the number of horizontal transitions is equal to four, and K is chosen after the second transition as shown in Figure 5.8c.

Once the initial point is determined, then the segmentation procedure described earlier is applied to determine the decision boundary. If the procedure fails after a prescribed number of horizontal scans, \( c_H \), as shown in Figure 5.8d, then step 2 is implemented.

**Step 2: (HT = 3)**

Search for three horizontal transitions as described in step 1. If there are not three transitions then the string is rejected. The
Fig. 5.8 Horizontal transitions.
number of horizontal transitions equal to three implies the following possibilities:

(a) one of the two numerals is a '0';
(b) the two numerals are not connected but overlapping or
(c) the two numerals are connected, touching and/or overlapping.

In all cases, L corresponds to the horizontal line where HT = 3 as shown in Figure 5.8d, while K is determined as follows:

Scan horizontally at L from left to right to generate the string \( S_j(i) \), \( i = 1, 2, \ldots, N; \ j = 1, 2, \ldots, M \); where \( N \) is the total number of symbols in the string, and \( M \) is the total number of states. The string \( S_j(i) \) is derived over the alphabet \( Z \) where

\[
Z = \{ Q, 1, 4, 6 \}
\]

'0' the label for the character
'1' the label for the background
'4' the label of the first detected border,
'6' the label of other borders.

Derivation of the string can be represented as a finite automaton, which will be denoted by a 5-tuple \((Q, Z, d, q_1, F)\), where

- \( Q \) is a finite set of states,
- \( Z \) is a finite input alphabet,
- \( q_1 \) in \( Q \) is the initial state,
- \( F \) in \( Q \) is the final state,
- \( d \) is the transition function mapping \( Q \times Z \) to \( Q \).

That is, \( d(q, a) \) is a state for each state \( q \) and input symbol \( a \).
As an illustration, consider the two numeral strings shown in Figure 5.9a and b. The string \( S_j^1(i) \) derived at \( L \) for the first numeral string is
\[
S_j^1(i) = 1406160416061
\]
while \( S_j^2(i) \) derived at \( L \) for the second numeral string is
\[
S_j^2(i) = 1406160414041
\]
where \( i = 1,2,\ldots,N; \) and \( N \) is the total number of symbols in the string \( S_j(i), \) and \( j = 1,2,\ldots,M; \) where \( M \) is the number of states and is equal to thirteen in both strings \( S_j^1(i) \) and \( S_j^2(i), \) and \( A \) is a string of pixels having the label \( \lambda. \) (\( \lambda = 0 \) or 1 or 4 or 6).

It should be mentioned that \( S_j^1(i) \) and \( S_j^2(i) \) will always be equal to '1' since the first and the last pixels in any horizontal scan belongs to the background.

Figures 5.10a and b show the transition graphs recognizing the strings \( S_j^1(i) \) and \( S_j^2(i) \) respectively. The initial state is \( q_1 \) and will always accept an input '1', and a transition will occur if \( a_n \neq a_{n+1}, \)
where \( a_n \) and \( a_{n+1} \) are two consecutive input symbols. Similarly, the final state is \( q_{13}, \) and will always accept an input symbol '1'.

The string \( S_j(i) \) is then examined and two different situations may arise:

**Case (1):**

\[
\text{if } [ (S_2(i)=4) \land (S_{M-1}(i)=6) ] \lor [ (S_2(i)=6) \land (S_{M-1}(i)=4) ]
\]

\( \implies \) numerals are overlapping and not connected

i.e., if the suffix is '14' and the prefix is '61' or if
Fig. 5.9 Numeral strings with three horizontal transitions.
Fig. 5.10 (a) Transition graph recognizing the string $S_j^1$ (i).
(b) Transition graph recognizing the string $S_j^2$ (i).
the suffix is '16' and the prefix is '41', then the two numerals are overlapping and not connected as shown in Figure 5.9a.

Case (2):

\[ \text{if } [(S_2(i)=4) \land (S_{M-1}(i)=4)] \lor [(S_2(i)=6) \land (S_{M-1}(i)=6)] \]

\[ \implies \text{numerals are connected.} \]

i.e., if the suffix is '14' (or '16') and the prefix is '41' (or '61'), then the two numerals are connected as shown in Figure 5.9b:

In cases (1) and (2) we count the number of transitions:

\[ (41)_p \triangleq \ '4' \text{ to } '1' \text{ transitions} \]

\[ (61)_p \triangleq \ '6' \text{ to } '1' \text{ transitions} \]

\[ (46)_p \triangleq \ '4' \text{ to } '6' \text{ transitions} \]

\[ (64)_p \triangleq \ '6' \text{ to } '4' \text{ transitions} \]

where \( p \) represents the number of transitions encountered in the string \( S_j(i) \). These two cases are discussed in detail below.

Case 1:

Overlapping but not connected numerals

In this case, \( S_2(i) \neq S_{M-1}(i) \). There are two cases to be considered:

(a) The suffix is '14' and the prefix is '61'

that is, \( (S_2(i) = 4) \land (S_{M-1}(i) = 6) \)

In this case the two numerals are not connected and the border of the left numeral is detected first.
(i) Existence of a '4' to '6' transition, i.e., \((46)_1\), means that the two outer borders of the left and right numerals are touching as shown in Figures 5.11a. The string \(S_j(i)\) generated at \(L\) for the two connected numerals of Figure 5.11a is
\[
S_j(i) = 1 4 0 4 6 0 6 1 6 0 6 1
\]
and \(K\) is the coordinate of the pixel corresponding to '4' of the '4' to '6' transition in \(S_j(i)\). In other words,
\[
K = \{ i \mid S_j(i) = 4 \in (46)_1 \}
\]
Once \(L\) and \(K\) are determined the decision boundary is then evaluated. If \((46)_0\) then go to (ii).

(ii) If the left numeral is a '0', as in the case of Figure 5.11b, the string \(S_j(i)\) generated at \(L\) is
\[
S_j(i) = 1 4 0 6 1 6 0 4 1 6 0 6 1
\]
In this case, there are one '4' to '1' transition and two '6' to '1' transitions, that is
\[
(41)_1 \land (61)_2
\]
and
\[
K = \{ i \mid S_j(i) = 4 \in (41)_1 \}
\]

(iii) If the left numeral is an open '0', as shown in Figure 5.11c, the string \(S_j(i)\) generated at \(L\) is
\[
S_j(i) = 1 4 0 4 1 4 0 4 1 6 0 6 1
\]
In this case, there are two '4' to '1' transitions and
Fig. 5.11 Nonconnected numeral strings with three horizontal transitions.
one '6' to '1' transition, that is

\[(41)_2 \land (61)_1\]

and

\[K = \{ i \mid S_j(i) = 4 \in (41)_2 \}\]

(iv) If the right numeral is an open '0', as shown in Figure 5.11d, the string \(S_j(i)\) generated at \(L\) is

\[S_j(i) = 1 \ 4 \ 0 \ 4 \ 1 \ 6 \ 0 \ 6 \ 1 \ 6 \ 0 \ 6 \ 1 \]

In this case, there are one '4' to '1' transition and two '6' to '1' transitions, that is

\[(41)_1 \land (61)_2\], as in (ii),

and

\[K = \{ i \mid S_j(i) = 4 \in (41)_1 \}\]

(b) The prefix is '16' and the suffix is '41'.

that is, \((S_2(i) = 6) \land (S_{n-1}(i) = 4)\).

