

NANYANG TECHNOLOGICAL UNIVERSITY



**Discriminative Power of Features Used by Forensic  
Document Examiners in the Analysis of  
Handwriting**

A THESIS

SUBMITTED TO

NANYANG TECHNOLOGICAL UNIVERSITY

BY

**Vladimir Pervouchine**

IN FULFILMENT OF THE REQUIREMENT FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

SCHOOL OF COMPUTER ENGINEERING

2006

# Acknowledgements

I would like to thank my supervisor in NTU, Prof. Graham Leedham for his invaluable guidance and support during my work on the project. Without him this work would have never become a reality. His vision, involvement, and attention was a great source of inspiration for me in the work. His insightful comments during our discussions of the project helped me to understand the project tasks in detail and focus on solving them.

I am very thankful to Prof. Alexei Sourin from SCE, NTU and Prof. Stanislav Klimenko from Department of General and Applied Physics, MIPT for their creation and support of the student exchange programme between Moscow Institute of Physics and Technology in Russia and Nanyang Technological University in Singapore. This programme gave me as well as many other students from MIPT an opportunity to study in Singapore and carry out research in Computer Engineering. I am also thankful to the dean of Department of General And Applied Physics, Prof. Kamenetz, for his support of the MIPT-NTU programme.

I would like to thank Ms. Tan Wei Kei, graduate student of Queensland University of Technology (Australia), and Mr. Jacob Arun, undergraduate student of Nanyang Technological University. Without their contribution to the project it would have been impossible to do this large amount of work.

Support of lab technicians Mr. Tan Swee Huat, Mr. Lim Hong Heng, and Mr. Lau Boon Chee from Centre for Computational Intelligence, SCE, NTU (formerly Intelligent Systems Laboratory) as well as Mrs. Ng-Lee Fui Chin, Christina, from Forensic and Security Lab was invaluable to the project. I thank them for providing me with cosy workplaces in the labs and graciously accommodating my various requests for software required during the course of the project.

I would like to thank Prof. Sargur Srihari from Center of Excellence for Document Analysis and Recognition, State University of New York at Buffalo, for providing the

samples of handwriting for this project.

I would like to thank all the lab technicians who helped me in manual segmentation of characters from handwriting images. Without help of those people I could never have done that huge amount of work which was necessary to get the data for the project.

I would like to thank School of Computer Engineering for providing space and first-class equipment to work with, which made it possible to create the necessary software. The facilities provided by Nanyang Technological University helped me to focus on the project, and the scholarship helped me to be financially secure in my research. This help is gratefully appreciated.

Vladimir Pervouchine.

# Contents

|   |           |
|---|-----------|
| <b>Acknowledgements</b>   | 1         |
| <b>Abstract</b>   | 7         |
| <b>1 Introduction</b>   | <b>9</b>  |
| 1.1 Background  | 9         |
| 1.2 Motivation  | 10        |
| 1.2.1 Need for scientific basis   | 10        |
| 1.2.2 Need for automated tools  | 12        |
| 1.3 Problem statement   | 13        |
| 1.3.1 Individuality validation  | 13        |
| 1.3.2 Validation of methods used by forensic document examiners                               | 14        |
| 1.4 Objectives of research  | 17        |
| 1.4.1 Investigation of document examiner features   | 17        |
| 1.4.2 Restoration of handwriting strokes (skeletonisation)                                    | 18        |
| 1.4.3 Development of tools for document examiners   | 19        |
| 1.5 Organisation of this thesis   | 20        |
| <b>2 Forensic document analysis: its validation and suitability for computational methods</b> | <b>22</b> |
| 2.1 Accreditation of forensic document examiners  | 22        |
| 2.2 Handwriting individuality validation  | 23        |
| 2.3 Models of handwriting   | 25        |
| 2.4 Features of handwriting and feature extraction  | 26        |
| 2.4.1 Computational features  | 28        |
| 2.4.2 Document examiner features  | 31        |
| 2.4.3 Distance measures for feature vectors   | 39        |
| 2.5 Preparation of handwriting image for feature extraction                                   | 41        |

|          |  |           |
|----------|--|-----------|
| 2.5.1    | Background separation . . . . .  | 41        |
| 2.5.2    | Image segmentation . . . . .   | 42        |
| 2.5.3    | Stroke extraction / skeletonisation . . . . .                          | 42        |
| 2.6      | Interactive tools for forensic document analysis . . . . .             | 44        |
| 2.6.1    | Overview of the existing systems . . . . .                             | 45        |
| 2.6.2    | Summary on the existing tools . . . . .                                | 48        |
| 2.7      | Writer verification / identification via pattern recognition . . . . . | 48        |
| 2.7.1    | Classifiers . . . . .  | 49        |
| 2.8      | Concept of feature usefulness . . . . .                                | 51        |
| 2.8.1    | Feature selection by filter approach . . . . .                         | 52        |
| 2.8.2    | Feature selection by wrapper approach . . . . .                        | 52        |
| 2.9      | Conclusions and identification of research issues . . . . .            | 53        |
| <b>3</b> | <b>Choice of characters and features for study</b>                     | <b>56</b> |
| 3.1      | Plan of the study . . . . .  | 56        |
| 3.2      | Handwriting sample database . . . . .                                  | 57        |
| 3.3      | Selection of characters and graphemes . . . . .                        | 58        |
| 3.4      | Selection of features . . . . .  | 60        |
| <b>4</b> | <b>Image preprocessing and extraction of character skeleton</b>        | <b>66</b> |
| 4.1      | Segmentation and binarisation . . . . .                                | 67        |
| 4.2      | Skeleton extraction . . . . .  | 68        |
| 4.2.1    | Thinning-based skeletonisation . . . . .                               | 69        |
| 4.2.2    | Vector skeletonisation . . . . .                                       | 69        |
| 4.3      | Comparison of skeletonisation methods . . . . .                        | 81        |
| <b>5</b> | <b>Extraction of features</b>  | <b>84</b> |
| 5.1      | Extraction of skeleton-independent features . . . . .                  | 84        |
| 5.2      | Detection of end points and junctions . . . . .                        | 85        |
| 5.3      | Raster skeleton tracing . . . . .                                      | 86        |
| 5.4      | Extraction of angular features . . . . .                               | 87        |
| 5.4.1    | Slant . . . . .  | 87        |
| 5.4.2    | Stroke angle . . . . .   | 88        |
| 5.4.3    | Fissure angle . . . . .  | 88        |
| 5.5      | Detection of loops and extraction of loop features . . . . .           | 89        |

|          |  |            |
|----------|--|------------|
| 5.6      | Relative height of ascender/descender . . . . .                | 90         |
| 5.6.1    | Descender of “y” . . . . .                                     | 90         |
| 5.6.2    | Ascender of “d” . . . . .                                      | 91         |
| 5.7      | Other features of “t” and “th” . . . . .                       | 92         |
| 5.7.1    | Stem and crossbar of “t” . . . . .                             | 92         |
| 5.7.2    | Top of stems of “t” and “h” . . . . .                          | 93         |
| 5.7.3    | Features of “th” extracted from vector skeleton only . . . . . | 93         |
| 5.8      | Feature extraction engine . . . . .                            | 95         |
| 5.9      | Feature extraction accuracy . . . . .                          | 96         |
| 5.9.1    | GUI tool for feature extraction examination . . . . .          | 96         |
| 5.9.2    | Results and discussion . . . . .                               | 97         |
| <b>6</b> | <b>Analysis of feature discriminative power</b>                | <b>101</b> |
| 6.1      | Methods of analysis . . . . .                                  | 101        |
| 6.1.1    | Assessment of the whole feature set . . . . .                  | 103        |
| 6.1.2    | One way analysis of variance test . . . . .                    | 103        |
| 6.1.3    | Wrapper approach: search for the best feature sets . . . . .   | 105        |
| 6.2      | Analysis of features extracted from 4 characters . . . . .     | 108        |
| 6.2.1    | Extend of separability . . . . .                               | 108        |
| 6.2.2    | Analysis of variance . . . . .                                 | 109        |
| 6.3      | Analysis of features of 3 characters and 1 grapheme . . . . .  | 111        |
| 6.3.1    | Best feature subset search . . . . .                           | 111        |
| 6.3.2    | Classification accuracy . . . . .                              | 113        |
| 6.4      | Analysis of features of grapheme ‘th’ . . . . .                | 116        |
| 6.4.1    | Comparison of the skeletonisation methods . . . . .            | 116        |
| 6.4.2    | Best feature subset search . . . . .                           | 118        |
| 6.5      | Summary of feature analysis . . . . .                          | 118        |
| <b>7</b> | <b>Interactive tools for handwriting analysis</b>              | <b>120</b> |
| 7.1      | Tools for visual comparison . . . . .                          | 120        |
| 7.1.1    | Image overlapping . . . . .                                    | 121        |
| 7.1.2    | Frame-by-frame image output . . . . .                          | 122        |
| 7.2      | Interactive skeletonisation tools . . . . .                    | 123        |

|          |   |            |
|----------|---|------------|
| <b>8</b> | <b>Conclusions and recommendations for further research</b>   | <b>126</b> |
| 8.1      | Conclusions . . . . .   | 126        |
| 8.1.1    | Extraction of document examiner micro features . . . . .      | 126        |
| 8.1.2    | Discriminative power of features . . . . .                    | 128        |
| 8.1.3    | Novel skeletonisation for handwriting characters . . . . .    | 130        |
| 8.1.4    | Interactive tools for forensic handwriting analysis . . . . . | 131        |
| 8.1.5    | Main conclusion . . . . .                                     | 131        |
| 8.2      | Research contribution . . . . .                               | 132        |
| 8.3      | Suggestions for future research . . . . .                     | 133        |
|          | <b>References</b>   | <b>134</b> |
| <b>A</b> | <b>Characters and grapheme extracted from CEDAR letter</b>    | <b>150</b> |
| A.1      | Character “d” (10 occurrences) . . . . .                      | 150        |
| A.2      | Character “y” (8 occurrences) . . . . .                       | 151        |
| A.3      | Character “f” (8 occurrences) . . . . .                       | 152        |
| A.4      | Grapheme “th” and character “t” (9 occurrences) . . . . .     | 153        |
| <b>B</b> | <b>Example of a handwritten CEDAR letter</b>                  | <b>154</b> |
| <b>C</b> | <b>List of Publications</b>                                   | <b>155</b> |

# Abstract

One of the tasks of a forensic document examiner is to analyse a questioned handwritten document in order to determine its author. The purpose of such analysis is to determine whether the document is genuine, forged, or disguised. Document analysis plays an important investigative and forensic role in many types of crime and is used as evidence of authorship or non-authorship in a court of law. The methods used by forensic experts in document analysis are intuitively reasonable and have been derived from experience. The experts look into characteristics of handwriting (features) that are consistent in a person's normal writing. These characteristics allow the experts to compare different handwritten samples and draw a conclusion about the authorship of the questioned sample. However, the methods used are based on experience rather than scientific facts. There is a lack of scientific basis to support their methods of analysis.

The aim of this thesis is to answer the question whether the methods used by forensic document examiners for handwriting analysis allow us to distinguish different writers with reasonable accuracy when the documents are written in normal (not deliberately changed) writing. In this work a number of features commonly used by forensic document examiners were automatically extracted from greyscale images of samples of characters "d", "y", "f", "t", and grapheme "th". The discriminative power of the features was assessed. The feature extraction was performed in several iterations; in each iteration new features were added, some of the previous features were removed, and extraction algorithms were improved.

Automatic feature extraction accuracy of 85% was achieved in the initial experiments and later was improved to 94% by using a novel method of handwriting stroke approximation (character skeletonisation).

A new method of skeletonisation for approximation of handwriting strokes on character images was developed. The method was designed to extract the skeleton which

is very close to human perception of the original pen tip trajectory. The need for such skeletonisation arises from algorithms of document examiner feature extraction, which are sensitive to inaccuracies in the positions of skeleton curves. The skeleton is constructed in three steps directly from the greyscale image and is represented as a set of curves, which, in turn, are represented as cubic B-splines. The new skeletonisation method enabled more accurate feature extraction as well as extraction of more document examiner features.

Analysis of the usefulness of features was conducted and three categories of features were identified: *indispensable*, *partially useful*, and *irrelevant* by searching for optimal feature sets using the wrapper method. A neural network was used as a classifier and a genetic algorithm was used to search for optimal feature sets. It was shown that at least some of the structural micro features similar to those used in forensic document analysis do possess discriminative power and thus their use in handwriting analysis to detect authorship is justified. It was found that the grapheme possessed significantly higher discriminating power than any of the single characters studied, which supports the opinion that a character form is affected by its adjacent characters.

Some interactive tools were developed as a by-product of this study to facilitate visual comparison of handwriting samples and aid in development of the skeletonisation method.

Study of document examiner features in forged and disguised handwriting is a necessary next step to provide a scientific support to forensic analysis of handwriting, and is suggested as a possible direction for further research.

# Chapter 1

## Introduction

### 1.1 Background

The use of handwriting as a means of identifying an individual has a long history. For many years individuals have been asked to sign documents so that others can ensure the document's authenticity. Crimes in which alteration or falsification of written documents is involved are dealt with by a branch of forensic science known as *forensic document examination*. Forensic analysis of documents plays an important investigative role in many types of crime, particularly those involving handwritten documents, and is used as evidence of authorship or non-authorship of documents in a court of law.

It is assumed that handwriting is a personal biometric and hence is unique to an individual. If that is true, handwriting can be used to identify an individual. When doubts of authenticity arise, forensic document examiners are asked to conduct an analysis of the questioned document. The analysis is usually seeking to determine whether the document under question is *genuine* (that is, written by the person who claims to have written it), *forged* (written by someone trying to present the document as being written by someone else), or *disguised* (written by a person in a different handwriting style to their own in order to deny writing the document in the future).

Figure 1.1 shows four classes of problems involving forensic document examination of handwritten documents: signature verification, writer identification, forgery identification, and disguised writing identification.

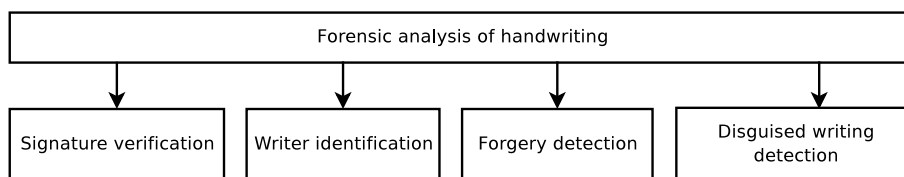


Figure 1.1: Classification of forensic document analysis of handwriting.

Signature verification stands alone from the other document examination problems. It has been the subject of extensive research as it is widely used for authentication (cheques, credit card transactions, etc.) (Plamondon & Lorette, 1989; Ammar et al., 1989). The aim of signature verification is to compare the questioned signature to a set of known signatures. The differences from the other handwriting identification and verification problems arise from the fact that certain factors need greater emphasis in signature verification than in other handwriting verification tasks. There often are some elements which are consistent in a person's handwriting but are not common in his/her signature. Also the identifying attributes of a signature may not receive the same special consideration as in the identification of normal handwriting (Hilton, 1993). Signature verification is not considered further in the current work, however, the techniques described could be applied to signature verification.

## 1.2 Motivation

### 1.2.1 Need for scientific basis

There is an important assumption, on which all forensic document examination is based. The assumption is that the handwriting of a person is individualistic and is as unique to the individual as a person's fingerprints. This individuality hypothesis relies on the fact that the process of handwriting is an unconscious act learnt over time. That is, for each person, some pen movements are invariable and are not easily changed or disguised when an attempt at forgery or disguise is made.

Forensic document analysis of handwriting is the examination of the design, shape, and structures of handwriting. Professional document examiners seek characteristics of handwriting that are consistent in a person's normal writing. These characteristics are called *features* (Lindblom, 1999; Harrison, 1981; Hilton, 1993). The individuality hypothesis basically states that those features are consistent in a person's handwriting and many of them are hard to change, hence they can be used to determine the authorship of questioned documents. The document examiners methods are intuitively reasonable and have been derived from experience.

However, there is lack of a sound scientific basis to support the handwriting individuality hypothesis as well as the possibility of using handwriting features to distinguish writers. While other current branches of forensic science, such as DNA analysis and the analysis of materials found at the scene of a crime are supported by well-established chemical and biological scientific knowledge which has been demonstrated, explained,

and proved by experimental study, the forensic techniques used to study handwriting have far less scientific support. The techniques employed in handwriting analysis are, indeed, not scientifically proven. The credibility of the document examiner as a credible expert witness, rather than through the scientific basis of the techniques, has been a key basis for the examiners believability and acceptability as an expert in a court of law.

In recent cases the scientific acceptability of forensic document examination that deals with handwriting analysis has been successfully challenged. Several landmark cases have appeared in which the use of expert forensic document examination testimony in the courts, as evidence, has been questioned on the basis that the conclusions drawn from forensic document analysis does not have a scientific basis which has been statistically validated. The most well-known cases are *the Daubert Hearing (Daubert et al. v. Merrell Dow Pharmaceuticals, 1993)* and *United States v. Starzecpyzel*. After the Daubert case the US Supreme Court came up with the following list of inquiries to be used by judges in order to determine the reliability of scientific evidence.

1. Has the scientific theory or technique been empirically tested?
2. Has the scientific theory or technique been subjected to peer review and publications?
3. What is the known or potential rate of error?
4. Do standards for controlling the use of the scientific technique exist and are they maintained?
5. Is there general acceptance of the technique by the scientific community?

There have been other similar court cases (Srihari, Cha, Arora, & Lee, 2001a). Thus there is a need to determine scientific validity of individuality in handwriting as well as methods used by forensic document examiners in their analysis of handwriting. It is necessary to prove or disprove that handwriting is individual by means of statistical methods. It is necessary to determine whether the features which document examiners rely upon indeed bear sufficient individual information to successfully discriminate writers even when attempts at forgery or disguise have been made. It is also necessary to obtain estimates for the probability of error in the decision about the authorship of a questioned document. Numerous factors that are believed to influence handwriting such as age, handedness, gender, physical condition, presence of chemicals, emotional condition etc. must be taken into account in order to determine whether they can change handwriting features and to what extent.

These tasks can be approached through a development of a computational theory of handwriting. Because the methods based on such a theory have the advantage of repeatability, the approach can be used to determine whether handwriting is unique and to what extent. It can also help to reveal whether it is the methods used by forensic document examiners that allow us to determine the authorship of a document and not something else (e.g. pure experience). To establish the scientific basis for the document examiner methods it is necessary but not enough to show that the experts perform better than lay people in the task of authorship detection (as attempted by Kam et al. (1994) and by Sita et al. (2001)). It is necessary to explain why this happens.

### 1.2.2 Need for automated tools

Forensic document examiners use various tools to establish the authorship of a document. Analysis of handwriting is a part of the overall analysis of handwritten documents. The other parts include chemical analysis of paper and ink, ultra-violet and infra-red photography etc. From the numerous tasks performed by forensic document examiners only a few are accomplished automatically or semi-automatically, while most of the steps in document examination are performed manually. Many aspects of this work are laborious, routine, and time consuming. This applies, for example, to the procedure of preparation of comparison charts that are used in a court of law as evidence. Preparation of such a chart usually takes several days. A computer system can ease many steps of document examiner work. Various measurements can be performed on a document image rather than on the hard copy of a document. Operations like cutting and arranging of document segments are also less tedious when working with images.

The methods of image processing as well as automatic or semi-automatic feature extraction developed in a computational theory of handwriting, can be employed to develop a set of tools for handwriting analysis. A computer system can aid feature measurement and hence enable different ways of handwriting comparison, provided robust algorithms for feature extraction have been developed as well as methods for adequate numerical representations for the feature values. Due to the laborious nature of some aspects of the work of the document examiners, such tools will certainly enhance the productivity of forensic document examiners (Leedham & Sagar, 1994).

## 1.3 Problem statement

### 1.3.1 Individuality validation

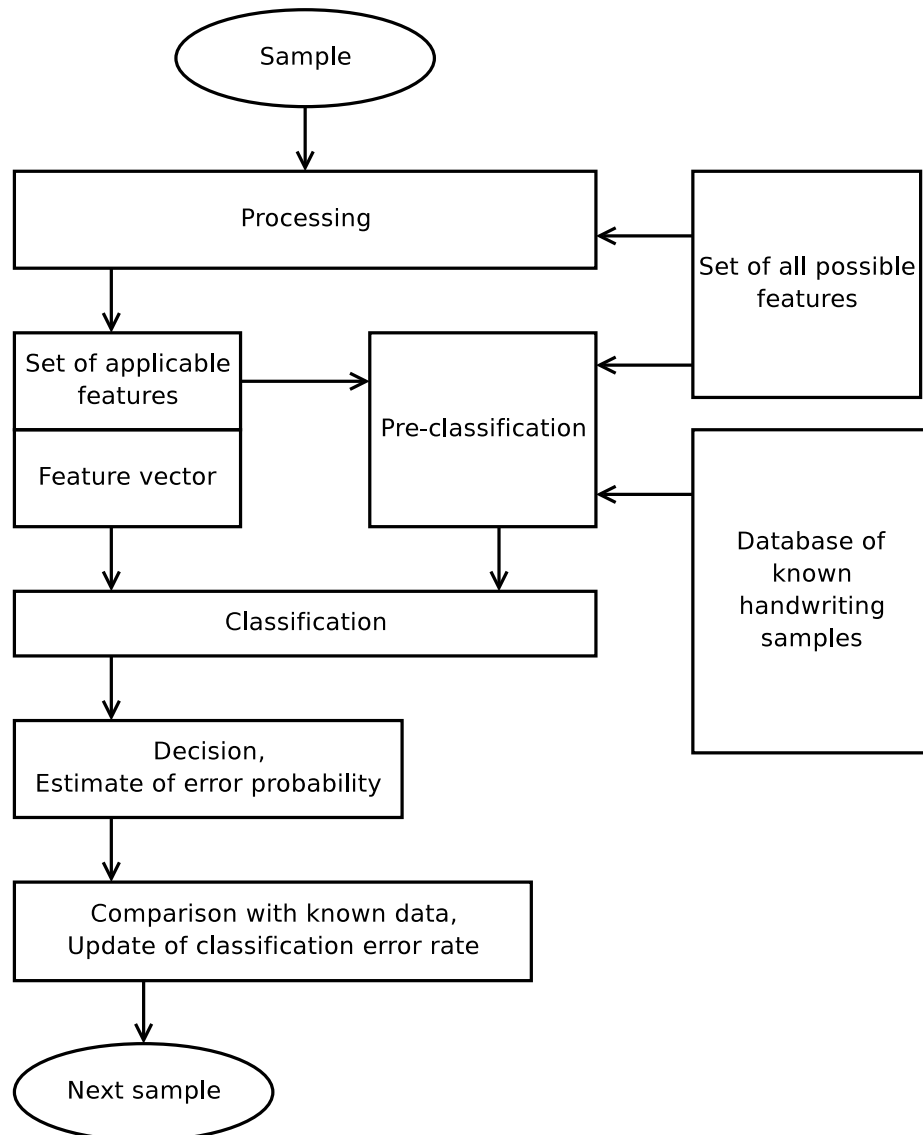


Figure 1.2: General layout of a system for handwriting individuality hypothesis validation.

Figure 1.2 shows the general layout of a system that can be used to test the hypothesis about handwriting individuality. Suppose that the aim is to test the hypothesis for some population which is a subset of the world population. This can be a population of a country, or a population of people speaking the same language. The system database of handwritten samples should be representative of the population in a statistical sense, i.e. if there is a possibility that handedness can influence a person's handwriting then the database should contain the same percentage of samples written by right-handed people as exists in the population. In practice it seems to be impossible to take all the potential factors into consideration. Thus it is necessary to

make some reasonable assumptions about the factors.

A questioned document can be either a sample from the existing database or a new sample of a writer whose samples are in the database. As some of the features that are present in one handwriting can be absent in another one, it is possible to perform a pre-classification using a list of applicable features. The classification stage makes use of the extracted feature vector to compare it with the feature vectors of the other writers in the database.

The decision about authorship made by the classifier is compared to the known data and the error rate value is updated. Provided the database is truly representative of the population, the features are extracted correctly and expressed adequately, and the error rate converges to some value, it is possible to obtain an estimate of the probability of wrong decision when such decision is made by professional document examiners using the same set of features.

In order to make the system work it is necessary to implement the document processing block, which has a feature extractor. It is assumed that a document is fed into the system as a grayscale image and the text content of the document is available (has manually been entered) and can be used by the system. These assumption are reasonable for the area of forensic document examination since a document is usually written in one colour and does not have a lot of text to be entered so the typing part is not tedious compared to the rest of the work the document examiners carry out.

Figure 1.3 shows the main steps in the document processing. According to the classification of features of handwriting in relation to the scale of extraction all features can be divided into two categories: macro features and micro features (Srihari, Cha, & Lee, 2001). The former ones are those extracted from the whole document image, from lines, and from words up to the character level. Micro features are those extracted from short consistent character combinations (graphemes), characters, their parts as well as between-character parts (ligatures).

### **1.3.2 Validation of methods used by forensic document examiners**

Two types of features that can be extracted from handwritten sample images are called document examiner features and computational features. Computational features are the features that can be automatically measured by computer algorithms but are not necessarily perceivable by humans. Normally such features are extracted by applying various filters to the document image or its parts. The advantage of

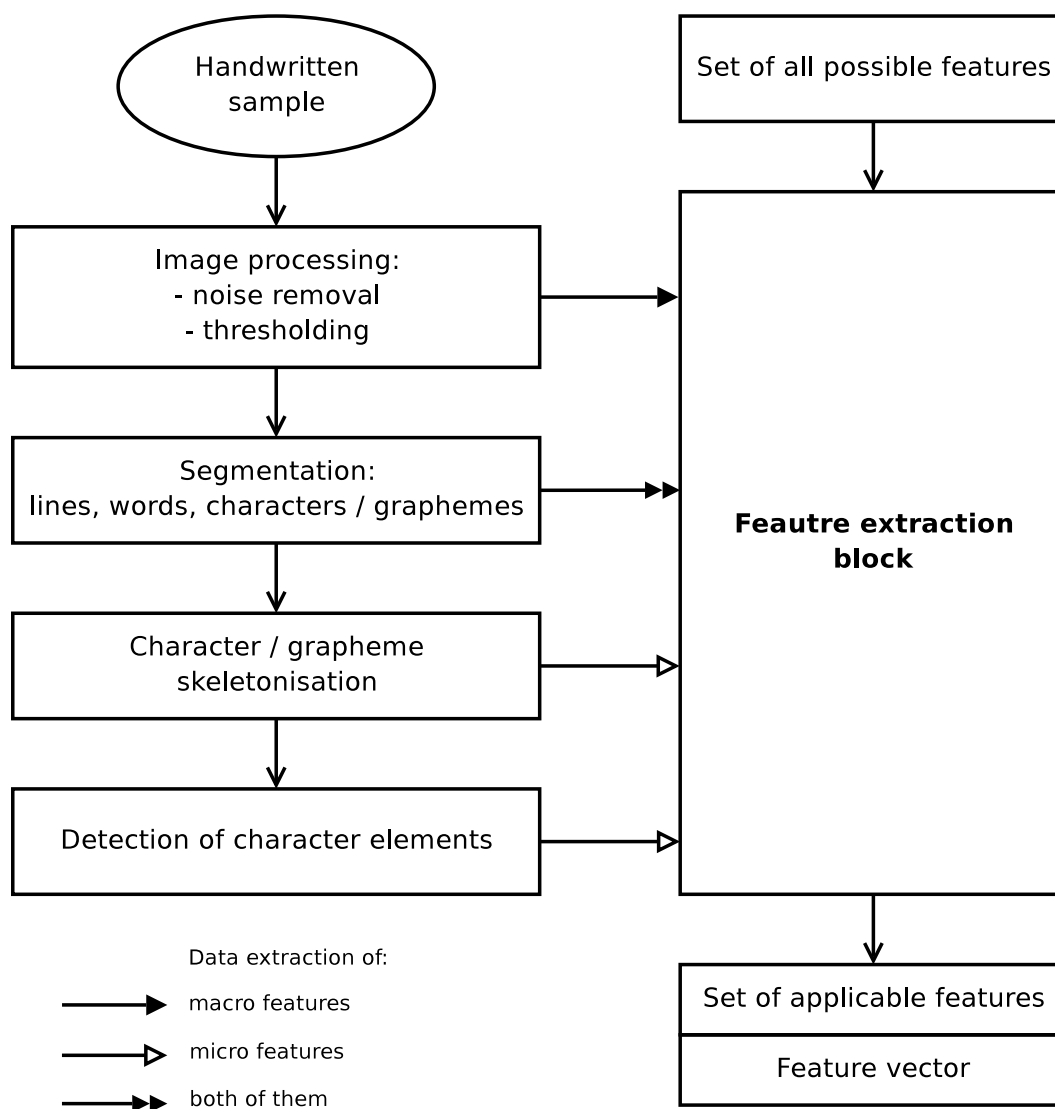


Figure 1.3: Extracting a feature vector from a document image.

using the computational features is that they are strictly defined and can usually be measured regardless of the content of a document. Extensive study of such features, their consistency, and discriminative power as well as their application to writer classification is being performed in the Center of Excellence for Document Analysis and Recognition (CEDAR) in the State University of New-York at Buffalo.

In establishing the scientific basis for forensic analysis of handwriting the first issue to solve is the questionable individuality of handwriting. Computational features are suitable enough to validate the individuality hypothesis (Srihari et al., 2002). However, many of the computational features are purely computational: they do not represent any features forensic document examiners use in their methods.

Document examiner features are those commonly used by forensic document examiners to determine the authorship of documents under question. Huber and Headrick

(1999) compiled those features in a set of *21 discriminating elements of handwriting*<sup>1</sup>:

1. Arrangement of handwriting.
2. Class of allograph / alphabet.
3. *Connections.*
4. *Design of allographs and their construction.*
5. *Dimensions (horizontal and vertical).*
6. *Slant and slope.*
7. Spacings — inter and intra word.
8. Abbreviations.
9. Baseline alignment.
10. *Initial and terminal strokes.*
11. Punctuation.
12. *Embellishments.*
13. Legibility or writing quality.
14. Line continuity.
15. *Line quality.*
16. Pen control.
17. Writing movement.
18. *Natural variations or consistency.*
19. Persistency.
20. *Lateral expansion.*
21. Word proportions.

If handwriting is indeed individual, then to establish the scientific basis for forensic document analysis it is necessary to determine whether it is the methods that forensic document examiners use that allow them to distinguish writers. Hence, it is important to determine whether writers can be distinguished using document examiner features of handwriting. Here arises a problem, however: many of the features in the list above are defined quite ambiguously and can be very subjective in terms of their validity and use. A possible solution is to formalise some of those features from the list, that is, to map them into other features which in turn are strictly defined, can be expressed in a numerical manner, and measured by computer algorithms.

Most likely it is not possible to formalise all features of handwriting used by forensic document examiners. Hence reasonable assumptions are that an expert can

---

<sup>1</sup>Elements that are studied in the current project are *italicised*

(i) effectively utilise more features of handwriting than a computer system and (ii) determine which features should and which should not be used in a particular case (i.e. having looked at handwritten samples an expert is able to pick only the important features of the handwriting under examination). Lack of important features can degrade the performance of a pattern classification machine; presence of unimportant features can also degrade its performance. As a consequence, it is assumed that an expert can distinguish writers or determine the authorship of questioned documents with higher accuracy than a computer system when only document examiner features are used.

The problem of validation of individuality of handwriting has been extensively studied during recent years (Srihari et al., 2002; B. Zhang et al., 2003; Srihari, Tomai, et al., 2003). Issues of techniques used in forensic analysis of handwriting, detection and measurement of features, comparison based on pictorial similarities between samples, etc. has been discussed (Found & Rogers, 1995, 1998). It has also been demonstrated that professional document examiners do their job better than lay people (Kam et al., 1994, 1997, 1998, 2001). Chapter 2 discusses these studies in detail. The work reported in this thesis is focused on the problem of validation of the methods used by forensic document examiners which, as is also shown in Chapter 2, has received little attention until now.

With this focus, three issues are addressed in this thesis.

1. Investigation of the discriminative power of features used by forensic document examiners.
2. Restoration of handwriting strokes from raster images.
3. Development of visual tools to assist document examination.

## **1.4 Objectives of research**

### **1.4.1 Investigation of document examiner features**

The first objective of the current work is to formalise some of the document examiner features, particularly micro features, i.e. to develop a way of measuring these features and expressing them in mathematical terms so that they can be used with the pattern classification techniques. According to the definitions of computational and document examiner features given in (Srihari, Cha, & Lee, 2001) the problem of document examiner feature formalisation can be seen as a problem of mapping the document examiner features into computational features. However, in this work the features

obtained via such mapping (all the features under investigation) are still referred to as document examiner features in order to distinguish them from purely computational features such as *GSC features* (Govindaraju et al., 1999).

An estimate of the extraction accuracy is to be made here by testing for correspondence of the results of feature extraction to the definitions of the features as formalised. Reasons for the feature extraction failures and inaccurate measurements are to be revealed in order to improve the feature extraction algorithms. Usefulness of the features is to be assessed and taken into account to determine whether the feature formalisation is appropriate or not, and how it can be changed. If a feature happens to be not useful according to some measure of usefulness, it may be an indicator of an inadequate formalisation of the corresponding document examiner features and further research into the issue of the appropriate formalisation may be needed. Thus, the following issues belong to the first objective:

1. Mapping of document examiner features into other features that are unambiguously defined (formalisation).
2. Design of the procedures to measure the feature values.
3. Design the feature extraction algorithms.
4. Understanding of the causes of extraction failures and inaccuracies and elimination of them.
5. Investigation of consistency of the features and their usefulness for writer discrimination.
6. Estimation of writer discrimination error rate.

The features from the list of the 21 discriminating elements of handwriting, which were fully or partially formalised and used in extraction are *italicised* in the list (Section 1.3.2).

### **1.4.2 Restoration of handwriting strokes (skeletonisation)**

The second objective of the current work is to develop a method to partially restore handwritten strokes from a scanned image of handwriting. Document examiner features are visible features such as character sizes, slants, stroke angles, etc. These features are mostly extracted from sample skeletons. Precision in measurement of values of most of such features strongly depends on the quality of the skeletons of handwritten elements used for feature extraction.

There are a number of techniques to obtain skeletons of handwritten samples. Applicability of a particular technique depends on the problem in hand. Thinning methods (Lam et al., 1992), methods based on thinning with correction of junction points (Amin & Singh, 1997), and many others have been proposed for character skeletonisation (see Section 2.5.3). However these methods have been developed to be used as a pre-processing stage for handwriting recognition algorithms. In handwriting recognition the features are aimed at distinguishing different characters or their combinations and thus need to represent dissimilarities between different alphabet characters and have similar values for the same characters written by different writers. In writer identification the features are aimed at distinguishing between writers and thus need to emphasise the differences in shapes of the same characters written by different writers. The methods proposed give good results for handwriting recognition purposes but they were not designed with the aim to preserve individual traits the writer endows on a character. Many methods produce very approximate skeletons (e.g. principal curves (Kégl & Krzyżak, 2002)). Displacement of end and junction points is also common. That is why for author identification these methods hardly give better results than simple and fast, but artifacts-prone thinning techniques.

Thus there exists a need for a skeletonisation method suitable for authorship detection based on document examiner features. In this work a new method that preserves some of the individual features of handwritten samples by modelling the middle line of strokes with high accuracy is presented. The method is also capable of detection of retraced strokes as well as restoration of hidden loops. The method is content-dependent, which is suitable for forensic document analysis since the text content of questioned documents is available and does not need to be recognised. Because of the dependence on the context the method is most likely not suitable for the area of handwriting recognition.

### **1.4.3 Development of tools for document examiners**

The third issue addressed in this thesis is the development of tools that can be used by forensic document examiners in their work. These tools can help to process images of handwritten documents, enabling an examiner to perform some manipulations like slant or slope changing as well as allow them to measure some feature values and make quantitative comparison of several document images along with a visual (qualitative) comparison.

A set of tools for some manipulations with handwriting images have previously

been developed (Leedham, 1999). In the current work the attention is focused mainly on investigation of document examiner features and restoration of handwriting strokes as an important preprocessing stage. Without solving these two problems development of tools for forensic analysis of handwriting is not too useful. A set of tools should incorporate stroke restoration, feature extraction, and quantitative comparison of handwriting in addition to image manipulation tools since the image manipulation and measurement tools alone are already implemented in a number of software products (e.g. *Adobe Photoshop*, *GIMP*). The tools developed during this research are a by-product of the main work rather than a separate objective.

The tools developed during this research include a visualisation tool that allows the examiner to perform qualitative comparison of many samples of handwriting by overlapping them and performing a “slide show” in user-defined positions as well as smooth scaling. The software has been used to better understand which parts of characters are consistent in a person’s handwriting and how to extract features from them. The visualisation software can be the basis for quantitative comparison tools which can be incorporated in it in the future once the appropriate feature formalisation and extraction techniques are developed. Another tool is the manual skeletonisation tool which allows the user to define the initial skeletal branches of a character. The set of branches is then fed into the automatic algorithm which combines them into strokes. This tool was used to develop and test the stroke formation and loop restoration parts of the skeletonisation algorithm.

## 1.5 Organisation of this thesis

The chapters of this thesis are organised as follows. Chapter 2 is a detailed review of the related research that has been conducted during recent years. In Chapters 3,4,5, and 6 the core of this research, the work conducted on document examiner micro features is described.

Chapter 3 describes the experimental setup and selection of characters, graphemes, and features. Chapter 4 demonstrated the importance of the appropriate skeletonisation algorithm, describes the novel skeletonisation method, and compares the results obtained by automatic feature extraction with those presented in Chapter 5. Chapter 5 contains formalisation of the selected features, methods of measuring of the feature values, description of the automatic extraction algorithms and the results of the automatic feature extraction when thinning-based skeletonisation of handwriting

was used. In Chapter 6 evaluation of the features is described: their discriminative power, usefulness, and accuracy of numerical representation.

Chapter 7 describes the tools for visual comparison of character samples that have been developed. It also describes the manual part of the skeletonisation tool which can be used instead of the automatic one when the latter produces an error during the initial branch detection stage.

Chapter 8 summarises the work that has been done during the years of my research in NTU, draws the conclusions and suggests possible directions for the future research.

Everywhere in this thesis handwriting based on the English alphabet is assumed unless specifically mentioned otherwise.