The two numerals are also not connected but the border of the right numeral is detected first. Table 5.2 illustrates the four possibilities encountered in this case.

Case 2:

Connected Numerals

In this case, \(S_2(i) = S_{n-1}(i)\), that is the prefix is '14' and the suffix is '41' as in Figure 5.12a, or the prefix is '16' and the suffix is '61' as in Figure 5.12b.

When the two numerals are connected, two different numeral strings can generate the same string \(S_j(i)\). Figure 5.13 illustrates a set of strings that fall within this category. As an example, the two strings
<table>
<thead>
<tr>
<th>Case No.</th>
<th>Example of Numeral String</th>
<th>Properties of $S_j(t)$</th>
<th>Value of $K$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><img src="image" alt="Numeral String 1" /></td>
<td>$S_j(t) = 6 \in (66)_1$</td>
<td>$t</td>
</tr>
<tr>
<td>2</td>
<td><img src="image" alt="Numeral String 2" /></td>
<td>$(66)_1$</td>
<td>$t</td>
</tr>
<tr>
<td>3</td>
<td><img src="image" alt="Numeral String 3" /></td>
<td>$(66)_1$</td>
<td>$t</td>
</tr>
<tr>
<td>4</td>
<td><img src="image" alt="Numeral String 4" /></td>
<td>$(66)_1$</td>
<td>$t</td>
</tr>
</tbody>
</table>
Fig. 5.12 Connected numerals.

Fig. 5.13 Samples of connected numeral strings with three horizontal transitions.

Fig. 5.14 Two numeral strings generating the same string $S_j(i)$. 
shown in Figures 5.14a and b, generate the same string \( S_j(i) \)

\[
S_j(i) = 1 \quad 4 \quad 0 \quad 6 \quad 1 \quad 6 \quad 0 \quad 6 \quad 1 \quad 6 \quad 0 \quad 4 \quad 1
\]

which creates ambiguities since \( K \) for the first string equal to

\[
\{ i \mid S_j(i) = 6 \in (61)_2 \}, \text{ while for the second string } K \text{ is equal to } \\
\{ i \mid S_j(i) = 6 \in (61)_1 \}.
\]

In order to resolve this ambiguity, the decision will be based on global contextual information. The profiles of the two connected numerals is computed over the whole frame. The smoothness of the left and right profiles is evaluated as the measure of discontinuity of that profile, and is given by \( \text{LPEAK} \) and \( \text{RPEAK} \), where \( \text{LPEAK} \) and \( \text{RPEAK} \) are the peak to peak value of the first difference of the left and right profiles respectively. The left (or right) profile is considered to be smooth if \( \text{LPEAK} \) (or \( \text{RPEAK} \)) is less than \( W/5 \), where \( W \) is the width of the frame.

If both profiles are not smooth [i.e., \( (\text{LPEAK} \text{ and } \text{RPEAK}) > W/5 \)], then divide the frame into two equal regions and evaluate the left profile from the left side of the frame to its middle, and the right profile from the right side of the frame to its middle. This is illustrated in Figures 5.15a and b respectively.

If the left and right profiles are still not smooth [i.e., \( (\text{LPEAK} \text{ and } \text{RPEAK}) > W/5 \)], then the numeral string is rejected. Otherwise, if the two profiles are smooth [i.e., \( (\text{LPEAK} \text{ and } \text{RPEAK}) < W/5 \)], then the left numeral is a '0' and \( K \) is chosen after the second horizontal transition. Consider as an example the numeral string of Figure 5.16a, the string \( S_j(i) \) evaluated at \( L \) is
Fig. 5.15a Global profiles evaluated over the whole frame.

Fig. 5.15b Global profiles evaluated till the middle of the frame.
\[ S_j(i) = 1 \ 4 \ 0 \ 6 \ 1 \ 6 \ 0 \ 4 \ 1 \ 4 \ 0 \ 4 \ 1 \]

In this case, \( [(\text{LPEAK and RPEAK}) < W/5] \land (41)_2 \land (61)_1 \), and 
\[ K = \{ i \mid S_j(i) = 4 \in (41)_1 \} \).

If the left profile is smooth while the right profile is not, and there are one '4' to '1' transition and two '6' to '1' transitions, that is

\[ \text{IF } (\text{LPEAK} < W/5) \land (\text{RPEAK} > W/5) \land (41)_1 \land (61)_2 \]

\[ \text{THEN} \]

\[ K = \{ i \mid S_j(i) = 6 \in (61)_2 \} \]

\[ \text{ENDIF} \]

as in the case of Figure 5.16b where the two numerals are connected and touching, and the left numeral is '0' and the string \( S_j(i) \) generated at \( L \) is

\[ S_j(i) = 1 \ 4 \ 0 \ 6 \ 1 \ 6 \ 0 \ 6 \ 1 \ 6 \ 0 \ 4 \ 1 \]

Table 5.3 illustrates all the possibilities encountered when the two numerals are connected and/or touching.

5.3 **Extension of the Algorithm**

Although only two-numeral strings were considered in the development of the segmentation scheme, the methods can be easily extended to connected strings of more than two numerals. The procedure will be illustrated by considering numeral strings with three digits. The numeral string is enclosed in a rectangular frame with height \( H \) and width \( W \). The frame is then divided into three equal regions. Tests for isolated numerals is first initiated. If a vertical line separating each two numerals is found,
Fig. 5.16 Connected strings.

(a) \(07\)  
(b) \(02\)

Fig. 5.17 Numeral strings with three digits.  
(a) Isolated numerals  
(b) One isolated and two connected numerals  
(c) Three connected numerals.
### Table 5.3 Determination of the Initial Scanning Point for Connected Numeral Strings

<table>
<thead>
<tr>
<th>Case No.</th>
<th>Example of Numeral String</th>
<th>( S_j(1) ) Generated at I.</th>
<th>Properties of the Numeral String</th>
<th>Value of K</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>L[07]</td>
<td>( 1 4 0 6 1 6 2 4 1 4 0 4 1 )</td>
<td>([\text{LPEAK and RPEAK} &lt; \bar{W}/5]) ((41)_2 \land (61)_1)</td>
<td>((1 \mid S_j(1) = 4 \in (41)_1))</td>
</tr>
<tr>
<td>2</td>
<td>L[08]</td>
<td>( 1 4 0 6 1 6 0 6 1 6 0 4 1 )</td>
<td>((\text{LPEAK} &lt; \bar{W}/5) \land (\text{RPEAK} &gt; \bar{W}/5)) ((41)_1 \land (61)_2)</td>
<td>((1 \mid S_j(1) = 6 \in (61)_2))</td>
</tr>
<tr>
<td>3</td>
<td>L[20]</td>
<td>( 1 4 0 6 1 6 0 6 1 6 0 4 1 )</td>
<td>((\text{LPEAK} &gt; \bar{W}/5) \land (\text{RPEAK} &lt; \bar{W}/5)) ((41)_1 \land (61)_2)</td>
<td>((1 \mid S_j(1) = 6 \in (61)_1))</td>
</tr>
<tr>
<td>4</td>
<td>L[02]</td>
<td>( 1 4 0 6 1 6 0 4 1 4 0 4 1 )</td>
<td>((\text{LPEAK} &lt; \bar{W}/5) \land (\text{RPEAK} &gt; \bar{W}/5)) ((41)_2 \land (61)_1 \land (S_4(1) = 6))</td>
<td>((1 \mid S_j(1) = 4 \in (41)_1))</td>
</tr>
<tr>
<td>5</td>
<td>L[20]</td>
<td>( 1 4 0 6 1 6 0 4 1 4 0 4 1 )</td>
<td>((\text{LPEAK} &gt; \bar{W}/5) \land (\text{RPEAK} &lt; \bar{W}/5)) ((41)_2 \land (61)_1 \land (S_4(1) = 6))</td>
<td>((1 \mid S_j(1) = 6 \in (61)_1))</td>
</tr>
</tbody>
</table>
Table 5.3 (Cont'd)

<table>
<thead>
<tr>
<th>Case No.</th>
<th>Example of Numeral String</th>
<th>$S_j(1)$ Generated at L.</th>
<th>Properties of the Numeral String</th>
<th>Value of K</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>L20</td>
<td>1404140616041</td>
<td>(LPEAK &gt; W/5) ∧ (RPEAK &lt; W/5) (41)_2 ∧ ((61)_1 ∧ (S_4(1) = 4))</td>
<td>(1 \mid S_j(1) = 4 \epsilon (41)_1)</td>
</tr>
<tr>
<td>7</td>
<td>L02</td>
<td>1404140616041</td>
<td>(LPEAK &lt; W/5) ∧ (RPEAK &gt; W/5) (41)_2 ∧ ((61)_1 ∧ (S_4(1) = 4))</td>
<td>(1 \mid S_j(1) = 6 \epsilon (61)_1)</td>
</tr>
<tr>
<td>8</td>
<td>L02</td>
<td>1404140414041</td>
<td>(LPEAK &lt; W/5) ∧ (RPEAK &gt; W/5) (41)_3 ∧ ((61)_0)</td>
<td>(1 \mid S_j(1) = 4 \epsilon (41)_2)</td>
</tr>
<tr>
<td>9</td>
<td>L20</td>
<td>1404140614041</td>
<td>(LPEAK &gt; W/5) ∧ (RPEAK &lt; W/5) (41)_3 ∧ ((61)_0)</td>
<td>(1 \mid S_j(1) = 4 \epsilon (41)_1)</td>
</tr>
<tr>
<td>10</td>
<td>L05</td>
<td>1606160616061</td>
<td>(LPEAK &lt; W/5) ∧ (RPEAK &gt; W/5) (41)_0 ∧ ((61)_3)</td>
<td>(1 \mid S_j(1) = 6 \epsilon (61)_2)</td>
</tr>
</tbody>
</table>
then the three numerals are isolated as shown in Figure 5.17a. If one numeral is isolated and two are connected, as shown in Figure 5.17b, then the procedure of non-linear segmentation of two numerals proposed in section 5.2 is applied to the connected strings after isolation of the left (or right) numeral.

If the three numerals are connected, then the bottom profiles are examined for peaks above a certain threshold. If such peaks exist then a vertical line passing through these peaks is taken as the decision boundaries. This is illustrated in Figure 5.17c for a typical specimen of 3-numeral string. Even if no peaks above a certain threshold are discernible, the presence of peaks can be used to locate an initial scanning point. The methods described previously for 2-numeral strings can then be used. When the bottom profile does not indicate clearly defined peaks, then horizontal scans at the middle of the frame are initiated and the number of horizontal transitions $HT$ is determined. If $HT$ is equal to three, then $L_1$ corresponds to the row where $HT = 3$, $K_1$ is chosen after the first transition, and $K_2$ is chosen after the second transition, where $(L_1, K_1)$ are the coordinates of the initial scanning point on the decision boundary between the first and second numerals, and $(L_1, K_2)$ are the coordinates of the initial scanning point on the decision boundary between the second and third numerals. Similarly, if $HT = 6$, then $L_1$ corresponds to the row where $HT = 6$, $K_1$ is chosen after the second transition, and $K_2$ is chosen after the fourth transition. If $HT$ is not equal to three or six after a prescribed number of horizontal scans, then search for four or five horizontal transitions. The string $S_j(i)$ is generated by
scanning from left to right at the row where HT is equal to four or five, \( L_1 \) corresponds to that row, while \( K_1 \) and \( K_2 \) are evaluated using the properties of the generated string \( S_3(i) \) as well as the smoothness of the left or right profile of the numeral string.

5.4 Recognition Algorithm

Once the numeral string has been properly segmented, the recognition algorithm developed earlier in Chapter IV is applied to yield an identification decision. It should, however, be noted that additional subclasses for each numeral are defined to account for the connecting tail that is still present after segmentation.

The syntactic technique developed in Chapter IV yielded a recognition accuracy of 92% when applied to isolated and connected numeral strings. Figures 5.18, 5.19 and 5.20 illustrate the left and right profiles of segmented numeral strings with two, three and four horizontal transitions respectively. Appendix IV illustrates the left and right profiles of several segmented numeral strings. It is clear that recognition of the individual numerals from their profile descriptions will be straightforward.

5.5 Summary

In this chapter, a technique for segmenting numeral strings is proposed. The segmentation algorithm is hierarchical in nature. It tests various hypotheses ranging from the case where the numerals are completely isolated and well separated to that where the numerals may be connected, touching, broken and overlapping, or they appear in a
combination of these attributes.

The technique utilizes semantic knowledge together with sequential reasoning to determine the decision boundary between the numerals in the string.

Once the numeral string has been segmented, the syntactic algorithm, developed in Chapter IV, is applied to recognize the numeral string. An accuracy of 92% was obtained when the test data consisted of isolated and connected numerals.
Fig. 5.18 Left and right profiles of segmented numerals with HT = 2.

Fig. 5.19 Left and right profiles of segmented numerals with HT = 3.
Fig. 5.20 Left and right profiles of segmented numeral string with HT = 4.
Chapter VI

SUMMARY AND CONCLUSIONS

The principal objective of the research work described in this thesis was to develop an algorithm for the recognition of handwritten numeral strings.

To achieve this goal, we first investigated the application of a new class of Fourier shape descriptors to the recognition of isolated handwritten numerals. The Fourier shape descriptors are invariant to rotation, translation, size and change in the starting point. The problem of variability within each group representing a certain numeral was resolved using a functional mapping approach together with an efficient version of the K-MEANS clustering algorithm. The functional mapping technique provided the initial conditions used in the K-MEANS algorithm. This procedure was applied to the feature vectors of each numeral and a total of 26 distinct classes were obtained for the 400 numerals used for training the classifier.

The recognition algorithm was applied to the 400 numerals representing the test set. The squared Euclidean distance when applied, yielded a recognition accuracy of 92%, while the Mahalanobis distance measure yielded 89.5% when the pooled covariance was used, and 97.3% when distinct covariance matrices were used.