## Chapter 2

# Forensic document analysis: its validation and suitability for computational methods

In this chapter the research that has been conducted in the area of forensic analysis of handwriting as well as in the related areas such as image processing, feature extraction, feature evaluation, and writer discrimination is presented. It is shown that while some areas, such as validation of handwriting individuality, have received extensive research during the recent years, other areas like extraction and study of document examiner features still require investigation.

### 2.1 Accreditation of forensic document examiners

As was mentioned in Chapter 1, there are two issues to be studied in order to establish a scientific basis for forensic document examination: it is necessary to determine whether handwriting is unique and whether the methods used by forensic document examiners allow writers to be distinguished with sufficient accuracy.

Before investigating the second issue in detail it is reasonable to first determine whether the professional document examiners perform better than ordinary people. Research of this issue has been conducted by Kam *et al.* A number of tests have been performed under various conditions to demonstrate that the accuracy of writer identification by forensic document examiners is significantly higher than that of lay people (Kam *et al.*, 1994, 1997, 1998). Later it was shown that professional document examiners are better at signature verification too (Found, Sita, & Rogers, 1999; Kam *et al.*, 2001; Sita *et al.*, 2001). Similar study that included both handwriting samples and signatures was performed among the experts from the New Zealand Police Document Examination Section (Found *et al.*, 2001b, 2001a).

A study of forensic document examiners' performance in detection of forgery and disguise was conducted (Found & Rogers, 2005) and the difference in error rates of detection of different types of forgery and disguise was demonstrated, the highest error rate being 16.7% for simulated forgery. Although the rejection rates (inconclusive opinions) were shown to be high for disguised writing (41.28%) and simulated forgery (43.8%), the error rates were reasonably low.

Establishing these facts means that techniques used by forensic document examiner may indeed allow them to identify authorship with observable accuracy. Hence it is necessary to verify the claim that it is the techniques and not something else (e.g. pure experience from having seen a lot of samples) that mainly contribute to the high writer identification accuracy demonstrated by the experts. In other words, it is necessary to address the question raised in the Daubert case and determine whether the techniques employed by forensic document examiners have a sound basis.

## **2.2 Handwriting individuality validation**

The problem of establishing the individuality of handwriting has received extensive study during recent years in the Center of Excellence for Document Analysis and Recognition (CEDAR) in the State University of New York at Buffalo. Although the study was limited to English language only, the methods used are applicable to other languages including those based on non-Latin alphabets. The individuality of handwriting is formally defined as "given two well-selected samples of handwriting, we can tell whether they were written by the same person or by two different people with a high degree of confidence." (Srihari, Cha, Arora, & Lee, 2001a).

The samples used in their study were drawn from the US population and several factors that are thought to influence the handwriting such as handedness, age, gender, and ethnic group were taken into account so that the set of samples was representative of the population. There were three samples of a handwritten letter collected from each individual. The text of the letter was designed so that it had all characters in the beginning, middle, and ending positions as well as many two-character graphemes of interest. The detailed description of the sample set is given in (Srihari, Cha, & Lee, 2001; Srihari et al., 2002).

The progress of the study, from the stage of feature extraction to the estimates of writer verification accuracy has been published in several papers (Cha & Srihari, 2000a; Srihari, Cha, Arora, & Lee, 2001b; Srihari, Cha, & Lee, 2001; Lee et al.,

2002; Srihari, 2003). From the many-class problem of identifying one writer by the handwritten sample from a database of 500 writers the problem has been transformed into a two-class problem: two handwritten samples are fed into the feature extractor and classifier and the expected answer is whether the two samples belong to the same or to different writers.

On the sample size of 1500 (500 writers, 3 samples of a handwritten letter given by each writer) the system has been demonstrated to perform with accuracy above 95%. It has been concluded that handwriting indeed possesses individual features and can be used as a means to identify an individual with a high degree of certainty. On the other hand, several issues still remain open. As all samples were taken within relatively short period of time, no study could be conducted to determine how a person's handwriting changes with his/her age and whether it is still possible to identify a writer with high accuracy if his/her handwritten samples were taken at different ages, which is not an uncommon case in the practice of forensic document examiners. Also the influence of factors stated above (gender, handedness, etc.) need to be considered. There is currently no scientific proof that any of those factors influence a person's handwriting and hence make it possible to determine particulars of the person. For example, different background, particularly educational systems in different countries, will influence a person's handwriting, as shown in Figure 2.1.

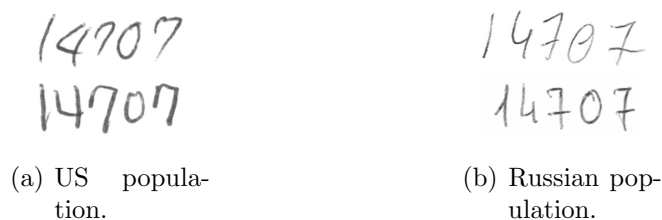


Figure 2.1: Presence/absence of crossbar in '7' is due to different school systems.

Further research has been carried out to study the discriminatory power of certain handwriting elements, particularly words (Tomai et al., 2004), characters (B. Zhang et al., 2003), and digits (Srihari, Tomai, et al., 2003). It has been shown that certain characters, especially those with ascenders and descenders<sup>1</sup> as well as capital letters bear more individual information than others which agrees with the statements of forensic document examiners (Huber & Headrick, 1999; Harrison, 1981).

In the study conducted in CEDAR pattern recognition techniques were employed

---

<sup>1</sup>Ascenders are the elements of characters extending above the upper level of character main part. Descenders are those elements extending beyond the baseline of writing. Examples of characters with ascenders: 'b', 'd', 't'; with descenders: 'y', 'g', 'q'.

to validate the handwriting individuality hypothesis. A feature vector was extracted from each sample and the feature vectors were fed into a classifier. The features extracted from handwriting were mostly purely computational features (Lee et al., 2002). Document examiner features used in the study were mostly macro features extracted from words and appeared to not contribute significantly to the writer identification accuracy (Srihari et al., 2002) (although shift to computational *GSC* features for words resulted in significant improvement of classification accuracy (B. Zhang & Srihari, 2003a)).

Thus it has been established that handwriting can be used to identify a person with high accuracy. The next step is to demonstrate that the techniques used by document examiners allow us to identify authorship with high accuracy. In order to do this it is necessary to show that features used by the document examiners are sufficient to distinguish writers. This leads to a problem of extraction of potentially useful document examiner features from handwriting elements. Consideration of computational models of handwriting can help to determine what features are important.

## 2.3 Models of handwriting

Models of handwriting represent the process of handwriting in a mathematical way. They are aimed at generation of human-like characters, which, in turn, can help to better understand the process of handwriting and result in improvements in handwriting recognition as well as writer identification and signature verification. Natural (fast) handwriting is generally accepted to be a ballistic process, when the motion of a pen is controlled by impulses without continuous visual feedback (Plamondon & Maarse, 1989). It is a result of the motor program learned over the years.

Several studies reveal that the best reconstruction of handwriting is achieved by velocity-controlled models (Schomaker et al., 1989; Feng et al., 2002), when the input to the system that generates handwriting is given in the form of some signal that models momentum of a pen tip. Second-order velocity models give good results while higher order models do not result in significant improvement of the generated handwriting over the second-order ones (Plamondon & Maarse, 1989). Often the Cartesian coordinate system is used in which the  $x$ -axis is directed along the baseline of writing. The momentum that drives the generation of handwriting consists of its  $x$ - and  $y$  components. Plamondon (1989) proposed a second-order velocity-controlled model based on differential geometry of handwriting curves. The whole process is repre-

sented using intrinsic parametrisation of curves rather than an external coordinate system. Such an approach allowed better understanding of how frequently occurring strokes are generated, such as *curvilinear strokes* and *angular strokes* in terms of the input signal.

Individual features of handwriting are often viewed in handwriting models as small distortions added to the driving signals (Kondo, 1989). Such distortions can be modelled for example by Gaussian noise. In the problem of extracting individual features of handwriting it is necessary to measure those distortions (in terms of computational models). It is reasonable to try to extract individual features from the differential geometry of certain strokes that represent shapes of those strokes in handwritten characters such as curvatures, slants, etc. These are the micro features of handwriting and they correspond to some of the features used by forensic document examiners (Huber & Headrick, 1999) (see section 1.3.2).

## 2.4 Features of handwriting and feature extraction

Both forensic document examiners and computer scientists extract features from handwritten samples in order to make the comparison of the samples possible. Features of handwriting can be viewed as elements of handwriting that can bear information that helps to identify the writer. The hypothesis about the individuality of handwriting means that each person's handwriting contains some elements from which a unique set of values can be formed. For example, the slant (measured in radians) of one person's handwriting can have a particular distribution of values. The slant of another person can have a different distribution of values. Thus, when the slant feature is extracted from questioned document samples, the decision about the authorship can be made provided the distribution of slant values is known for both writers. Also the estimate of error can be made. In real cases taking into account the slant feature only may lead to intolerably high decision error. Apart from the fact that two writers may have either a similar or overlapping (Figure 2.2) distribution of slant feature values, in real cases those distributions themselves are not known precisely and rather are estimated from several samples provided by each writer. Thus, the first reason why it is necessary to extract a number of features is the possible insufficient discriminative power of a small feature set.

While the first reason is straightforward and arises in every pattern classification problem, the second reason for extracting as many features as possible is specific to

handwriting as well as to some other areas. Many writers tend to write some character in different shapes as shown in Figure 2.3. That is why it is nearly impossible for a professional document examiner to give a complete list of features that are used by him/her for document analysis. Usually it is possible to say that this writer has one set of handwriting features while another writer has another set of features, and these two feature sets generally do not coincide.

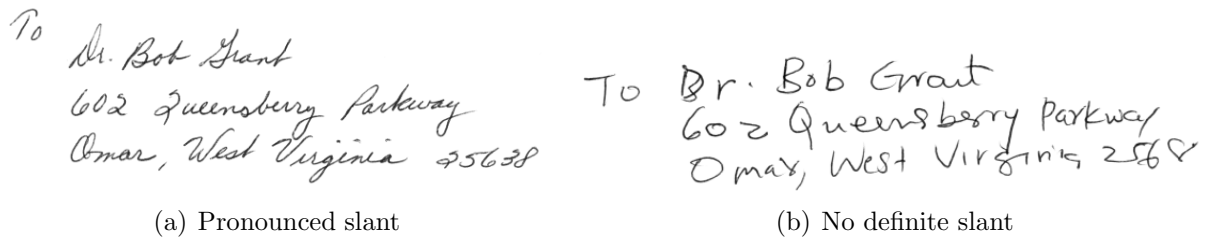


Figure 2.2: Overlapping slant value distributions.



Figure 2.3: Different ways of writing — different sets of features.

Different categories of handwriting features, such as micro and macro features, and computational and document examiner features have been discussed above. The following definitions summarise the discussion.

**Definition 2.1** *Macro features of handwriting are those features extracted from a document sample as a whole, from lines and words up to the level of characters.*

**Definition 2.2** *Micro features of handwriting are those features extracted from characters and character parts as well as from ligatures connecting the characters.*

The following definitions of computational and document examiner features differ from those given in (Srihari & Cha, 2001).

**Definition 2.3** *Document examiner features are those that 1) are generally used by forensic document examiners regardless of whether such features are defined strictly and can be measured and expressed as a value/set of values or not; 2) measurable features that correspond to those used by document examiners.*

**Definition 2.4** *Computational features are those that are strictly defined and can be measured and expressed as a value/set of values.*

Thus, some document examiner features are also computational features, but computational features also include so-called *purely computational features* which are not document examiner features. They do not correspond to anything a person can see when looking at a handwriting sample.

Formalisation of document examiner features results in features that are both computational and document examiner features. For example, the slant of letter “f” is a document examiner feature. If the slant of letter “f” is defined, for example, as the angle between a vertical line and the line best fitted to the set of points of the stem of the letter in the least square error sense, the feature becomes both a document examiner and computational feature since it is possible now to measure slant of any sample of letter “f” and the feature extraction is not subjective anymore.

In the current research the term *computational feature* is used to refer to either purely computational features or to the measurable document examiner features. When it is necessary to stress that a feature is not a document examiner feature, the term *purely computational feature* is used. Unless otherwise stated the term *document examiner feature* refers to the measurable document examiner features which are the focus of the current work.

### 2.4.1 Computational features

Use of purely computational features is attractive to the design of a system of automatic writer identification because such features are defined so that they can be extracted regardless of the shape of certain characters. Quite often the document-level features are extracted independently of the document content. Said et al. (1998) developed a writer identification system by using two sets of purely computational macro features extracted from scanned document pages using Gabor filtering and computation of gray scale co-occurrence matrices. Further development of the method and use of the feature sets with different types of classifiers is reported in (Said et al., 2000) and the identification accuracy of 96% on samples of 1000 documents from 40 writers was achieved. Bulacu et al. (1995) extracted edge-direction distribution and edge-hinge distribution features from images of handwriting preprocessed by an edge detection operator. The rationale behind the use of such features was the significance of distribution of directions in handwritten curves which could be useful for writer identification (Maarse et al., 1988) and classification of writing style (Crettez, 1995). Identification accuracy of around 97% was reported on the data of 500 written samples from 250 different writers (Schomaker et al., 1995).

Extraction of another purely computational *fractal features* was performed by Tang et al. (2002). The features can be extracted from any two-dimensional images and are aimed to be used for handwriting recognition, writer identification, and off-line signature verification. Evaluation of performance of a system based on such features was made for the Chinese character recognition problem and some results were presented for signature verification. Another use of fractal features of handwriting has been demonstrated by H.-C. Chen et al. (2003). The feature called *wrinkliness* is aimed at forgery detection and based on a reasonable assumption that although one can copy the shape of another person's handwriting, it is difficult to make it naturally fast and with the same speed and acceleration pattern of the pen. When an attempt at forgery is made, the strokes are often drawn slowly, which results in more wiggly or less fluent handwriting and this difference is captured in the wrinkliness feature value. This feature is related to the *writing quality* feature from the 21 discriminating list of handwriting (see Section 1.3.2).

Extensive study of extraction and use of computational features has been carried out at CEDAR. The background of the research on handwriting individuality was a project on automatic mail sorting system, which required dealing with handwritten text on mail pieces. Such problems as locating the address block, segmentation of address lines and interpretation of an address arose in the work. Algorithms for locating addresses exploited probabilistic rules of address positions for different types of mail (Srihari, 1988). There were also feature extraction algorithms for discrimination between handwritten and printed addresses. The next step in the work was the recognition of handwritten and machine-printed text.

In the work of Cohen et al. (1991) and Srihari (1993) feature extraction for address recognition was implemented. Features, which are similar to what are referred to here as document examiner features (arcs, caves, holes, strokes) were used for classification of digits along with computational features, namely *gradient features*, Gabor coefficients and contour chain code. Different types of classification algorithms were applied to different feature vectors. For computational features a neural network classifier was used. Use of only gradient features resulted in an accuracy rate of 96.4%.

Character recognition was a more complicated problem. A recogniser that used *structural features* was implemented there. These features were extracted by use of morphological operators. Similar to the gradient features, structural features were computational features too. Later it was demonstrated that such features could be used for discrimination of different handwriting styles (Govindaraju et al., 1999). A

detailed description of gradient and structural features extraction is available (Srikan-  
tan et al., 1996).

For the purpose of writer identification several different features were extracted. Some of them were macro features. All macro features were grouped into three categories: darkness features, contour features (connectivity and slope features), and averaged line-level features. In total 11 macro features were extracted.

The darkness feature category consisted of gray-level distribution, gray-level thresh-  
old values and the number of black pixels. The contour features category included the  
number of interior and exterior contours extracted from a chaincode representation of  
a document image (Freeman, 1961; Kim & Govindaraju, 1997). The average number  
of interior and exterior contours could be used as a measure of writing movement,  
to distinguish between cursive handwriting and hand-printed texts. There were also  
four contour slope features in this category: vertical, horizontal, positive, and nega-  
tive slope components. Such features indicated different stroke formation. Averaged  
line-level feature category consisted of slant and height features extracted at the line  
level.

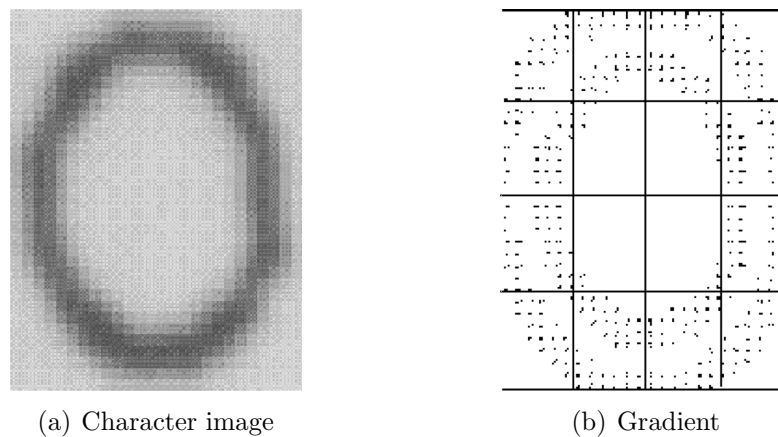


Figure 2.4: Sample image and its gradient direction map (taken from (Cha, 2001))

Micro features were those used in the problem of address recognition for auto-  
matic mail sorting. These features were gradient (Figure 2.4), structural, and con-  
cavity (GSC) features. Later several other features were added to the feature set.  
The complete list of features used for the purpose of handwriting classification and  
handwriting individuality hypothesis validation is shown in Table 2.1. According to  
the definitions given above all of the features were computational features but only a  
part of them were also document examiner features.

| Features                              | Document | Paragraph | Word | Character |
|---------------------------------------|----------|-----------|------|-----------|
| Gray level entropy                    | Y        |           |      |           |
| Gray level threshold                  | Y        |           |      |           |
| No. of black pixels                   | Y        | Y         | Y    | Y         |
| No. of interior & exterior contours   | Y        | Y         | Y    |           |
| No. of 4-directional slope components | Y        | Y         | Y    |           |
| Average height                        | Y        | Y         | Y    |           |
| Average slant                         | Y        | Y         | Y    |           |
| Aspect ratio                          |          | Y         |      | Y         |
| Margin width                          |          | Y         |      |           |
| Length                                |          |           | Y    |           |
| Upper & lower zone ratio              |          |           | Y    |           |
| Centroid height & width ratio         |          |           |      | Y         |
| Spatial features                      |          |           |      | Y         |
| GSC                                   |          |           |      | Y         |

Table 2.1: Features extracted from handwritten document at four levels of coarseness: document, paragraph, word, and character (adapted from (Srihari, Cha, Arora, & Lee, 2001a))

## 2.4.2 Document examiner features

Extraction of only document examiner features was performed in several studies. Use of document examiner features for detection of forgery in signatures was demonstrated in (Ammar et al., 1989). A set of features was studied that include shape, slant, length of strokes and density features. It was demonstrated that use of only geometrical features result in accuracy of signature verification of about 86% on the data of 400 samples from 20 people.

Estimation of slant of handwriting has been a topic of active research in both writer identification and handwriting recognition areas. In the latter, elimination of slant and slope (also called normalisation of handwriting) often helps to improve the recognition rate. Example of slant estimation algorithm applied to separate words is presented in (Kavallieratou et al., 2001). A number of methods of slant and baseline slope (also known as skew) estimation for whole handwriting images are discussed in several papers (Poddar, 1997; Chin & Jennings, 1997; Vinciarelli, 2002; Shi & Govindaraju, 2003). The problem with slant as well as slope estimation is that there is no clear definition of how to determine those values from a handwritten sample. Hence in designing a method for slope estimation, for example, it is a norm to use subjective measures such as “what a person feels the slope is”. Once the method gives a reasonable estimate of slope from the subjective point of view, the slope feature can be strictly defined as exactly the value produced by the method.

Loop features are considered to be important for both handwriting recognition and

writer identification. However, it is often hard to identify loops in a scanned sample of handwriting because loops can be incomplete or hidden. The latter occur when the strokes that form the loop are drawn too close to each other so that there is little or no white space between them. The amount of white space can also be reduced by a scanning process. Such a hidden loop looks like a blob on the handwriting image and is hard to distinguish from other blobs. Several methods have been proposed to reconstruct close strokes and hence hidden loops. For example, a three-stage method based on analysis of blob shapes is presented in (Doermann et al., 2002). Other methods exist mostly as parts of stroke recovery techniques and are discussed in Section 2.5.3.

Formalisation and extraction of other complex document examiner features has been discussed. H.-C. Chen et al. (2003) presented a formalisation of the stroke quality feature by measuring *wrinkliness* of handwriting strokes from images of handwriting samples. Found and Rogers (1996) discussed a feature commonly referred to as “complexity” of handwriting. This feature reflects the perceived ease or difficulty of successful simulating the handwriting by a forger. The feature was mapped into a set of four other features: the number of turning points in the line, the number of line intersections and retracings, the number of line portions crossed by other lines, and the presence of feathering of the line (measure of fluency of handwriting). These four features were used to classify signature samples into one of three complexity categories close to the classification performed by professional document examiners. The agreement of the results with those produced by the examiners was reported to be 73.5%.

Further in this section systems are considered that were developed mainly to automate or assist the work of forensic document examiners. The system include automatic or interactive extraction of document examiner features and do not deal with purely computational features.

### **The Forensic Information System for Handwriting (FISH)**

This system was developed for use in the Federal Bureau of Criminal Investigation (Bundeskriminalamt) in Germany. The system was capable of scanned image enhancement, interactive extraction of some features and comparing the questioned document with documents in existing database (Franke et al., 2003). The features that could be extracted automatically or semi-automatically were:

- letter isolation

- line tracking
- width
- size
- slant
- basic ovals
- loop shape
- zone emphasis
- line spacing

Further development of the system introduced more document examiner features such as *writing style* and *connectivity*.

### Writer Identification System (WIS)

This system was a further development of FODES (Forensic Document Examination Software) system developed in the UK (Leedham et al., 1995; Holcombe & Leedham, 1995). The FODES system was a set of tools that allowed a document examiner to work with a scanned image instead of working with a hard copy (Holcombe et al., 1996). In the WIS system several new blocks were added, including feature extraction. An overview of the complete system is shown in Figure 2.5 (Greening et al., 1995a, 1995b). Feature extraction blocks are emphasised. The system was capable of extracting the following sets of features:

- **Global features:** average slant of text, height of ascenders and descenders, character size, style of writing, fluidity of writing, and margin behaviour.
- **Local features:** (partially interactive extraction) features of several letters (Eldridge et al., 1984).
- **Texture features:** these were purely computational features extracted by applying transforms to a whole document image (Steinke, 1981).

Extraction of some features from letters were done automatically and estimation of the extraction accuracy was obtained (Greening et al., 1995a). All the letters were divided into components and the features were extracted according to components construction. Components of the letters are shown in Figure 2.6 (adapted from (Greening et al., 1995a)). Below is the list of features for which automatic extraction algorithms were implemented.

- **Construction order.** What part was constructed first: base or staff. Not applicable for “t” and “f”.

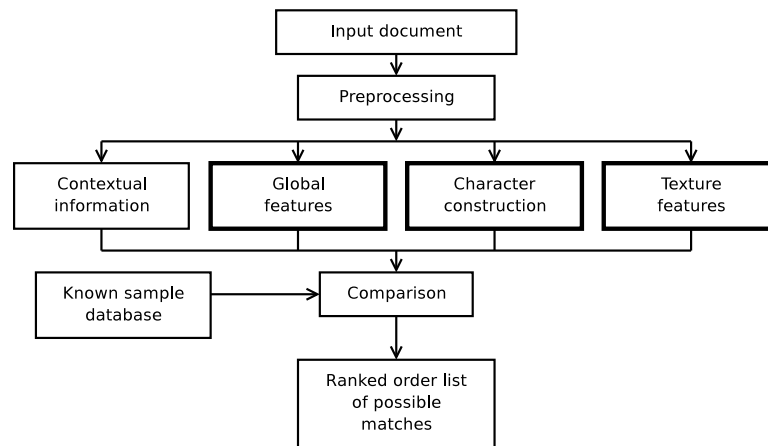


Figure 2.5: Writer Identification System (adapted from (Greening et al., 1995a))

- **Number of strokes.** One of three choices: 1, 2, or 3 strokes.
- **Base direction.** This feature had 6 possible choices and was used for base construction description. Applicable only to the letters with base component.
- **Base closure.** Described if a base is open or closed. The feature was always inapplicable to “f”, “t” and “h” and sometimes to “k”.
- **Top of staff closure.** Retrace, loop or straight stroke at the top of staff were coded by this feature.
- **Top of staff direction.**
- **Bottom of staff closure.**
- **Bottom of staff closure.**
- **Bottom of staff direction.** This was a measure of the direction of curvature at the bottom of staff.
- **Preceding letter join.**
- **Following letter join.**
- **Crossbar position.**
- **Crossbar curvature.** Possible choices were straight line, concave up or down, or both up and down.

### Forensic Document Examination Toolset (FOX)

The FOX system was developed in the Nanyang Technological University, Singapore (NTU) with assistance from the Singapore Institute of Science and Forensic Medicine (Chong, 1996). The system contained both interactive and fully automatic tools, which allowed a forensic document examiner to extract both global and local features for analysis purposes. The diagram of the system is shown in Figure 2.7 (taken from (Leedham, 1999)).

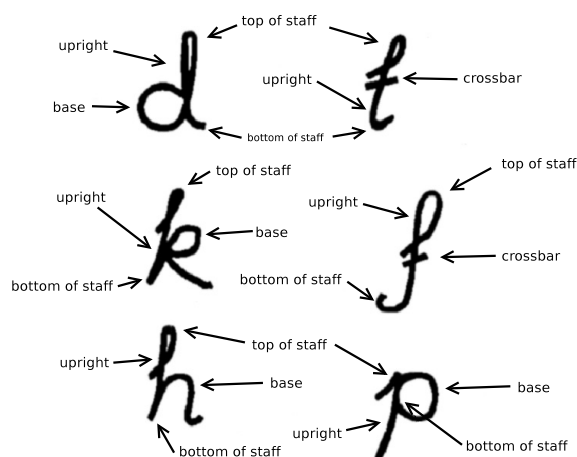


Figure 2.6: Components of the letters “d”, “t”, “k”, “f”, “h”, “p” (adapted from (Greening et al., 1995a))

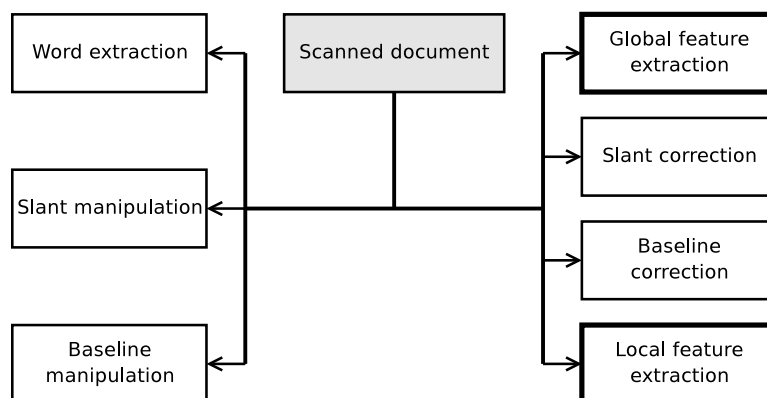


Figure 2.7: Diagram of FOX system.

The extraction algorithms were developed for the following set of features.

1. Global features.

- Word spacing
- Line spacing
- Height of main body, upper zone, and lower zone of words
- Slant and baseline angle (slope)

2. Local features.

- Slant and baseline of a word
- Loop detection (interactive)

Baseline alignment, also known as slope or skew, refers to lining up the base of writing along an imaginary line. It is seldom rigidly straight though some people can produce quite straight lines of handwriting. Variability in baselines can result from individual habits, visual difficulties, and from unaccustomed writing position (Robertson, 1991). Therefore, it is possible that when an attempt at forgery or disguise is

made the person will try to change the baseline patterns. The alteration of baseline patterns can occur as change in the overall baseline, introduction of baseline variation, and ruler guided baseline alignment.

Handwriting slant is also a feature which is said to be very consistent for many people. Changing handwriting slant is the most common way for a person to disguise his handwriting (Robertson, 1991). For the purpose of slant measurement a bounding box of non-horizontal strokes was used. For each word any row which contained horizontal runs of black pixels greater than some threshold value were removed. Then each connected component of the word was isolated. Depending on the component sizes some components were removed. For others components slant of each of them was estimated as  $\tan^{-1}(\text{height}/\text{width})$ . The word slant was calculated as the average of all the corresponding component slants.

For each global feature the feature values were measured across the whole document image and three values were produced, namely the average feature value, the variance and the standard deviation.

Tools for image manipulation were also implemented in this system. The tools allowed the user to manipulate word slant and word baseline in order to help detect forgery.

### Further development of FOX system

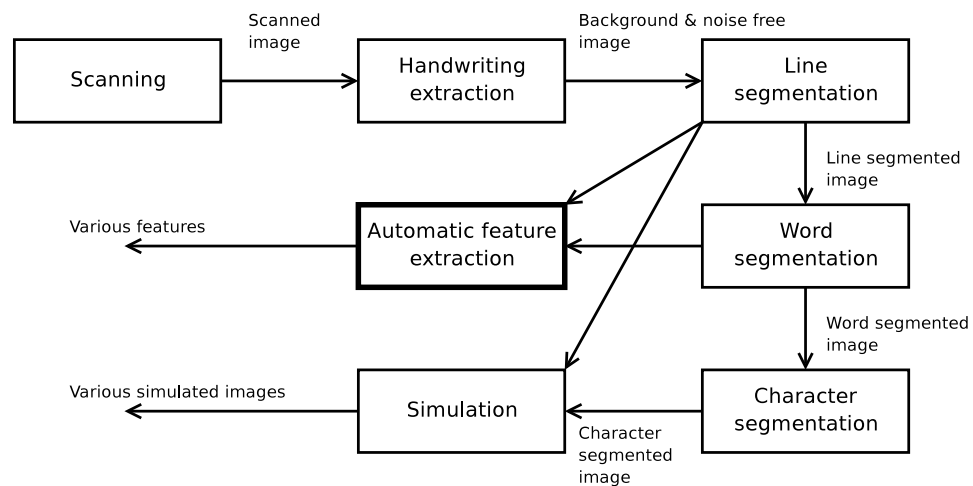


Figure 2.8: Diagram of the improved FOX system (adapted from (Solihin, 1997)).

A number of improvements were introduced in the FOX system by Yan Solihin (Solihin, 1997). The improved system was capable of automatic extraction of global and local features as well as manipulating of some handwriting parameters, like slant and slope. The algorithms used for feature extraction and handwriting

manipulation were different from those used in the FOX system and gave promising results. An attempt was made to formalise loop forms through the measurement of several loop features.

The diagram of the whole system is presented in Figure 2.8. The list of features the system was capable of extracting of are:

- Baseline (slope)
- Slant
- Stroke width
- Loop features: area, slant and circle dissimilarity index.

In contrast to the existing methods (Bozinovic & Srihari, 1989; Caesar et al., 1995) in which a linear approximation of baseline was employed, the current system used a new method which exploited the information of south contours of handwritten words and lines. The baseline of a word was divided into several segments and a linear approximation was employed on each segment. Such a method was able to represent the baseline with high accuracy. It was also able to remove the baseline variability of a slanted baseline handwriting as well as hill and dale handwriting.

There was also a new method for slant measurement (Sagar et al., 1996). Measurement of a word slant in FOX system (Chong, 1996) had several disadvantages. It was inaccurate in cases of thick handwriting and it was unable to measure slant in cases of words for which all connected components had been removed as they had failed the tests. Such cases occurred when a word was too long, or a word contained no letters with ascenders and descenders, or a word contained too many letters with ascenders and descenders. The method used in the current system was a further development and improvement of the method of Bozinovic and Srihari (1989). The possible inaccuracy that arose from the way of slant estimation used in the FOX system and the new way of slant estimation are best shown on Figures 2.9(a) & 2.9(b).

Stroke width was measured in order to assist work of other algorithms, including image enhancement algorithms. It was a very important measure because it could make other algorithms independent of the resolution of an image and of the handwriting thickness.

It was also one of the first reported attempts of automatic extraction of several loop characteristics. Loop area was measured as the number of pixels inside the loop contour. Slant of a loop was measured by dividing the loop into the upper half and the lower half and measuring the slant of the line connecting the centres of gravity

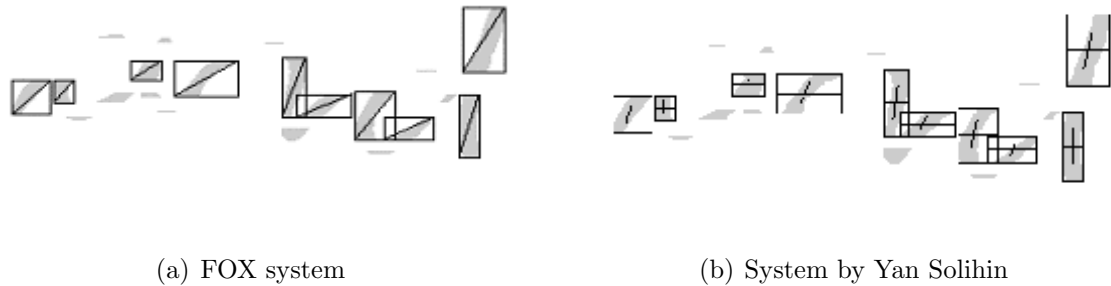


Figure 2.9: Estimation of component slants in FOX systems (adapted from (Solihin, 1997))

of the obtained loop parts. Circle dissimilarity index was measured as the average deviation of the distance of the loop contour pixels to the centre of the best-fit circle from the circle's radius.

### WANDA measurement plug-in (WAM)

WANDA is a system designed to automate document examiner's work and is considered in more detail in Section 2.6. It consists of the main system and several plug-ins to perform various operations with document images. A measurement plug-in is developed to extract some document examiner features, both automatically and interactively. The following set of features can be extracted:

- Heights of characters and their parts: ascenders, descenders, base.
- Slant of characters.
- Width of characters.
- Loops sizes.
- Average distance between baselines.
- Allograph measurement.

The allograph measurement is based on interactive tracing of handwritten characters using a digitiser and comparison of the resulting on-line trace with the known traces in the database. The detailed description of the feature extraction capabilities is presented in (Franke et al., 2004; Van Erp et al., 2003).

### CEDAR-FOX system

CEDAR-FOX system has been developed in CEDAR and is based on the research in handwriting individuality validation and handwriting recognition (Srihari, Zhang, et al., 2003; Srihari & Leedham, 2003). The system allows the extraction of several document examiner features at document, word, and character levels, such as slant,

stroke width, word gaps as well as purely computational features. Apart from feature extraction, the system allows the user to perform search of similar handwriting using image-image similarity based on the extracted computational features (Srihari, Huang, & Srinivasan, 2005). A detailed description of the system is presented in (Srihari, Huang, Srinivasan, & Shah, 2005).

In all the works considered above a number of document examiner features, both macro and micro features, were formalised and extracted. Some of them are used in the current project, although most extraction algorithms are different.

### 2.4.3 Distance measures for feature vectors

When extracting features it is possible that different features have different types of values. Some features like height of a character are expressed in real numbers. Other features like angles are also expressed as real numbers, but some values are equivalent: for example, slant of 0 is the same as slant of  $\pi$  radians. There are binary features which denote presence of some elements such as t-bar, and features that can take a certain number of integer values. In addition to the features of element type just mentioned, there are other features that can be expressed as histograms or strings. Comparison of handwriting samples means comparison of their feature values. Hence it is important to have a correct distance measure that can handle different feature types.

Extensive study of this issue has been carried out in CEDAR. Below some of the distance measures for non-trivial feature types are considered shortly. Figure 2.10 show the different types of features used in CEDAR studies according to the applicability of distance measures.

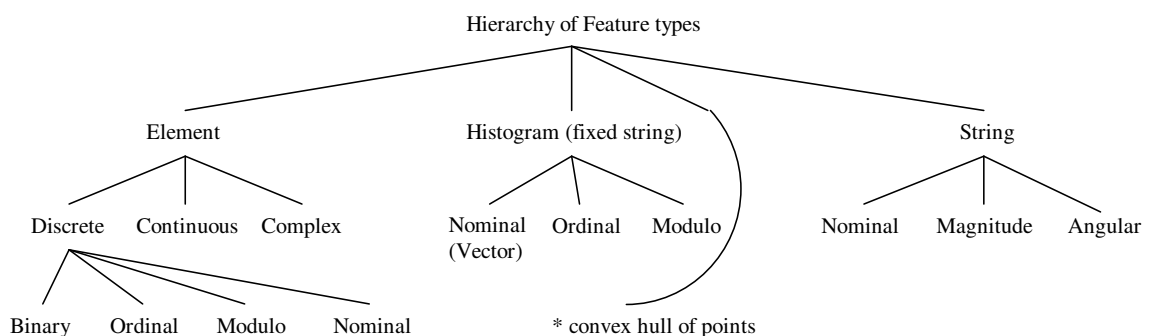


Figure 2.10: Different types of features used in CEDAR (adapted from (Cha & Srihari, 2000b))

## Measure of distance between histograms

Histograms can be treated either as fixed-dimensional vectors or as approximations of probability density functions (pdf). In the former case any vector distance measure is applicable. Computing the distance between two pdf's can be regarded as computing the Bayes probability. However, known distance measures do not take into account the similarity of non-overlapping parts of the two histograms, that is why a new distance measure was introduced that used the notion of the *Minimum Difference of Pair Assignments* (Cha, 2001). Three types of histograms were considered, *nominal type*, *ordinal type*, and *modulo (angular) type* (Cha et al., 1999a). A comparison of the new distance measures with the conventional types revealed that the new measures represented histogram similarity better for the task of handwriting feature comparison (Cha, 2001).

## String distance measures

Another important distance measure was introduced for measuring similarity between two strings. A string is a sequence of symbols drawn from the alphabet and it is one of the popular representations of patterns. Three types of strings were considered: *nominal type* (each symbol value is named), *angular type* (elements are angular degrees), and *linear type* (elements are integer or real values) (Cha et al., 2000, 1999b).

## Binary vector distance measures

Several popular binary dissimilarity measures, both metric and non-metric, are considered in (B. Zhang & Srihari, 2003b, 2003c) from the viewpoint of their performance in writer classification based on binary computational micro features. For each dissimilarity measure its weighted version was considered by multiplying the number of occurrences of matches of zero in one vector with zero at the same position in the other vector  $S_{00}$ . The rationale behind such a weighting scheme was that '0' bit provided less information about classes separability than '1' bit in the problem at hand. It was shown that appropriate choice of the weighting factor can significantly improve classification accuracy.