However, the rotational invariance property of the Fourier descriptors does create difficulties in resolving numerals that are similar in shape and whose differences can be attributed to rotation and/or
reflection. Thus, Fourier descriptors do not distinguish between 2 and 5, 6 and 9, etc. Hence, it becomes necessary to assume that the orientation of the numerals is known apriori. Moreover, difficulty arises when the numerals are broken, since more than one border has to be expanded.

In order to resolve this dilemma, the classification obtained from Fourier descriptors is followed by a second stage that utilizes a set of topological features called "Border Transition Features" (BTFs). These features provide shape sensitive parameters for resolving the difficulties inherent in the first stage based on Fourier descriptors. The overall recognition accuracy obtained over the two stages using squared Euclidean distance was 95%. When Mahalanobis distance measure was applied, it yielded 97% recognition accuracy using the pooled covariance matrix and 99% accuracy using distinct covariance matrices.

Although the Fourier descriptors are shown to be sensitive to variations in shapes, they require a relatively high degree of computational effort. In order to improve the speed of recognition while maintaining high recognition accuracy and reliability, a new set of topological features (primitives) derived from a global description of the characters was proposed. These primitives are derived from the left and right profiles of the thresholded image of the numerals and can be summarized as follows:

(a) Character width
(b) Height to width ratio
(c) Locations of extrema
(d) Discontinuities in character profiles.
An attribute is then assigned to each primitive. These attributes take the form of logical predicates and play the role of semantic rules. The recognition algorithm is developed to deal with continuous as well as broken numerals and utilizes a tree classifier to determine the identity of the test numeral. Tests with 450 numeral specimens (including broken numerals) yielded an overall recognition accuracy of 99%.

Although a variety of algorithms has been developed for character recognition, these algorithms assume that the numerals are completely isolated, i.e., the characters must be drawn in boxes, and each character is assumed to be isolated from its neighbours. Such assumptions are restrictive since the handwritten numerals encountered in the real world may be isolated, broken, touching, connected, or they appear with a combination of these attributes.

In view of these difficulties an algorithm that utilizes the unique characteristics of handwritten numeral string was proposed. This technique utilizes semantic knowledge together with sequential reasoning to determine the decision boundary between the numerals in the string. The segmentation algorithm is hierarchical in nature. It tests various hypotheses ranging from the case where the numerals are completely isolated and well separated, to that where numerals may be connected, touching, broken or overlapping, or they appear in a combination of these attributes.

Once the numeral string has been segmented, the syntactic algorithm (tree classifier) was applied to recognize the individual numerals. An accuracy of 92% was obtained when the test data consisted of isolated and connected numerals.
In conclusion, the contributions stemming from this dissertation can be summarized as follows:

1) A new set of size, shift and rotation invariant Fourier shape descriptors has been shown to provide a reliable shape representation of handwritten numerals.

2) A new set of topological features termed the border transition features (BTFs) has been shown to successfully resolve the ambiguities in the recognition caused by the rotational invariance of the Fourier descriptors.

3) A new set of topological features for a fast recognition of isolated and possibly broken handwritten numerals has been shown to yield high accuracies (≥ 99%).

4) An algorithm for segmenting connected and possibly broken numeral strings has been developed and shown to be feasible and reliable. After segmentation, the individual numerals in the string are shown to be recognizable by the syntactic recognition scheme of Chapter IV.
BIBLIOGRAPHY


PUBLICATIONS BASED ON THIS RESEARCH

(a) Papers in Refereed Journals


2. M. Shridhar and A. Badredlin, "Feature evaluation and sub-class determination through functional mapping," Pattern Recognition Letters. (Accepted)


5. M. Shridhar and A. Badredlin, "Recognition of isolated and simply connected handwritten numerals," Pattern Recognition. (Accepted)

Papers in Refereed Conference Proceedings


8. M. Shridhar and A. Badreldin, "Machine recognition of handwritten numeral strings," 1985, Conf. on Intelligent Systems and Machines,
APPENDIX I

Properties of the Functional Mapping Technique

Given a finite set of parameters \( \{S_i, i = 1, 2, \ldots, N\} \) the functional mapping technique is applied to yield a function of one variable 't' as follows:

\[
F(t) = [S_1, S_2, \ldots, S_N] \begin{bmatrix}
q_1(t) \\
q_2(t) \\
\vdots \\
q_N(t)
\end{bmatrix}
\]

\[\text{(Al.1)}\]

for \( t_1 \leq t \leq t_2 \)

In the above realization \( \{q_i(t), i = 1, 2, \ldots, N\} \) are a set of mutually orthogonal functions of 't' defined over the closed interval \( [t_1, t_2] \).

Thus

\[
\int_{t_1}^{t_2} q_i(t) q_j(t) \, dt = k_i \delta_{ij}
\]

\[\text{(Al.2)}\]

where \( \delta_{ij} \) is the kronecker delta. A typical set of such functions could be defined as follows:

- \( q_1(t) = \sin t \), \quad q_2(t) = \cos t \),
- \( q_3(t) = \sin 2t \), \quad q_4(t) = \cos 2t \),
- \[ \cdots \]
- \( q_N(t) = \sin [(N+1)t/2] \) \quad \text{N odd} \quad \[\text{(Al.3)}\]
- \( q_N(t) = \cos [Nt/2] \) \quad \text{N even} \]
The range of $t$ is $(-\pi, \pi)$ and $k_i = \pi$ for all $i$.

The function $s(t)$ is the one-dimensional projection of the vector $[S_1, S_2, \ldots, S_N]^T$ in the 't' space.

The functional mapping technique preserves the following properties of the feature set $\{S_i\}$:

1) The mean and the mean-squared value of the elements of the feature set:

(a) $E[S_i] = \frac{1}{\pi} \int_{-\pi}^{\pi} E[F(t) q_i(t)] \, dt$ (Al. 4)

(b) $E[S_i^2] = \frac{1}{N\pi} \int_{-\pi}^{\pi} E[F^2(t)] \, dt$ (Al. 5)

(where $E[S_i] = 0$ and $\text{Var}[S_i] = \text{Var}[S_j]$)

2) The Euclidean distance between two feature sets $(a_i)$ and $(b_i)$.

Thus

$$|a - b|^2 = \sum_{i=1}^{N} (a_i - b_i)^2$$

$$= \frac{1}{\pi} \int_{-\pi}^{\pi} \left[ F_1(t) - F_2(t) \right]^2 \, dt$$ (Al. 6)
where

\[ F_1(t) = a^T q \]

and

\[ F_2(t) = b^T q \]

3) The transformation is linear and yields a one-dimensional projection of the feature set.
APPENDIX II

An Efficient Version of the K-MEANS Algorithm:

The main objective of the K-MEANS clustering algorithm is to find optimal partitions of the data such that the sum-of-squared error criterion is minimized. This can be described mathematically as follows:

$$\text{minimize } J \triangleq \sum_{j=1}^{K} \sum_{i=1}^{n} ||x_i - z_j||^2$$

where $K$ = number of clusters
$n$ = number of patterns
$x_i$ = $i$th pattern
$z_j$ = $j$th cluster center
$c_j$ = $j$th cluster

The ordinary K-MEANS can be described by the following steps:

Begin

Step 1: Select an initial partition of $n$ patterns into $K$ clusters.

Loop

Step 2: Calculate cluster centers $z_j$, $1 \leq j \leq K$.

Step 3: Allocate each pattern to the cluster of nearest center

Until (there is no change in cluster centers)

End
A more efficient version of the K-MEANS algorithm was proposed by Duda and Hart [33] and is given by:

**Begin**

**Step 1:** Select an initial partition of $n$ patterns into $K$ clusters, and compute $J$ and the cluster centers $Z_j$, $1 \leq j \leq K$.

**Loop**

**Step 2:** Select the next candidate pattern $\hat{x}$

Suppose that $\hat{x}$ is currently in cluster $C_i$.

**Step 3:** Compute the effect of moving pattern $\hat{x}$ from its present cluster $C_i$ to any other cluster $C_j$,

$j = 1, 2, \ldots, K, j \neq i$.

**Step 4:** Transfer pattern $\hat{x}$ to cluster $C_j$ which reduces the objective function most.

**Step 5:** Update the objective function $J$, and cluster centers $Z_i$ and $Z_k$.

**Until** (no pattern transfer is possible)

**end**
### Appendix III

**Table A3 Subclasses of the Numerals '0' through '9'**

<table>
<thead>
<tr>
<th>Numeral</th>
<th>No.</th>
<th>Subclasses of the Numeral</th>
</tr>
</thead>
<tbody>
<tr>
<td>'0'</td>
<td>1</td>
<td>$a_1 A \bar{b}<em>1 A b_4 A \bar{a}<em>1 V \bar{a}</em>{12} A a_8 A (d_1 V d_2) A \bar{a}<em>4 A \bar{a}</em>{16} A \bar{a}</em>{11} A \bar{a}_7$</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>$a_1 A b_1 A b_3 A \bar{a}<em>5 A \bar{a}</em>{10} A (\bar{a}_{13} A b_8) A f_1$</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>$a_1 A \bar{b}_1 A h A (g_2 A g_3) A e_3 A f_1$</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>$a_1 A \bar{b}_1 A h A (g_2 A g_3) A e_3 A f_1 A b_4$</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>$a_1 A \bar{b}_1 A \bar{h} A (g_2 A g_3) A e_3 A \bar{f}_1 A \bar{a}_4 A \bar{a}_8 A s_1$</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>$a_1 A \bar{b}_1 A \bar{h} A (g_2 V g_3) A \bar{f}_2 A e_3 A f_1$</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>$a_1 A \bar{b}_1 A \bar{h} A (g_2 V g_3) A \bar{f}_2 A e_3 A \bar{f}_1 A b_4$</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>$a_1 A \bar{b}_1 A \bar{h} A (g_2 V g_3) A \bar{f}_2 A e_3 A \bar{f}_1 A \bar{b}_4 A \bar{a}_8 A s_1$</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>$a_1 A b_1 A \bar{a}<em>3 A (\bar{a}</em>{13} A b_8) A e_3 A \bar{f}_1 A b_4$</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>$a_1 A b_1 A \bar{a}<em>3 A (\bar{a}</em>{13} A b_8) A e_3 A f_1$</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>$a_1 A b_1 A \bar{a}<em>3 A (\bar{a}</em>{13} A b_8) A e_3 A \bar{f}_1 A \bar{b}_4 A \bar{a}_8 A s_1$</td>
</tr>
</tbody>
</table>

| '1'     | 1   | $\bar{a}_1 A \bar{b}_1 A b_4 A (\bar{b}_7 V \bar{a}_{12}) A a_8 A (d_1 V d_2) A \bar{a}_4 A \bar{a}_{16} A \bar{a}_{11} A \bar{a}_4$ |
|         | 2   | $a_1 A \bar{b}_1 A h$ |
|         | 3   | $a_1 A \bar{b}_1 A \bar{h} A (g_2 V g_3) A a_2 A (\bar{a}_{17} V b_{11}) A \bar{a}_8$ |