## 2.5 Preparation of handwriting image for feature extraction

Before extraction of features from a document image is attempted it is necessary to have the image in an appropriate form. Firstly, the image needs to be separated from the background. Depending on the scale at which features are extracted, the image needs to be segmented to paragraphs, lines, words, or characters. Often micro features represent shapes of certain strokes and are extracted from a skeletonised version of the character image. Hence, an appropriate skeletonisation (or stroke extraction) method is needed. Several studies on background separation, segmentation, and stroke extraction are discussed below.

### 2.5.1 Background separation

Background removal is a preprocessing stage that is often crucial for further stages to produce correct results. Non-uniform background often occurs in scanned images of handwriting because of many factors such as old paper, paper with pattern, faint ink, thin paper, and double-sided writing as well as their combinations. Scanning itself introduces noise too that affects even perfectly white background. Automatic separation of the foreground from the background can be a complicated problem that requires sophisticated algorithms (Leedham, Chen, et al., 2003). It arises in many applications that involve document scanning. In many areas this problem is investigated and a large number of different algorithms have been proposed (Otsu, 1979; Abutaleb, 1989; Chi & Hong, 1993; Savakis, 1998; Solihin & Leedham, 1999; Sauvola & Pietikäinen, 2000; Shah-Hosseini & Safabakhsh, 2002; Y. Chen & Leedham, 2004).<sup>2</sup> Quite often background separation methods are problem-specific: they work well only when certain conditions hold. The area of background separation is out of the scope of the current research project; it is assumed that input images are already separated from background well enough, that is, without affecting the document examiner features of interest. Sets of interactive and automatic image processing tools including background separation that have been developed within systems to facilitate forensic document analysis are discussed in Section 2.6.

---

<sup>2</sup>As it is almost impossible to reference all key thresholding/background separating methods here only some papers are cited; the bibliographies in the cited papers contain more references.

## 2.5.2 Image segmentation

Segmentation of document images usually comprise two areas: page decomposition, which is a separation of text from non-text regions (images, etc.) and segmentation of text regions themselves into lines, words, and characters. Handwriting images often need to be segmented down to the character level for both handwriting recognition and forensic analysis. Segmentation of scanned documents of machine printed text only is a well-explored area nowadays. Segmentation of handwritten document images containing only text is a far more complicated problem because of irregular and diverse shapes of handwritten characters. Even automatic segmentation of lines poses significant problems caused by touching and overlapping lines (Doermann & Rosenfeld, 1993; Takru & Leedham, 2002). There has been even less success in automatic segmentation of words and characters. Several studies have been performed in this area and different methods proposed (Dunn & Wang, 1992; Lecolinet & Crettez, 1991; Sagar et al., 1996; Anquetil & Lorette, 1997; Eu, 1999; Ho, 2001; De Stefano & Marcelli, 2002; Nicolas et al., 2004). The main problem of fully automatic handwriting segmentation is recognition of strokes from raster images and it is closely related to the problem of transition from raster image of handwriting to a document where handwriting is represented as a set of strokes (Liao & Huang, 1990; Liu et al., 1997; Nakajima, Mori, & Tagekami, 1999; Plamondon & Privitera, 1999; Mestetskii et al., 2002). In the current project features of characters are studied and it is assumed that the characters have already been segmented from the documents. A study of stroke extraction is important for the success of feature extraction and is performed in the current project, so the background of the problem of handwriting skeletonisation and stroke extraction is discussed in detail in the next section.

## 2.5.3 Stroke extraction / skeletonisation

For both off-line handwriting recognition and author identification many of the features are extracted from skeletonised images. Moreover, a person perceives handwriting as a set of strokes rather than as a set of pixels and many of the document examiner features describe shapes of the strokes. Precision in measurement of values of most of such features strongly depends on the quality of skeletons of handwritten elements from which features are extracted.

There are a number of techniques to obtain skeletons of handwritten samples. Applicability of a particular technique depends on the problem in hand. Thinning meth-

ods (Lam et al., 1992; Suen & Wang, 1994) are very popular for skeletonisation mainly because of their implementation simplicity and high computational speed. However, the results they produce have many drawbacks such as erroneous branches and other artifacts, affecting the subsequent feature extraction (Figure 2.11(c), 2.11(d)). Thinning also destroys stroke order information, which can be used in handwriting recognition to partially employ the online handwriting recognition methods as well as in author identification to add more features representing individual character formation.

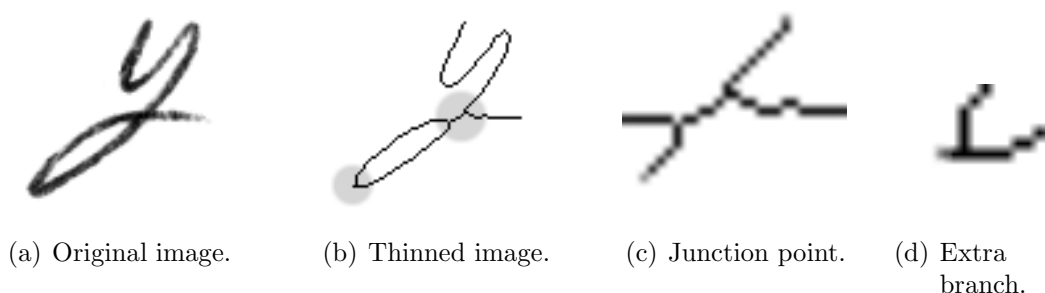


Figure 2.11: Artefacts produced by thinning.

Because of the drawbacks of thinning methods a number of skeletonisation and stroke extraction methods have been proposed which are based on thinning. Such methods use the thinned image as the first approximation to the desired skeleton and then use various methods to correct junctions and branches, recover loops (Doermann et al., 2002) and sometimes also to recover the writing sequence (Doermann & Rosenfeld, 1995). The latter is especially important in recognition of handwritten Chinese characters and several methods have been proposed recently (Amin & Singh, 1997; Liu et al., 1999; Chang & Yan, 1999; Su & Wang, 2003).

Some methods of skeletonisation are based on contour tracing (L'Homer, 2000; Zou et al., 2001; Plamondon & Privitera, 1999; Hew & Alder, 1999), fitting of curves (Kégl & Krzyżak, 2002; Wenyin & Dori, 1998; Liao & Huang, 1990), use of self-organising map (Datta et al., 2001), wavelet transform (Tang & You, 2003), bi-tangent circle (Mestetskii et al., 2002).

Recovery of stroke sequence from thinned handwriting images of characters of the Latin alphabet has also been presented (Jäger, 1998; Lau et al., 2003; Qiao & Yasuhara, 2004). Such recovery is desired for forensic document analysis because it allows to extract additional document examiner features (Doermann & Rosenfeld, 1992). Both reconstruction of writing sequence and correction of junction points is based on the observation that people normally produce smooth curves when writing and tend to write characters using the least effort. These facts inspire use of op-

timisation algorithms to reconstruct the handwriting sequence by searching for the minimum of a cost function with local (e.g. smoothness) and a global (e.g. number of pen-ups) cost components (Jäger, 1996, 1997; Bunke et al., 1997; Nakajima, Mori, & Taguekami, 1999).

Many of those methods produce results that have sufficient accuracy for handwriting recognition purposes and can work quite fast. However, there is a problem in applying the methods to forensic analysis of handwriting. In handwriting recognition the features are aimed at distinguishing different characters or their combinations and thus need to represent dissimilarities between different alphabet characters and have similar values for the same characters written by different writers. In writer identification the features of interest are aimed at distinguishing between writers and thus need to emphasise the differences in shapes of the same characters written by different writers. Unfortunately, all the methods of stroke extraction/skeletonisation have been designed for application to handwriting recognition, and do not preserve most of individual traits the writer endows to a character. That is why for author identification these methods hardly give better results than the thinning techniques (Pervouchine et al., 2005a).

In this project a new skeletonisation method is proposed based on calculation of B-spline-approximated handwriting branches from grayscale images of characters and then use of optimisation techniques to produce correct junctions, loops, and hidden loops. The method is intended to be used in forensic document examination. A detailed description of this method is given in Chapter 4.

## **2.6 Interactive tools for forensic document analysis**

Currently most of the work in forensic document analysis is performed manually. Copying of document parts, pasting them into comparison charts and other routine tasks can be performed on scanned documents with the help of computer tools thus saving a substantial amount of time and effort for document examiners. Tools to facilitate the work of a forensic document examiner include image processing tools such as noise removal and background separation, segmentation tools, feature extraction and measurement tools, visual comparison of document parts, quantitative comparison of documents and storage and retrieval of documents with similar handwriting.

### 2.6.1 Overview of the existing systems

Since 1975 several systems have been proposed and some of them are now in use. Some of them are considered below.

#### **The Forensic Information System for Handwriting (FISH)**

FISH system is probably one of the oldest systems of this kind; its development started in 1975. The system allows some image processing like noise removal and thresholding to be performed. It also stores scanned processed documents in a database. The system was designed to handle 10000 new investigation cases every year and by 1995 it had about 77000 handwritten documents stored. Some document examiner features can be measured manually and some can be extracted automatically (Franke et al., 2003).

Another computer-based system with powerful interactive image processing tools have been developed by request of the Bundeskriminalamt, the main user of the FISH system (Franke & Köppen, 2001). The goal was to employ new algorithms in image processing and pattern recognition that had been developed. The system is able to perform sophisticated interactive multi-stage processing of images, and examples of successful background separation in complicated cases of patterned background have been reported. The set of tools have been used in the WANDA system developed to replace the FISH system in the German central police bureau (see below). The software is written in C and can be compiled for a number of platforms.

#### **Writer Identification System (WIS)**

The system has interactive tools that allow user to trace over the strokes that make up characters. The information obtained during the tracing is then used to classify the characters according to Eldridge classification scheme (Eldridge et al., 1984).

#### **Pattern Evidence Analysis Toolbox (PEAT)**

The PEAT system was developed in La Trobe University, Australia (Found et al., 1994). The purpose of the system is to assist forensic document examiners by providing a set of interactive tools for different measurements as well as facilitating the comparison of handwriting samples. The tools enable the user to measure the total line length, the total area enclosed by the line, the area of any enclosed region, including the regions separated by other lines, the path length between any two points, and angles. Also magnification of a small region under the cursor is provided. In order

to measure line (stroke) features, the software implements line tracing according to a direction specified by the user, so that crossing and overlapping strokes can be treated correctly.

For each measured value the number of measurements, the mean, the standard deviation, and the range are calculated. The results can be imported to a spreadsheet software. The related 'Angular Differential' software (Found et al., 1997) enables to detect the points of extreme curvature on handwriting samples. Application of the software to the problem of handwriting analysis has been reported (Found et al., 1998; Found, Rogers, & Metz, 1999). The software runs on Macintosh computer series II and above.

### **Forensic Document Examination toolset (FOX) and its improved version**

In this toolset automatic thresholding of document images was implemented using integral ratio techniques (Solihin & Leedham, 1999). Automatic segmentation of text was implemented via connected components mapping. Connected components that consist of parts of several lines are split and components that are the parts of the same word are merged. This makes possible automatic line and word level segmentation. The system also has visualisation tools that allow user to change word slant and baseline, and simulate ruler guided writing (Leedham, 1999). The software runs under the HP-UX operating system.

### **Pikaso Write-On system**

This system, which is a commercial software (*Pikaso Write-On Handwriting Comparison Software*, n.d.) is a set of tools that helps document examiners to visually compare handwriting from several questioned documents. The program performs context based interactive letter level segmentation of handwritten text. It allows the user to find all occurrences of a single character or combination of characters in the documents and put them into a comparison chart. The version available for free evaluation runs under the Microsoft Windows operating system.

### **WANDA system**

The WANDA system is being developed to replace the current FISH system. Design of the WANDA system allows it to be used in various forensic laboratories. The system is implemented as a kernel and a number of plug-ins that can be extended and modified as new algorithms are developed. It uses the XML language to describe the

data and processing instructions between the components of the system. The system consist of “(a) clients with client plug-ins, which implement graphical user interface, (b) servers with server plug-ins, which implement the processing modules, (c) data repositories and working sets, and (d) web clients for remote data retrieval” (Franke et al., 2004). Use of Java programming language and XML data interchange language allows the system to be used on a variety of platforms.

The system has an image processing module which is mostly described in (Franke & Köppen, 2001) and has document-independent, document-specific (with adapted parameters), and document-dependent filters that allow documents with coloured and textured backgrounds to be handled. The system has an annotation module that describes a document in terms of features, including nominal features. A number of nominal features are designed so that they allow the user to express a degree of membership of a document to a certain category. The system also enables some features to be measured using interactive tools and supports automatic classification of writers to help to identify documents with similar handwriting.

### **CEDAR-FOX system**

The system is capable of automatic image preprocessing, segmentation, and recognition of handwritten document. The segmentation can be made up to the word level. The system allows some document examiner macro features to be extracted at the document level as well as computational GSC features at the character level. Comparison of handwriting of two writers (writer verification) can be performed. Also storage and retrieval of samples along with their metadata is implemented. The retrieval module allows to perform search of document samples by using whole document image, a region of interest, or words as a query (Srihari & Shi, 2004; Srihari, Huang, & Srinivasan, 2005). The handwriting word recognition module allows the user to use word images for search. The CEDAR-FOX system is developed to run under Microsoft Windows.

### **Computer Assisted Visual Interactive Recognition (CAVIAR) system**

This system, although not designed for as a forensic document examiner tool, uses the perceptual ability of humans to classify different patterns (Nagy & Zou, 2002). The user helps the system in feature extraction by pointing to certain objects in an image, essentially facilitating the image analysis. As more features are extracted by the system, the candidate classes are presented to the user with the corresponding

confidence intervals. The final decision of classification is made by the user. The system architecture seems promising for interactive forensic document examination tool.

## 2.6.2 Summary on the existing tools

Below is the list of tools for document examination that have been developed and implemented in various systems.

- *Document image pre-processing.* There exist a number of algorithms for noise removal and background separation. Some of the tools incorporated into systems for forensic document examiners are able to successfully handle even complicated cases of textured and coloured backgrounds. The problems still exist in background separation of images of documents written on textured and/or very old paper with faint ink.
- *Segmentation of lines, words, and characters.* Although a number of methods for segmentation on different levels have been proposed, the accuracy of the methods is quite low. Interactive and semi-automatic segmentation algorithms seem to be promising solutions.
- *Tools for visual comparison.* The algorithms already implemented are those manipulating baseline and slant of handwriting. Comparison of images side-by-side is implemented in almost all existing systems.
- *Tools for quantitative analysis.* In the FOX system a report is generated with the values of the features extracted. For each feature its average, standard deviation, and variance are produced. CEDAR-FOX and WANDA systems are able to extract features and perform automatic comparison based on the extraction.

## 2.7 Writer verification / identification via pattern recognition

In order to identify the writer a feature vector representing the writing is fed to a classifier. The problem of classification is: given a feature vector  $\vec{f}$  as an input it is necessary to recognise the class label  $\omega$ , which corresponds to one of the known writers. Provided the probabilities  $prob(\omega_i|\vec{f})$  are known, the Bayes' decision rule can be stated as follows: decide for the class label  $\omega_i$  if  $prob(\omega_i|\vec{f}) = \max_j\{prob(\omega_j|\vec{f})\}$  and  $prob(\omega_i|\vec{f}) > \beta$ , where  $\beta$  is a threshold. Otherwise the decision should be "the

pattern is not from class  $\omega_i$ ” (Figure 2.12). Since the distributions of feature values in feature space are unknown in advance, the only practical way to the design of the classification system is to rely on the paradigm of learning from examples (Duda & Hart, 1974). Generally a mapping  $\vec{f} \rightarrow \vec{d}$  is established for classification, where  $\vec{d}$  consists of class discriminant functions  $d_i$ , which can be the probability functions (as above), or their estimates: some kind of distance-measuring function evaluating the dissimilarity of  $\vec{f}$  with each of  $I$  classes  $\omega_i$  (Schrümann, 1996).

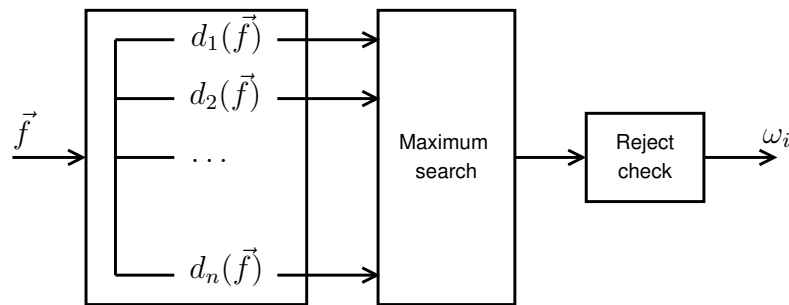


Figure 2.12: Diagram of a statistical pattern classifier.

### 2.7.1 Classifiers

Below some classifiers are discussed that have been used in studies of handwriting individuality.

#### k-NN classifier

One way to get an estimate of  $prob(\omega_i|\vec{f})$  is to determine the set of  $k$  nearest neighbours of a data point  $\vec{f}$  and assign the class label to the data point based upon the most numerous class with the set. In the simplest case  $k$  is equal to 1 and the class label assigned is the same as the class label of the nearest sample, ties being broken arbitrarily. Such a classifier has been used in CEDAR and methods of reduction of computational cost caused by many samples and high dimensionality of the feature space have been proposed (Srihari, Cha, & Lee, 2001). The proposed k-NN search algorithm calculates partial distances between samples and builds an *additive binary tree (ABT)* to pre-structure the distances (Cha & Srihari, 2002). Methods of computational cost reduction when only binary features are used has also been proposed in (B. Zhang & Srihari, 2004).

#### Multilayer perceptron based classifiers

Multilayer perceptron (an artificial neural network, ANN) is one of the most often used approaches to classifier construction in the area of writer classification. The approach

is based on the fact that the multilayer perceptron can be used to approximate an arbitrary mapping  $\vec{f} \rightarrow \vec{d}$ . It is possible to show that back propagation learning leads to the approximation of discriminant functions  $d_i$  (Schrümann, 1996).

**Backpropagation networks** are among the most popular choices of the ANN classifiers (Duda, 1994). One drawback of the approach is that there is no definite way of choosing the network topology. When too few elements (perceptrons) are used the classification performance of the network is poor, when too many elements are used the network has poor generalisation ability, the effect also known as overfitting. Currently trial-and-error method is normally used to determine a suitable topology of an ANN classifier. Examples of use of ANN for validation of handwriting individuality hypothesis are (Cha, 2001; Srihari et al., 2002).

**Constructive networks** are the networks which have the advantage that they do not have a predefined topology. Instead, elements are added during the learning phase as long as a necessity arises to classify more samples correctly. Such an approach offers an incremental construction of a network until some pre-defined stop condition is met. Unfortunately, the risk of overfitting is still present and one way to lessen it is to restrict the maximum number of elements in the network (or to adjust a stop condition). A number of neural network construction algorithms such as *Pyramid*, *Tower* (Gallant, 1990), *DistAl* (Yang et al., 1997) and their modifications have been proposed and some of them have been shown to converge to zero classification errors under certain assumptions (such as clear separability of classes). Several of the algorithms were evaluated by Parekh et al. (2000) and it was shown that their performance was comparable to that of other learning algorithms on standard data sets.

### **Classifier combination**

Use of several classifiers and combining their output has been proposed as a means to improve the classification accuracy (Xu et al., 1992). It has been shown that when separate classifiers have low classification error rates and are diverse (make different errors) combinations of such classifiers significantly improves performance of a pattern recognition system (Kittler et al., 1998; Sharkley et al., 2000; Kuncheva, 1998; Kuncheva et al., 2000). Approaches to automatic selection of combination scheme have been also proposed (Giacinto & Roli, 2001a, 2001b; Roli & Giacinto,

2002). Comparison of the classifier combination approach to other approaches to classifier design such as Support Vector Machines (SVM) and consideration of their pros and cons is presented in (Kittler, 2000).

### **Summary of classifiers**

Although a number of classifiers have been proposed and this area is an active topic of research, it seems that the biggest difference in the performance of a pattern recognition system is introduced by choice of appropriate features and their successful extraction. Indeed, with a good set of features almost any classifier is suitable, and with a poor feature set even the most appropriate classifier (or their combination) will give an average performance (Pavlidis, 2003). In the author's opinion, the choice of classifier is important, but only after the selection of feature is exhausted and no big improvement seems to be possible there. As the current project is focused on features and their usefulness relative to each other, the choice of classifier is only considered to ensure that the classifier is "good enough".

## **2.8 Concept of feature usefulness**

A traditional approach to formation of feature sets in pattern recognition problems is to extract all features that might be useful and then select a smaller subset of them by estimating usefulness of the features. John et al. (1994) analysed several definitions of feature relevance which have been presented in the literature and proposed a definition that includes two degrees of relevance: strong and weak relevance. Strong relevance means that a feature cannot be removed from the feature set without loss of classification accuracy. Weak relevance means that a feature can sometimes contribute to classification accuracy. A feature is irrelevant if it is neither strongly nor weakly relevant, and thus can be excluded from the feature set without loss of classification accuracy.

Proper choice of features leads to reduced learning time and a number of samples necessary for the learning process, and can also result in higher classification accuracy. Simple approaches like Fisher criteria, ANOVA statistics, or information theory based measures allow us to find some of useless features. To reduce the feature set further it is necessary to perform exhaustive search of all possible feature subsets that can be formed from the original feature set. As this method is infeasible for most problems, other methods have been proposed that can be divided into two groups: filter and wrapper methods.

### 2.8.1 Feature selection by filter approach

The filter approach includes the methods that perform feature selection independently of the learning algorithm. It is a popular way of selecting a good feature subset mostly because of its computational speed (Setiono & Liu, 1997; Novovičová et al., 1996; Almuallim & Dietterich, 1991). Most current filter methods assume monotonicity of some estimate of feature subset performance: it is assumed that if removal of either feature  $f_1$  or feature  $f_2$  does not worsen the performance estimate then removal of both  $f_1$  and  $f_2$  will not worsen the classification performance either. However, this is often not the case. To eliminate this drawback, for example, sequential floating search methods have been proposed (Jain & Zongker, 1997; Somol et al., 1999).

### 2.8.2 Feature selection by wrapper approach

The wrapper approach includes the methods that select features using the classification accuracy of the learning algorithm as an indicator of how good a feature subset is. Exhaustive search is the simplest example of such methods. Due to computational complexity of the approach, other methods based on some heuristics have been proposed (Brill et al., 1992; Nunes et al., 2004). Application of genetic algorithms has been tried (Vafaie & De Jong, 1992) and become one of the most popular methods of feature selection using the wrapper approach (S. Chen et al., 1999; Kuncheva & Jain, 1999; Yang & Honavar, 1998b). Indeed, genetic algorithms are an attractive choice of heuristics for wrapper methods because of a number of characteristics: they use only fitness function and not its derivatives, they use a population of candidate solutions, they incorporate random walk into search by using probabilistic transition rules, and they are known to be robust to parameter selection: non-optimal choice of parameters of a GA will most likely still allow us to find a solution although after a greater number of iterations (Goldberg, 1989). Moreover, representation of feature subsets as binary strings is straightforward. It is also possible to find several different optimal or near-optimal feature sets by using GA with sharing (Deb & Goldberg, 1989; Miller & Shaw, 1996). This method is used in the current project.

Depending on the problem in hand, selection of a feature subset may be needed that is optimal in the sense of several criterion, for example, small number of features and high classification accuracy. The problem of feature selection thus becomes a multi-objective optimisation problem and a special class of genetic algorithms (Multi-objective GA, MGA) have been developed for such problems (Schaffer, 1985). Exam-

ples of successful application of MGA to the problem of feature selection have been reported (Oliveira et al., 2002; Oliveira & Sabourin, 2003).

## **2.9 Conclusions and identification of research issues**

Figure 2.13 shows the areas and sub-areas of research that are related to the problem of establishing a scientific basis for forensic document examination based on the overview above. Terminal boxes are marked by gray level according to the amount of research the area has received (Figure 2.13(c)).

There has been much progress in the validation of handwriting individuality hypothesis using mainly computational features. Although some questions like influence of age on handwriting remains open, it is fairly conclusively established that handwriting can be used to distinguish writers with high degree of accuracy.

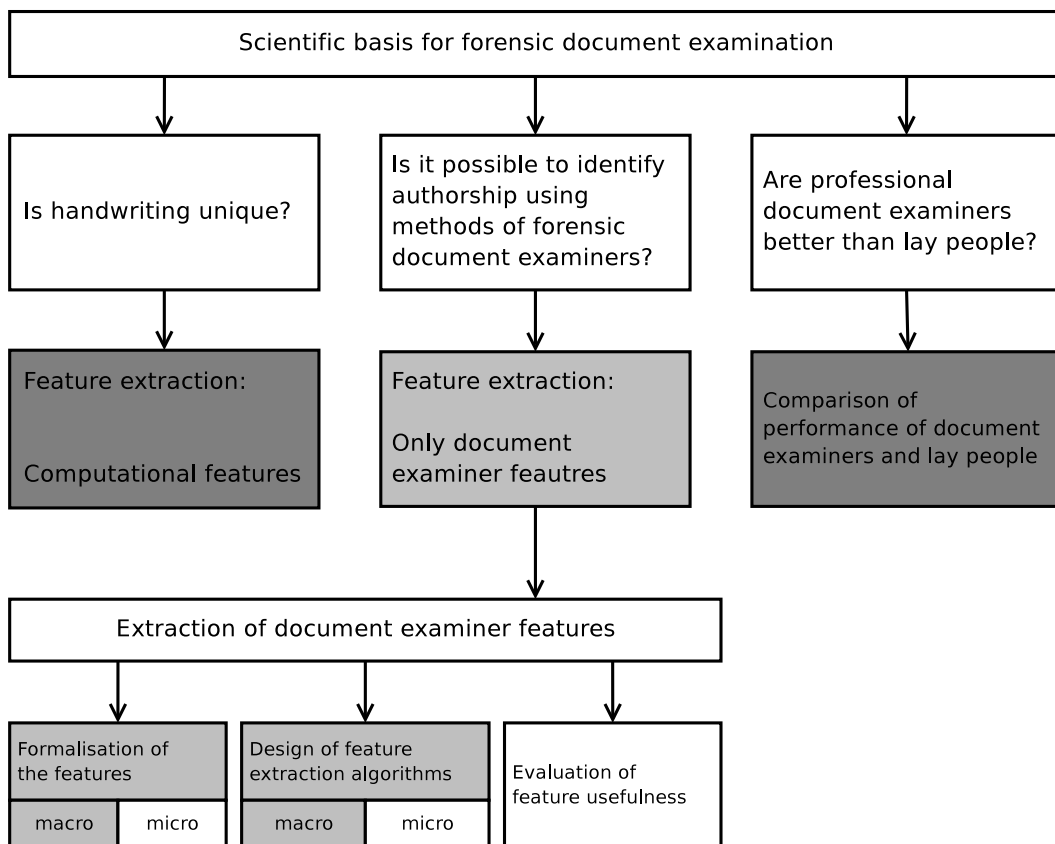
Validation of methods of forensic document examiners has received little attention yet. Formalisation and extraction of document examiner features has been performed for a few macro features and some micro features like character size, but most features from the list of 21 discriminating elements of handwriting have not been formalised and measured.

Image processing tools have been developed that allow both automatic and interactive work with a document image and produce good results for noise removal and background separation. Automatic image segmentation tools that produce consistently accurate results for lines that do not overlap much are developed and studies of separation of overlapping lines is going on; few studies report tools for automatic word segmentation too. Character segmentation is available only in the form of computer-assisted manual tools. Some forms of visual comparison of handwriting images are available in many systems but manipulation of handwriting such as alteration of slant and baseline is only available in a few systems and still requires more research. This research is related to extraction of document examiner features since it is the difference in those features resulting from alterations of handwriting that is of interest for forensic document examiners.

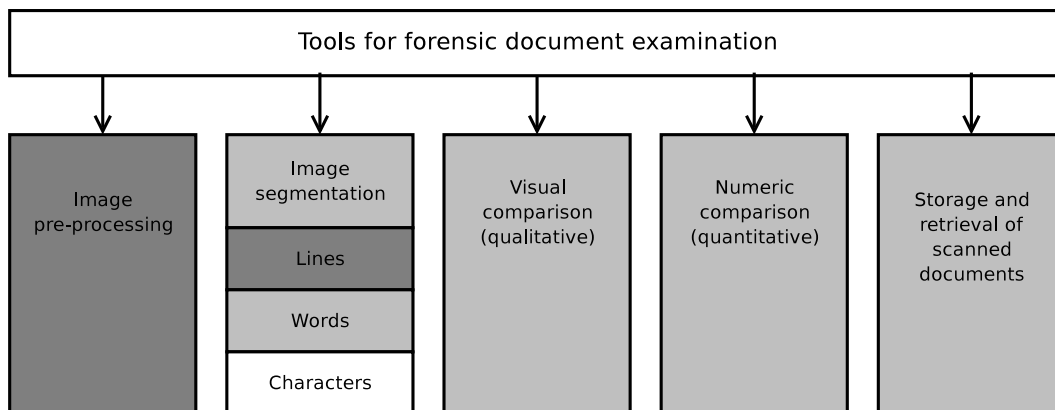
Storage and retrieval of documents for forensic analysis of handwriting has been implemented in several systems (CEDAR-FOX, FISH, WANDA) and tools are available that allow the databases to be searched. Again, formalisation and extraction of document examiner features may enhance the search interface since queries can be

made using similarities of certain features.

Summarising all said above, it seems that at this conjecture research into the document examiner features has a high priority since (i) it is needed to validate methods of forensic document examination, (ii) it is necessary to make measurement and comparison of document examiner features possible and hence usable in interactive tools. It is the gap in document examiner feature formalisation, extraction, and evaluation that the current research is aimed to lessen.

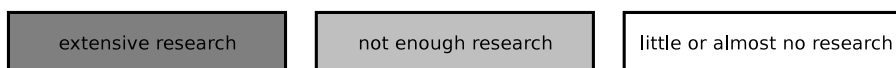


(a) Establishing scientific basis to forensic document analysis.



(b) Tools for computer-assisted forensic document analysis.

Areas that have received



(c) Legend.

Figure 2.13: Taxonomy of the research areas related to forensic document analysis.

## Chapter 3

# Choice of characters and features for study

In this chapter the rationale behind the selection of certain characters and graphemes as well as their features for study is presented.

### 3.1 Plan of the study

The purpose of the study is to determine whether it is possible to distinguish people by their handwriting using a subset of document examiner features. It seems reasonable to assume that an experienced person, that is, an expert in forensic document analysis can (i) effectively utilise more features than our system does, for it is very hard to express in strict mathematical terms many of the document examiner features, and (ii) the person can determine which features should, and which should not, be used in a particular case (that is, having looked at handwritten samples, an expert is able to select only the important features for handwriting comparison). As a consequence, an expert can distinguish writers or establish authorship of questioned documents better on average, than an automatic system.

The research was also aimed at determining a lower bound on the accuracy of writer discrimination when only document examiner features are used (it is not a lower bound in strict terms, because the two assumptions above, although seemingly reasonable, are difficult to verify). Only *unconstrained genuine handwriting* was considered in the study. The problem of authorship identification in court cases usually involves forged and disguised handwriting, however, such documents have not been considered for two reasons. The main reason is that before studying complicated cases of deliberately changed handwriting it is important to study the general case of people's normal handwriting. If the results achieved from the study are negative, that is, the considered features cannot help to distinguish writers, there is no point

in applying the same approach to the more complicated cases. The other reason is a lack of data on forged and disguised handwriting available for such studies.

It was decided to formalise some of the features from “21 discriminating elements” (Huber & Headrick, 1999), learn how to measure those features and how to extract them automatically, and evaluate their discriminating power by building a classifier. The following experimental plan was proposed.

1. Select features to be extracted.
2. Develop the rules how to measure the feature values (formalisation).
3. Develop automatic extraction algorithms.
4. Evaluate the feature extraction accuracy.
  - (a) Determine the main causes of errors.
  - (b) If necessary, go back to item 3 to improve the extraction algorithms.
5. Use ANOVA<sup>1</sup> to filter out some features with little or no discriminating power.
6. Build a classifier.
7. Evaluate usefulness of the features and writer identification accuracy.
8. Determine whether low usefulness of some features can be caused by inadequate formalisation. Try a different formalisation, if necessary, and repeat the study from item 3.

## 3.2 Handwriting sample database

Since shape of a character is usually affected by the preceding and following characters as well as by writing conditions such as type of pen and paper, constrains, etc. it is desirable to extract features of handwriting from samples written in similar conditions and having similar content so that the influence of those factors is minimised.

Handwriting samples of the *CEDAR letter* are suitable for this kind of study. The text of the letter was specially designed in the Center of Excellence for Document Analysis and Recognition at the State University of New-York at Buffalo for research purposes and contains each character at the beginning of a word as a capital and a small letter, and as a small letter in the middle and at the end of a word. In addition it also contains numerals, punctuation and various frequent letter combinations (graphemes) (Cha, 2001). The writers were asked to provide three samples of handwritten copies of the given text using the given pen and paper. Thus all samples were written on the same paper with similar pens and without constraints. The complete

---

<sup>1</sup>One-way analysis of variance.

text of the letter can be found in Appendix A and an example of written sample is shown in Appendix B. The database is representative of the US population, that is, it contains approximately the same proportion of samples from male and female writers, right- and left-handed people, etc. as in the United States. Since such factors as handedness, age, gender, etc. are thought to possibly influence handwriting, this feature of the database is important to make results of the research statistically sound and avoid using convenience sample<sup>2</sup>. The CEDAR database contains samples collected from 1568 writers.

The database used in the current study consists of 8-bit grayscale images of 600 samples from 200 writers scanned at 300 dpi and separated from the background. It is a part of the CEDAR handwriting database, and was kindly provided for this study by CEDAR.

### 3.3 Selection of characters and graphemes

According to Srihari, Cha, and Lee (2001) there are two categories of features classified by the extraction level: macro features (up to the word level) and micro features (character level). The current study is focused on micro features, because *(i)* they are thought to be more endowed with individual traits and are thought to be harder to change under attempt of forgery or disguise, and *(ii)* extraction of document examiner macro features has been partially done in previous work (WIS, FOX, etc., see Section 2.4.2).

It was necessary to make a decision on three important issues. 1. Because the work is too large to undertake on all letters a subset had to be selected for feature extraction. 2. Having selected the letters to study it was necessary to select which features to extract from the chosen letters. 3. Resolve the issue of how to express the features in a numerical way. The first and the second issues are discussed in this chapter. The third issue is discussed in Chapter 5.

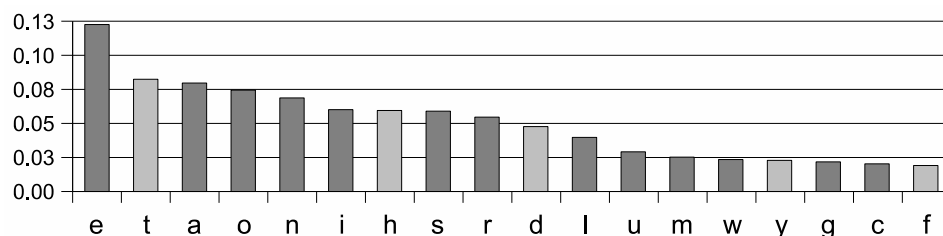
In order to decide which letters or letter combinations (graphemes) to use for micro feature extraction, several considerations were taken into account. *(i)* The letters and/or graphemes must be frequent enough so that a number of them can be found in most handwriting samples and hence it is possible to obtain the statistically reliable feature values. Each feature value is calculated as a simple mean of the values

---

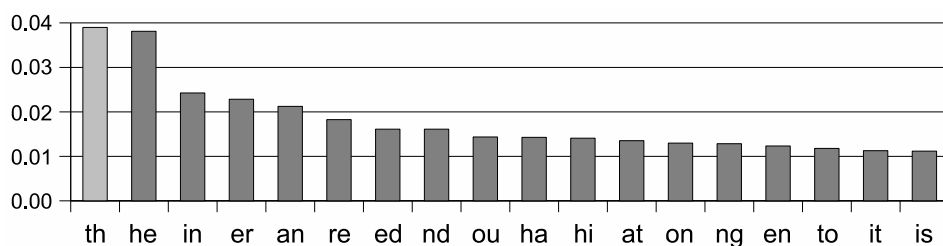
<sup>2</sup>That is, a set of samples may be drawn from a sub-population so that it possesses some characteristics that are pertinent to that sub-population but are not pertinent to the population in general. Example of this may be collecting handwriting samples only from left-handed people and extrapolating the results of analysis conducted on these samples to the whole population.

obtained from all the samples of the letter in a document. (ii) According to Eldridge et al. (1984) letters with ascenders or descenders are particularly useful for the purpose of writer identification. (iii) Besides, there are the existing techniques for extraction of some features which can be used. Based on the existing techniques ease of extraction of the potential features was assessed.

For the analysis of letter and grapheme frequencies several novels available in the public domain online libraries were used. The total number of words was about 220,000. The frequencies of occurrence were calculated for each letter as the number of occurrences of the letter normalised to the total number of letters. For grapheme analysis only two-letter combinations were considered. Figures 3.1(a) and 3.1(b) display the frequency of occurrences of most frequent letters and graphemes correspondingly.



(a) Normalised frequencies of character occurrences.



(b) Normalised frequencies of grapheme occurrences.

Figure 3.1: Frequencies of characters and graphemes. The characters and graphemes chosen are highlighted.

| Character / Grapheme | Percentage of occurrence |
|----------------------|--------------------------|
| “f”                  | 1.9%                     |
| “y”                  | 2.3%                     |
| “d”                  | 4.8%                     |
| “t”                  | 8.2%                     |
| “th”                 | 3.9%                     |

Table 3.1: Frequency of occurrence of some characters in an analysis of several literature works.

Based on these observations it was first decided to choose four letters for feature extraction: “d”, “y”, “f” and “t”. Subsequently letter “t” was changed to grapheme “th”, which is the most frequent grapheme in English texts.

Table 3.1 shows the percentage of occurrence of each of the three letters and one grapheme selected for the study. It should be emphasised that the frequency of the letter and grapheme occurrence is limited to English. In other languages (e.g. German) the distribution is different and other letters or graphemes may be preferred.

The samples of the characters and grapheme were extracted manually from CEDAR letter samples. In order to minimise the influence of the adjacent characters it was decided to extract samples of “d” and “y” only from the end positions as shown in Appendices A.1, A.2. Samples of grapheme “th” were extracted from the starting positions, namely from 8 words “the” and 1 word “through” as shown in Appendix A.4. Samples of characters “t” were extracted from the corresponding grapheme samples. All “f” characters were extracted since there were only 8 of them in the CEDAR letter (Appendix A.3).

### 3.4 Selection of features

Selection of features to extract from each character and grapheme was motivated by discussion of micro features in several books on forensic document analysis (Huber & Headrick, 1999; Harrison, 1981; Hilton, 1993). Particularly, the list of *21 discriminating elements of handwriting* was used as a reference. The list of features extracted in the study is shown in Tables 3.2–3.6. Each feature is denoted as  $f_i$ , where  $i$  is the unique feature index. The total number of features extracted from the 4 characters and 1 grapheme was 67, but not all of them were extracted at once: initially only 40 features from the 4 characters were extracted and then some features were filtered out and some added. Extraction of the grapheme features was added later too. Some of the features of grapheme “th” could not be extracted reliably when thinning-based skeletonisation was used ( $f_{57} \dots f_{67}$ ). More details about which features were extracted at which stage of the research can be found in Section 5.9.2.

An important question arises at this point: how adequate are the feature extraction methods developed in this research for the purpose of the study? To answer this question, two types of features need to be clearly distinguished here. One type of features is the original document examiner features from the *21 discriminating element of handwriting* (“type 1”). The problem with those features is their quite vague definition. As long as it is unclear what the feature really is, it is not possible to extract it objectively correctly. The other type of features (“type 2”) is those features that were actually studied. These features were defined so that their extraction was

straightforward from their definition (Table 3.2–3.6).

The purpose of the feature extraction was, basically, to mimic the extraction of features as performed by document examiners, that is, to extract and study type 1 features from the list of *21 discriminating element of handwriting*. The mapping from features of type 1 to features of type 2 was the formalisation of the document examiner features. The question “how suited the extraction methods of type 2 features were for extraction of type 1 features” is, hence, equivalent to question “how to evaluate the adequateness of the formalisation.” It is not clear how to make such an evaluation in general. For some features, like dimensional features, the formalised features are exactly the same as the originals, so the answer is obvious. For others, like line quality, I believe, it is not possible to give the answer at all because the original feature is not clearly defined. How suitable is the set of features “straightness of t-bar, t-stem, and h-stem” to describe “quality of line”? In order to answer this question one needs to understand what “quality of line” actually means, and that is not at all clear. If it were clear, the formalisation itself would be unnecessary.

| Features extracted from character “d” |       |
|---------------------------------------|-------|
| Height                                | $f_1$ |
| Width                                 | $f_2$ |
| Height to width ratio                 | $f_3$ |
| Relative height of ascender           | $f_4$ |
| Slant of ascender                     | $f_5$ |
| Final stroke angle                    | $f_6$ |
| Fissure angle                         | $f_7$ |

Table 3.2: “d”.

The feature set consists of geometrical characteristics of characters, various angular measures, loop characteristics, and stroke features. There are several binary features which denote presence or absence of some elements, like t-bar and loops. Such features are assigned the value of 1 if the corresponding element is found in the character image and the value of 0 otherwise. *Presence of point D* in “y” character corresponds to the intersection of the descender and the base as in Figure 3.2(d). *Position of t-bar* feature of grapheme “th” is binary too: it is equal to 1 when the t-bar is crossing the stem and 0 in the cases of touching, detached, or absent t-bar. It was observed that writers who tend to produce t-bars touching the stem in grapheme “th” also tend to produce disconnected t-bars in that same grapheme and vice versa.

Characters “d” and “y” were divided into two parts: the base and the ascender (descender). The border between the parts was defined by the upper point of the loop

forming the base part of “d” and the lower point of the base part of “y” correspondingly. When no clear base part could be found, the relative height of the ascender (descender) became equal to 1. If the base part was detected, the feature value was calculated as  $a/f_1$  for character “d” and  $1 - d/f_8$  for character “y”.

For each character its *slant* was measured. For characters “d”, “h”, and “y” slant was defined as the slant of the ascender (descender). For characters “f”, “t” the slant was defined as the slant of the stem. Such definitions of slant seems reasonable as people usually make their judgement about the slants of these five characters by the slants of the long nearly-vertical strokes of those characters.

| Features extracted from character “y” |          |
|---------------------------------------|----------|
| Height                                | $f_8$    |
| Width                                 | $f_9$    |
| Height to width ratio                 | $f_{10}$ |
| Relative height of descender          | $f_{11}$ |
| Descender loop completeness           | $f_{12}$ |
| Completeness of loop at $Y_T$         | $f_{13}$ |
| Slant at $Y_T$                        | $f_{14}$ |
| Slant of descender                    | $f_{15}$ |
| Final stroke angle                    | $f_{16}$ |
| Presence of point $D$                 | $f_{17}$ |
| Descender loop length                 | $f_{18}$ |

Table 3.3: “y”.

| Features extracted from character “f” |          |
|---------------------------------------|----------|
| Height                                | $f_{19}$ |
| Width                                 | $f_{20}$ |
| Height to width ratio                 | $f_{21}$ |
| Slant                                 | $f_{22}$ |
| Number of loops                       | $f_{23}$ |
| Main loop area                        | $f_{24}$ |
| Main loop length                      | $f_{25}$ |
| Main loop slant                       | $f_{26}$ |
| Presence of loop at $F_T$             | $f_{27}$ |
| Presence of loop at $F_B$             | $f_{28}$ |

Table 3.4: “f”.

A *final stroke* was defined as the angle between the tangent at the end point of the stroke and a horizontal line. *Fissure angle* for character “d” ( $f_7$ ) was defined as the angle between the two tangents to the two strokes that form the fissure, as shown in Figure 3.2(a). *Slant at point  $Y_T$*  of character “y” was defined as the angle between the tangent line at point  $Y_T$  and a vertical line.

| Features extracted from character “t” |          |
|---------------------------------------|----------|
| Height                                | $f_{29}$ |
| Width                                 | $f_{30}$ |
| Height to width ratio                 | $f_{31}$ |
| Slant                                 | $f_{32}$ |
| Position of t-bar                     | $f_{33}$ |
| Bar width                             | $f_{34}$ |
| Left part of bar                      | $f_{35}$ |
| Right part of bar                     | $f_{36}$ |
| Top part of stem                      | $f_{37}$ |
| Bottom part of stem                   | $f_{38}$ |
| Number of loops                       | $f_{39}$ |
| Main loop area                        | $f_{40}$ |
| Main loop length                      | $f_{41}$ |
| Main loop slant                       | $f_{42}$ |

Table 3.5: “t”.

| Features extracted from grapheme “th”                          |          |
|--|----------|
| Height   | $f_{43}$ |
| Width  | $f_{44}$ |
| Height to width ratio  | $f_{45}$ |
| Distance $HC$  | $f_{46}$ |
| Distance $TC$  | $f_{47}$ |
| Distance $TH$  | $f_{48}$ |
| Angle between $TH$ and $TC$                                    | $f_{49}$ |
| Slant of t   | $f_{50}$ |
| Slant of h   | $f_{51}$ |
| Position of t-bar  | $f_{52}$ |
| Connected / disconnected t and h                               | $f_{53}$ |
| Average stroke width   | $f_{54}$ |
| Average pseudo-pressure  | $f_{55}$ |
| Standard deviation of pseudo-pressure                          | $f_{56}$ |
| <i>features below were extracted from vector skeleton only</i> |          |
| Standard deviation of stroke width                             | $f_{57}$ |
| Number of strokes  | $f_{58}$ |
| Number of loops and retraced strokes                           | $f_{59}$ |
| Straightness of t-stem   | $f_{60}$ |
| Straightness of t-bar  | $f_{61}$ |
| Straightness of h-stem   | $f_{62}$ |
| Presence of loop at top of t-stem                              | $f_{63}$ |
| Presence of loop at top of h-stem                              | $f_{64}$ |
| Maximum curvature of h-knee                                    | $f_{65}$ |
| Average curvature of h-knee                                    | $f_{66}$ |
| Relative size (diameter) of h-knee                             | $f_{67}$ |

Table 3.6: “th”.

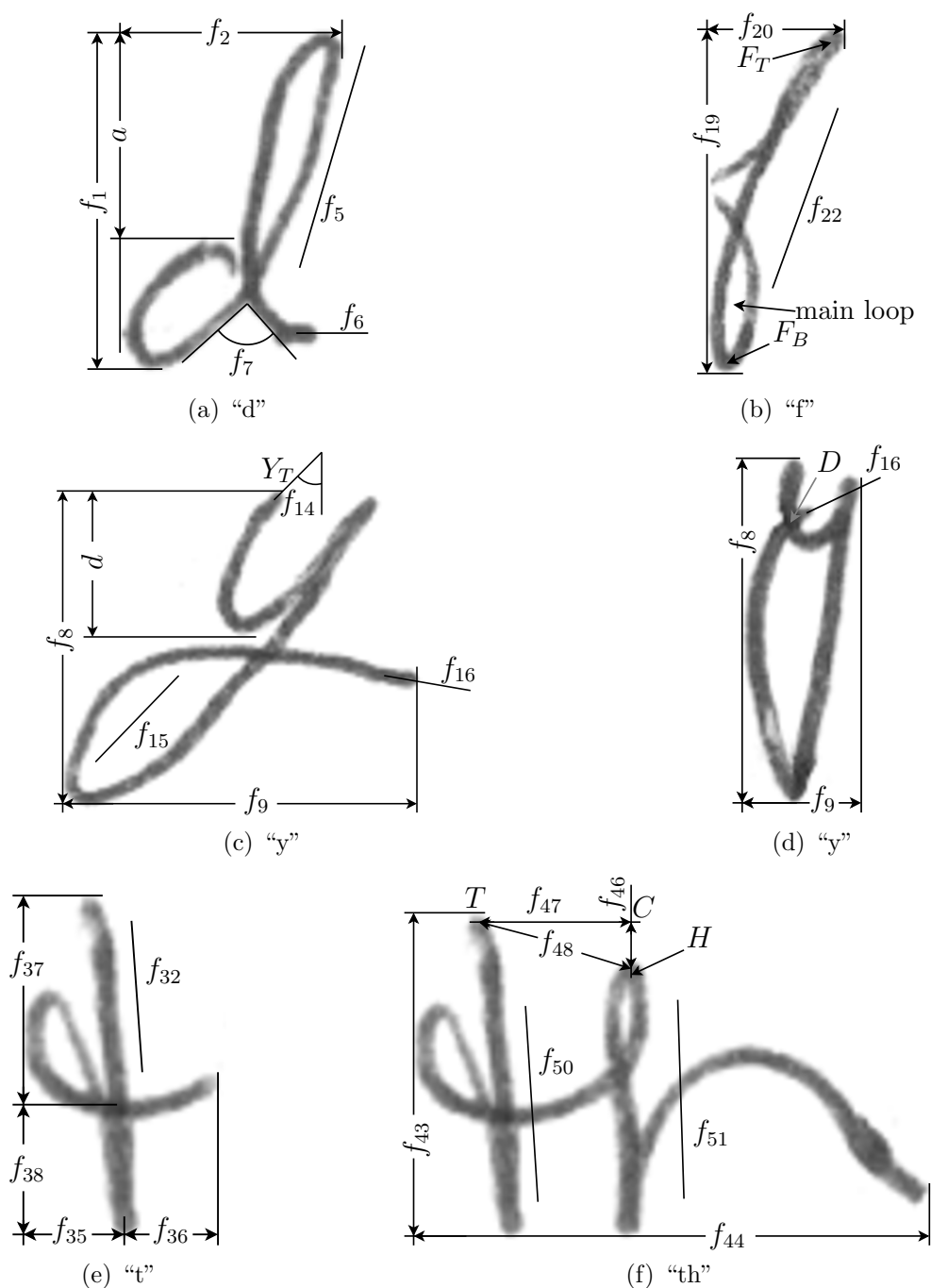


Figure 3.2: Some of the features extracted from the four characters.

*Loop area* was approximated by the number of pixels inside the loop, and *loop length* was approximated by the number of border pixels. *Loop slant* was defined the same way as in the FOX system (Solihin, 1997). *Loop completeness* was defined through the length of the loop and the distance between the starting and ending points. If a loop is complete, the latter is 0.

*Length of descender loop* of character "y" was calculated as the distance between the descender self-intersection point and the bottommost point of the descender.

*Average stroke width* was calculated in two ways: via the number of border pixels

and the total number of black pixels in a binarised image and via tracing the strokes with a small step and evaluating the stroke width at each sample point using the distance map of the binarised image. The latter method could be used only with a vector skeleton. Although the difference in the resulting feature values was insignificant, the latter method also enabled extraction of *the standard deviation of the stroke width* ( $f_{57}$ ). *Pseudo-pressure* was calculated by averaging the gray values of all the pixels that form the strokes.

*Straightness of t-stem* and *h-stem* (ascender) were defined as the ratio of the length of the curve to the distance between its end points. *Maximum* and *average curvatures of h-knee* as well as *relative size (diameter) of h-knee* were calculated in a straightforward way from the B-spline representation of the corresponding stroke.

Definitions of other features can be derived from Figures 3.2(c)–3.2(f).

## Chapter 4

# Image preprocessing and extraction of character skeleton

Several image preprocessing steps are necessary for feature extraction. The extraction engine requires the grayscale image of a character or grapheme sample, the binarised version of this image, and the skeleton of it. In addition, to extract variation of the stroke width feature from images grapheme “th”, their distance maps are necessary, which can be easily obtained from the binarised images.

This chapter describes the preprocessing steps: manual segmentation of handwriting sample images to extract the characters and grapheme of interest, binarisation of the resulting images, and extraction of skeletons. The latter is discussed in detail here as it is the most important preprocessing step, and the feature extraction algorithms highly depend on it. The algorithm described in this chapter has been published in (Pervouchine et al., 2005a, 2005b).

In handwriting recognition the features are aimed at distinguishing different characters. The features need to emphasise the dissimilarities between different characters and be insensitive to the writer-related differences in the same characters. In contrast, for writer identification the features of interest need to emphasise the differences in shapes of the same characters written by different writers. Hence, the preprocessing algorithms need to work so that the individual traits a writer endows to characters are preserved. For example, to extract final stroke angle as defined in Section 3.4 correctly from a skeletonised image of a character it is necessary that the final stroke of the skeleton has the same tangent in its end point as the original stroke. If some features are extracted by making use of junction points then the junction points must not change their positions due to the preprocessing. In quantitative terms this means that the values of document examiner features are not changed significantly due to the preprocessing steps.

## 4.1 Segmentation and binarisation

The images of characters “d”, “y”, “f”, and grapheme “th” were extracted manually from the CEDAR letter images. The extraction positions are shown in Appendices A.1–A.4.

Characters “d” and “y” were extracted from the end positions so that there is no following character to affect their final strokes, ascender/descender, and fissure angle of “d”. When a character was not attached to its preceding neighbour the segmentation was straightforward. Otherwise the segmentation point for character “d” was chosen so that either its base part or its ascender became separated from the connecting ligature as shown in Figure 4.1(a). For character “y” the segmentation point was chosen so that either the connecting ligature was eliminated entirely (Figure 4.1(b)) or only the part of the ligature that forms a loop, hidden loop, or retracing with the left stroke of the base part of “y” was preserved (Figure 4.1(c)). Such a segmentation method was aimed at the base part shape preservation of characters “d” and “y”.

Grapheme “th” was extracted from words “*the*” and “*through*”. Segmentation point was chosen somewhere on the final stroke of character “h”, usually in its lowest point, as shown in Figure 4.1(d). Because of such segmentation, the final stroke feature value was not reliable and hence was not measured for grapheme “th”.

Character “t” was segmented from samples of grapheme “th” by separating the two characters in the point where the ligature touches the h-stem. The ligature was considered to be a part of “t” (Figure 4.1(d)).

All samples of character “f” were extracted because there were only 8 of them. In cases of a single “f” surrounded by one or two neighbour characters the segmentation point was chosen to be the point where the connecting ligature touches the neighbour character base as in Figure 4.1(f). When such a base was not distinct (combination “*ef*” in word “*referred*” and “*lf*” in “*halfway*”), the lowest point of the ligature was chosen as a segmentation point as in Figure 4.1(e). Separation of a two-character sample from word “*affected*” was done by choosing the segmentation point approximately in the middle of the ligature as shown in Figure 4.1(f).

The entire segmentation workflow is shown in the diagram in Figure 4.2.

Binarisation of the images was simple since the background had been already separated on the CEDAR samples. A constant threshold of 250 was used: a pixel was marked as white if its value exceeded the threshold. The small allowance of 4 gray levels (254...251) was given because on some images the surrounding strokes from

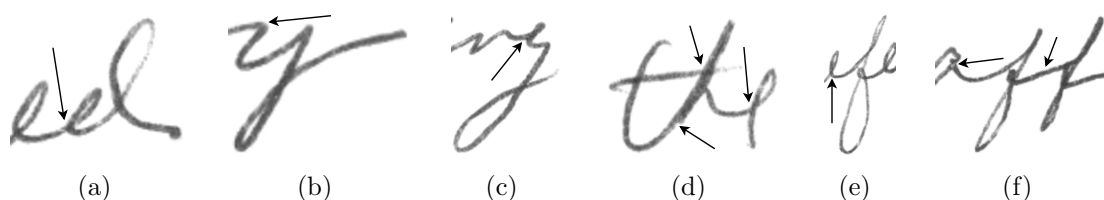


Figure 4.1: Segmentation points (marked by arrows).

other characters were not erased completely.

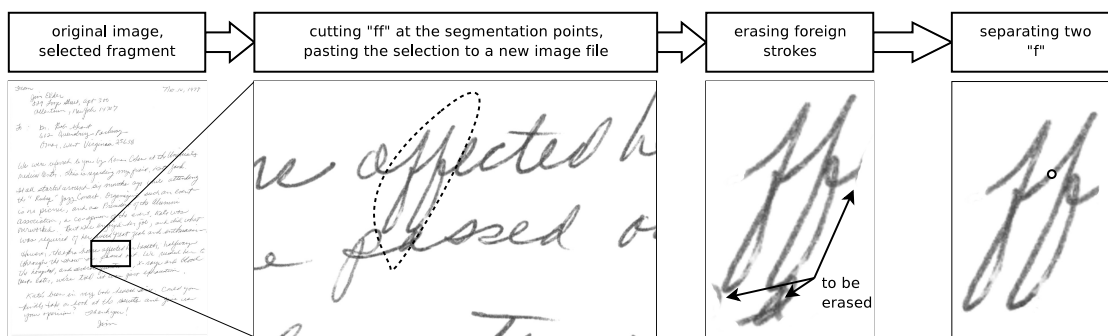


Figure 4.2: Character image segmentation workflow.

## 4.2 Skeleton extraction

In the initial experiments with document examiner feature extraction a skeletonisation method based on thinning was used. The resulting skeleton was a raster image and each junction point of the original handwriting image was normally represented by two close junction points. Positions of some branches were changed too. In order to eliminate some artefacts produced by thinning a correcting algorithm was used. In order to compensate for changes in junction points and branch positions, feature extraction algorithms were designed that took such changes into account. Section 4.2.1 describes the thinning-based skeletonisation in detail.

Later a new skeletonisation algorithm was developed. The algorithm was able to preserve the original junction points and approximate the original handwriting strokes with smooth B-spline curves. The resulting skeleton was in a vector form (a set of B-splines). The feature extraction algorithms were changed so that they could extract features from the new skeletons. The skeletonisation algorithm is described in detail in Section 4.2.2 and in papers (Pervouchine et al., 2005a, 2005b).

### 4.2.1 Thinning-based skeletonisation

The thinning-based skeletonisation method was developed on the base of the thinning function provided in *Matlab Image Processing Toolbox* (Guo & Hall, 1989), which is a modified version of Zhang and Suen thinning algorithm (T. Y. Zhang & Suen, 1984). Other thinning methods were tried too (Suen & Wang, 1994) but no significant difference was found for the problem at hand. The method requires a binary image for input and produces a 1-pixel thick image by erosion of the outer black pixels that satisfy the pixel removal conditions. First thinning algorithm was applied to a character image and then correction of some artefacts produced by thinning was performed as shown in the pseudo-code below:

```
do {  
    remove small connected components  
    find junction points  
    find end points  
    correct spurious loops  
    prune short branches  
} while there are some changes in the skeleton image
```

Figure 4.3(c) shows some artefacts that were removed: (1) small connected components, (2) spurious loops, (3) and extra branches. Removal of small connected components was necessary because of noise in the image and because binarisation sometimes produced small disconnected parts at the ends of handwriting strokes. Removal of spurious loops was accomplished by analysis of each loop with the area smaller than a pre-defined threshold and removal of the loops that resulted from small white “holes” in the binarised image as in Figure 4.3(c), arrow 2. Unfortunately not all such loops could be removed correctly. Short branches that did not exist on the original image were pruned if their length was less than a pre-defined threshold. All these operations could make some connected components that had been too big to be removed before the operations, small enough to be removed after them. Hence the operations were repeated until one application of them did not result in any changes in the skeleton. Figures 4.3(a)–4.3(d) show the original image and the resulting image after each stage.

### 4.2.2 Vector skeletonisation

The ideal skeleton of images of handwriting, which is suitable for extracting information about the writer, is the curve or set of curves that model the original trajectory of the pen tip. The ideal skeleton should store the sequence in which the strokes

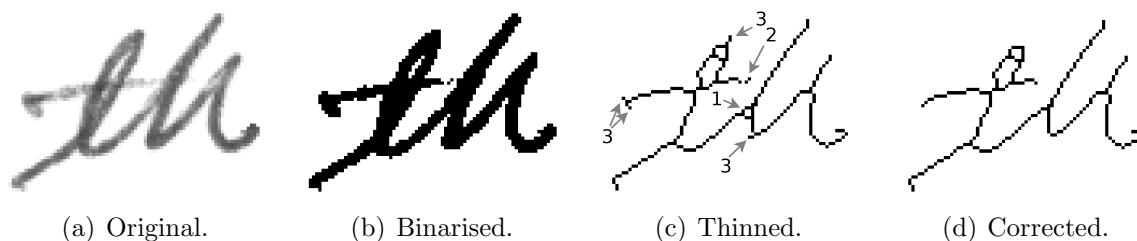


Figure 4.3: Stages of thinning-based skeletonisation.

were produced as well. It is assumed that the input image is grayscale scanned at the resolution most commonly used in handwriting processing (150–600 dpi). It is also assumed that no background noise is present in the images, or it is easily removable. The approximation to the ideal skeleton developed in this study does not contain the sequence of the original strokes. For the document examiner features considered in this study the true sequence of strokes is not necessary. Although recovery of the stroke sequence would enable extraction of more features, it is not clear how to perform it and whether it can be performed at all.

Below two important issues are discussed in detail: use of grayscale image rather than binarised image for skeletonisation and vector representation of a skeleton.

### Binary vs. grayscale image

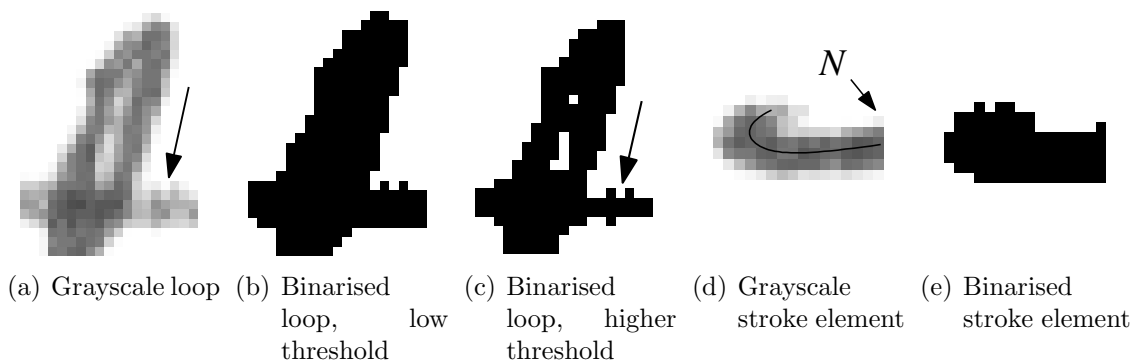


Figure 4.4: Loss of pen tip trajectory information due to binarisation.

Most methods of skeletonisation and stroke extraction make use of binarised images of handwriting samples. Such an approach has some benefits, like strict division of an image into handwriting (foreground) and background, which allows use of well-developed thinning algorithms and eases the image processing at other stages. However, binarisation of images has many drawbacks too, the main one being the loss of information due to noise. No matter how well the binarisation threshold is chosen, some details like small loops at junction points and hooks at the ends of

strokes become lost in the binary image although they can clearly be seen in the grayscale image. Figure 4.4 demonstrates this effect. Figures 4.4(b) and 4.4(c) show that changing the threshold value in an attempt to save loop forms becoming lost results in distortion of other parts of the image like strokes (marked with an arrow). As it is impossible to detect strokes before thresholding (if it were possible than the problem of skeletonisation would be solved making thresholding unnecessary) it is not possible to make use of adaptive thresholding techniques either. In addition, when a person looks at a grayscale image of a handwritten character they can easily restore the original pen tip trajectory with high precision (Plamondon et al., 1994). This is not always the case for a binary image (Figure 4.4(d), 4.4(e)). It seems reasonable to assume that a computer program cannot outperform humans in this task and restore the lost information from a binary image.

In both cases loss of information is caused by boundary pixels. There is noise introduced by digitisation of an image caused by the finite spatial resolution of the scanner. It causes the boundary of handwriting to consist of distinct elements — pixels, and, hence, be non-smooth. As seen from Figures 4.4(d), the colour (gray level) of the pixels that contribute to the noise more than to the useful information is very close to white (e.g. pixel  $N$  in the figure), whereas dark pixels carry mostly useful information. This consideration suggests to use the gray level of a pixel as its level of significance when restoring the pen tip trajectory. The proposed skeletonisation method works on grayscale images and uses gray levels of pixels in the described manner.

### **Raster vs. vector skeleton**

There are two ways to represent a skeleton. The raster representation is the most common one when dealing with handwriting images. A raster skeleton is basically a binary image of the same size as the original, in which pixels of one colour represent the background and pixels of the other colour represent the branches of the skeleton, which are usually (but not necessarily) 1-pixel wide. The vector representation of skeleton consists of a set of curves, which in turn can be represented as lines (Kondo, 1989), Bézier curves (Liao & Huang, 1990), B-splines (Nakajima, Mori, Takegami, & Sato, 1999), etc.

Use of raster skeleton in the early experiments on document examiner feature extraction revealed some drawbacks of the raster representation. One of the main drawback is low accuracy of calculation of curvatures and tangents of skeletal branches

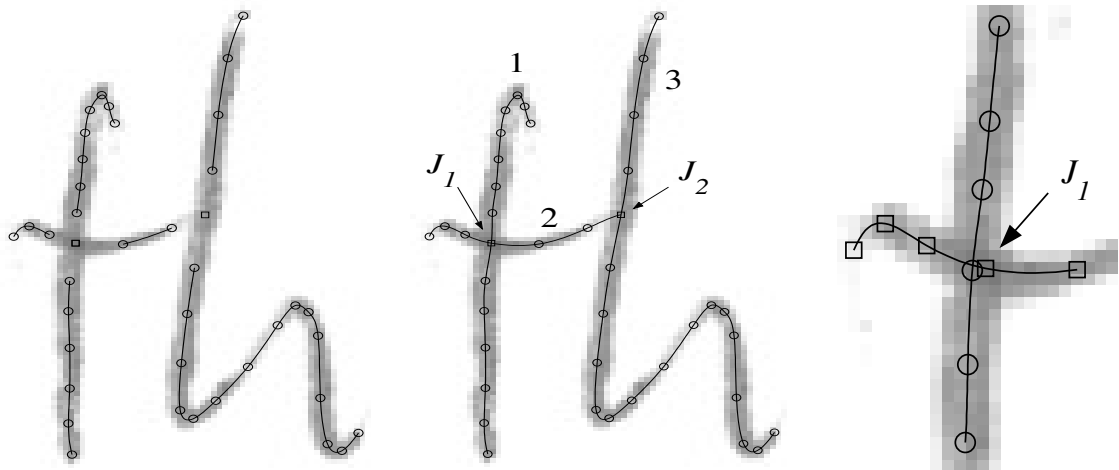
caused by discrete representation. The existing methods of calculation of digital curve differential geometry parameters (Fischler & Wolf, 1994; Rattarangsi & Chin, 1992; Worring & Smeulders, 1993; Found et al., 1997) work well when the purpose is to find the points of extreme curvature. When the task is to calculate the actual values of curvature at skeleton points as well as tangent angles, these methods give results which are often far from what humans perceive when looking at the image. Another source of errors in feature extraction arises from the constraints imposed on positions of skeleton points in the raster skeleton representation. Coordinates of any pixel are integer numbers; hence the positions of all points of a skeleton are always determined within 1-pixel error. For an image scanned at 300 dpi the typical width of a stroke is around 6 pixels, even less at the ends. Thus the relative error in position of skeleton points is about 20% and for some strokes can be even higher. Discreteness of skeleton representation also results in imprecise junction points — an angle between any two lines at a junction point is always a multiply of  $\pi/4$ . The vector representation of a skeleton is free of these drawbacks.

### **The model**

Taking into account the issues discussed above, it was decided to develop a skeletonisation algorithm according to the following requirements.

- Skeleton branches should consist of basic points interpolated with cubic B-splines (vector representation). The choice of cubic splines is just a preference here; other representations may also be possible.
- The basic points of the skeleton should be found directly from the source grayscale image of a handwritten sample. Gray level of pixels should be considered as the level of their significance in terms of contribution to the useful information.
- The whole skeleton should be represented as a set of branches with connections between them.
- Information about retracing of some branches should be stored.
- The correct restoration of stroke order is desirable but not necessary.

Such a representation should not only result in improvement of accuracy of extraction of the currently used document examiner features but also allow the extraction of more potentially important features, the extraction of which was impossible because of incorrect representation of junction points, generation of extra branches and junction points, and inaccurate representation of skeletal branches.



(a) Stage 1: Spline knots are marked as circles outside junctions and as boxes in junctions; those in junctions are shared among all the incoming branches.  
 (b) Stage 2: Branches have been merged. Spline points in junctions are still the same for branches 1 and 2 in  $J_1$ , and for branches 2 and 3 in  $J_2$ .  
 (c) Stage 3: After fine tuning of near-junction spline points. Points of branch 1 are marked with circles and points of branch 2 with boxes.

Figure 4.5: Three stages of skeletonisation. The image of ‘th’ is faded to make skeleton line more visible.

The proposed method of handwritten element skeletonisation consists of three stages. At the first stage, called vectorisation, the positions of points of B-splines are defined and the junction points are marked. The positions of spline points near junctions and in junctions may not be accurate at this conjecture (Figure 4.5(a)). The second stage merges the skeletal branches obtained in the first stage in an attempt to restore the correct connections between branches in junction points as well as restore hidden loops from retraced segments. A global optimisation technique is used for this purpose. After the two stages the skeleton branches in the vicinity of junction points are still far from the actual pen trajectory in that area (Figure 4.5(b)). The third stage, the fine tuning of skeletal curves, is aimed at adjustment of positions of spline points near junctions (Figure 4.5(c)). The program was written in Matlab. All the three stages are described in detail below.

### Stage 1: Vectorisation

The basic idea of the method is to divide the gray strokes into a set of approximately rectangular segments and define spline knots as the centres of the segments. It has been observed that there is no significant information loss when a skeleton is represented as a B-spline with knots at a distance of about the widths of a stroke. Such representation is also not sensitive to the noise caused by discreteness of an image (Melikhov et al., 2004).

Two vectorisation methods were tried. The first method determines an approximate tangent to the edge. This tangent becomes one side of the rectangle whose centre is a spline knot. Suppose the set of all foreground pixels of a stroke be  $S = \{s_n\}$  and the set of the border pixels be  $D = \delta S$ . Approximate width of a stroke is computed as  $w = 2N_S/N_D$ , where  $N_S$  is the amount of pixels in  $S$  and  $N_D$  is the amount of pixels in  $D$ .

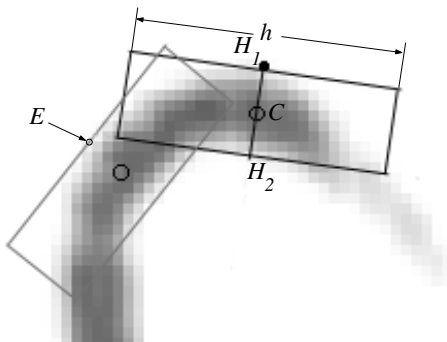


Figure 4.6: Calculation of spline points, first method.

The algorithm starts with point  $H_1$ ,  $H_1 \in D$  as in Figure 4.6. The tangent angle is defined from the nearest pixels in  $D$  on the distance of  $w$  from  $H_1$ . Although such detection does not give the exact value of tangent angle, the algorithm is not sensitive to the error. Point  $H_2$  is defined from the second intersection of a line orthogonal to the tangent at  $H_1$  with border  $D$  (the opposite side of the stroke). Point  $C$  is defined as the centre of mass of line segment  $H_1H_2$ , mass being the intensity of the pixels belonging to the segment. A rectangle at  $H_1H_2$  is initially defined to be of width  $h = 3w$ . After that the width is adjusted so that the error

$$E = \sum_{r_n} I(r_n) \quad (4.1)$$

is decreased until its value is not more than  $2w$ . Here  $R = \{r_n\}$  is the set of all pixel inside the rectangle,  $I(r_n)$  is the intensity of pixel  $r_n$ , black pixel has the intensity of 0 and white has the intensity of 1. Thus the error represents the difference between the stroke segment and the current rectangle. The error is smaller when the rectangle overlaps the bigger stroke area. When the rectangle is determined, the stroke pixels inside are marked as checked, point  $C$  becomes the next spline point, and the next position of  $H_1$  is chosen as a point on the same side of  $D$  at a distance of approximately  $h/2$  from the current position of  $H_1$  (point  $E$  in Figure 4.6). In a case the new  $H_1$  appears in an already checked area, the corresponding point  $C_1$  of that area is linked with the current centre point  $C$ . Thus all the centre points become linked with their

neighbours. Finally when there are no unchecked areas left the junction points are defined as those having more than two neighbours while the end points are defined as those with only one neighbour. The B-spline interpolation is applied to each sequence of spline points between either ends or junctions or an end and a junction.

The second method of vectorisation was proposed because the first method gave too biased spline knots in some junctions on a number of samples. This method uses a square instead of an arbitrary rectangle and instead of determining the square angle from the tangent estimate and changing the size, the method searches for the largest square  $R(x, y, \alpha)$  with centre at point  $(x, y)$  and rotation angle  $\alpha$  as shown in Figure 4.7(a) so that the sum of intensities of pixels inside the square do not exceed  $2w$ . Thus the square covers a segment of the actual stroke.



Figure 4.7: Fitting of squares: (a) larger; (b) smaller.

If the distribution of intensity across strokes is approximated with a normal distribution  $N(\mu, \sigma)$  as shown in Figure 4.8, then the actual edge lies approximately on the distance of  $3\sigma$  from the centre of the stroke. In optics the visible edge of a peak is defined as the position where the intensity drops two times compared to the peak intensity. For normal distribution this visible edge is on the distance of  $\sqrt{2\sigma \ln 2}$  from the centre of the stroke and the area between the visible edges includes around 80% of the intensity.

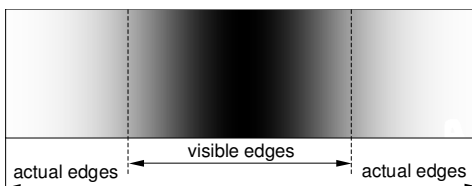


Figure 4.8: Stroke approximated by normal distribution.

The second square is fit within the pixels covered by the first square and is aimed to cover a segment of the visible stroke. This is achieved by gradually decreasing the size of the square and adjusting its position and angle until the pixels within the square contribute 80% of the total intensity of the original square (Figure 4.7(b)). The centre of this smaller square becomes the spline knot, and the area covered by

the bigger square is marked as visited. The next point on the edge is taken within the distance of  $1.5w$  from the current edge point and the fitting is repeated. In case the next point belongs to the visited area, the algorithm starts again from an arbitrary edge point which has not been visited yet. During the fitting the spline knots are saved along with the information about their neighbours. When no more unchecked pixels left, skeletal branches are formed. Spline knots that have only one neighbour become end points and those with more than two neighbours become junction points. Skeletal branches are formed between end points and junction points.

Unfortunately sometimes both algorithms do not put a spline knot in the point of peak curvature of the stroke when the curvature is too high as shown in Figure 4.9. Thus post-processing is applied to add one more spline knot.

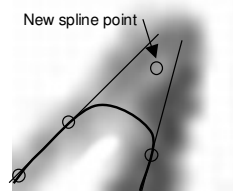


Figure 4.9: Post-processing of sharp corners.

## Stage 2: Merging of skeletal branches

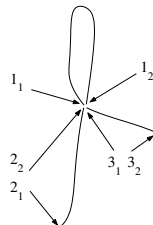


Figure 4.10: An example of junction point.

This stage focuses on correct restoration of interconnections between pairs of found skeletal branches in each junction point. The possibility of retracing of some branches is also taken into account. Branches are represented as arrays of spline points; the first point in an array is the first end of the branch and the last point is the second end of it. After the first stage each branch is represented as an array of spline knots. Let one branch end be  $b_i$  where  $b$  is the branch index and  $i$  can be 1 or 2 and represents the branch end. Each junction point has  $n$  involved branch ends and is represented by a  $n(n-1)/2$ -bit long binary string which can encode all possible connections between the involved branches. For example if there are three branches involved in  $n = 4$ -end

junction as in Figure 4.10, a connection between them can be encoded into a 6-bit string which represents a configuration of the junction:

|           |                |                |                |                |                |                |
|-----------|----------------|----------------|----------------|----------------|----------------|----------------|
| $b_{k_1}$ | 1 <sub>1</sub> | 1 <sub>1</sub> | 1 <sub>1</sub> | 1 <sub>2</sub> | 1 <sub>2</sub> | 2 <sub>2</sub> |
| $b_{k_2}$ | 1 <sub>2</sub> | 2 <sub>2</sub> | 3 <sub>1</sub> | 2 <sub>2</sub> | 3 <sub>1</sub> | 3 <sub>1</sub> |
| $S_J$     | 0              | 0              | 1              | 1              | 0              | 0              |

This configuration corresponds to the situation when branch 2 is connected to the right part of the loop (branch 1) and the left part of the loop continues in branch 3. The string can have any combination of bit values.