<p>| '2'     | 1   | $\bar{a}_1 A \bar{b}<em>1 A b_4 A (\bar{b}<em>7 V \bar{a}</em>{12}) A a_8 A (d_1 V d_2) A \bar{a}<em>4 A \bar{a}</em>{16} A \bar{a}</em>{11} A \bar{a}_7$ |
|         | 2   | $\bar{a}_1 A \bar{b}<em>1 A b_4 A (\bar{b}<em>7 V \bar{a}</em>{12}) A a_8 A (d_1 V d_2) A \bar{a}<em>4 A \bar{a}</em>{16} A \bar{a}</em>{11} A e_3$ |
|         | 3   | $\bar{a}<em>1 A \bar{b}<em>1 A b_4 A (\bar{b}<em>7 V \bar{a}</em>{12}) A a_8 A (d_1 V d_2) A a_4 A \bar{a}</em>{13} A \bar{a}</em>{16}$ |
|         | 4   | $\bar{a}_1 A \bar{b}_1 A b_4 A (\bar{b}<em>7 V \bar{a}</em>{12}) A a_8 A (d_1 V d_2) A \bar{a}<em>4 A \bar{a}</em>{13} A f_3$ |</p>
<table>
<thead>
<tr>
<th>Numeral</th>
<th>No.</th>
<th>Subclasses of the Numeral</th>
</tr>
</thead>
<tbody>
<tr>
<td>'2'</td>
<td>5</td>
<td>( \bar{a}_1 \land \bar{b}_1 \land b_4 \land (\bar{b}<em>7 \lor \bar{a}</em>{12}) \land a_8 \land (\bar{a}_1 \land \bar{a}<em>2) \land a</em>{13} \land \bar{e}_3 )</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>( \bar{a}_1 \land \bar{b}_1 \land \bar{b}_4 \land e_4 \land b_9 \land \bar{e}_3 )</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>( \bar{a}_1 \land \bar{b}_1 \land \bar{b}_4 \land e_4 \land b_9 \land \bar{e}_3 \land b_6 )</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>( \bar{a}_1 \land \bar{b}<em>1 \land b_3 \land a_5 \land (\bar{a}</em>{15} \lor b_9) \land \bar{e}_3 )</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>( \bar{a}_1 \land \bar{b}<em>1 \land b_3 \land a_5 \land (\bar{a}</em>{15} \lor b_9) \land \bar{e}_3 \land b_6 )</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>( \bar{a}_1 \land \bar{b}_1 \land b_3 \land \bar{a}<em>5 \land a</em>{10} )</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>( \bar{a}_1 \land \bar{b}<em>1 \land b_3 \land \bar{a}<em>5 \land \bar{e}</em>{10} \land (a</em>{13} \lor b_8) \land \bar{e}_2 )</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>( \bar{a}_1 \land \bar{b}_1 \land \bar{h} \land (\bar{e}<em>2 \lor \bar{e}<em>3) \land a_2 \land (\bar{a}</em>{17} \lor b</em>{11}) \land a_8 \land \bar{e}_4 )</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>( \bar{a}_1 \land \bar{b}_1 \land \bar{a}_3 )</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>( \bar{a}<em>1 \land \bar{b}<em>1 \land \bar{a}<em>3 \land (a</em>{13} \lor b_8) \land a</em>{14} \land b</em>{10} \land \bar{e}_2 )</td>
</tr>
<tr>
<td>'3'</td>
<td>1</td>
<td>( \bar{a}_1 \land \bar{b}<em>1 \land \bar{b}<em>3 \land (\bar{a}</em>{15} \lor b_9) \land a_7 \land a</em>{13} \land \bar{e}_1 )</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>( \bar{a}_1 \land \bar{b}_1 \land \bar{b}<em>3 \land a_5 \land (\bar{a}</em>{15} \lor b_9) \land \bar{e}_3 \land \bar{b}_6 \land \bar{e}_1 )</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>( \bar{a}_1 \land \bar{b}_1 \land \bar{b}_4 \land e_4 \land a_7 )</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>( \bar{a}_1 \land \bar{b}_1 \land \bar{b}_4 \land e_4 \land b_9 \land \bar{e}_3 \land \bar{b}_6 \land \bar{e}_1 )</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>( \bar{a}_1 \land \bar{b}_1 \land \bar{b}_4 \land e_4 \land \bar{b}_9 \land \bar{e}_3 \land a_4 \land c_2 )</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>( \bar{a}_1 \land \bar{b}_1 \land \bar{b}_4 \land e_4 \land \bar{b}_9 \land \bar{e}_3 \land c_2 )</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>( \bar{a}_1 \land \bar{b}_1 \land b_4 \land (\bar{b}<em>7 \lor \bar{a}</em>{12}) \land a_8 \land (d_1 \lor d_2) \land \bar{a}<em>4 \land a</em>{16} \land \bar{e}_2 )</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>( \bar{a}_1 \land \bar{b}<em>1 \land b_4 \land (\bar{b}<em>7 \lor \bar{a}</em>{12}) \land a_8 \land (d_1 \lor d_2) \land a_4 \land a</em>{13} \land \bar{e}_3 )</td>
</tr>
<tr>
<td>Numeral</td>
<td>No.</td>
<td>Subclasses of the Numeral</td>
</tr>
<tr>
<td>---------</td>
<td>-----</td>
<td>---------------------------</td>
</tr>
<tr>
<td>'3'</td>
<td>9</td>
<td>(\bar{a}_1 \land \bar{b}_1 \land \bar{a}_4 \land (\bar{b}<em>7 \lor \bar{a}</em>{12}) \land a_8 \land (\bar{d}_1 \land \bar{d}<em>2) \land a</em>{13} \land \bar{f}_3)</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>(\bar{a}_1 \land \bar{b}_1 \land \bar{a}_4 \land (\bar{b}<em>7 \lor \bar{a}</em>{12}) \land a_8 \land (\bar{d}_1 \land \bar{d}<em>2) \land \bar{a}</em>{13} \land c_3 \land a_4 \land c)</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>(\bar{a}_1 \land \bar{b}_1 \land \bar{a}_4 \land (\bar{b}<em>7 \lor \bar{a}</em>{12}) \land a_8 \land (\bar{d}_1 \land \bar{d}<em>2) \land \bar{a}</em>{13} \land c_3 \land c_2)</td>
</tr>
<tr>
<td>'4'</td>
<td>1</td>
<td>(a_1 \land \bar{b}_1 \land \bar{h} \land (g_2 \lor g_3) \land e_3 \land \bar{f}_1 \land \bar{b}_4 \land a_9)</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>(a_1 \land \bar{b}_1 \land \bar{h} \land (g_2 \lor g_3) \land f_2 \land e_3 \land \bar{f}_1 \land \bar{b}_4 \land a_9)</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>(\bar{a}_1 \land \bar{b}<em>1 \land \bar{b}<em>3 \land (\bar{a}</em>{15} \lor b_9) \land a_7 \land a</em>{13} \land e_1 \land \bar{e}_6)</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>(\bar{a}_1 \land \bar{b}<em>1 \land \bar{b}<em>3 \land (\bar{a}</em>{15} \lor b_9) \land a_7 \land a</em>{13} \land \bar{a}_6 \land \bar{b}_5)</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>(\bar{a}_1 \land \bar{b}_1 \land \bar{b}<em>3 \land (\bar{a}</em>{15} \lor b_9) \land \bar{a}_6 \land \bar{b}_5)</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>(\bar{a}_1 \land \bar{b}_1 \land \bar{b}<em>3 \land (\bar{a}</em>{15} \lor b_9) \land \bar{a}_7 \land \bar{b}_7 \land \bar{a}_5 \land \bar{b}_5)</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>(\bar{a}_1 \land \bar{b}_1 \land \bar{b}_4 \land (\bar{b}<em>7 \lor \bar{a}</em>{12}) \land \bar{a}_8)</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>(a_1 \land \bar{b}<em>1 \land \bar{a}<em>3 \land (a</em>{13} \lor b_8) \land \bar{a}</em>{14})</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>(a_1 \land \bar{b}<em>1 \land \bar{a}<em>3 \land (a</em>{13} \lor b_8) \land a</em>{14} \land \bar{a}_{10} \land e_2)</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>(a_1 \land \bar{b}_1 \land \bar{a}<em>3 \land (a</em>{13} \lor b_8) \land e_3 \land \bar{f}_1 \land \bar{b}_4 \land a_9)</td>
</tr>
<tr>
<td>'5'</td>
<td>1</td>
<td>(\bar{a}_1 \land \bar{b}_1 \land \bar{b}<em>3 \land (\bar{a}</em>{15} \lor b_9) \land \bar{a}_7 \land \bar{b}_7 \land \bar{a}_5 \land \bar{b}_2 \land \bar{a}_6)</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>(\bar{a}_1 \land \bar{b}<em>1 \land \bar{b}<em>3 \land (\bar{a}</em>{15} \lor b_9) \land a_7 \land a</em>{13} \land e_1 \land \bar{a}_6)</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>(\bar{a}_1 \land \bar{b}<em>1 \land \bar{b}<em>3 \land (\bar{a}</em>{15} \lor b_9) \land a_7 \land \bar{a}</em>{13} \land \bar{a}_6 \land b_2)</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>(\bar{a}_1 \land \bar{b}_1 \land \bar{b}<em>3 \land (\bar{a}</em>{15} \lor b_9) \land \bar{f}_3 \land \bar{b}_6 \land a_1)</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>(\bar{a}_1 \land \bar{b}_1 \land \bar{b}_3 \land \bar{e}_4 \land \bar{a}_7)</td>