The configuration of the whole skeleton is represented by a string which is obtained by concatenation of strings of each junction. A certain cost is associated with a skeleton configuration. The number of possible configurations is calculated and depending on the value either an exhaustive search or genetic algorithm optimisation is used to find the configuration with the minimum cost.

Calculation of skeleton configuration cost consists of three stages. A configuration string of bits is mapped into a branch ends adjacency matrix of size  $2N \times 2N$ , where  $N$  is the total number of branches. The matrix contains 1 if the two corresponding ends are connected and 0 otherwise. After that a configuration correctness check is performed. The overall number of end points in the skeleton should be even. This condition arises from the fact that every pen-down event is followed by a pen-up in handwriting. The number of connections for each branch end should be either 0 (end point), or 1 (connection to another branch), or 2 (retracing and connection to two other branches). If either of the two conditions is not satisfied the configuration cost is assigned the highest possible value (penalty). After that another test is performed to determine whether such a configuration can result in a correct skeleton. If the test fails the configuration is also assigned the penalty value.

For those configurations which passed both tests, paths are searched between two end points. Calculation of the cost is based on the same principles used in other stroke restoration work (Bunke et al., 1997). Global and local costs are taken into account. Global cost is higher for greater number of separate paths which corresponds to many pen-ups and pen-downs in writing. Local cost contains several components. One component corresponds to a junction and its value depends on the number of branch ends in the junction as well as on the number of ends left disconnected. Another component corresponds to the smoothness of connection of branches in junction points and is proportional to the integral of the absolute value of curvature of the resulting curve taken in the vicinity of the junction point. The other component represents the cost of retracing of branches. It takes into account the smoothness of the branch, its

direction, and its length. The cost of retracing is lower for straight and short vertical branches. The total cost is a linear combination of the components:

$$Cost = a_0 C_{global} + a_1 \sum_i C_{M,i} + a_2 \sum_j C_{R,j} \quad (4.2)$$

where  $a_0$ ,  $a_1$ , and  $a_2$  are constants.

For each configuration possible paths (strokes) are searched between any two end points using a backtracking algorithm. While a stroke is formed the corresponding local cost components are calculated. In cases when a configuration does not result in several strokes or when some end points are left after all strokes are found the configuration cost is assigned a high value (penalty). For every retraced segment (it may include several branches) an attempt at hidden loop recovery is made (Figure 4.11(a)). In case of success the cost of the loop is calculated. It depends on the original width excess along the segment as well as on the remaining width excess. The costs of connection of the loop to the other segments are also recalculated. If the cost of having the loop is less than the cost of simple retracing, the loop is saved otherwise the retraced segment is saved. The loop restoration algorithm duplicates the retraced segment and moves knots of the first and the second copies apart from each other taking into account the underlying grayscale image as shown in Figures 4.11(b), 4.11(c). The resulting loops usually have excessive spline knots and are adjusted on the third stage.

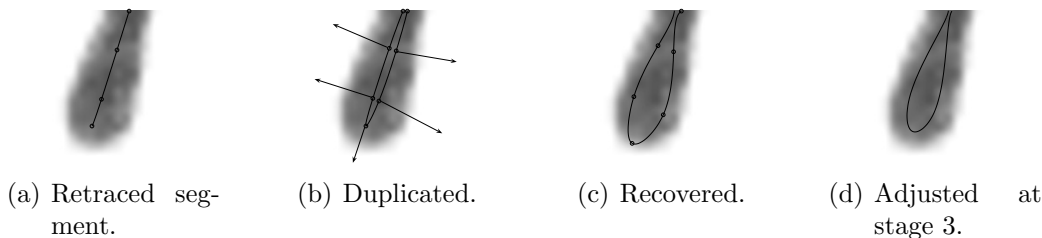


Figure 4.11: Loop recovery steps.

The costs are of different order of magnitude: connection cost is of the order of 0.1, the retracing cost and the loop cost are of the order of 1, the cost of leaving disconnected ends in junctions is of the order of 10 as well as the cost associated with the total number of strokes.

If a genetic algorithm is used for minimum cost configuration search, the parameters of the genetic algorithm are: population size 50, number of generations 100, uniform crossover with probability 0.6, mutation with probability 0.03, and replacement with elitism. Fitness function is taken as  $Cost^{-1}$  and a linear scaling with

factor 2 is applied to it.

### Stage 3: Adjustment of some spline knots

Because of inaccuracies in spline knot placement in the vectorisation stage and loop recovery, it is necessary to adjust the positions of spline knots in junctions, adjacent to the junctions, and those that belong to recovered loops. After the vectorisation stage each junction has one spline point which is shared among all branches of this junction. This changes after the second stage, when each stroke has spline knots independent of those of the other strokes.

Adjustment is performed for each curve separately. The spline knots whose positions are to be adjusted were the junction knots, their neighbours, and the knots on the restored hidden loops. For each non-loop knot to be adjusted the pixels within a circle of radius of one local stroke width with the centre in it are marked. Figure 4.12(a) shows the example of circles around the spline knots to be adjusted. For each of the marked pixels  $i$  the shortest distance to the curve  $d_i$  is calculated (Pottmann & Hofer, 2002). More details on the calculation of  $d_i$  are given below.

For knots of the restored hidden loops the calculation of  $d_i$  is performed separately for the two parts of a loop as shown in Figures 4.12(b) and 4.12(c). The radius of the circles is  $1.5w$  where  $w$  is the local width of the original stroke without the restored loop. For points within the circles on part 1 of the loop (Figure 4.12(b) the shortest distance to part 1 of the curve is calculated, while for points within the circles of part 2 the shortest distance to part 2 of the curve is calculated.

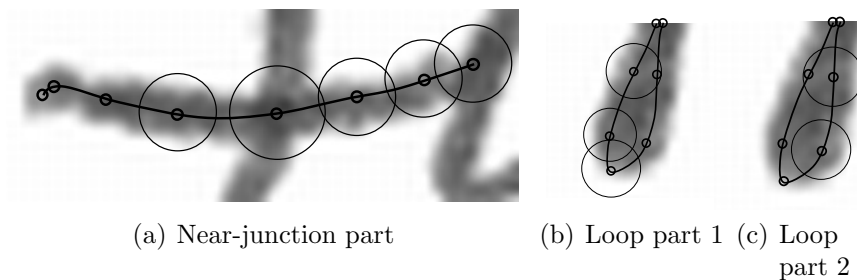


Figure 4.12: Pixels inside the circles are used for spline knots adjustment.

The smoothness of the curve is estimated as

$$\int ||D^2 f(t)|| dt, t = 0 \dots 1 \tag{4.3}$$

where  $f(t) = (x(t), y(t))$  is the curve and  $D$  is operator of differentiation, the integral is taken along the whole curve.

Adjustment of the spline knots is performed by minimisation of the function

$$F = (1 - \lambda) \sum_i \bar{I}(P_i) d_i^2 + \lambda \int \|D^2 f(t)\| dt \quad (4.4)$$

where  $\bar{I}(P_i) = 1 - I(P_i)$  is the reversed intensity of the pixel (*black* =  $0 \leq P_i \leq 1 =$  *white*). By changing  $\lambda \in [0..1]$  different degree of stroke smoothness can be achieved. For the current work  $\lambda = 0.9$  was chosen. The search for the minimum of (4.4) was performed via the steepest descend method. The method is similar to that described in (Wang et al., 2004), the difference being a different quadratic form in the first term of equation 4.4.

### Calculation of the shortest distance to a curve

Let curve  $\Gamma$  be defined as  $f(t) = (x(t), y(t))$ ,  $t = 0$  for one end of the curve and  $t = 1$  for the other end. Let  $I_i$  be the point that corresponds to a marked pixel. It is necessary to determine the distance from point  $I_i$  to  $\Gamma$ .

Point  $I_i$  can be either outer or inner point in respect to the curve. In the former case the normal from  $I_i$  to  $\Gamma$  does not intersect  $\Gamma$  within  $0 \leq t \leq 1$  but rather intersects its continuation as shown for point  $I_1$  in Figure 4.13(a). When point  $I_i$  is an outer point the nearest distance to the curve is simply the distance to the nearest end of the curve.

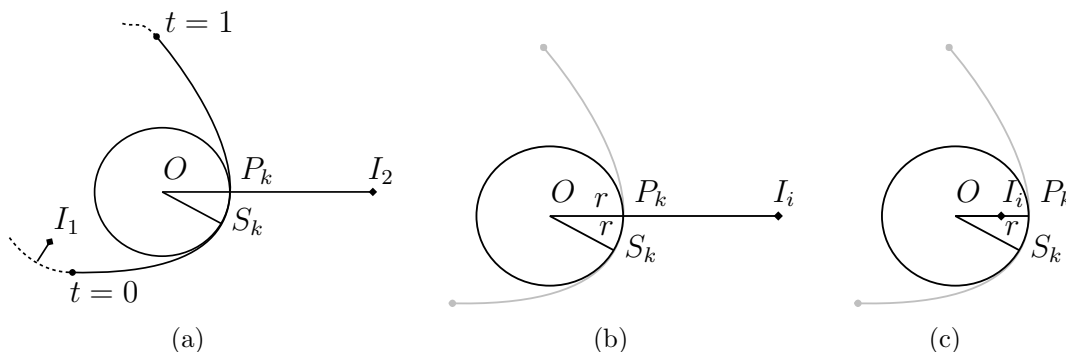


Figure 4.13: Calculation of distance to curve.

When  $I_i$  is an inner point as point  $I_2$  in Figure 4.13(a) an approximate distance calculation is possible. Let a set of sample points  $S_j$  is taken along the curve, and  $S_k$  is the closest sample point to  $I_i$ . Curve  $\Gamma$  can be approximated by a circle in the vicinity of  $S_k$ . The radius of the circle  $r$  is known from the curvature at point  $S_k$ <sup>1</sup> and the coordinates of the centre of the circle  $O$  can be easily calculated from the coordinates of  $S_k$  and the radius. The distance from  $I_i$  to  $\Gamma$  is approximated by the distance from  $I_i$  to the circle.

<sup>1</sup>If the curvature is 0 then the shortest distance is the same as the distance between  $I_i$  and  $S_k$ .

There are two possible cases here as shown in Figures 4.13(b) and 4.13(c) (a trivial case of point  $I_i$  on  $\Gamma$  is not considered). In the first case the distance is given by

$$d_i = OI_i - r \quad (4.5)$$

and in the second case it is given by

$$d_i = r - OI_i \quad (4.6)$$

where  $OI_i$  is the distance between  $O$  and  $I_i$ . This two formulas can be united into one:

$$d_i = |OI_i - r| \quad (4.7)$$

### 4.3 Comparison of skeletonisation methods

Examples of the original images and the skeletons of grapheme “th” are shown in Figure 4.14. It is seen that the new vector skeletonisation method provides better approximation to the pen tip trajectory than the thinning-based method. For the purpose of forensic document examination “the better method” means the one which allows the extraction of document examiner features with higher accuracy and/or enables extraction of more features. Thus, the only definite way to estimate whether the vector skeletonisation is better than the thinning-based one is to compare the accuracy of feature extraction and accuracy of writer classification when each of the methods is used.

In order to make a quantitative comparison of the two skeletonisation methods before feature extraction, human subjects were involved in skeleton comparison. Five people were asked to mark several points on grapheme “th” that are important in feature extraction: top points of stems of “t” and “h”, t-bar crossing points, and t-bar left and right end points (when possible). Each person was given 150 samples of grapheme “th” taken from 30 different writers. The resulting positions of the points were compared to the positions of the corresponding points on the skeletons produced by thinning-based and the vector skeletonisation methods. Coordinates of each of the test points were calculated as the average of the coordinates of points marked by the people. For each test point the error of its position was calculated as the average of the distances between the test point and the corresponding point resulting after application of skeletonisation (Figure 4.15).

The results are shown in Table 4.1. As seen from the table as well as from Figure 4.14, application of the new skeletonisation method results in a skeleton that

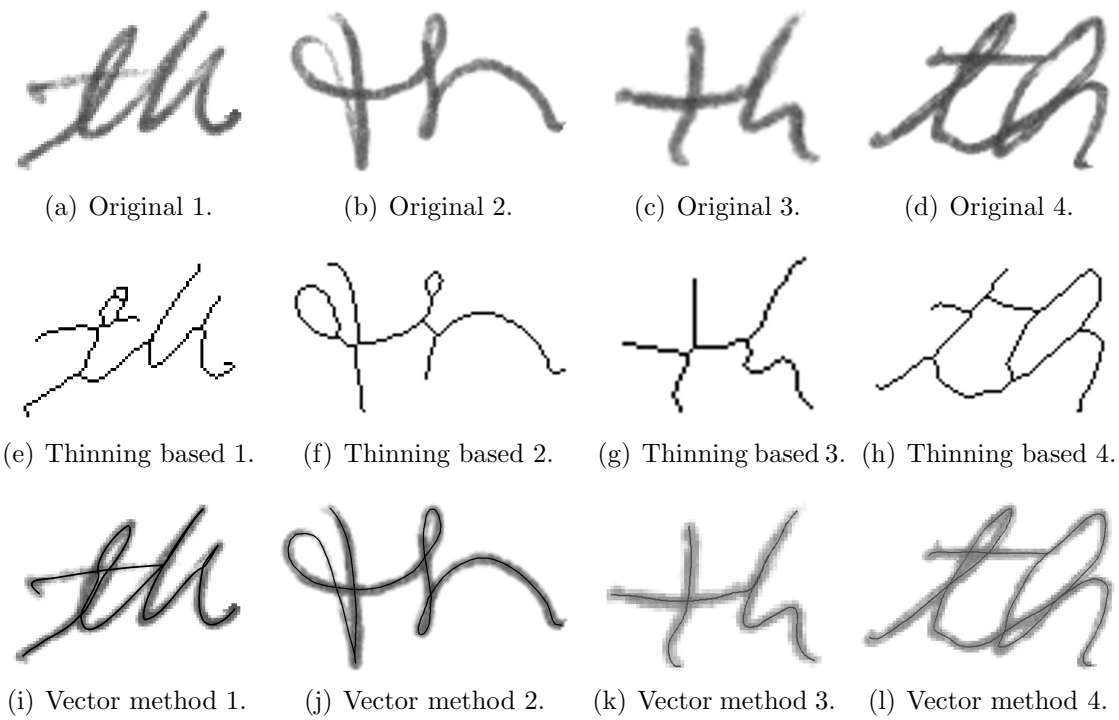


Figure 4.14: Examples of skeletonised images: original images are in the upper row, skeletonised by the thinning-based method are in the middle row and skeletonised by the vector method are in the lower row. Some images are faded to make the skeleton more visible.

is much closer to human perception than that obtained by thinning. Particularly, junction points are not split into two points and end points are at right positions.

| Point         | Skeletonisation method |        |
|---------------|------------------------|--------|
|               | Thinning-based         | Vector |
| top of h-stem | 2.1                    | 0.6    |
| top of t-stem | 2.1                    | 0.6    |
| t-crossing    | 5.0                    | 1.0    |
| t-bar left    | 2.3                    | 0.6    |
| t-bar right   | 2.4                    | 0.6    |

Table 4.1: Average errors of positions of test points (in pixels).

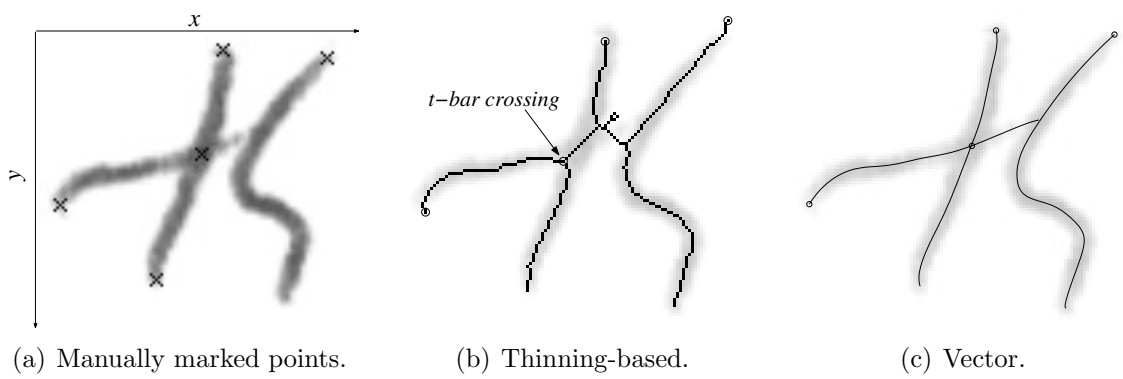


Figure 4.15: Skeleton comparison. Test points are marked as crosses and circles.

# Chapter 5

## Extraction of features

This chapter provides a detailed description of feature extraction algorithms when both raster and vector skeletons are used. The sequence in which features were extracted and evaluated is also discussed. All feature extraction algorithm as well as the front-end for feature extraction control were written in Matlab. The work presented in this chapter is also presented in (Leedham, Pervouchine, et al., 2003; Leedham et al., 2004; Leedham & Pervouchine, 2005).

Further in this chapter coordinates of certain points are used to express the feature values. The origin of the coordinate system is in the upper-left corner, the abscissa ( $x$ ) axis is horizontal and directed to the right, the ordinate ( $y$ ) axis is vertical and directed downwards.

### 5.1 Extraction of skeleton-independent features

Several features were extracted from either the original or the binarised image of handwriting samples. Height, width, and height to width ratio were measured from the binarised image by determining the bounding box of the image. The bounding box coordinates  $x_1, y_1, x_2, y_2$  correspond to the topmost, leftmost, bottommost, and rightmost black pixels on the image correspondingly. The feature values were calculated as

$$height = y_2 - y_1 + 1, width = x_2 - x_1 + 1, ratio = height/width \quad (5.1)$$

Height and width were measured in pixels, which can be converted into inches. Height to width ratio is dimensionless.

Average stroke width was estimated from the number of foreground and edge pixels in the binarised image. Calculation of the stroke width is based on the fact that if ribbon-like shapes of handwriting strokes are unfolded they can be approximated by a  $w \times l$  rectangle, where  $w$  is the stroke width and  $l$  is the total length of the strokes.

The area of such rectangle is  $w \cdot l$  and is approximately equal to the number of the foreground pixels  $N_s$ . The perimeter of the rectangle is  $2(l + w) \approx 2l$  since  $w \ll l$ . It can also be approximated by the number of edge pixels  $N_d$ . Thus the stroke width  $w$  can be calculated as

$$w = 2 \frac{w \cdot l}{2l} \approx 2 \frac{w \cdot l}{2(l + w)} \approx \frac{2N_s}{N_D} \quad (5.2)$$

The stroke width thus measured is expressed in pixels.

Pseudo-pressure is estimated from the grey levels of the image pixels. Let the set of the foreground pixels be  $S$  and the intensity of a pixel is  $I(x, y)$ , *black* =  $0 \leq I(x, y) \leq 1 =$  *white*. The average pseudo-pressure is thus

$$\langle p \rangle = \frac{1}{N_S} \sum_{(x_i, y_i) \in S} I(x_i, y_i) \quad (5.3)$$

And the standard deviation of the pseudo-pressure is

$$\sigma_p = \sqrt{\frac{1}{N_S - 1} \sum_{(x_i, y_i) \in S} (I(x_i, y_i) - \langle p \rangle)^2} \quad (5.4)$$

In order to save computational time and avoid scanning the image twice, both pseudo-pressure and its standard deviation were calculated from the image histogram. Both features are dimensionless provided the grey level of pixels is dimensionless too.

## 5.2 Detection of end points and junctions

Detection of end points and junctions is important because the feature extraction algorithm used shape analysis to find the areas of interest. For vector skeleton all end points and junctions were known from the skeletonisation algorithm. For the raster skeleton, detection of end points and junctions was performed by considering the 3x3 neighbourhood of each skeleton pixel and counting the number of transitions from black to white when the neighbourhood pixels were examined in the counter-clockwise direction as in Figure 5.1(a). If the number of transitions was equal to one, the pixel represented an end point. If the number of transitions was more than two, the pixel represented a junction. The complexity of the detection algorithm was thus  $O(n)$ ,  $n$  being the number of skeleton pixels. Figure 5.1(b) shows a skeleton example, and Figures 5.1(c) and 5.1(d) show the detected end points and junctions correspondingly.

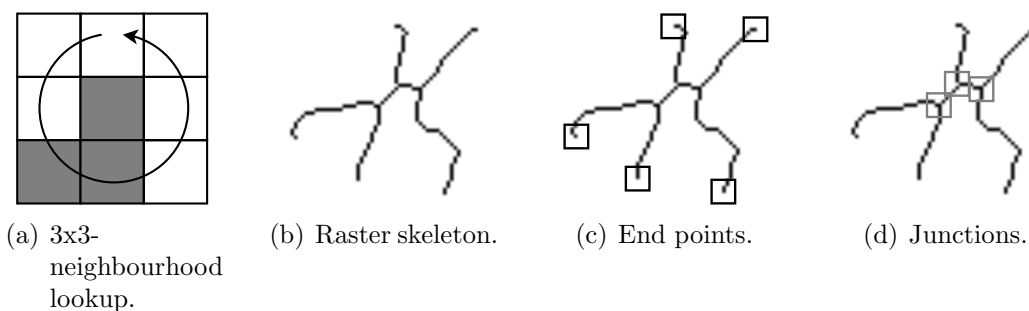


Figure 5.1: End point and junction point detection for raster skeleton.

### 5.3 Raster skeleton tracing

Following a 1-pixel wide line on a raster skeleton is an important subroutine of many feature extraction algorithms. In order to make the next step and visit the next pixel it was necessary to determine the direction to go. Even if the current pixel is not a junction point it is still possible that its 3x3-neighbourhood has several black pixels that have not been visited yet (Figure 5.2(a)). An even more complicated case is when the current pixel is a junction point. In order to trace skeleton branches correctly the tracing algorithm allows several possible sequences in which the pixels in the 3x3-neighbourhood of the current pixel were examined. Such parameter as *gravity* was used for tracing. Figures 5.2(b)–5.2(e) show the possible sequences in which the neighbourhood was examined for each of the four gravity values. The tracing algorithm examined the neighbourhood pixels in the sequence determined by the gravity and made the next step to the first non-visited black pixel found. For each encountered junction point the tracing algorithm located all possible branches and then called itself recursively for each of the found branches. Thus, when, for example, a tracing with up gravity was used, the tracing of branches was performed so that when a junction was encountered, the branch that leads up was traced first.

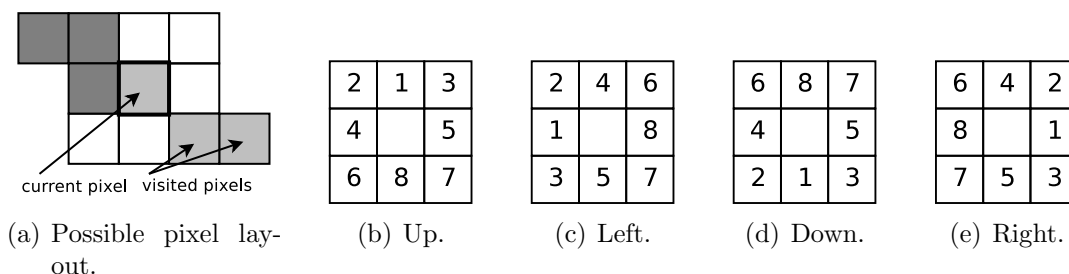


Figure 5.2: Following a raster skeleton branch.

## 5.4 Extraction of angular features

### 5.4.1 Slant

Slant was calculated in two different ways for raster and vector skeletons. In both cases a set of strokes or skeletal branches were located first, which represent the element from which slant feature was extracted: ascender for character “d”, descender for “y”, stems for “t”, “f”, and “h”, etc. After that, for raster skeleton the pixels that belong to the located set of branches were taken and a regression line  $x = ky + b$  was fitted into that set of points as in Figures 5.3(a) and 5.3(b). The choice of the fitting line is explained by the fact that slant is normally closer to a vertical line than to a horizontal line. If the set of points to which a line is to be fitted is denoted by  $(x_i, y_i), i = 1 \dots N$ , factors  $k$  and  $b$  are calculated as

$$k = \frac{N \sum_{i=1}^n x_i y_i - \sum_i x_i \sum_i y_i}{N \sum_i y_i^2 - (\sum_i y_i)^2} \quad (5.5)$$

$$b = \frac{\sum_i x_i}{N} - k \frac{\sum_i y_i}{N} \quad (5.6)$$

Slant value was taken as

$$\alpha_{slant} = \arctan k \quad (5.7)$$

The slant value was within range  $-\pi/2 \leq \alpha_{slant} < \pi/2$  and expressed in radians. The value of  $b$  (5.6) was used to control the correctness of the slant extraction (see Section 5.9.1). The value of slant for a vertical line was equal to 0.

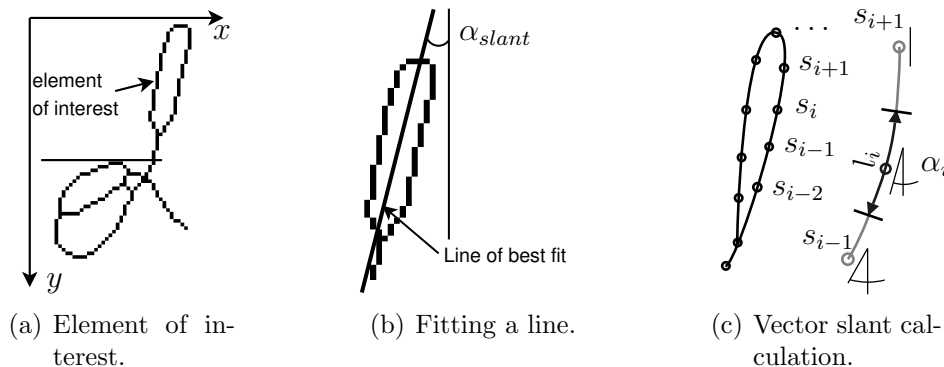


Figure 5.3: Slant calculation.

For vector skeleton slant value was calculated by taking a set of sample points  $s_i$  along each spline-approximated stroke that represented the element of interest (ascender, descender, etc.) and calculating the angles of tangents in these points  $\alpha_i = \arctan k_i$ . The slant was calculated as the weighted average of those angles:

$$\alpha_{slant} = \frac{\sum_i l_i \alpha_i}{L} \quad (5.8)$$

where  $l_i$  is the length of the corresponding curve segment (see Figure 5.3(c)),  $L = \sum_i l_i$  is the total curve length.

### 5.4.2 Stroke angle

Contrary to slant, final strokes can point to any direction, hence the range of possible values should comprise an entire circle:

$$-\pi \leq \alpha_{stroke} < \pi \quad (5.9)$$

Final stroke angle was represented as a vector whose angle is equal to that of the final stroke and which points towards the continuation of the stroke as in Figure 5.4(a).

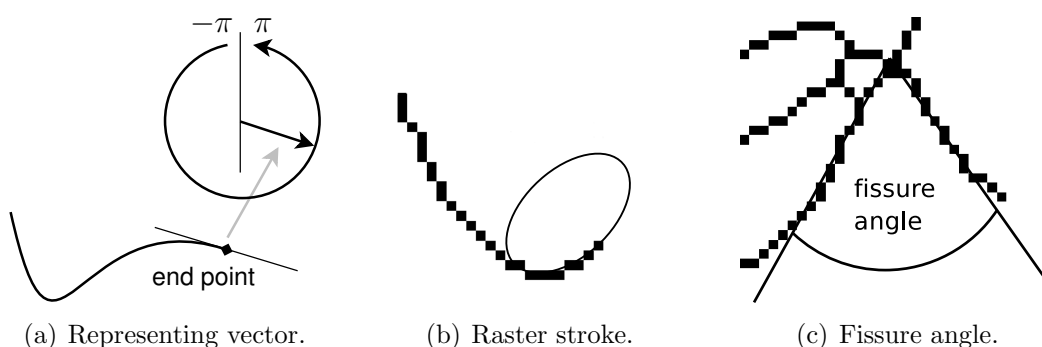


Figure 5.4: Final stroke and fissure angle extraction.

For the case of vector skeleton the final stroke angle was calculated as the angle of the tangent to the corresponding end point. For the case of raster skeleton a straight line was fitted into a set of pixels initially, but later that was changed to fitting an ellipse into a set of pixels representing the final stroke and the tangent to the ellipse in the point nearest to the end point of the stroke was taken for calculation. In the cases ellipse fitting algorithm produced high residual error, a straight line was fitted into the set of pixels instead of an ellipse.

### 5.4.3 Fissure angle

Fissure angle measurement was performed similarly to the final stroke measurement. Once the fissure point was located, extraction of fissure angle in case of a vector skeleton was straightforward. In the raster skeleton case two regression lines were fitted, each one approximating the tangent line to the corresponding stroke (see Figure 5.4(c)). The angle between the regression lines was taken as the fissure angle.

## 5.5 Detection of loops and extraction of loop features

Detection of loops was performed in a different way for different loops. All algorithms below were implemented for the raster skeleton only since the extraction of features from the vector skeleton was performed only for grapheme “th”.

Descender of character “y” was considered as a loop in general, hence no special loop detection algorithm was necessary. The bottommost pixel on the raster skeleton of a character “y” sample was detected and the skeleton was traced from this pixel into one or two possible directions. The tracing stopped when either an end point or a junction was met. The visited pixels formed either a complete descender loop (Figure 5.5(b)) or an open descender (Figure 5.5(a)). In both cases the set of the visited pixels was considered to be a loop and the loop completeness was calculated as

$$f_{12} = 1 - \frac{\sqrt{(x_1 - x_n)^2 + (y_1 - y_n)^2}}{\sum_{i=1}^{n-1} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2}} \quad (5.10)$$

where  $(x_i, y_i)$  were the pixels of the loop,  $i = 1$  corresponded to one end and  $i = n$  to the other end. Thus the loop completeness feature had a value of 0 for a straight line, a value of 1 for a closed loop, and for an open loop the feature value was between 0 and 1. In the case of a closed loop in descender of “y”, the loop length was calculated as

$$f_{18} = \frac{y_b - y_1}{\cos f_{15}} \quad (5.11)$$

where  $f_{15}$  is the descender slant and  $(x_y, y_b)$  is the bottommost point of the loop.

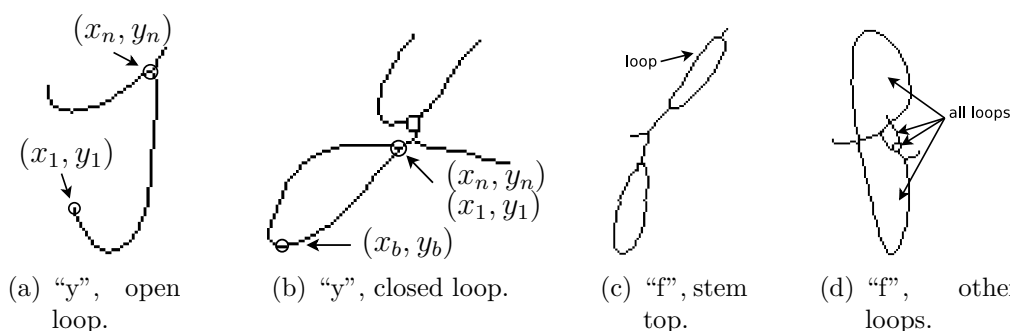


Figure 5.5: Loop features.

Loops at the top and bottom points of “f”-stem as well as loop at point  $Y_T$  of character “y” were detected in a similar manner. First the top and bottom stem points were detected and then tracing of the skeleton was performed in order to determine whether there was a loop. Since thinning-based skeletonisation sometimes

produced extra branches and not all such branches could be removed by the correction algorithm, the possibility of presence of a short branch on top of the loop as in Figure 5.5(c) was taken into account.

Other loops were detected using the loop detection algorithm described in (Cheriet et al., 1992). The algorithm was capable of detecting only complete loops (Figure 5.5(d)). The number of loops was calculated as the number of complete loops detected. For each detected loop its area was calculated as the number of the enclosed white pixels and the biggest loop was considered as the main loop. For the main loop its slant and perimeter (length) were measured. Loop perimeter was measured as the number of pixels that constituted the loop contour. Loop slant was measured by dividing the loop into two parts: the upper and the lower half. The centres of gravity were calculated for each of the two halves. The angle between the line connecting the two centres of gravity and a vertical line was taken to be the loop slant (Leedham, 1999).

## 5.6 Relative height of ascender/descender

To measure relative height of an ascender or descender it was necessary to find a horizontal line that separates the base part of a character from its ascender/descender part. It was done in two different ways for characters “y” and “d”.

### 5.6.1 Descender of “y”

Descender of “y” was detected by tracing it in both directions from the bottommost black pixel, which always belonged to the descender, as discussed earlier in Section 5.5. After tracing of the descender all the visited pixels were erased and the horizontal distance from point  $(0, y)$  to the leftmost black pixel was calculated for all  $y$  on the image. The horizontal level at which the distance function had a sharp increase was taken as the bottommost lowest base level. In case when no sharp increase was observed, the bottommost black pixel was taken as the lowest base level. If the ordinate of this pixel was  $y_{base}$ , the relative height of descender was calculated as

$$f_{11} = \frac{y_b - y_{base}}{f_8} \quad (5.12)$$

where  $f_8$  was the character height feature.

Examples of various forms of character “y” are presented in Figures 5.6. The eliminated descender is shown in grey; the bottommost point of descender is marked. Figure 5.6(a) presents an example when the lowest base level is detected by the sharp

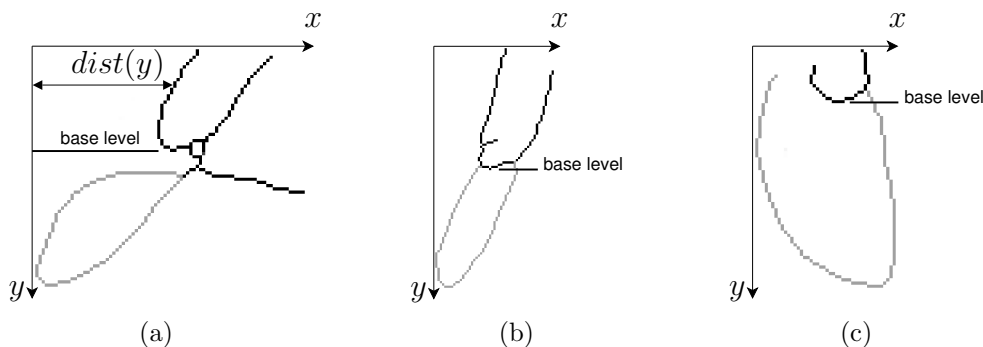


Figure 5.6: Relative height of descender.

increase of the horizontal distance function, and Figures 5.6(b) and 5.6(c) present an examples when the bottommost black pixel is the base level.

### 5.6.2 Ascender of “d”

Due to frequent presence of spurious loops resulting from skeletonisation of hidden or small loops in the ascender of “d”, tracing of the ascender part of the skeleton proved to be not as robust as it was for the descender of “y”. In order to find the horizontal level that corresponded to the top of the base, two algorithms were used. In the first algorithm the distance between the left edge of the image and the leftmost black pixel was calculated the same way as for the base part of “y”. Sharp increases in the distance function value  $dist(y)$  were marked and sorted according to their increase value. The largest increase was compared to a threshold value. If its magnitude was more than the threshold, ordinate  $y$  that corresponded to that increase was taken to be the top of the base as in Figure 5.7(a). If none of the increases was larger than the threshold the second detection algorithm was run. The value of the threshold was determined from experiments.

The second algorithm extracted the width of the character skeleton as the function of  $y$  (see Figure 5.7(b)). The value of  $y$  at which the resulting function  $width(y)$  reached its minimum was taken to be the top of the base. In cases when there were several minimums the lowest one was taken.

Once the top level of the base  $y_t$  was detected, the feature values was calculated as

$$f_4 = \frac{y_t - y_1}{f_1} \quad (5.13)$$

where  $y_1$  was the top level of the bounding box of the skeleton and  $f_1$  is the character height feature.

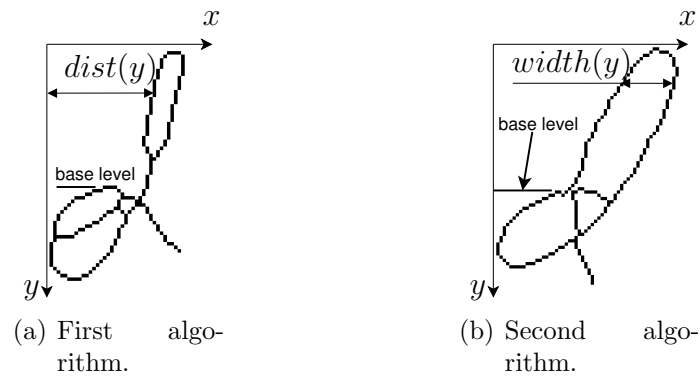


Figure 5.7: Relative height of ascender.

## 5.7 Other features of “t” and “th”

### 5.7.1 Stem and crossbar of “t”

A necessary but not sufficient condition for the existence of t-bar is the presence of a junction point on a stem. Detection of the stem was simple as the lowest black pixel belonged to the stem for almost all possible shapes of the character. Detection of t-bar branches was subdivided into 5 categories. The categories were divided depending on the number of junction points detected on a stem and the number of loops which contain the junction points.

1. *Presence of one junction point and zero loops.*

This was a simple case when the only junction was formed by a t-bar.

2. *Presence of a junction point due to a loop in the upper or lower region of the stem.*

This loop could be a continuation of a connecting ligature from the previous character, or could be a part of a t-bar if it was located in the lower region of the stem.

3. *Presence of 2 junctions on a stem and one loop.*

This situation occurred when a loop in the upper region was present as well as a t-bar.

4. *Presence of 2 junctions on the stem and 2 loops.*

This situation occurred when a loop in the upper region and a loop in the lower region were present. A t-bar was part of the loop in the lower region of the stem.

5. *Presence of 2 junction points and 3 loops.*

Here the loops were formed by intersection of t-bar with loops that belonged to the stem.

The algorithm located extrema points of black pixel density when scanning an image in the horizontal direction. The extrema points could be the end points of the t-bar. Closeness of the located extrema points to each other and tracing of the skeletal branches from the corresponding end points and from the top and bottom points of the stem were used as a test of correspondence of the extrema points and the t-bar end points.

Once t-bar end points were detected (and, hence, a point of crossing was detected), extraction of the crossbar width, crossbar left part, crossbar right part, top part of the stem, and bottom part of the stem of character “t”,  $f_{34} \dots f_{38}$ , became straightforward.