</tr>
<tr>
<td>Numeral</td>
<td>No.</td>
<td>Subclasses of the Numeral</td>
</tr>
<tr>
<td>---------</td>
<td>-----</td>
<td>---------------------------</td>
</tr>
<tr>
<td>'5'</td>
<td>6.</td>
<td>$\bar{a}_1 \Lambda \bar{b}_1 \Lambda \bar{b}_4 \Lambda \bar{e}_4 \Lambda \bar{b}_9 \Lambda \bar{g}_3 \Lambda \bar{b}_6 \Lambda \bar{c}_1$</td>
</tr>
<tr>
<td></td>
<td>7.</td>
<td>$\bar{a}_1 \Lambda \bar{b}_1 \Lambda \bar{b}_4 \Lambda \bar{e}_4 \Lambda \bar{b}_9 \Lambda \bar{c}_3 \Lambda \bar{a}_4$</td>
</tr>
<tr>
<td></td>
<td>8.</td>
<td>$\bar{a}_1 \Lambda \bar{b}_1 \Lambda \bar{b}<em>3 \Lambda (\bar{a}</em>{15} \Lambda \bar{b}_9 \Lambda \bar{a}_6 \Lambda \bar{b}_5$</td>
</tr>
<tr>
<td></td>
<td>9.</td>
<td>$\bar{a}_1 \Lambda \bar{b}_1 \Lambda \bar{b}<em>3 \Lambda (\bar{a}</em>{15} \Lambda \bar{b}_9 \Lambda \bar{a}_6 \Lambda \bar{c}_2 \Lambda \bar{e}_5 \Lambda \bar{b}_2$</td>
</tr>
<tr>
<td></td>
<td>10.</td>
<td>$\bar{a}_1 \Lambda \bar{b}_1 \Lambda \bar{b}<em>3 \Lambda (\bar{a}</em>{15} \Lambda \bar{b}_9 \Lambda \bar{a}_7 \Lambda \bar{b}_7$</td>
</tr>
<tr>
<td></td>
<td>11.</td>
<td>$\bar{a}_1 \Lambda \bar{b}_1 \Lambda \bar{b}<em>3 \Lambda (\bar{a}</em>{15} \Lambda \bar{b}_9 \Lambda \bar{a}_7 \Lambda \bar{g}_7 \Lambda \bar{a}_5 \Lambda \bar{b}_5 \Lambda \bar{b}_2$</td>
</tr>
<tr>
<td></td>
<td>12.</td>
<td>$\bar{a}_1 \Lambda \bar{b}<em>1 \Lambda \bar{b}<em>4 \Lambda (\bar{a}</em>{7} \Lambda \bar{g}</em>{12} \Lambda \bar{a}<em>8 \Lambda \bar{a}</em>{13} \Lambda \bar{c}_3 \Lambda \bar{a}_4$</td>
</tr>
<tr>
<td>'6'</td>
<td>1.</td>
<td>$\bar{a}_1 \Lambda \bar{b}_1 \Lambda \bar{h} \Lambda (\bar{g}_2 \Lambda \bar{g}_3 \Lambda \bar{e}_3 \Lambda \bar{f}_1 \Lambda \bar{g}_4 \Lambda \bar{a}_9 \Lambda \bar{g}_1$</td>
</tr>
<tr>
<td></td>
<td>2.</td>
<td>$\bar{a}_1 \Lambda \bar{b}_1 \Lambda \bar{h} \Lambda (\bar{g}_2 \Lambda \bar{g}_3 \Lambda \bar{f}_2 \Lambda \bar{e}_3 \Lambda \bar{f}_1 \Lambda \bar{g}_4 \Lambda \bar{a}_9 \Lambda \bar{g}_1$</td>
</tr>
<tr>
<td></td>
<td>3.</td>
<td>$\bar{a}_1 \Lambda \bar{b}_1 \Lambda \bar{b}<em>3 \Lambda (\bar{a}</em>{15} \Lambda \bar{b}_9 \Lambda \bar{a}<em>7 \Lambda \bar{a}</em>{13} \Lambda \bar{b}_2$</td>
</tr>
<tr>
<td></td>
<td>4.</td>
<td>$\bar{a}_1 \Lambda \bar{b}_1 \Lambda \bar{b}<em>3 \Lambda \bar{a}<em>5 \Lambda \bar{a}</em>{10} \Lambda (\bar{a}</em>{13} \Lambda \bar{b}_8) \Lambda \bar{f}_1$</td>
</tr>
<tr>
<td></td>
<td>5.</td>
<td>$\bar{a}_1 \Lambda \bar{b}_1 \Lambda \bar{b}<em>3 \Lambda (\bar{a}</em>{15} \Lambda \bar{b}_9 \Lambda \bar{a}_7 \Lambda \bar{b}_7 \Lambda \bar{a}_5 \Lambda \bar{b}_5 \Lambda \bar{b}_2$</td>
</tr>
<tr>
<td></td>
<td>6.</td>
<td>$\bar{a}_1 \Lambda \bar{b}<em>1 \Lambda \bar{a}<em>3 \Lambda (\bar{a}</em>{13} \Lambda \bar{b}<em>8 \Lambda \bar{a}</em>{14} \Lambda \bar{b}</em>{10}$</td>
</tr>
<tr>
<td></td>
<td>7.</td>
<td>$\bar{a}_1 \Lambda \bar{b}_1 \Lambda \bar{a}<em>3 \Lambda (\bar{a}</em>{13} \Lambda \bar{b}_8 \Lambda \bar{e}_3 \Lambda \bar{f}_1 \Lambda \bar{b}_4 \Lambda \bar{a}_9 \Lambda \bar{g}_1$</td>
</tr>
<tr>
<td>'7'</td>
<td>1.</td>
<td>$\bar{a}_1 \Lambda \bar{b}_1 \Lambda \bar{h} \Lambda (\bar{g}_2 \Lambda \bar{g}_3 \Lambda \bar{f}_2 \Lambda \bar{a}_2$</td>
</tr>
<tr>
<td></td>
<td>2.</td>
<td>$\bar{a}_1 \Lambda \bar{b}_1 \Lambda \bar{b}<em>3 \Lambda (\bar{a}</em>{15} \Lambda \bar{b}_9 \Lambda \bar{a}_7 \Lambda \bar{b}_7 \Lambda \bar{a}_5 \Lambda \bar{b}_2$</td>
</tr>
<tr>
<td></td>
<td>3.</td>
<td>$\bar{a}_1 \Lambda \bar{b}_1 \Lambda \bar{b}<em>3 \Lambda (\bar{a}</em>{15} \Lambda \bar{b}_9 \Lambda \bar{a}_7 \Lambda \bar{b}_7 \Lambda \bar{a}_5 \Lambda \bar{b}_2 \Lambda \bar{e}_6$</td>
</tr>
<tr>
<td></td>
<td>4.</td>
<td>$\bar{a}_1 \Lambda \bar{b}_1 \Lambda \bar{b}_3 \Lambda \bar{a}_5 \Lambda (\bar{b}<em>9 \Lambda \bar{a}</em>{15}$</td>
</tr>
<tr>
<td>Numeral</td>
<td>No.</td>
<td>Subclasses of the Numeral</td>
</tr>
<tr>
<td>---------</td>
<td>-----</td>
<td>--------------------------</td>
</tr>
<tr>
<td>'7'</td>
<td>5</td>
<td>( \bar{a}_1 \bar{a}_2 \bar{b}_4 \bar{b}_9 \bar{c}_3 \bar{a}_4 \bar{a}_9 \bar{a}_2 )</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>( \bar{a}_1 \bar{a}_2 \bar{b}_4 \bar{b}_9 \bar{c}_3 \bar{a}_2 )</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>( \bar{a}_1 \bar{a}_2 \bar{b}<em>3 \bar{a}</em>{15} \bar{b}_9 \bar{a}_6 \bar{a}_2 )</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>( \bar{a}_1 \bar{a}_2 \bar{b}<em>3 \bar{a}</em>{15} \bar{b}_9 \bar{a}_6 \bar{a}_2 \bar{e}_5 \bar{b}_2 )</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>( \bar{a}_1 \bar{a}_2 \bar{b}_4 \bar{b}<em>9 \bar{a}<em>8 \bar{a}</em>{12} \bar{a}</em>{13} \bar{a}_4 \bar{a}_6 \bar{a}_2 \bar{e}_5 \bar{b}_2 )</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>( \bar{a}_1 \bar{a}_2 \bar{b}_4 \bar{b}<em>9 \bar{a}<em>8 \bar{a}</em>{12} \bar{a}</em>{13} \bar{c}_3 \bar{a}_4 \bar{a}_6 \bar{e}_5 \bar{b}_2 )</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>( \bar{a}_1 \bar{a}_2 \bar{b}_4 \bar{b}<em>9 \bar{a}<em>8 \bar{a}</em>{12} \bar{a}</em>{13} \bar{c}_3 \bar{c}_2 )</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>( \bar{a}_1 \bar{a}_2 \bar{b}_4 \bar{b}_9 \bar{a}<em>8 \bar{a}</em>{12} \bar{a}_7 )</td>
</tr>
<tr>
<td>'8'</td>
<td>1</td>
<td>( \bar{a}_1 \bar{a}_2 \bar{a}_4 \bar{a}_6 \bar{g}_2 \bar{g}_3 \bar{e}_3 )</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>( \bar{a}_1 \bar{a}_2 \bar{a}_4 \bar{a}_6 \bar{g}_2 \bar{g}_3 \bar{e}_3 )</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>( \bar{a}_1 \bar{a}_2 \bar{a}_4 \bar{a}_6 \bar{g}_2 \bar{g}_3 \bar{e}_3 )</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>( \bar{a}_1 \bar{a}_2 \bar{a}_4 \bar{a}_6 \bar{g}_2 \bar{g}_3 \bar{e}_3 )</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>( \bar{a}_1 \bar{a}_2 \bar{a}_4 \bar{a}_6 \bar{g}_2 \bar{g}_3 \bar{e}_3 )</td>
</tr>
<tr>
<td>'9'</td>
<td>1</td>
<td>( \bar{a}_1 \bar{b}_1 \bar{b}_4 \bar{b}_9 \bar{a}<em>2 \bar{a}<em>2 \bar{a}</em>{12} \bar{a}</em>{11} \bar{c}_2 )</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>( \bar{a}_1 \bar{b}_1 \bar{b}_4 \bar{b}_9 \bar{a}<em>2 \bar{a}<em>2 \bar{a}</em>{12} \bar{a}</em>{11} \bar{f}_4 )</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>( \bar{a}_1 \bar{b}_1 \bar{b}_4 \bar{b}_9 \bar{a}<em>2 \bar{a}<em>2 \bar{a}</em>{12} \bar{a}</em>{11} \bar{f}_4 )</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>( \bar{a}_1 \bar{b}_1 \bar{b}_4 \bar{b}_9 \bar{a}<em>2 \bar{a}<em>2 \bar{a}</em>{12} \bar{a}</em>{11} \bar{f}_4 )</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>( \bar{a}_1 \bar{b}_1 \bar{b}_4 \bar{b}_9 \bar{a}<em>2 \bar{a}<em>2 \bar{a}</em>{12} \bar{a}</em>{11} \bar{f}_4 )</td>
</tr>
</tbody>
</table>
APPENDIX IV