### 5.7.2 Top of stems of “t” and “h”

Features of grapheme “th” were extracted after experiments with extraction of features from the four characters. Geometrical features of t-bar and t-stem were eliminated by that time because of both high extraction failure rate and low discriminating power (as was indicated by the ANOVA test, see Chapter 6). To extract some features that represent relative position of characters “t” and “h” in the grapheme it was decided to extract slant features for each of the characters and also measure the relative position of the top points of their stems. Detection of the top points was made by first detecting all the end points in the upper half of a sample image and then tracing the branches from those end points to determine which of them correspond to which elements of “th”. Cases of “t” and “h” sharing the top stem point were taken into account. Once the top of the stem points were detected, extraction of the related distance features and the angle (Section 3.4, Figure 3.2(f)) was simple.

All connected components in a sample image were detected. If the top of the stem points belonged to a different connected components, characters “t” and “h” were disconnected, otherwise they were connected.

### 5.7.3 Features of “th” extracted from vector skeleton only

When the vector skeletonisation was implemented, a number of additional features were extracted from grapheme “th”, which could not be extracted reliably from the raster skeleton.

- Detection of the top points of “t” and “h” stems was conducted similarly to that on a raster skeleton. All end points were taken and the corresponding branches were analysed. Due to absence of extra branches and junctions on vector skeletons detection of the top points of the stems was robust.

- *Number of strokes and number of loops and retraced strokes* were available directly from the skeleton. Since each retraced stroke can be considered as a small loop, or a hidden loop, the number of loops, the number of hidden restored loops and the number of retraced strokes were summed to give the feature value.
- *Standard deviation of stroke width* was extracted by taking a set of sampling points on the skeleton curves with a small step and measuring the cross-section of the strokes on the underlying binarised image as shown in Figure 5.8(a). The obtained measurements were then used to calculate the feature value.
- *Presence of loops* at the top of t-stem and h-stem could be extracted from the raster skeleton as well. However, the experiments with extraction of similar features from character “f” showed that extraction of these features from the skeleton produced by the thinning-based method was unreliable due to spurious loops introduced by thinning. Extraction of these features from a vector skeleton was straightforward once the corresponding end points were determined.
- *Straightness* of a stroke (t-bar, t-stem, h-stem) was calculated similarly to the descender of “y” completeness feature: if the distance between the two end points was  $d$  and the length of the stroke was  $L$ , the straightness of the stroke was given by

$$\text{straightness} = \frac{L}{d} \quad (5.14)$$

It was close to 1 for a straight stroke, and significantly larger for a curved stroke.

- *Maximum and average curvature* of a stroke (h-knee) was calculated by taking sample points with a small step along the curve, calculating the curvature in each point, and taking the largest value and the weighted average.
- *Relative size of h-knee* was calculated as the largest distance from h-stem to the h-knee curve as in Figure 5.8(b). It was approximately calculated as the largest horizontal distance between the curves representing the stem and the knee.

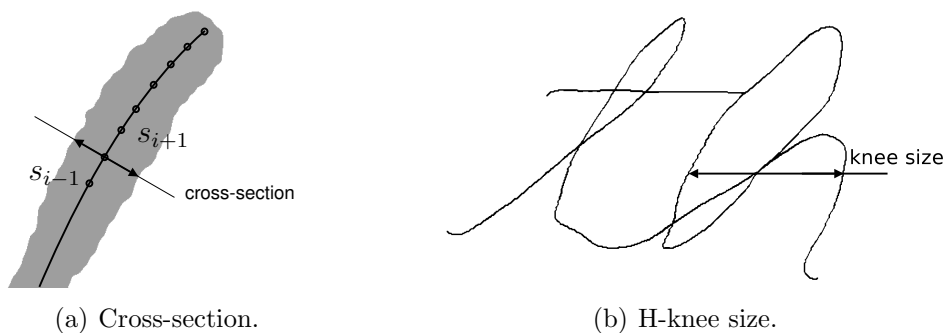


Figure 5.8: Some features of “th” extracted from a vector skeleton.

## 5.8 Feature extraction engine

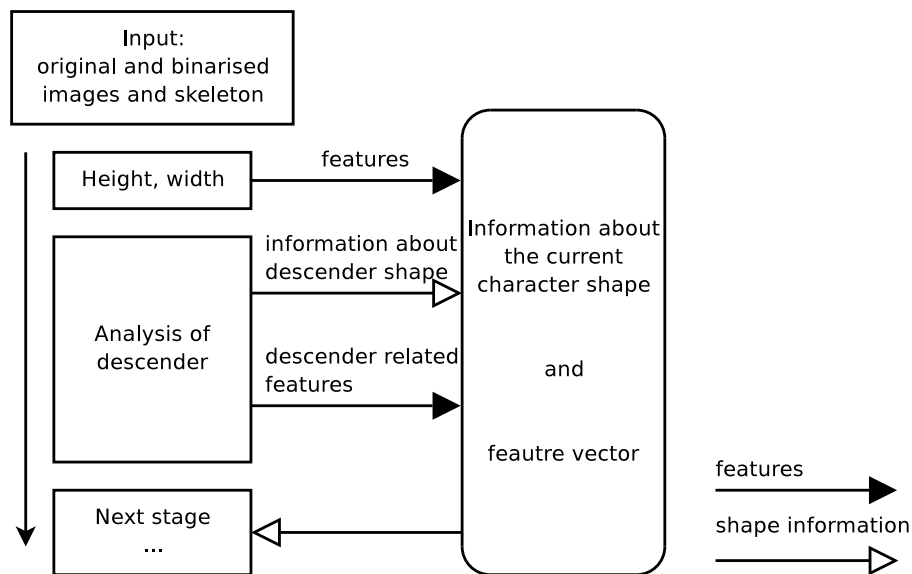


Figure 5.9: Extraction of features of character “y”.

Algorithms for feature extraction consisted of a main program and subroutines for extraction of particular features. The input to the algorithms was a character image, the output was the feature vector along with additional information which was later used to verify correctness of the feature values. Obviously it was impossible to write algorithms for all possible shapes of each character. However, it was observed that for most samples of each character and grapheme a wide variety of shapes could be taken into account.

The main program was different for each character and the grapheme. Each of the four character-specific extraction algorithms was designed so that it could handle many different possible character forms. In each stage some features were extracted and information about the character form, such as position of certain points and correspondence of branches to certain parts of the character, was refined. This information was then used in further stages of the feature extraction.

Figure 5.9 shows part of the feature extraction algorithm for character “y”; the transparent arrows indicate information about a character form. First the skeleton-independent features were extracted. After that the extraction algorithm proceeded with a study of the descender. Results of the descender tracing, namely, what kind of points (end or junction) were encountered on each side during the descender tracing, provide the information about the character shape. Depending on the current information, the main program selected an appropriate method to locate important elements of a character for extraction of features in the next stage. Once the ele-

ments were located a subroutine was called to calculate feature values and get more information about the form of the character.

Similar algorithms were used for the other character and the grapheme. Extraction of skeleton-related features of character “d” began by locating the topmost black pixels and tracing the skeleton downwards. Extraction of features of “f” and “t” and grapheme “th” began by locating a bottommost point of the stems and analysis of the end points in the upper half of the image by tracing the corresponding branches.

## 5.9 Feature extraction accuracy

### 5.9.1 GUI tool for feature extraction examination

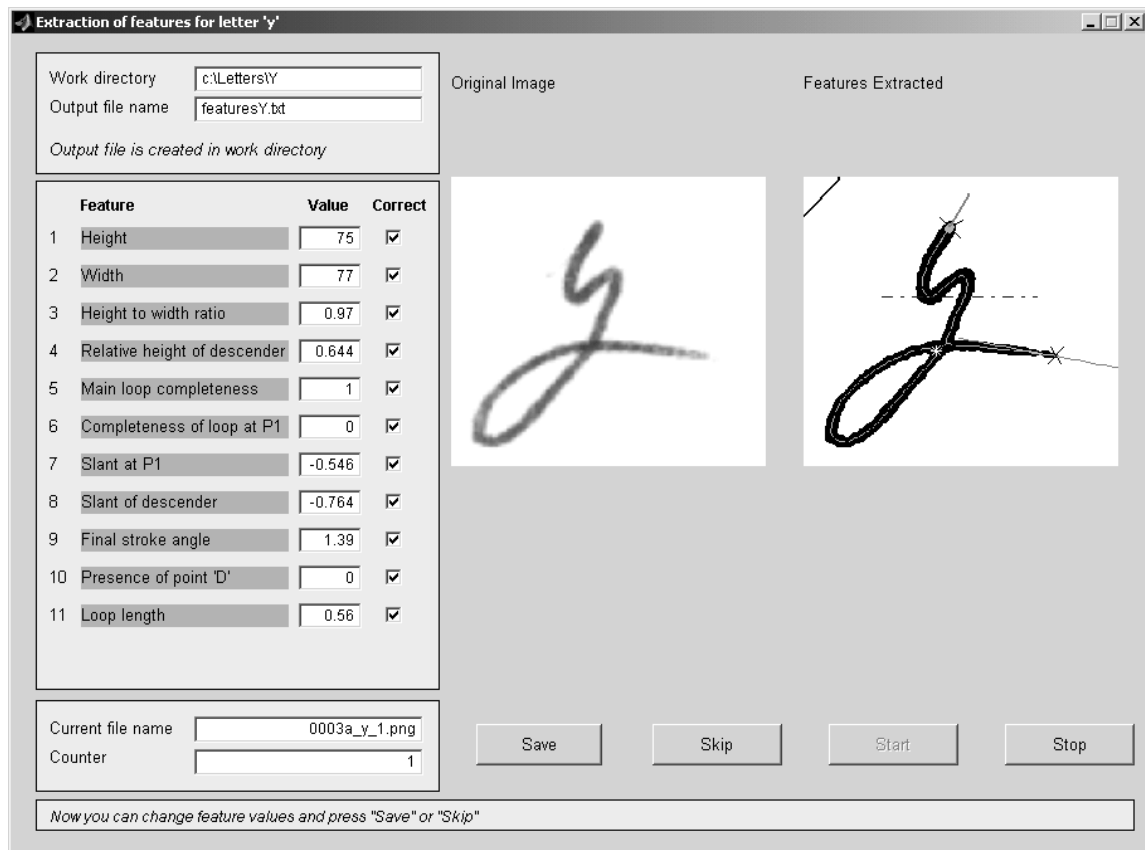


Figure 5.10: Front-end for feature extraction control.

In order to check the results of feature extraction, a GUI tool was written for easy examination and, if possible, correction, of feature values. First, the feature extraction was run in a batch mode. For each input image the binarised image and the skeletons were saved on disk. Feature values were saved in a separate data file. Additional information such as position of certain points and branches extracted during feature extraction and used to calculate feature values was saved in a separate data file too.

After that, the GUI-based feature extraction examination tool read the sample images one after another and let the user mark the samples for which some features were extracted incorrectly. The tool window is shown in Figure 5.10.

For each sample image the image itself was shown on the left and the binarised image with the skeleton over it was shown on the right. The feature values were read from the feature data file and shown in the corresponding boxes. Each feature value could be edited by changing the value in the box. The correct/incorrect feature extraction checkbox was shown as “checked” for each feature value and the user could “uncheck” it if the visual examination shown that the value was incorrect. If a feature was extracted incorrectly, the user could deselect the corresponding checkbox thus marking the sample as “incorrectly processed”. Alternatively, for example, in cases of incorrectly extracted binary features, the user could change the feature value to the correct one.

To make visual examination easy, data from additional data files were read and used to represent the feature values on the image on the right (binarised image with skeleton). On the example in Figure 5.10 the horizontal dash-dot line indicates the bottommost point of a base part. Slants and stroke angles were also represented by lines of different colour. Points used in feature extraction such as final stroke ends, top/bottom of stems, fissure point, etc. were shown as well.

If some feature values were extracted incorrectly for a sample, the GUI tool allowed the user to approximately determine the stage at which the feature extraction went wrong. The binarised image and the skeleton were available, and indication of reference points used in feature extraction allowed the user to see which points were determined correctly and which were not. Thus, the tool not only was used to mark correctly and incorrectly processed samples, but also provided the feedback on the quality of feature extraction and character shape analysis algorithms. Thus, the tool proved to be useful for both examination of the results of feature extraction and debugging of the feature extraction programs.

## 5.9.2 Results and discussion

Feature extraction experiments were conducted in several iterations with different feature sets as shown in Figure 5.11. At first, the four characters, “d”, “y”, “f”, and “t”, were used and the feature set consisted of 40 features in total. After evaluation of the discriminative power of those features using ANOVA and measure of separability of classes (Schrümann, 1996), some features were eliminated.

Further, character “t” was changed to grapheme “th”, and the new feature set contained 31 features. Also some improvements in feature extraction algorithm were introduced such as approximation of a final stroke with an ellipse rather than with a straight line. Analysis of the new feature set was performed by building a classifier and searching for the feature subsets that yielded the highest classification accuracy. Experiments with classification accuracy estimation demonstrated that features of grapheme “th” introduced the main improvement in writer classification accuracy as compared to features of other characters.

In the third series of experiments only features of grapheme “th” were used. Some new structural features ( $f_{53} \dots f_{56}$ ) were also added. To improve the accuracy of feature extraction the new skeletonisation method was implemented. The new method enabled the extraction of more document examiner features that could not be extracted reliably when thinning-based raster skeletonisation was used ( $f_{57} \dots f_{67}$ ). The final set of features contained only features of grapheme “th”. In total 14 features were extracted for the grapheme when raster skeletonisation was used, and 25 features were extracted when vector skeletonisation was used.

A detailed description of the analysis of the feature discriminative power is presented in Chapter 6.

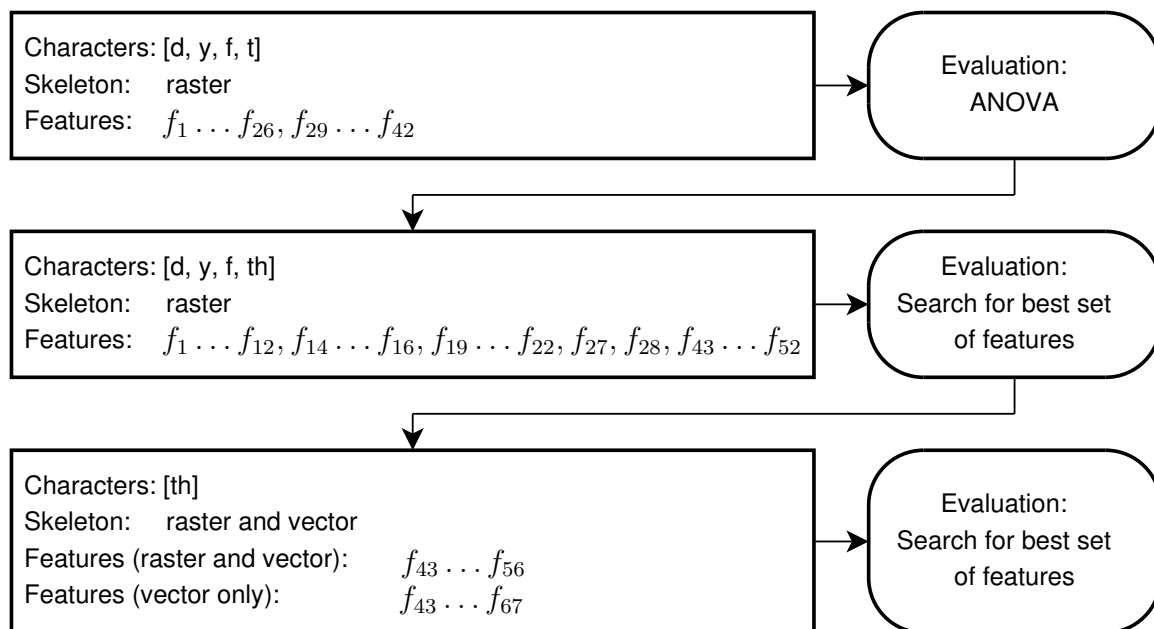


Figure 5.11: Iterations of feature extraction and analysis.

Accuracy of feature extraction was evaluated for each feature separately in the first series of experiments. This estimation allowed the identification of the main problems in the feature extraction algorithms. Figure 5.12 shows the feature extraction error

rates for each feature of the first set of 40 features. Features that are not shown on the chart are those that were always extracted correctly, such as height and width. As seen from the chart, final stroke angle extraction produced a relatively high failure rate of 23%. Such a high error rate is due to two reasons: inaccuracy in final stroke angle measurement (characters “y” and “d”) and errors in detection of character shape after analysis of the descender (character “y”). After changing the final stroke extraction method from fitting a line to fitting an ellipse or a line, the accuracy of the final stroke angle extraction was improved to 89% (see Section 5.4.2). The algorithm for analysis of descender of “y” was also improved so that it could take into account more different “y” shapes and thus produce less errors in detection of strokes and points, which are used for feature value calculations.

Features related to t-bar and t-stem also produced high rate of failure. The errors were mostly due to incorrect detection of the branches of t-bar and t-stem. Detection of those branches was based on skeleton tracing and a search for the shortest paths between the crossing point and the end points of the stem and crossbar. The correct detection was difficult due to the fact that the crossing point was normally split into two or more junction points due to the raster skeletonisation. Another problem was created by the presence of spurious loops on the stem and crossbar around the crossing area. Thus, it was later decided to eliminate the features of t-bar and t-stem from the feature set.

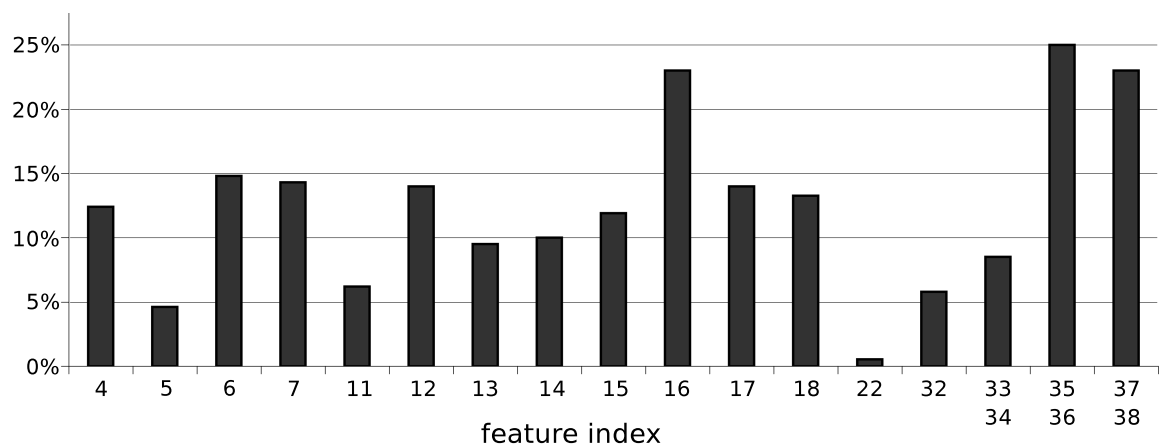


Figure 5.12: Average error rates of feature extraction for the initial set of 40 features.

In further experiments the accuracy of feature extraction was estimated separately for each character (grapheme) rather than for each feature. From the viewpoint of classification the feature extraction algorithm fails if at least one feature is extracted incorrectly. Thus the values of extraction accuracy in Table 5.1 present the cases

when all features were extracted correctly. It shows the results for the second feature set (31 features from characters “d”, “y”, “f”, and grapheme “th”) as well as for the third feature set (“th” only) when the vector was used.

| Character         | “d” | “y” | “f” | “th” (raster) | “th” (vector) |
|-------------------|-----|-----|-----|---------------|---------------|
| Number of samples | 861 | 707 | 712 | 797           | 150           |
| Accuracy, %       | 85  | 85  | 92  | 87            | 94            |

Table 5.1: Feature extraction accuracy.

Improvements in the feature extraction algorithms resulted in a reasonable accuracy of feature extraction when raster skeletonisation was used. Those samples that were processed incorrectly were excluded from the feature usefulness analysis. Implementation of a vector-based skeletonisation led to significant improvement in detection of important points and strokes of character samples (see Table 4.1), and led to an increase in the accuracy of feature extraction as seen from the last two columns of Table 5.1. It is reasonable to conclude that greater precision in handwriting stroke approximation offered by the vector skeletonisation method resulted in greater accuracy in measurements of feature values. Remaining errors in feature extraction from vector skeletons were mostly due to a variety of character shapes. Changing the feature extraction algorithm further to take even more possible character shapes into account seems unreasonable because it will complicate the feature extraction greatly while producing only a small increase in the feature extraction accuracy. Besides, there will always be rare character shapes which will cause failures of the feature extraction. Further improvement of feature extraction is most likely possible if interactive extraction algorithms are implemented on the base of the existing ones, when the user is asked for assistance in determining various character shapes.

## Chapter 6

# Analysis of feature discriminative power

This chapter discusses analysis of usefulness of features. Various methods of analysis and their purposes are described here. Application of those methods to feature sets obtained in the experiments is discussed in the same sequence in which the experiments were conducted: 40 features from 4 characters first, then 31 features from 3 character and 1 grapheme, and then features of the grapheme extracted using raster and vector skeletonisation methods, as shown in Figure 5.11. Some parts of this chapter are also presented in (Leedham et al., 2004; Pervouchine & Leedham, 2004; Leedham & Pervouchine, 2005).

### 6.1 Methods of analysis

There are several methods of analysis that can be applied to determine how significant a particular feature is or how good a feature set is. To make further discussion easier, the following variables are introduced and used throughout this chapter:

- $f$  denotes a feature value and is written with three indices:  $f_{ijk}$ ;
- index  $i$  is the feature number;
- index  $j$  is the writer number;
- index  $k$  is the sample number for a particular writer. Thus,  $f_{ijk}$  denotes  $i$ -th feature extracted from  $k$ -th sample of  $j$ -th writer. For example,  $f_{123}$  denotes the value of *height of “d”* extracted from the third sample of writer #2;
- $N_{ij}$  is the number of values of  $i$ -th feature extracted for writer  $j$ ;
- $m$  is the total number of writers;
- $M$  is the number of features.

For feature  $f_i$  the writer mean  $\overline{f_{ij}}$  and the total mean  $\overline{f_i}$  are calculated as

$$\overline{f_{ij}} = \frac{1}{N_{ij}} \sum_{k=1}^{N_{ij}} f_{ijk} \quad \overline{f_i} = \frac{1}{m} \sum_{j=1}^m \overline{f_{ij}} \quad (6.1)$$

Several definitions of feature relevance has been proposed (John et al., 1994). Normally they define a feature to be either relevant or irrelevant (Figure 6.1(a)). This dichotomy is not always adequate: for example, if there are two correlated features and exclusion of both of them from the set degrades the classifier performance, either of them can be excluded from the set without loss of classification accuracy. Depending on the definition used, such features are classified as either both relevant or both irrelevant. Either classification is misleading.

A variation of the division of features into two classes of relevant and irrelevant features is use of a value for each feature, which indicates a measure of usefulness (Figure 6.1(b)). For example, decrease in classification accuracy when a feature is excluded from a feature set can be used as an estimate of the feature relevance. In this case features are ranked according to the indicator value.

John et al. (1994) proposed to divide features into three rather than two categories: strongly relevant, weakly relevant, and irrelevant (see Section 2.8). In the example above with two correlated features those features are weakly relevant: either of them can be excluded without loss of classification accuracy, but not both of them. The scheme in Figure 6.1(c) presents the three-category division of features according to their relevance.

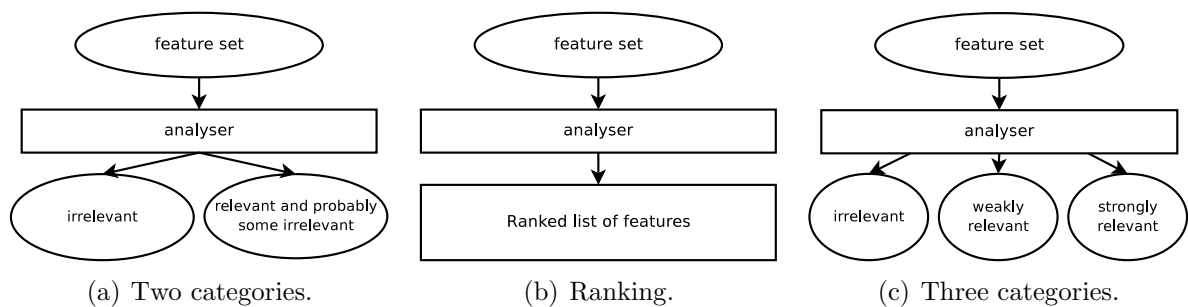


Figure 6.1: Division of features by relevance.

Methods such as principal component analysis and Fisher discriminant analysis are used to reduce the dimensionality of a feature set. However, these methods perform a mapping from the original feature space into a new feature space. Methods of feature selection that perform such a mapping are not suitable for the study because the features in the new space have no actual representation as document examiner features.

### 6.1.1 Assessment of the whole feature set

Discriminative power of a feature set as a whole can also be assessed. Such a test allows an assessment of whether the feature set is at all useful. If the test shows that the set has a discriminative power, it makes sense to conduct further analysis of features.

A value called *an extent of separability* can be calculated to assess a feature set (Schrümann, 1996). Assuming that given a feature vector, each class (writer) is approximately equiprobable, the mean-squared within-writer distances are calculated as

$$V_j^2 = E\{\|\vec{f}_{jk} - \vec{f}_j\|^2 \mid j\} \approx \frac{1}{N_{ij}} \sum_{k=1}^{N_{ij}} \|\vec{f}_{jk} - \vec{f}_j\|^2 \quad (6.2)$$

where  $\vec{f}_{jk} = (f_{1jk}, f_{2jk}, \dots, f_{Mjk})$ ,  $M$  is the number of features. The mean of the  $V_j^2$  is calculated as

$$V^2 = \frac{\sum_j N_{ij} V_j^2}{\sum_j N_j} \quad (6.3)$$

The mean-squared between-writer distance is calculated as

$$D^2 = \frac{1}{m(m-1)} \sum_{j_1, j_2} \|\vec{f}_{j_1} - \vec{f}_{j_2}\|^2 \quad (6.4)$$

The extent of separability

$$Q = \frac{V^2}{V^2 + D^2} \quad (6.5)$$

indicates optimum separability as  $Q \rightarrow 0$  and inseparability as  $Q \rightarrow 1$ . The extent of separability approaches 0 when the points in feature space corresponding to different writers are grouped in clusters according to the writers and the distance between the clusters is significantly larger than their size.

Application of the method is presented below in Section 6.2.

### 6.1.2 One way analysis of variance test

Evaluation of feature usefulness can be done for each separate feature of a set by analysis of the feature value distributions for different classes (writers). Such an evaluation allows some of the irrelevant features to be filtered out and divides all features of the set into two classes: those that failed the test (irrelevant) and those that passed it. The latter class may include both relevant and some of the irrelevant features from the original set.

One of the methods to perform such an evaluation is to ensure that for a particular feature  $f_i$  the feature values for different writers are not all equal. This is a necessary

condition for a feature to be relevant, and it gives a confidence that the feature can be used to separate the group of all writers into at least two subgroups. The null hypothesis to be tested for each  $f_i$  is

$$H_0 : \{\text{All } \overline{f_{ij}} \text{ are equal}\} \quad (6.6)$$

and the alternative hypothesis is

$$H_1 : \{\text{At least one } \overline{f_{ij}} \text{ differs from the others}\} \quad (6.7)$$

For the necessary condition to be met the null hypothesis has to be rejected. One-way analysis of variance (ANOVA) is used to test the null hypothesis. Although for results of ANOVA to be meaningful it is necessary that the feature value distributions are close to a normal distribution, the analysis is known to be robust to modest violations of the normality condition (Chatfield, 1981). Usually it is assumed that feature values are distributed approximately normally with about the same variances. Alternatively, this assumption can be tested.

Usually the ANOVA method is presented in books for a case of equal number of measurements of each testing value. For the case of different  $N_{ij}$  obtained due to different number of character samples and failures of feature extraction algorithms, the variations and numbers of degrees of freedom (d.f.) are recalculated.

Unbiased estimates of standard deviations of feature values for feature  $f_i$  are calculated as

$$s_{ij}^2 = \frac{1}{N_{ij} - 1} \sum_{k=1}^{N_{ij}-1} (f_{ijk} - \overline{f_{ij}})^2 \quad (6.8)$$

for each writer  $j$ , and the number of degrees of freedom is  $N_{ij} - 1$ . The within-writer variation is given by

$$s_{iW}^2 = \frac{1}{m} \sum_{j=1}^m s_{ij}^2 \quad (6.9)$$

and is based on  $\sum_j N_{ij} - m$  d.f. Variation between writers is calculated as

$$s_{iB}^2 = \sum_{j=1}^m N_{ij} \frac{(\overline{f_{ij}} - \overline{f_i})^2}{m - 1} \quad (6.10)$$

and is based on  $m - 1$  d.f.;  $\overline{f_{ij}}$  and  $\overline{f_i}$  are given by (6.1). The total variation

$$\sum_{j=1}^m \sum_{k=1}^{N_{ij}} (f_{ijk} - \overline{f_i})^2 \quad (6.11)$$

must be a sum of within-writer and between-writer variations based on the total number of d.f. of  $\sum_j N_{ij} - 1$ . These results are summarised in Table 6.1.

| Source of variation | Sum of squares   | d.f.             | Mean square |
|---------------------|--|------------------|-------------|
| Between authors     | $\sum_{j=1}^m N_j (\bar{f}_j - \bar{f})^2$             | $m - 1$          | $s_B^2$     |
| Within authors      | $\sum_{j=1}^m \sum_{i=1}^{N_j} (f_{ij} - \bar{f}_j)^2$ | $\sum_j N_j - m$ | $s_W^2$     |
| Total variation     | $\sum_{j=1}^m \sum_{i=1}^{N_j} (f_{ij} - \bar{f})^2$   | $\sum_j N_j - 1$ |             |

Table 6.1: One-way ANOVA for feature relevance test.

If the null hypothesis is true, the ratio of the squared between-writer variation (6.10) to the squared within-writer variation (6.9) for feature  $f_i$ , that is

$$F_i = s_{iB}^2 / s_{iW}^2 \quad (6.12)$$

follows an  $F$ -distribution with  $(m - 1, \sum_j N_{ij} - m)$  degrees of freedom. If the null hypothesis is not true, the  $F$ -ratio should be significantly larger. Given a certain level of confidence the threshold value of  $F$  can be taken from the table of  $F$ -distribution. If the value of  $F_i$  is less than the threshold, the null hypothesis must be accepted at the given confidence level, which means that the feature has no discriminative power — within the variance its values for all writers are indistinguishable. If  $F_i$  is higher than the threshold, the null hypothesis must be rejected, which means that the feature may be relevant.

Results of the application of ANOVA to the feature set is described in Section 6.2.

### 6.1.3 Wrapper approach: search for the best feature sets

In practice the definition of feature relevance given in (John et al., 1994) is hard to apply, since it is necessary to know the conditional probabilities  $P(\vec{f}_k | j)$ . Another approach can be applied though. Suppose there is a set of features and all subsets are found that are equally good for writer classification. Equally good means that the classification accuracies achieved when those subsets are used, do not differ significantly — their average values are indistinguishable. There are three classes of features according to their inclusion in the found feature sets: some features are included in all feature sets, some are not included at all, and the rest are included in some of the sets. The first category of features comprises of *indispensable features*, the second category contains *irrelevant features*, and the rest of the features are *partially relevant*. The categories generally do not coincide with those defined in (John et al., 1994), except for the irrelevant features, hence the terminology is different to prevent confusion.

As discussed earlier (Section 2.8.1), there are two approaches for determining the best feature sets and thus dividing features according to their relevance: filter and wrapper. In the current study the wrapper approach was used. For each feature subset

its performance was measured in a series of experiments with writer classification, producing the average accuracy of classification along with its standard deviation. The experiments used  $n$ -fold cross-validation (Weiss & Kulikowski, 1991). The data were divided into  $n$  approximately equal parts and the classification experiments were then performed  $n$  times. Each time another part of the data was taken out. The rest  $n - 1$  parts were used to train the classifier and the other part was used for test. The  $n$  values of the classification accuracy obtained were averaged.

### The classifier

DistAl, a constructive learning algorithm based on the multi-layer perceptron with spherical threshold units, was chosen as a classification system (Yang et al., 1997). There were several reasons for this choice over other possibilities:

- DistAl does not require any a priori assumptions about network topology. Network topology is determined dynamically in the learning process.
- It is fast in learning because it does not use an iterative algorithm to compute perceptron parameters (weights, thresholds). The most time-consuming part is the calculation of inter-pattern distances for each pair of patterns. However, this needs to be performed only once.
- DistAl is based on a distance metric between patterns which means it can easily be adapted to handle patterns with missing feature values.
- Experiments conducted on both artificial and real data (Yang et al., 1997) demonstrated results of classification comparable to those obtained by other commonly used learning algorithms.

All feature values were treated as real numbers. Having performed several experiments the normalised Manhattan distance was chosen as a distance measure for DistAl because this measure proved to give the best classification results for the problem at hand:

$$d(\vec{f}_{k_1}, \vec{f}_{k_2}) = \frac{1}{M} \sum_{i=1}^M \frac{|f_{ik_1} - f_{ik_2}|}{\max f_i - \min f_i} \quad (6.13)$$

where  $M$  was the number of features,  $\min f_i$  and  $\max f_i$  were the minimum and maximum values of the  $i$ -th feature in the data set respectively.

### Genetic algorithm search

Thus a value was associated with a feature subset that indicated how good the subset was. The total number of different feature subsets that can be formed from a set of

$M$  features is  $2^M - 1$ . When this number was low, an exhaustive search was possible. Otherwise it was decided to use genetic algorithm (GA) for the search.

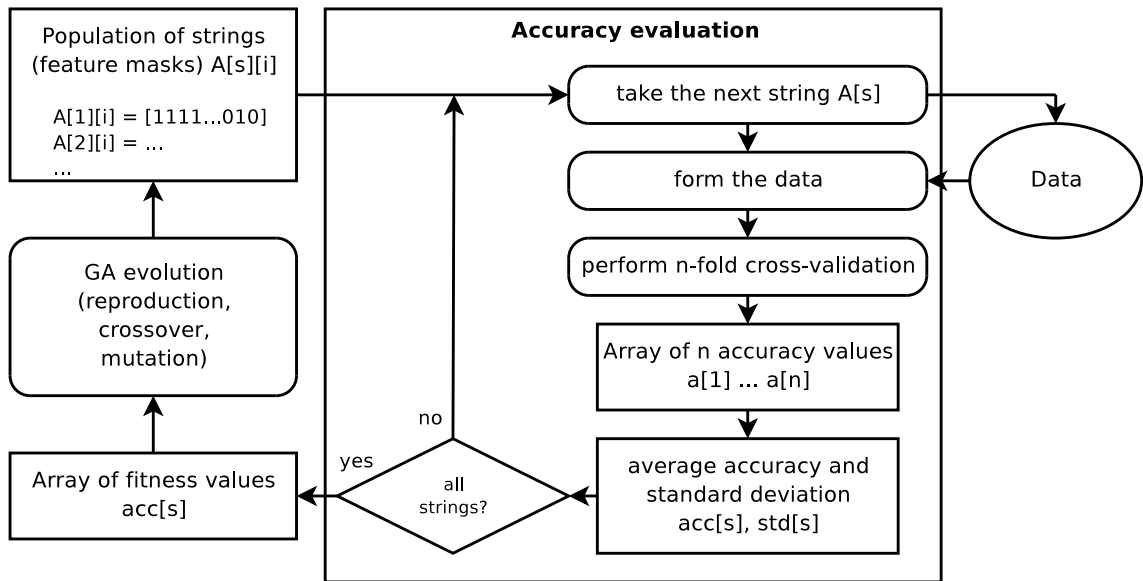


Figure 6.2: Wrapper feature subset search with GA.

From the studies of De Jong (Vafaie & De Jong, 1992) GAs have been extensively used to solve problems of feature selection in pattern recognition. Successful use of a GA together with the DistAl algorithm has also been demonstrated (Yang & Honavar, 1998b). Use of a GA has several advantages to other commonly used methods:

- GAs perform search from a number of starting point rather than from a single one. This makes the search
- GAs have the capability of finding a “good” or even optimal solution for complex problems relatively quickly. Since they perform search from a number of starting point rather than from a single one, they are less likely to get stuck in a local extremum than gradient-based techniques (Goldberg, 1989), or methods like backward and forward selections (Fukunaga, 1990) and branch and bound procedure.
- GAs use only the fitness function itself and not any additional information such as the derivatives. This is very suitable for the case of the feature selection problem by the wrapper method.
- Representation of a feature subset as a string in a GA is straightforward. Fixed length binary strings are used; the number of elements is equal to the total number of features  $M$ ; a value of 1 (0) in a string corresponds to the presence (absence) of the feature in the subset associated with the string.

- GAs are not very sensitive to the values of their parameters. Even when the values are far from optimal a good solution can still be achieved although it may require a larger number of generations to reach the solution. This property is very useful because it means that one need not to be over-concerned about adjusting the values of several parameters which cannot be calculated a priori.

Comparison of performance of GA-based feature selection to that using the sequential backward selection (SBS) is discussed in (Vafaie & De Jong, 1992). Comparison with a number of other feature selection methods is reported in (Yang & Honavar, 1998a).

Figure 6.2 shows the schema of feature selection used in the current study. Results of applications of the feature selection method are presented in Sections 6.3 and 6.4. The GA search, cross-validation, and DistAl network required to be fast and hence were implemented in C++.

## 6.2 Analysis of features extracted from 4 characters

Initial experiments on feature extraction were performed on characters “d”, “y”, “f”, and “t”. The feature set contained the 40 document examiner features listed in Table 6.2. The ANOVA result (column *A*) is explained in Section 6.2.2.

### 6.2.1 Extend of separability

To calculate the extent of separability the feature vectors  $\vec{f}_k$  were formed. For each writer a different number of characters were available. Let the number of samples per writer be  $n_d$ ,  $n_y$ ,  $n_f$ , and  $n_t$ . The number of feature vectors formed is  $\min(n_d, n_y, n_f, n_t)$ . The vectors were formed by combining features from a sample of “d” with features from samples of “y”, etc. of the same writer. The samples were chosen randomly. The samples were taken from 30 writers, the maximum number of feature vectors per writer was 24, and the minimum number was 15.

The extent of separability was calculated as in equations (6.2)–(6.5). A normalised Manhattan distance (eq. (6.13)) was used. The value of the extent of separability calculated according to eq. (6.5) was 0.1, which indicated a good separability of the 40 writers in the domain of the 40 features, as explained in Section 6.1.1.

|     | <i>feature</i>                | $f_i$    | <i>A</i> |     | <i>feature</i>        | $f_i$    | <i>A</i> |
|-----|-------------------------------|----------|----------|-----|-----------------------|----------|----------|
| 'd' | Height                        | $f_1$    | r        | 'f' | Height to width ratio | $f_{21}$ | r        |
| 'd' | Width                         | $f_2$    | r        | 'f' | Slant                 | $f_{22}$ | r        |
| 'd' | Height to width ratio         | $f_3$    | r        | 'f' | Number of loops       | $f_{23}$ | u        |
| 'd' | Relative height of ascender   | $f_4$    | r        | 'f' | Main loop area        | $f_{24}$ | u        |
| 'd' | Slant of ascender             | $f_5$    | r        | 'f' | Main loop length      | $f_{25}$ | u        |
| 'd' | Final stroke angle            | $f_6$    | r        | 'f' | Main loop slant       | $f_{26}$ | u        |
| 'd' | Fissure angle                 | $f_7$    | r        | 't' | Height                | $f_{29}$ | r        |
| 'y' | Height                        | $f_8$    | r        | 't' | Width                 | $f_{30}$ | r        |
| 'y' | Width                         | $f_9$    | r        | 't' | Height to width ratio | $f_{31}$ | r        |
| 'y' | Height to width ratio         | $f_{10}$ | r        | 't' | Slant                 | $f_{32}$ | r        |
| 'y' | Relative height of descender  | $f_{11}$ | r        | 't' | Position of t-bar     | $f_{33}$ | r        |
| 'y' | Descender loop completeness   | $f_{12}$ | r        | 't' | Bar width             | $f_{34}$ | r        |
| 'y' | Completeness of loop at $Y_T$ | $f_{13}$ | i        | 't' | Left part of bar      | $f_{35}$ | u        |
| 'y' | Slant at $Y_T$                | $f_{14}$ | r        | 't' | Right part of bar     | $f_{36}$ | u        |
| 'y' | Slant of descender            | $f_{15}$ | r        | 't' | Top part of stem      | $f_{37}$ | r        |
| 'y' | Final stroke angle            | $f_{16}$ | r        | 't' | Bottom part of stem   | $f_{38}$ | r        |
| 'y' | Presence of point $D$         | $f_{17}$ | u        | 't' | Number of loops       | $f_{39}$ | u        |
| 'y' | Descender loop length         | $f_{18}$ | u        | 't' | Main loop area        | $f_{40}$ | u        |
| 'f' | Height                        | $f_{19}$ | r        | 't' | Main loop length      | $f_{41}$ | u        |
| 'f' | Width                         | $f_{20}$ | r        | 't' | Main loop slant       | $f_{42}$ | i        |

Table 6.2: First feature set: character from which the feature was extracted, feature name, feature index, and ANOVA result (in “A” column): relevant, irrelevant, undefined. Note that term “relevant” here means only “passed the ANOVA test”, which means the feature satisfies a necessary condition of being relevant.

### 6.2.2 Analysis of variance

ANOVA was performed for each of the 40 features. The levels of significance were chosen to be 0.1 and 0.01. Thus, if the value of  $F_i$  (eq. (6.12)) was more than that of the  $F$ -distribution (taken from a table), the null hypothesis was rejected with the probability of error of 10% (1%). Otherwise the null hypothesis was accepted.

The number of d.f. of the numerator in eq. (6.12) is the same for each feature and equal to 29. The number of d.f. of the denominator is large (of the order of several hundreds) for all features. That is why threshold values  $F_{0.01}$  and  $F_{0.1}$  calculated for different features with different numbers of d.f. in the denominator of  $F_i$  did not differ much from  $F_{0.1}(29, \infty) \approx 1.46$  and  $F_{0.01}(29, \infty) \approx 2.034$  correspondingly.

There were three classes of features according to the value of  $F_i$ .

1. Features for which  $F_i < F_{0.1}$  and hence the null hypothesis was accepted.

There were two features for which the null hypothesis was accepted:  $f_{13}$ , loop completeness at  $Y_T$  of “y” and  $f_{42}$ , main loop slant of “t”. They are marked with *i* in Table 6.2.

2. *Features for which  $F_{0.1} \leq F_i \leq F_{0.01}$ .*

Rejection of the null hypothesis could be a decision in this case, but the probability of error in making such a decision is considerably high. It was decided to not classify these features. In Table 6.2 they are marked with *u*. Loop features of “f” and “t”  $f_{23} \dots f_{26}, f_{39} \dots f_{41}$  were in this category as well as  $f_{18}$ , *descender loop length of “y”*. Other features included in this category were  $f_{35}$  and  $f_{36}$ , *left part of t-bar* and *right part of t-bar*, *presence of point D of “y”*  $f_{17}$ .

3. *Features for which  $F_i > F_{0.01}$  and the null hypothesis was rejected.*

All other features belonged to this category and are marked with *r* in Table 6.2.

It was found that loop at point  $Y_T$  of character “y” could not be detected reliably because it was a hidden loop in a number of cases and when it wasn’t a hidden loop it was often too small and was transformed into a blob by binarisation. It was decided to not extract this feature in further experiments unless a new preprocessing algorithm and a reliable extraction method are found.

Presence of point *D*, that is, crossing of the base part of “y” by the descender was also excluded from further experiments. Also right and left parts of t-bar and top and bottom parts of t-stem were abandoned because of unclear discriminative power and/or high extraction error rate. It seems that successful measurement of these features depend on accurate extraction of pen strokes. Hence, a more accurate skeletonisation is a necessary condition for these features to be useful (although this may not be sufficient).

Extraction of the number of loops, detection of the largest (main) loop, and extraction of its features was also given up. Instead of comparing the largest loop which could correspond to different strokes in different samples, it was decided to extract document examiner features always from the corresponding strokes on different samples. For character “f” it was decided to extract presence of a loop at the top and bottom points of the stem. Although it may seem that the problem of such an extraction is similar to that of extraction of loop features at point  $Y_T$  of “y”, extraction of loops from top and bottom points of “f” stem is more robust because the loops tend to be more prominent and are not often changed to blobs by preprocessing. Besides, extraction of a binary feature (presence/absence of a loop) is more accurate than extraction of the loop completeness since no accurate tracing of the loop contour is needed.

Extraction of the length of descender loop of “y” was abandoned because of high extraction error rate.

Extraction of final stroke angle was improved (see Section 5.9.2), which enabled to extract such features with considerably higher accuracy.

A frequent combination of characters (grapheme) was used in feature extraction. The grapheme of choice was “th”, hence it was decided to substitute character “t” with grapheme “th” in future experiments.

### 6.3 Analysis of features of 3 characters and 1 grapheme

Features extracted from characters “d”, “y”, and “f”, and grapheme “th” using raster skeletonisation are listed in Table 6.3. There were in total 31 features in the set. In this section grapheme “th” is sometimes referred to as a character.

|     | <i>feature</i>               | $f_i$    | <i>set</i> |      | <i>feature</i>              | $f_i$    | <i>set</i> |
|-----|------------------------------|----------|------------|------|-----------------------------|----------|------------|
| ‘d’ | Height                       | $f_1$    | 1          | ‘f’  | Height                      | $f_{19}$ | 1          |
| ‘d’ | Width                        | $f_2$    | 1          | ‘f’  | Width                       | $f_{20}$ | 2          |
| ‘d’ | Height to width ratio        | $f_3$    | 2          | ‘f’  | Height to width ratio       | $f_{21}$ | 1          |
| ‘d’ | Relative height of ascender  | $f_4$    | 2          | ‘f’  | Slant                       | $f_{22}$ | 2          |
| ‘d’ | Slant of ascender            | $f_5$    | 2          | ‘f’  | Presence of loop at $F_T$   | $f_{27}$ | 3          |
| ‘d’ | Final stroke angle           | $f_6$    | 3          | ‘f’  | Presence of loop at $F_B$   | $f_{28}$ | 3          |
| ‘d’ | Fissure angle                | $f_7$    | 3          | ‘th’ | Height                      | $f_{43}$ | 1          |
| ‘y’ | Height                       | $f_8$    | 2          | ‘th’ | Width                       | $f_{44}$ | 2          |
| ‘y’ | Width                        | $f_9$    | 2          | ‘th’ | Height to width ratio       | $f_{45}$ | 2          |
| ‘y’ | Height to width ratio        | $f_{10}$ | 2          | ‘th’ | Distance $HC$               | $f_{46}$ | 1          |
| ‘y’ | Relative height of descender | $f_{11}$ | 1          | ‘th’ | Distance $TC$               | $f_{47}$ | 1          |
| ‘y’ | Descender loop completeness  | $f_{12}$ | 2          | ‘th’ | Distance $TH$               | $f_{48}$ | 1          |
| ‘y’ | Slant at $Y_T$               | $f_{14}$ | 2          | ‘th’ | Angle between $TH$ and $TC$ | $f_{49}$ | 1          |
| ‘y’ | Slant of descender           | $f_{15}$ | 2          | ‘th’ | Slant of t                  | $f_{50}$ | 1          |
| ‘y’ | Final stroke angle           | $f_{16}$ | 2          | ‘th’ | Slant of h                  | $f_{51}$ | 1          |
|     |                              |          |            | ‘th’ | Position of t-bar           | $f_{52}$ | 1          |

Table 6.3: Second feature set: character from which the feature was extracted, feature name, feature index, and how this feature was classified according to its relevance: indispensable (1), partially relevant (2), irrelevant (3).

#### 6.3.1 Best feature subset search

Feature extraction was performed from samples of 200 different writers. The feature sets that gave the lowest classification error rates were found for feature sets of each character by exhaustive search. Genetic algorithm search was used to find the best

feature sets for 2-character and 3-character combinations as well as for the full feature set.

For some writers the amount of patterns obtained for one of the four characters was too small because of errors in the feature extraction stage. To make the results of experiments comparable for all single characters and the four-character set, patterns from 165 different writers were used (writers who had less than 15 patterns for any of the character due to errors in feature extraction stage were excluded from the study).

Feature vectors for multi-character combinations were formed by merging feature vectors for separate characters together. Each character gave its feature vector for one and only one combination. Let the feature vectors obtained for a particular writer be  $\vec{f}_1^d \dots \vec{f}_{n_d}^d$ ,  $\vec{f}_1^y \dots \vec{f}_{n_y}^y$ ,  $\vec{f}_1^f \dots \vec{f}_{n_f}^f$ ,  $\vec{f}_1^{th} \dots \vec{f}_{n_{th}}^{th}$  extracted from the writer's samples of characters "d", "y", "f", and "th" respectively,  $n_d$ ,  $n_y$ ,  $n_f$ , and  $n_{th}$  being the number of vectors extracted for each character. In order to form the full vectors for the writer, four character-specific vectors were drawn from the set, without replacement, until any of the character-specific feature vector set was empty. The number of feature vectors formed for the four-character combination was thus  $\min(n_d, n_y, n_f, n_{th})$ . For example, a feature vector might be  $\vec{f} = \{\vec{f}_1^d \vec{f}_{10}^y \vec{f}_3^f \vec{f}_{20}^{th}\}$ . Several pattern sets differing by permutations of character-specific feature vectors for each writer were formed by this method.

For the full feature set GA with sharing was used to find several optimal feature subsets that gave the same classification accuracy. Sharing was performed by modifying the fitness function (Goldberg, 1989). Original fitness  $fit_s$  for feature subset  $x_s$  (represented as a binary string) was equal to the average correct classification rate  $acc_s$ . Modified fitness was

$$\widetilde{fit}_s = \frac{fit_s}{\sum_j (1 - d(x_s - x_j))} \quad (6.14)$$

where  $d(x_s, x_j)$  is a dissimilarity measure:

$$d(x_s, x_j) = \frac{\sum_m |x_{sm} - x_{jm}|}{M} \quad (6.15)$$

Here  $M$  is the number of features, or the length of binary strings  $x_s$  and  $x_j$  representing feature subset, and  $x_{sm}$  is the  $m$ -th bit of string  $x_s$ . Summation in (6.14) is performed across the whole population of strings.

The following parameters were used for GA:

- population size of 50;
- uniform crossover (Syswerda, 1989) with probability of 0.6;

- mutation with probability of 0.03;
- replacement strategy in which best strings from parents and offspring form the next generation;
- linear scaling of fitness function with the factor of 2;
- stop condition for 2- and 3-character combination feature sets was reaching 100 generations, stop condition for the best feature sets was “no new good feature subset during last 50 generations”.

DistAl neural network was used as a classifier and 5-fold cross-validation method was used to estimate the writer classification accuracy.

Table 6.4 shows the feature subsets of the whole 4-character feature set that gave the same highest classification accuracy. The accuracy values presented in the table are indistinguishable from each other at 1% significance level.

| ‘d’-part<br>$f_1 \dots f_7$ | ‘y’-part<br>$f_8 \dots f_{12} f_{14} \dots f_{16} f_{18}$ | ‘f’-part<br>$f_{19} \dots f_{22} f_{27} f_{28}$ | ‘th’-part<br>$f_{43} \dots f_{52}$ | accuracy | $\sigma_{accuracy}$ |
|-----------------------------|---|---|------------------------------------|----------|---------------------|
| 1111100                     | 01111001  | 111001  | 111111111                          | 0.58     | 0.04                |
| 1101100                     | 10111011  | 111001  | 111111111                          | 0.57     | 0.04                |
| 1110100                     | 11011010  | 111001  | 111111111                          | 0.55     | 0.04                |
| 1111100                     | 11111001  | 111001  | 110111111                          | 0.54     | 0.05                |
| 1110100                     | 11111101  | 101000  | 111111111                          | 0.54     | 0.05                |
| 1111000                     | 11111010  | 111001  | 111111111                          | 0.53     | 0.04                |
| 1111100                     | 11111001  | 111001  | 111111111                          | 0.53     | 0.04                |
| 1111100                     | 11110011  | 111001  | 101111111                          | 0.53     | 0.04                |

Table 6.4: Optimal feature sets, accuracy values, and standard deviations. Bit 1 corresponds to presence of the feature in the subset, bit 0 to absence of it.

Division of the feature set into three categories of indispensable, partially relevant, and irrelevant features was done according to the results presented in Table 6.4. The results are shown in Table 6.3; 1 corresponds to the set of indispensable features, 2 corresponds to the set of partially relevant features, and 3 corresponds to the set of irrelevant features. As seen from the results, the highest number of indispensable features belong to grapheme “th”. The next section discusses this observation in more detail.

### 6.3.2 Classification accuracy

Dependence of the classification accuracy on the number of writers whose samples needed to be classified was measured on best feature subsets of separate characters feature sets “d”, “y”, “f”, “th”, and on the best subsets of the feature sets of 2-, 3-, and 4-character combinations. Figure 6.3 shows the degradation of the classifier

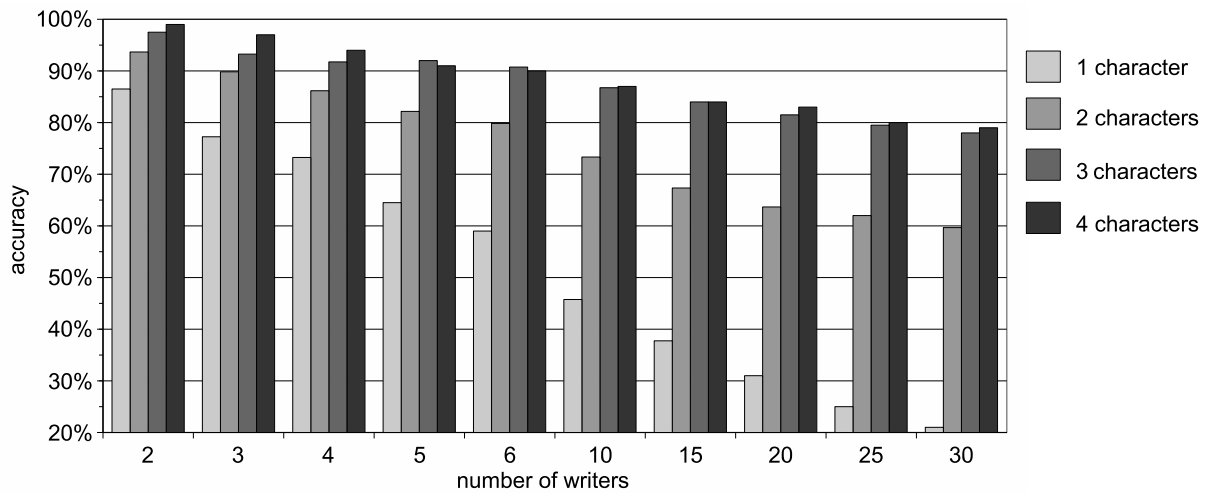


Figure 6.3: Degradation of writer classification accuracy with increasing number of writers when features of different characters/grapheme are included in the feature set.

performance due to the increasing number of different writers. For 2- and 3-character combinations the accuracies obtained on different combinations were averaged. As seen from the figure, inclusion of more features from more characters into the feature set used for writer classification both increases the classification accuracy and makes the decrease of the performance less rapid. It is worth noticing that there is little difference in the classification accuracy between feature sets of 3-character and 4-character combinations. This resulted from the fact that inclusion of features of grapheme “th” made the most significant contribution to the performance of the classifier compared to inclusion of features of any of the three single characters. Indeed, among the four possible 3-character combinations only one combination did not have “th”, whereas among the six 2-character combinations only a half included “th”. If the classification accuracy was significantly higher with features of “th” included, the average accuracy of 3-character combination should be almost the same as that achieved on the full 4-character feature set.

Figures 6.4(a) and 6.4(b) confirm this conclusion. Figure 6.4(a) shows the writer classification accuracy when feature sets of each single character were used. Accuracy values achieved on the feature sets of each of the 3 single characters were averaged and shown in the chart in the light gray, whereas the accuracy values achieved on the feature set of “th” are shown in the dark gray. Results for 2-character combinations are shown in Figure 6.4(b). In both cases significantly higher classification accuracy was achieved if features of “th” were used.

As the number of writer increases, the writer classification accuracy drops. This shows that either the classes are not completely separable in the feature space (and

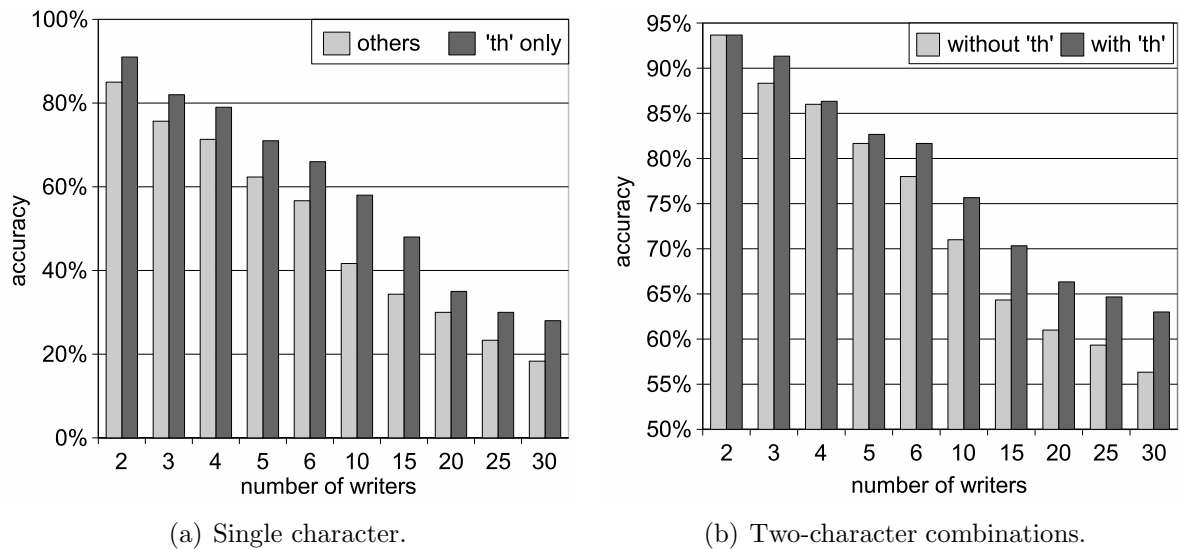


Figure 6.4: Dependence of the writer classification accuracy on the number of writers with features of grapheme “th” in the set and without them.

as their number increases so does the overlap between them), or the classifier fails to find good approximations to the decision boundaries. Recently the problem has been addressed by transformation of the  $N$ -class classification problem into a 2-class one. For each pair of patterns the distance between them is calculated. This gives two types of inter-pattern distances: those calculated between patterns of the same class and those calculated between patterns of different classes. Such a transformation is referred to as dichotomy transformation (Cha & Srihari, 2000c). Instead of the identification problem, where the output of the classifier is the class label, the problem becomes that of verification: given two patterns the classifier returns the answer about whether the patterns belong to the same class or not. The method has its drawbacks though: for example, if two classes in feature space are clearly separable but the distance between them is comparable to their size, the dichotomiser necessarily demonstrates poor verification accuracy (Cha, 2001). One of the main advantages of the method is that it can be used even when the number of classes is very large or infinite. In problems like person recognition based on biometrics it has been used for fingerprints (Pankanti et al., 2002) and iris (Daugman, 1993).

Table 6.5 shows the values of writer classification accuracy achieved when features of each of the character and grapheme were used. The full set of 165 writers were classified. As seen from the table, different characters possess different discriminating power. The experiments revealed that accuracy of writer identification obtained when three separate characters were used was the highest for character “f” and the lowest for character “d”. This result is in agreement to that obtained by Zhang and

Srihari (Srihari et al., 2002) although they used a completely different feature set (purely computational features, see Section 2.4.1). The experiments also showed that the discriminating power of grapheme “th” is significantly higher than that of any of the three single characters considered.

|             |    |    |    |    |     |
|-------------|----|----|----|----|-----|
| Character   | d  | y  | f  | th | all |
| Accuracy, % | 16 | 20 | 26 | 36 | 58  |

Table 6.5: Accuracy of writer identification using features of different characters.

Because of that, it was decided to focus on feature extraction from grapheme “th” only in further experiments. To improve the feature extraction accuracy as well as to make extraction of more document examiner features possible, the study was focused on a new skeletonisation method. The new skeletonisation method was described in detail in Chapter 4. The next section discusses the discriminative power of features extracted when the vector skeletonisation was used. It also compares the results to those obtained with the raster skeletonisation.

## 6.4 Analysis of features of grapheme ‘th’

### 6.4.1 Comparison of the skeletonisation methods

Using the new skeletonisation method described in Chapter 4, extraction of the features listed in Table 6.6 was performed. Preliminary test had already shown that new vector skeletons were more precise than the corresponding raster ones (Section 4.3). However, the difference that is indeed significant is the difference in writer classification accuracy when features extracted from vector skeletons are used.

To find the difference in classification accuracy, feature extraction was performed using 150 images of grapheme “th” written by 30 different writers. Only samples of writers with all 5 graphemes processed correctly in both old thinning-based skeletonisation followed by feature extraction and the new vector skeletonisation followed by feature extraction were used in the test, which made a total of 100 samples from 20 writers.

Features extracted from both raster and vector skeletons were called the “original feature set” ( $f_{43} \dots f_{56}$ , 14 features). This set together with 11 more features  $f_{57} \dots f_{67}$  extracted from vector skeleton formed the full feature set. Table 6.7 shows the results of writer classification accuracy obtained when raster skeletonisation was used, when vector skeletonisation was used but when the data containing only the features of the original feature set was fed into the classifier, and when vector skeletonisation was

| <i>Feature</i>                        | $f_i$    | <i>Skeleton</i> | <i>Set</i> |
|---------------------------------------|----------|-----------------|------------|
| Height                                | $f_{43}$ | both            | 1          |
| Width                                 | $f_{44}$ | both            | 2          |
| Height to width ratio                 | $f_{45}$ | both            | 2          |
| Distance $HC$                         | $f_{45}$ | both            | 1          |
| Distance $TC$                         | $f_{47}$ | both            | 1          |
| Distance $TH$                         | $f_{48}$ | both            | 1          |
| Angle between $TH$ and $TC$           | $f_{49}$ | both            | 1          |
| Slant of t                            | $f_{50}$ | both            | 1          |
| Slant of h                            | $f_{51}$ | both            | 1          |
| Position of t-bar                     | $f_{52}$ | both            | 1          |
| Connected/disconnected t and h        | $f_{53}$ | both            | 1          |
| Average stroke width                  | $f_{54}$ | both            | 1          |
| Average stroke pseudo-pressure        | $f_{55}$ | both            | 2          |
| Standard deviation of pseudo-pressure | $f_{56}$ | both            | 3          |
| Standard deviation of stroke width    | $f_{57}$ | vector          | 3          |
| Number of strokes                     | $f_{58}$ | vector          | 1          |
| Number of loops and retraced strokes  | $f_{59}$ | vector          | 1          |
| Straightness of t-stem                | $f_{60}$ | vector          | 1          |
| Straightness of t-bar                 | $f_{61}$ | vector          | 2          |
| Straightness of h-stem                | $f_{62}$ | vector          | 1          |
| Presence of loop at top of t-stem     | $f_{63}$ | vector          | 2          |
| Presence of loop at top of h-stem     | $f_{64}$ | vector          | 2          |
| Maximum curvature of h-knee           | $f_{65}$ | vector          | 1          |
| Average curvature of h-knee           | $f_{66}$ | vector          | 1          |
| Relative size (diameter) of h-knee    | $f_{67}$ | vector          | 1          |

Table 6.6: Third feature set: feature name, feature index, skeleton used for its extraction (vector only or both raster and vector), and how this feature was classified according to its relevance when extracted from vector skeleton: indispensable (1), partially relevant (2), irrelevant (3).

used with the full set of features. The experimental setup was the same as described in Section 6.3.1. DistAl neural network was used as a classifier, 5-fold cross-validation was used to estimate the classification accuracy, and GA (without sharing) was used to search for one optimal feature subset.

| <i>Feature set</i> | <i>Skeletonisation method</i> | <i>Accuracy</i> | $\sigma_{accuracy}$ |
|--------------------|-------------------------------|-----------------|---------------------|
| Original           | Thinning-based raster         | 0.78            | 0.04                |
| Original           | Vector                        | 0.83            | 0.04                |
| Full               | Vector                        | 0.98            | 0.05                |

Table 6.7: Accuracy of writer classification and its standard deviation.

All three values of classification accuracy in Table 6.7 are different from each other at the 1% significance level. As seen from the results, the accuracy of classification when the original feature set was used was slightly higher with the vector skeleton-

isation method. This was most likely caused by more precise measurement of some feature values, especially the angular features like slants and stroke angles. The accuracy of writer classification improved significantly when more features were taken into account. The latter improvement was important because it was the new skeletonisation algorithm that made reliable extraction of those additional features possible. Thus, the advantage of the new skeletonisation is twofold: (i) it allows the extraction of structural features with higher precision than it is possible when the thinning-based skeletonisation is used, and (ii) it allows the extraction of more features which contribute to writer discrimination.

### 6.4.2 Best feature subset search

Optimal feature subset search was performed using a GA with sharing the same way as was performed for feature subsets of 3 characters and 1 grapheme (Section 6.3.1). The feature data extracted from samples of 165 writers using vector skeletonisation were used. The resulting feature subsets that gave the highest classification accuracy are shown in Table 6.8. The accuracy values presented in the table are indistinguishable from each other at the 1% significance level.

| feature set $f_{43} \dots f_{67}$ | accuracy | $\sigma_{accuracy}$ |
|-----------------------------------|----------|---------------------|
| 11111 11111 11000 11101 11111     | 0.67     | 0.04                |
| 10111 11111 11000 11101 01111     | 0.67     | 0.04                |
| 11111 11111 11100 11101 01111     | 0.65     | 0.05                |
| 11011 11111 11100 11111 11111     | 0.64     | 0.04                |
| 11111 11111 11100 11101 01111     | 0.64     | 0.04                |
| 11111 11111 11000 11101 00111     | 0.64     | 0.04                |

Table 6.8: Optimal feature subsets of “th” feature set, accuracy values, and standard deviations. Bit 1 corresponds to presence of the feature in the subset, bit 0 to absence of it. Features are divided into groups of five for convenience.

Division of the features into indispensable, partially relevant, and irrelevant is shown in Table 6.3. A feature is marked as 1 if it is an indispensable, as 2 if it is a partially relevant, and as 3 if it is an irrelevant feature.

## 6.5 Summary of feature analysis

It is necessary to note that the features for which relevance was assessed were not exact document examiner features but a quantifiable representation of them (formalisation). That is why it is incorrect to say that final stroke shape of “d” has no discriminative power. Rather, the final stroke angle measured as the tangent angle to its end point is

not useful for writer classification. It is possible that the current formalisation is not sufficiently descriptive, and another formalisation of a final stroke shape can change this situation. The same applies to the fissure of character “d”. The main purpose of using features that represented loops at the top and bottom points of the f-stem as well as at point  $Y_T$  of “y” was to distinguish between the hand-printed and cursive forms of characters. From the results obtained it is concluded that these features do not help to discriminate between the two character forms effectively.

It was shown that a number of the considered document examiner (structural) features indeed possess discriminating power and thus use of these features for the purpose of writer identification is justified. It was demonstrated that different characters have different discriminative power (Table 6.5), and there exist a noticeable difference in discriminative power between characters in graphemes. Use of features of grapheme “th” resulted in significantly more accurate identification of writers than the use of any of the features of the three single characters. This supports the suggestion that the shape of a character can (and usually is) greatly affected by its adjacent characters — the suggestion which is used in forensic document analysis.

The most important (indispensable), partially relevant, and irrelevant features were identified under the assumption that the data were all genuine unconstrained handwriting. The identification of indispensable features provides the information about which features should be given priority in handwriting analysis when the aim is to identify writers. It is possible though that under conditions different from those used in the current study (normal unconstrained handwriting) the distribution of the features into the three categories according to their relevance may be different. For example, height to width ratio of a character of grapheme is thought to be a more useful feature than both height and width when handwriting is constrained as in cases where handwriting is extracted from forms. Also when grapheme “th” is extracted not from the beginning of words, the presence of loop at the top of the t-stem may more often be included in best feature subsets.

It was demonstrated that accuracy in approximation of handwriting strokes has significant influence on the accuracy of feature measurements and thus proper skeletonisation method is crucial for automatic extraction of document examiner micro features.

## Chapter 7

# Interactive tools for handwriting analysis

This chapter describes some interactive tools developed in the project. As noted before (see Section 1.4.3), the tools are a by-product of the research on document examiner feature extraction and evaluation, rather than a separate objective.

One interactive tool that was used to monitor feature extraction and to correct feature values was presented in Chapter 5, Section 5.9.1. Here two more tools are presented: a tool that facilitates visual comparison of handwritten characters and a tool that was used for implementation and debugging of the skeletonisation program. The latter can also be used to correct the results of skeletonisation.

### 7.1 Tools for visual comparison

Visual comparison of handwriting samples is a part of forensic document analysis. Visual comparison allows the examiner to determine what handwriting elements are consistent in a person's handwriting and hence can be used as features.

In the current project two simple tools for visual comparison of samples were developed. They allow the user to load a number of handwriting samples, overlap them with different levels of opacity, scale them smoothly, and show them one after another in a sequence with adjustable speed. The program was written in C++, and *Trolltech Qt* library was used to build the GUI.<sup>1</sup> Thus the program was cross-platform, although it needed to be recompiled to run on different platforms.

The tools were used to determine the potential document examiner features to extract. Use of both tools made it easier to find similar and different elements among several character or grapheme samples as well as give ideas of how to formalise certain structural features of handwriting.

---

<sup>1</sup>Non-commercial version (2.3) under Windows and open-source GPL version under X11 and Mac.

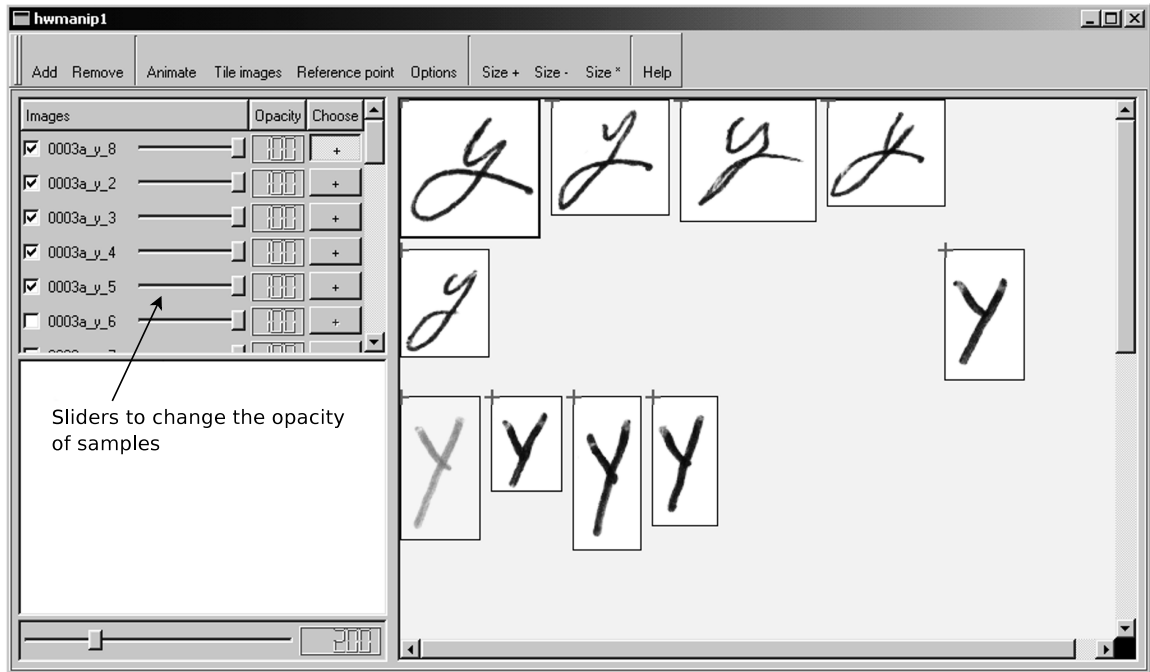


Figure 7.1: Screenshot of the program.

Figure 7.1 shows the screenshot of the main window of the program. It contains three frames: one where all the loaded images are listed, the one below it where selected images are shown in a sequence one after another, and the larger one (main) where the images can be dragged, resized, and overlapped.

### 7.1.1 Image overlapping

Image overlapping allows the user to put several images of characters one on top of another to see the similarities in their shapes. Figure 7.2(a) shows two overlapped images of character “y” written by the same writer, and Figure 7.2(b) shows two overlapped images from different writers. The images were displayed in the main frame if the corresponding checkbox on the left of the image file name was checked. The opacity of each image was adjusted by the corresponding slider next to the image file name as in Figure 7.1. The transparency mechanism did not use the alpha-channel of 32-bit image mode; instead, if an image with opacity  $\alpha$  had to be shown, the colour of each pixel was calculated as

$$colour = \alpha \times colour_1 + (1 - \alpha) \times colour_2 \quad (7.1)$$

where  $colour_1$  is the level of gray of the image pixel and  $colour_2$  is the level of gray of the corresponding background pixel,  $0 \leq \alpha \leq 1$ . Figures 7.3(a)–7.3(c) show an example of two images of character “t” overlapped with different levels of opacity. All images were converted to 8-bit grayscale mode. When several images with over-

lapping parts needed to be shown on the screen, they were shown according to their  $z$ -coordinate. Each image could be smoothly scaled by clicking on it and using “Size+” and “Size-” buttons of the toolbar. Clicking “Size\*” button returned the image to its original size. Changing the size of images allowed to compensate for possible difference in character sizes due to different writing conditions.

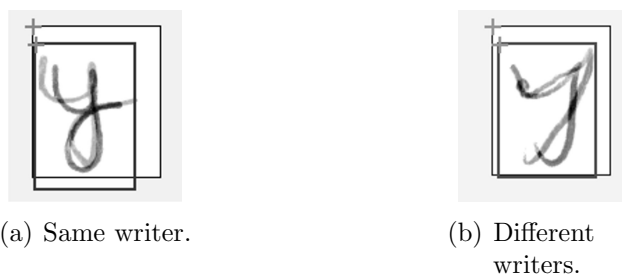


Figure 7.2: Overlapping of handwriting samples.

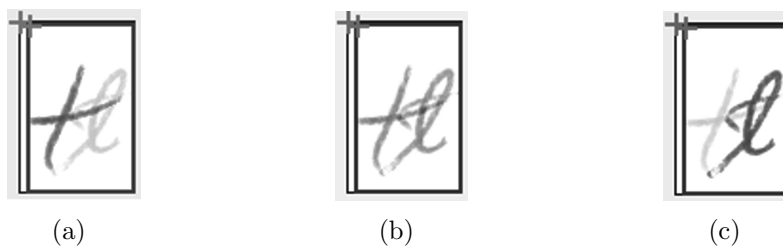


Figure 7.3: Different levels of opacity for sample overlapping.

### 7.1.2 Frame-by-frame image output

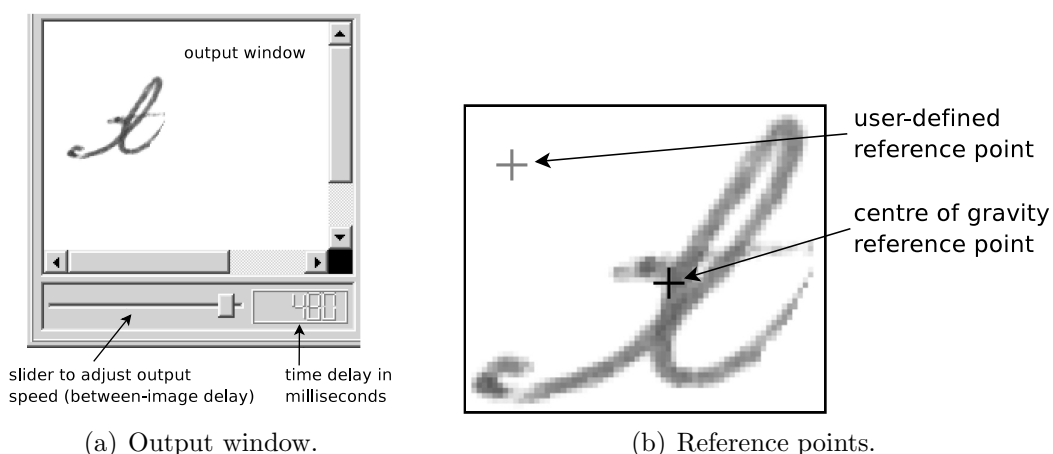


Figure 7.4: Image sequence output window and reference points.