PROFILES OF SEGMENTED NUMERAL STRINGS
Fig. A4.1 Profiles of segmented string with HT = 2.

Fig. A4.2 Profiles of segmented string with HT = 2.
Fig. A4.3 Profiles of segmented string with HT = 4.

Fig. A4.4 Profiles of segmented string with HT = 3.
Fig. A4.5 Profiles of segmented string with HT = 3.

Fig. A4.6 Profiles of segmented string with HT = 3.
Fig. A4.7 Profiles of segmented string with HT = 3.

Fig. A4.8 Profiles of segmented string with HT = 3.
Fig. A4.9 Profiles of segmented string with HT = 3.

Fig. A4.10 Profiles of segmented string with HT = 3.
Fig. A4.11 Profiles of segmented string with HT = 3.

Fig. A4.12 Profiles of segmented string with HT = 3.
Appendix V
DATA BASE

(a) **Training Set**

<p>| | | | | | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
</tbody>
</table>

<p>| | | | | | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
</tbody>
</table>

142
(b) Test Set

0 1 2 3 4 1 2 3 4 0 1 2 3 4 5 1 2 3 4 5
4 5 6 7 8 9 5 6 7 8 9 0 1 2 3 4 5
8 7 9 5 2 4 2 1 5 0 2 5 3 6 1 5 6 3 1 2
1 2 3 4 5 7 9 3 5 4 0 1 2 3 9 1 2 3 4
6 7 8 9 0 8 2 1 6 9 9 4 2 0 8 7 9 0 5 8
2 3 4 5 6 5 7 8 6 5 0 0 7 6 2 3 4 5 6
0 1 2 3 4 6 8 7 4 2 1 2 3 4 5 7 8 0 1 2
5 6 7 8 9 0 3 9 1 6 7 8 5 0 3 5 6 7 9
7 8 9 6 5 0 7 6 2 4 2 0 0 7 9 0 0 3 4
6 4 5 6 7 1 2 3 4 8 9 6 5 4 7 8 9 1 2
8 2 5 1 6 5 6 7 7 8 1 7 0 6 3 4 5 6 7
7 5 8 4 3 9 0 5 3 4 7 2 1 3 0 8 9 0 1 2
1 2 3 4 5 3 5 6 7 4 0 2 3 4 0 1 2 3 4
6 7 8 9 0 2 1 8 9 0 5 6 7 8 9 5 6 7 8 9
6 0 5 4 2 7 8 2 5 6 0 1 2 3 4 0 7 5 6 1
1 2 3 4 5 1 2 3 4 6 5 0 3 2 4 7 6 1 2 1
6 7 8 9 0 6 7 8 9 6 8 9 2 3 2 4 5 6
0 5 4 9 8 9 1 4 3 3 8 6 2 7 9 7 8 9 0 2
1 2 3 4 5 7 8 9 5 6 3 2 4 4 3 5 6 7
6 7 8 9 2 9 1 4 3 4 7 8 9 0 0 8 9 0 1 2
0 1 2 3 4 5 6 7 8 9 1 2 7 8 9 3 4 5 6 7
<table>
<thead>
<tr>
<th>Year</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>1952</td>
<td>Born on July 20, Alexandria, Egypt.</td>
</tr>
<tr>
<td>1975</td>
<td>Graduated from the University of Alexandria, Egypt, with the degree of Bachelor of Science in Electrical and Computer Engineering.</td>
</tr>
<tr>
<td>1980</td>
<td>Graduated from the University of Waterloo, Waterloo, Ontario, with the degree of M.A.Sc. in Systems Design.</td>
</tr>
<tr>
<td>1982-</td>
<td>Candidate for the degree of Doctor of Philosophy in Electrical Engineering, University of Windsor, Windsor, Ontario, Canada.</td>
</tr>
</tbody>
</table>