This tool showed the selected images one at a time in a pre-defined sequence and in a pre-defined positions relative to each other. The output window is shown in Figure 7.4(a). The program allows the user to change a time delay between the

consecutive frames, thus adjusting the speed of image display. In order to control relative positions in which images were displayed six reference points were used. Image output routine ensured that the selected reference points have the same coordinates for all selected images during the image output. Four reference points corresponded to the four corners of an image making it possible to align images so that they all have one corner at the same position. Another reference point was taken to be the centre of gravity of each image. Images could thus be displayed with coinciding centres of gravity. An example of this reference point is displayed as a black cross in Figure 7.4(b) (in the program the colour is purple). The other reference point was user defined. The user could define the reference point for an image by clicking on it while holding “Ctrl” key. Alternatively, the user could overlap several images in the main frame and click “*Reference point*” button in the toolbar to define such a reference point that the selected images would be displayed in exactly the same relative positions as they were arranged in the main frame. An example of the user defined reference point is shown in Figure 7.4(b) as a gray cross (green in the program).

Image output options accessible by clicking “*Options*” button in the toolbar allowed the user to change the currently used reference point and adjust other parameters of the program such as the highest and the lowest values of the time delay and the image scaling step.

## 7.2 Interactive skeletonisation tools

Interactive skeletonisation tools were developed in the process of developing the skeletonisation algorithm. The tools enabled easy creation of test skeletons for tuning and debugging of the skeletonisation program as well as correction of some errors occurred during stage 1 (vectorisation). The interactive programs were written in Matlab.

A screenshot of one of the two programs is shown in Figure 7.5. The program allows the user to open a sample image, open and save a skeleton, create a new skeleton or change the existing one by adding and removing branches and adjusting knots. For example, to adjust a knot the user needed to click “*Adjust knot*” button and then click on the knot to be adjusted. The knot became highlighted and with the second click the new position of the knot was determined. Figure 7.5 shows an example of such knot adjustment.

A skeleton was stored in a structure that contained the original image of a character, the initial skeletal branches resulted from stage 1, the strokes resulted from

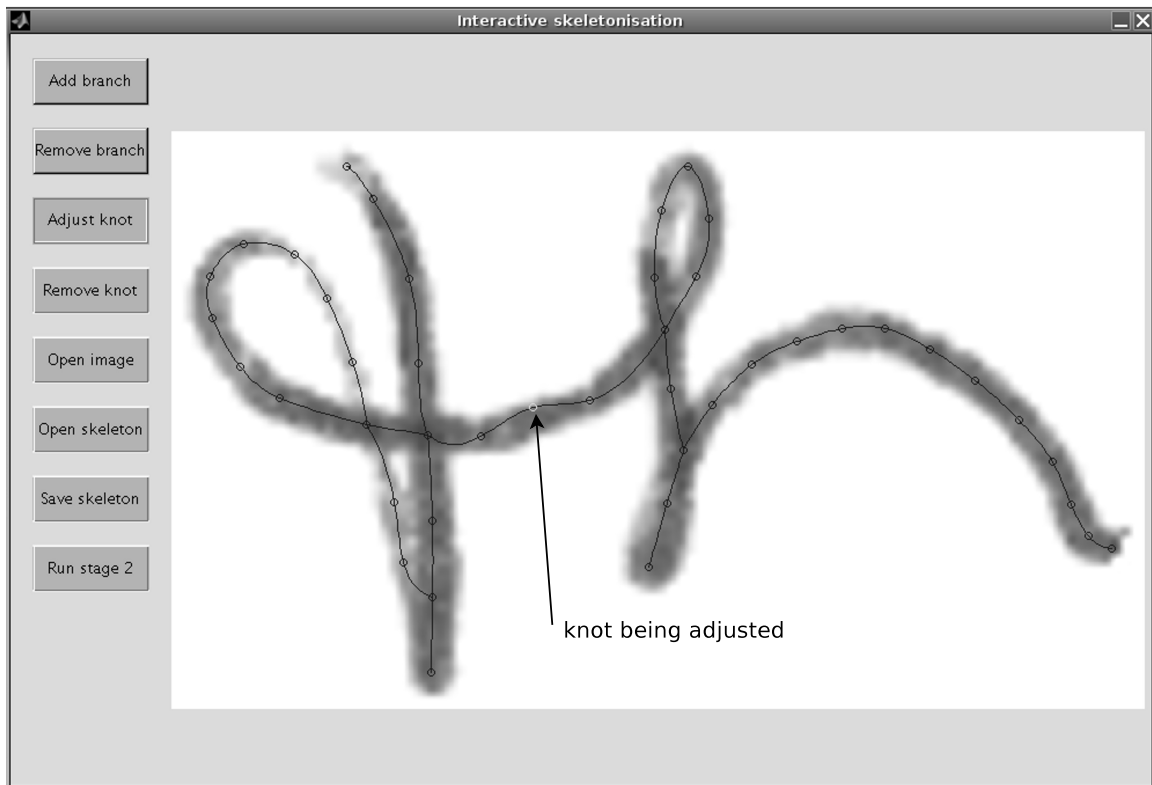


Figure 7.5: Interactive skeletonisation window.

stage 2, the adjustments of coordinates for each spline knot resulted from stage 3 (see Section 4.2.2), and additional information that helped to determine what went wrong and where, in case of an error. The structure could be saved to and read from a file.

The interactive program allows the user to either perform manual vectorisation or correct it, thus decoupling stages 1 and 2 so that each one could be developed assuming correct input data. The program also allows the user to run stage 2 on the current sample with the corresponding initial skeletal branches and see the results in two separate windows: one that showed the growth of maximum of fitness function and the other that displayed the found optimum connection of the skeletal branches. The second window appeared only when the search was over.

Fine adjustment of spline knots of each curve in a skeleton was performed by stage 3 of the skeletonisation method (Section 4.2.2), and an interactive program with GUI was written to conveniently adjust the value of  $\lambda$  in the optimisation function in eq. (4.4). A screenshot of the program is shown in Figure 7.6. The program allowed the user to open a sample (both a skeleton and the corresponding image), and try stage 3 adjustment of spline knots with different values of  $\lambda$ , and save the results.

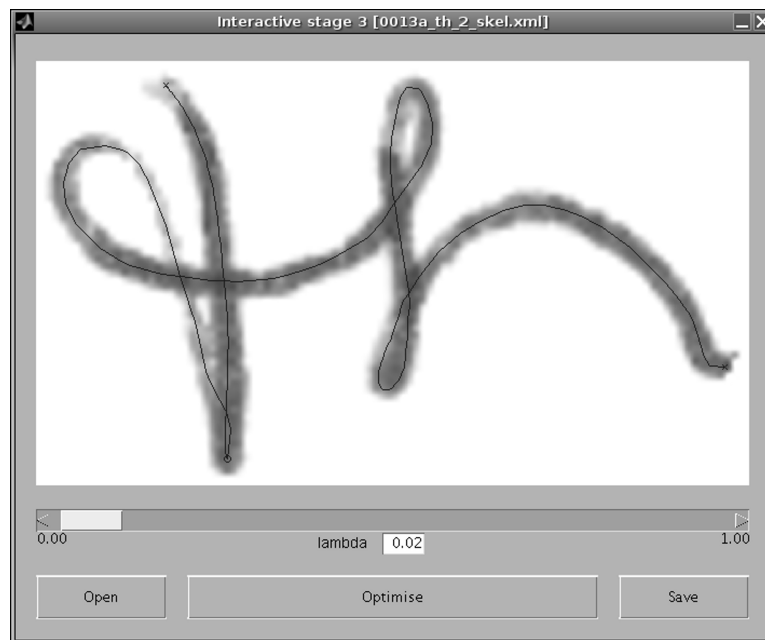


Figure 7.6: Interactive stage 3.

# Chapter 8

## Conclusions and recommendations for further research

This chapter summarises the thesis and states the conclusions drawn from the research investigations. Contributions of the project are highlighted in Section 8.2. Section 8.3 discusses possible directions of future research.

### 8.1 Conclusions

To evaluate the methods used by forensic document examiners in authorship detection, extraction and analysis of structural (document examiner) features were performed during the project. The features studied were micro features, that is, features extracted at a character level. Four characters (“d”, “y”, “f”, “t”) and one grapheme (“th”) of the Latin alphabet were chosen for the study. The rationale behind the choice of the characters and grapheme was their high frequency of occurrence in normal English writing as well as the presence of either ascenders or descenders (see Section 3.3). Selection of features was based on the list of features commonly used by professional document examiners, particularly the list of *21 discriminating elements of handwriting* (Huber & Headrick, 1999).

Automatic extraction of the document examiner features was carried out, followed by a study of the discriminative power of the features. The results and conclusions drawn from the study of feature extraction and feature discriminative power assessment are presented below.

#### 8.1.1 Extraction of document examiner micro features

Samples of characters “d”, “y”, “f”, “t”, and grapheme “th” were manually extracted from handwriting samples of the CEDAR letter (Section 3.2). Samples of characters “d” and “y” were extracted from the ending positions in the words to minimise the

influence of the neighbour characters; for the same reason samples of character “t” and grapheme “th” were extracted from the beginning positions in the words. All available samples of character “f” were extracted since there were only 8 of them. Appendices A.1–A.4 show the exact position from which character samples were extracted.

Automatic extraction algorithms were developed to extract the micro features. The complete list of the extracted features is presented in Tables 3.2–3.6.

A number of document examiner features were formalised, that is, defined in strict mathematical terms. This allowed those features to be measured from character images by means of automatic computer programs. The formalisation removed subjectivity from the process of feature extraction and enabled repeatability of the measurements. The list of the *21 discriminating elements of handwriting* presented on page 16 shows those document examiner features that were either fully or partially formalised and studied in this research in *italic*.

It was demonstrated that the selected characters/grapheme are not all equal in their discriminative power. Features of the chosen grapheme (“th”) made a significantly larger contribution to the discriminative power of the whole feature set than features of any other character: classification accuracy of 36% was achieved on a set of samples from 165 different writers when the features of grapheme “th” were used, compared to an accuracy of 26% achieved when the features of character “f” were used (see Section 6.3.2). This supports the hypothesis that shapes of handwritten characters are influenced by the adjacent characters, and short frequent character combinations extracted from the same position in words are more consistent in their shape than separate characters from different positions in words.

Assessment of the discriminative power of features allowed the identification of problems in formalisation of some features. A set of simple loop features is not sufficient to represent the variety of loop shapes. The shape of loops is an important feature in forensic document analysis and most likely an appropriate set of features to represent various loop shapes can give a good formalisation of that particular document examiner feature.

Automatic feature extraction algorithms were developed that took into account different possible shapes of characters. The algorithms are character-specific, which means the knowledge of the character is necessary for successful feature extraction. This satisfies the usual conditions of forensic handwriting analysis where the content of documents is assumed to be known. A correct feature extraction rate of about 85%

was achieved for each character and grapheme when a traditional thinning-based skeletonisation algorithm was used for handwriting stroke approximation (see Section 5.9.2). Extraction algorithms were improved for *final stroke angle* features ( $f_6$  and  $f_{16}$ ) that produced a high extraction error rate of 23%. Instead of fitting a line into a set of points corresponding to the pixels of a final stroke, a line or an ellipse was fitted depending on the stroke curvature, which resulted in an improvement of the final stroke angle extraction accuracy to a more acceptable level of 11%.

When a specially designed vector skeletonisation algorithm was used, the automatic feature extraction improved from 85% to 94% accurate on the same data set. The observed failure of feature extraction in 6% of cases was due to unusual character formation styles and would be reduced further by creating more sophisticated algorithms that take into account greater number of character shapes. However, the problem is that there exists a high variability in character shapes among those 6% of character samples, and taking into account all of them may be difficult. For practical use of the extraction algorithms to help forensic document examiners it makes sense to allow the user to aid the feature extraction in cases when automatic feature extraction fails. The user may be asked to identify several key points on a sample image and thus virtually eliminate the chances of an error of extraction algorithms due to incorrect conclusions about the character shape. This technique was actually used in the current study as a temporarily solution when it was necessary to focus on debugging of a certain part of the feature extraction program under the assumption that the character shape was analysed correctly.

### 8.1.2 Discriminative power of features

Analysis of the discriminative power of the features examined was conducted in several stages. The initial feature set extracted features from four characters “d”, “y”, “f”, and “t”. The extent of separability, which indicates the difference between intra-writer and inter-writer variation, was 0.1 for a set of 30 writers, which indicates good separability<sup>1</sup>. However, a check of a necessary condition of a feature significance conducted for each feature of the set using the ANOVA method revealed several features that did not possess any discriminative power and were thus useless — at least in the way they were specified and measured in this research. It is important to note that when a feature is said to be significant, relevant, irrelevant, etc. this refers

---

<sup>1</sup>Separability of classes increases as the value of the extent of separability approaches 0, see Section 6.1.1

to the feature that was actually measured, not to the document examiner feature it seeks to represent. Thus it is possible that while document examiner feature “*loop shape*” may be relevant, the loop features that were specified and measured are not relevant because they do not adequately represent detailed loop shapes. Following the results of the ANOVA test, extraction of several features (number of loops, length, area, and slant of loops, presence of point *D* in character “y” (see Figure 3.2(d))) was abandoned, and character “t” was changed to grapheme “th”. This analysis stage showed that the feature set indeed can be used for writer discrimination, although some features do not possess a discriminative power. A new (second) set of 31 features extracted from characters “d”, “y”, “f”, and grapheme “th” was formed for further study. Some of the features in the new set were the same as in the first set, and some were new (see Figure 5.11).

Analysis of the features of the second feature set was conducted in more detail. Samples from 165 different writers were used, and a search for the best feature subsets giving the highest writer classification accuracy was implemented. Firstly, discriminative power of each character and grapheme was assessed by searching for the best feature subset among the features of each character/grapheme. It was found that the discriminative power of grapheme “th” exceeds that of any of the three characters (see Section 8.1.1 above for details). It was also found that the degradation of classification accuracy when the classification was performed among different number of writers was less rapid when features of all three characters and one grapheme were used, as compared to feature sets of three-character combinations, two-character combinations, and single characters (see Figure 6.3). When the optimal feature sets of any single character/grapheme were used, the average writer classification accuracy fell below 20% for sample sets containing samples from more than 30 writers, while the optimal feature set of all three characters and the grapheme gave the classification accuracy of about 87% under the same conditions.

Secondly, the search for the best feature subsets was conducted using the wrapper approach. A genetic algorithm with sharing was used to search for the feature subsets that give the highest classification accuracy. The fitness function was taken to be the accuracy of writer classification measured by 5-fold cross-validation. The DistAl constructive neural network was used as a classifier (see Section 6.1.3).

In order to avoid ranking of features by a single measure, which implies monotonicity (“feature with higher rank is better than feature with lower rank”), the features were sorted into three sets according to the number of times they were included in

feature subsets that gave the highest classification accuracy. Indispensable features, partially relevant features, and irrelevant features were identified from the set of features extracted from “d”, “y”, “f”, and “th”. Indispensable features are those that were included in every found optimal feature subset. Thus, removal of any of those features leads to degradation of writer classification accuracy. Partially relevant features are those features that need to be included in an optimal feature set, but inclusion of all of them is not necessary. Inclusion of features from the set of partially relevant features may or may not increase the writer classification accuracy. In forensic document analysis this means that the optimal analysis method is to measure indispensable features first and then add partially relevant features if necessary. The choice of the features can be done based on the experience of a forensic examiner.

It was found that most features in the set of indispensable features are those of grapheme “th” — 8 out of 10 features were classified as indispensable, leaving only *width* ( $f_{44}$ ) and *height to width ratio* ( $f_{45}$ ) in the group of partially relevant features (see Table 6.3). This again suggests that features of graphemes are more helpful in distinguishing writers than features of single characters.

The third feature set was formed from features of grapheme “th” only. A new skeletonisation method was developed that enabled extraction of more structural features. The results are discussed below.

### 8.1.3 Novel skeletonisation for handwriting characters

A new and novel skeletonisation method was developed to approximate handwriting strokes in character images with high accuracy. The method is content-dependent: weight factors in the cost function depend on what character is being skeletonised. For example, it is typical for grapheme “th” to have up to 4 separate strokes, whereas character “y” normally consists of 1–2 strokes. Hence, the cost dependence on the number of strokes needs to be different for grapheme “th” and character “y”.

It was found that use of the new skeletonisation method allowed the extraction of features with less failure rate as compared to the use of a traditional thinning-based skeletonisation method. Representation of handwriting strokes in a vector form also allowed the extraction of more features which could not be extracted reliably from a raster stroke representation. Use of the new skeletonisation method resulted in a significant improvement in the writer classification accuracy even on the same feature set. The writer classification accuracy based on classification of 100 samples from 20 different writers and measured by 5-fold cross-validation improved from 78%

when raster skeletonisation was used to 83% when vector skeletonisation was used. Addition of new features improved the writer classification accuracy further to 98%. Writer classification accuracy using the optimal feature set on a sample set from 165 different writers was 67% (see Section 6.4.2). Thus the new method indeed proved to be very useful for the purpose of writer classification.

Because the current method needs to “know” what character/grapheme is being skeletonised, it is not practical for the area of handwriting recognition.

#### **8.1.4 Interactive tools for forensic handwriting analysis**

Some interactive tools were developed to facilitate this research. A program to compare samples of handwriting characters by overlapping them in different ways was developed and proved to be useful in helping to identify the significant features to extract. Inclusion of such tools into more comprehensive systems for forensic document examination is very useful.

Also, interactive tools for skeletonisation were created. These tools can be used to aid the automatic skeletonisation in difficult cases and thus may also be integrated in a system for forensic document examination if that system uses character skeletons for automatic or interactive feature extraction.

#### **8.1.5 Main conclusion**

The main conclusion drawn from this research is that the methods of analysis of handwriting used by forensic document examiners based on features of handwriting indeed make distinguishing of writers possible. A number of the features that correspond to the structural features of characters commonly used in forensic analysis of handwriting possess discriminative power. Hence, the use of these features to distinguish writers is justified.

It was demonstrated that features of grapheme “th” are more helpful in distinguishing writers than features of the other 3 studied characters. It is reasonable to conclude that features extracted from frequent graphemes or even short words such as “the” exhibit lower intra-writer variation than features of single characters, especially when the character samples are taken from various words and positions in words. It is likely that these features are also more stable under attempts at forgery and disguise.

If the number of writers to distinguish is low and the handwriting is normal the writers can be distinguished with a high accuracy of 67% for 165 different writers and 90% and higher for 6 and less different writers. This means that for a task of writer

identification that involves a small number (under 10) candidate writers that provide their handwriting samples, and one or several questioned handwriting samples, use of the feature set studied in this thesis would give a conclusion about the authorship of questioned samples with the probability of error of about 10%. As mentioned in Section 1.3.2, it is assumed that an expert in forensic handwriting analysis can use more features as well as pick only important features, and, hence, can most likely draw a conclusion about the authorship of questioned samples with far less probability of mistake.

It is an open question whether some of the document examiner features change their values and to what extent in forged or disguised handwriting. It is also not clear how stable these features are in respect to a person's ageing, and how these feature values are affected by illness, drugs, etc. Since the results achieved on normal handwriting samples are encouraging, further research is needed to provide answers to the questions. The method applied to the analysis of the discriminative power of features in this research can be used in studies of feature invariance in forged and disguised handwriting. The method is also applicable in the study of feature stability with respect to ageing. Methods of image processing, especially the skeletonisation method, can also be used in further research.

## 8.2 Research contribution

The following contributions were made in this thesis.

- Formalisation of a number of document examiner micro features and development of algorithms for automatic feature extraction. Some of document examiner features have been used in previous studies, but none of the studies used a large set of such features.
- Development of a novel skeletonisation method aimed at use in forensic analysis of handwriting.
- Comprehensive analysis of the discriminative power of formalised document examiner features. The analysis of feature discriminative power was conducted in several iterations, each iteration preceded by feature extraction. Such an approach enabled the feature set to be refined and the feature extraction algorithms to be improved. Identification of indispensable, partially relevant, and irrelevant features was done. Such division of features into the three categories gives a better idea of feature usefulness than feature ranking.

- Demonstration of the fact that at least some of the features used by forensic document examiners to identify writers indeed possess discriminative power. Using these features writers can be distinguished from one another with high degree of confidence, provided the number of writers is small and the handwriting is normal. This also suggests that cases of forgery and disguise needs to be analysed in a similar manner to find whether some features are consistent under attempts of forgery/disguise.

Appendix C shows the list of publications that resulted from this research.

### 8.3 Suggestions for future research

The next logical step in establishing a scientific basis for forensic analysis of handwriting is a study of consistency of document examiner features when an attempt of forgery or disguise has been made. The main hindrance however is lack of data to conduct such a study and obtain statistically sound results. Also there are different types of forgery such as skilled and unskilled forgery, and imitation of the the original handwriting by means of retracing it. Use of the real data from court cases would be quite suitable, but unfortunately this is not feasible. Collection of the data is a tedious task, but it seems that it needs to be done to make the further study possible.

Several studies have been conducted to determine the influence of various illnesses and influence of drugs on handwriting (Caligiuri et al., 2005; Van Roon et al., 2005; Van Gemmert et al., 1999). A study is needed to establish how stable document examiner features of a person are when the person grows older. It is important to evaluate quantitatively which features are less prone to change with ageing and hence can still be used in forensic analysis, and which features become unreliable. Unfortunately the problem of data availability is present here as well.

Development of interactive software to automate some parts of a forensic examiner's work is another possible direction of further research. Image processing and manipulation is available nowadays in some software for forensic document analysis, as well as extraction of some features (mostly computational). Automatic or interactive extraction of document examiner features will make such software more powerful and convenient to use. The extraction algorithms may be based on those developed in this research project.

# References

- Abutaleb, A. S. (1989). Automatic thresholding of gray-level pictures using two-dimensional entropy. *Computer Vision, Graphics, and Image Processing*, 47, 22–32.
- Almuallim, H., & Dietterich, T. G. (1991). Learning with many irrelevant features. In *Ninth National Conference on Artificial Intelligence*, (pp. 547–552). Anaheim, California: MIT Press.
- Amin, A., & Singh, S. (1997). Machine recognition of hand-printed Chinese characters. *Intelligent Data Analysis*, 1(2), 101–118.
- Ammar, M., Yoshida, Y., & Fukumura, T. (1989). Feature extraction and selection for simulated signature verification. In R. Plamondon, C. Y. Suen, & M. L. Simner (Eds.), *Computer Recognition and Human Production of Handwriting* (pp. 61–76). World Scientific.
- Anquetil, E., & Lorette, G. (1997, August). Perceptual model of handwriting drawing — application to the handwriting segmentation problem. In *Proc. 4th Int'l Conf. Document Analysis and Recognition (ICDAR'97)* (pp. 112–117). Ulm, Germany.
- Bozinovic, R. M., & Srihari, S. N. (1989). Offline cursive script word recognition. *IEEE Trans. Pattern Anal. Machine Intell.*, 2, 68–83.
- Brill, F. Z., Brown, D. E., & Martin, W. N. (1992, March). Fast genetic selection of features for neural network classifiers. *IEEE Trans. Neural Networks*, 3(2), 324–328.
- Bulacu, M., Schomaker, L., & Vuurpijl, L. (1995, August). Writer identification using edge-based directional features. In *Proc. 3rd Int'l Conf. Document Analysis and Recognition (ICDAR'95)* (pp. 937–941). Montreal, Canada.
- Bunke, H., Ammann, R., Kaufmann, G., Ha, T. M., Schenkel, M., Seiler, R., et al. (1997, August). Recovery of temporal information of cursively handwritten words for on-line recognition. In *Proc. 4th Int'l Conf. Document Analysis and Recognition (ICDAR'97)* (pp. 931–935). Ulm, Germany.
- Caesar, T., Gloger, J. M., & Mandler, E. (1995, August). Estimating the baseline for written material. In *Proc. 3rd Int'l Conf. Document Analysis and Recognition (ICDAR'95)* (pp. 382–385). Montreal, Canada.
- Caligiuri, M. P., Teulings, H. L., Filoteo, J. V., Song, D., & Lohr, J. B. (2005, June).

- Quantitative measurement of handwriting in the assessment of drug-induced parkinsonism. In A. Marcelli & C. De Stefano (Eds.), *Proc. 12th Conf. Int'l. Graphonomics Society (IGS2005)* (pp. 300–304). University of Salerno, Salerno, Italy.
- Cha, S.-H. (2001). *Use of distance measures in handwriting analysis*. Doctoral dissertation, State University of New-York at Buffalo, NY, USA.
- Cha, S.-H., Shin, Y.-C., & Srihari, S. N. (1999a). *Algorithm for the edit distance between angular type histograms* (Technical Report No. CEDAR-TR-99-1). SUNY at Buffalo.
- Cha, S.-H., Shin, Y.-C., & Srihari, S. N. (1999b, November). Approximate character string matching algorithm. In *Proc. 5th Int'l Conf. Document Analysis and Recognition (ICDAR'99)* (pp. 53–56). Bangalore, India.
- Cha, S.-H., Shin, Y.-C., & Srihari, S. N. (2000). Approximate string matching for stroke direction and pressure sequences. In *Proceedings of SPIE, Document Recognition and Retrieval VII* (pp. 2–10). San Jose, CA, USA.
- Cha, S.-H., & Srihari, S. N. (2000a). Assessing the authorship confidence of handwritten items. In *Proceedings of 5th Workshop on the Application of Computer Vision (WACV2000)* (pp. 42–47). Palm Springs, CA, USA.
- Cha, S.-H., & Srihari, S. N. (2000b, September). Multiple feature integration for writer verification. In *Proc. 7th Int'l Workshop on Frontiers in Handwriting Recognition (IWFHR-7)* (pp. 333–342). Amsterdam, The Netherlands.
- Cha, S.-H., & Srihari, S. N. (2000c). Writer identification: Statistical analysis and dichotomizer. In *Proceedings International Workshop on Structural and Syntactical Pattern Recognition (SSPR2000)* (pp. 123–132). Alicante, Spain.
- Cha, S.-H., & Srihari, S. N. (2002). A fast nearest neighbor search algorithm by filtration. *Pattern Recognition*, *35*, 515–525.
- Chang, H.-H., & Yan, H. (1999, February). Analysis of stroke structures of handwritten chinese characters. *IEEE Trans. Syst., Man, Cybern. B*, *29*(1), 47–61.
- Chatfield, C. (1981). *Statistics for technology*. London, New York: Chapman and Hall.
- Chen, H.-C., Cha, S.-H., Chee, Y.-M., & Tappert, C. C. (2003, November). The detection of forged handwriting using a fractal number estimate of wrinkliness. In H. L. Teulings & A. W. A. Van Gemmert (Eds.), *Proc. 11th Conf. Int'l. Graphonomics Society (IGS2003)* (pp. 312–315). Scottsdale, AZ, USA.
- Chen, S., Smith, S., Guerra-Salcedo, C., & Whitley, D. (1999, July). Fast and accurate feature selection using hybrid genetic strategies. In *Proc. Congress on Evolutionary Computation (CEC99)*. Washington DC, USA.
- Chen, Y., & Leedham, C. G. (2004, October). Decompose-threshold approach to handwriting extraction in degraded historical document images. In *Proc. 9th*

- Int'l Workshop on Frontiers in Handwriting Recognition (IWFHR-9)* (pp. 239–244). Tokyo, Japan.
- Cheriet, M., Huang, Y. S., & Suen, C. Y. (1992). Background region-based algorithm for the segmentation of connected digits. In *Proc. 11th Int'l. Conf. pattern recognition* (pp. 619–622). The Hague, The Netherlands.
- Chi, Z., & Hong, Y. (1993). Map image segmentation based on thresholding and fuzzy rules. *Electronic Letters*, 29, 1841–1843.
- Chin, A. H. W., & Jennings, A. (1997). Skew detection in handwritten scripts. In *Proc. IEEE Conf. on Speech and Image Technologies for Computing and Telecommunications* (pp. 310–322).
- Chong, S. W. (1996). *Tools for forensic questioned document examination* (Honours Year Project Report). School of Applied Science, Nanyang Technological University.
- Cohen, E., Hull, J. J., & Srihari, S. N. (1991). Understanding handwritten text in a structured environment: Determining ZIP Codes from addresses. *Int'l Journal on Pattern Recognition and Artificial Intelligence*, 5(1 & 2), 221–264.
- Crettez, J.-P. (1995, August). A set of handwriting families: Style recognition. In *Proc. 3rd Int'l Conf. Document Analysis and Recognition (ICDAR'95)* (Vol. 1, pp. 489–494). Montreal, Canada.
- Datta, A., Parui, S. K., & Chaudhuri, B. B. (2001, March). Skeletonization by a topology-adaptive self-organizing neural network. *Pattern Recognition*, 34(3), 617–629.
- Daubert et al. v. Merrell Dow Pharmaceuticals.* (1993). 509 U. S. 579.
- Daugman, J. G. (1993, November). High confidence visual recognition of persons by a test of statistical independence. *IEEE Trans. Pattern Anal. Machine Intell.*, 15(11), 1148–1161.
- Deb, K., & Goldberg, D. E. (1989). An investigation of niche and species formation in genetic function optimization. In *Proceedings of the Third International Conference on Genetic Algorithms* (pp. 42–50). Washington, DC, USA.
- De Stefano, C., & Marcelli, A. (2002, August). From ligatures to characters: A shape-based algorithm for handwriting segmentation. In *Proc. 8th Int'l Workshop on Frontiers in Handwriting Recognition (IWFHR-8)* (pp. 473–478). Ontario, Canada.
- Doermann, D., Intrator, N., Rivlin, E., & Steinherz, T. (2002, August). Hidden loop recovery for handwriting recognition. In *Proc. 8th Int'l Workshop on Frontiers in Handwriting Recognition (IWFHR-8)* (pp. 357–380). Ontario, Canada.
- Doermann, D., & Rosenfeld, A. (1992, June). Recovery of temporal information from static images of handwriting. In *Proc. Conf. Computer Vision and Pattern Recognition (CVPR'92)* (pp. 162–168). Champaign, IL, USA.

- Doermann, D., & Rosenfeld, A. (1993, May). The interpretation and reconstruction of interfering strokes. In *Proc. 3rd Int'l Workshop on Frontiers in Handwriting Recognition (IWFHR-3)* (pp. 41–50). Buffalo, NY, USA.
- Doermann, D., & Rosenfeld, A. (1995). Recovery of temporal information from static images of handwriting. *International Journal of Computer Vision*, *15*, 143–164.
- Duda, R. O. (1994). Elements of pattern recognition. In J. M. Mendel (Ed.), *A prelude to neural networks: Adaptive and learning systems* (pp. 3–34). Englewood Cliffs, New Jersey: PTR Prentice Hall.
- Duda, R. O., & Hart, P. E. (1974). *Pattern classification and scene analysis*. New York: John Wiley and Sons.
- Dunn, C. E., & Wang, P. S. P. (1992). Character segmentation techniques for handwritten text — a survey. In *Proc. 11th IAPR Int'l. Conf. Pattern Recognition* (Vol. II, pp. 577–580). The Hague, The Netherlands: IEEE Computer Society Press.
- Eldridge, M. A., Nimmo-Smith, I., Wing, A. M., & Totty, R. N. (1984). The variability of selected features in cursive handwriting — categorical measures. *Journal of Forensic Science Society*, *24*, 179–219.
- Eu, E. C. M. (1999). *Accurate segmentation of lines and words in unconstrained handwriting* (Final Year Project Report). School of Applied Science, Nanyang Technological University.
- Feng, C., Woch, A., & Plamondon, R. (2002, August). A comprehensive study of two velocity profile models for rapid stroke analysis. In *Proc. 16th Int'l. Conf. Pattern Recognition* (Vol. 4, pp. 52–55). Quebec, Canada.
- Fischler, M. A., & Wolf, H. C. (1994, February). Locating perceptually salient points on planar curves. *IEEE Trans. Pattern Anal. Machine Intell.*, *16*(2), 113–129.
- Found, B., Roger, D., & Schmittat, R. (1994). A computer program designed to compare the spatial elements of handwriting. *Forensic Science International*, *68*, 195–203.
- Found, B., & Rogers, D. (1995). Contemporary issues in forensic handwriting examination. a discussion of key issues in the wake of the starzecpyzel decision. *Journal of Forensic Document Examination*, *8*, 1–31.
- Found, B., & Rogers, D. (1996). The forensic investigation of signature complexity. In M. L. Simner, C. G. Leedham, & A. J. W. M. Thomassen (Eds.), *Handwriting and drawing research: Basic and applied issues* (pp. 483–492). Amsterdam, The Netherlands: IOS Press.
- Found, B., & Rogers, D. (1998). A consideration of the theoretical basis of forensic handwriting examination. *Journal of Forensic Document Examination*, *4*(2), 109–118.
- Found, B., & Rogers, D. (2005, June). Problem types of questioned handwritten

- text for forensic document examiners. In A. Marcelli & C. De Stefano (Eds.), *Proc. 12th Conf. Int'l. Graphonomics Society (IGS2005)* (pp. 8–12). University of Salerno, Salerno, Italy.
- Found, B., Rogers, D., & Herkt, A. (2001a). Comparison of document examiners' opinions on original and photocopied signatures. *Journal of Forensic Document Examination, 14*, 1–13.
- Found, B., Rogers, D., & Herkt, A. (2001b). The skill of a group of forensic document examiners in expressing handwriting and signature authorship and production process opinions. *Journal of Forensic Document Examination, 14*, 15–30.
- Found, B., Rogers, D., & Metz, H. (1999). The objective static analysis of spatial errors in simulations. *Journal of Forensic Document Examination, 12*, 81–99.
- Found, B., Rogers, D., & Schmittat, R. (1997). Recovering dynamic information from static handwriting traces using 'angular differential' software. *Journal of Questioned Document Examination, 6*(1), 17–38.
- Found, B., Rogers, D., & Schmittat, R. (1998). 'Matrix Analysis': a technique to investigate the spatial properties of handwritten images. *Journal of Forensic Document Examination, 11*, 54–74.
- Found, B., Sita, J., & Rogers, D. (1999). The development of a program for characterizing forensic handwriting examiners' expertise: Signature examination pilot study. *Journal of Forensic Document Examination, 12*, 69–80.
- Franke, K., & Köppen, M. (2001). A computer-based system to support forensic studies on handwritten documents. *International Journal on Document Analysis and Recognition, 3*(4), 218–231.
- Franke, K., Schomaker, L., Veenhuis, C., Vuurpijl, L. G., Erp, M. van, & Guyon, I. (2004). WANDA: A common ground for forensic handwriting examination and writer identification. *ENFHEX news — Bulletin of the European Network of Forensic Handwriting Experts, 1/04*, 23–47.
- Franke, K., Schomaker, L., Vuurpijl, L., & Giesler, S. (2003). FISH-new: A common ground for computer-based forensic writer examination. In *Proc. 3rd European Academy of Forensic Science Triennial Meeting* (p. 84). Istanbul, Turkey.
- Freeman, H. (1961). Techniques for the digital computer analysis of chain-encoded arbitrary plane curves. In *Proc. Nat. Electronics. Conf.* (Vol. 17, pp. 412–432).
- Fukunaga, K. (1990). *Introduction to statistical pattern recognition* (2nd ed.). Academic Press.
- Gallant, S. I. (1990, June). Perceptron-based learning algorithms. *IEEE Trans. Neural Networks, 1*(2), 179–191.
- Giacinto, G., & Roli, F. (2001a, August). Design of effective neural network ensembles for image classification purposes. *Image and Vision Computing, 19*(9–10), 699–707.

- Giacinto, G., & Roli, F. (2001b, October). Dynamic classifier selection based on multiple classifier behaviour. *Pattern Recognition*, 34(10), 1879–1881.
- Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley.
- Govindaraju, V., Srihari, S. N., & Shin, Y.-C. (1999). Use of handwriting recognition features in handwriting identification. In *Proceedings 9th Biennial Conference of the International Graphonomics Society* (pp. 73–78). Singapore.
- Greening, C. M., Sagar, V. K., & Leedham, C. G. (1995a, July). Automatic feature extraction for forensic purposes. In *Proc. 5th IEE Int'l Conf. image processing and its applications* (pp. 409–414). Edinburgh, UK.
- Greening, C. M., Sagar, V. K., & Leedham, C. G. (1995b). Handwriting identification using global and local features for forensic purposes. In *Proc. European Convention on Security and Detection* (pp. 272–278). Brighton, UK.
- Guo, Z., & Hall, R. W. (1989). Parallel thinning with two-subiteration algorithms. *Comm. ACM*, 32(3), 359–373.
- Harrison, W. R. (1981). *Suspect documents, their scientific examinations*. Illinois, USA: Nelson-Hall.
- Hew, P. C., & Alder, M. D. (1999, March). *Strokes from pen-opposed extended edges* (Technical report No. TR99-02). The University of Western Australia.
- Hilton, O. (1993). *Scientific examination of questioned documents*. Florida, USA: CRC Hall.
- Ho, B. H. (2001). *Document image segmentation system* (Final Year Project Report). School of Computer Engineering, Nanyang Technological University.
- Holcombe, G., & Leedham, C. G. (1995). An experimental imaging environment for examination of questioned documents. In *Proc. 7th International Graphonomics Conference* (pp. 80–81). London, Ontario, Canada.
- Holcombe, G., Leedham, C. G., & Sagar, V. K. (1996). Computer aided creation of handwriting comparison charts in the forensic examination of questioned documents. In M. L. Simner, C. G. Leedham, & A. J. W. M. Thomassen (Eds.), *Handwriting and drawing research: Basic and applied issues* (pp. 493–507). Amsterdam, The Netherlands: IOS Press.
- Huber, R. A., & Headrick, A. M. (1999). *Handwriting identification: Facts and fundamentals*. CRC Press, LCC.
- Jäger, S. (1996, August). Recovering writing traces in off-line handwriting recognition: Using a global optimization technique. In *Proc. 13th Int'l Conf. Pattern Recognition* (Vol. 2, pp. 150–154). Vienna, Austria.
- Jäger, S. (1997, August). A psychomotor method for tracking handwriting. In *Proc. 4th Int'l Conf. Document Analysis and Recognition (ICDAR'97)* (pp. 528–531). Ulm, Germany.

- Jäger, S. (1998). *Recovering dynamic information from static handwritten word images*. Ph.d. thesis, Albert-Ludwigs-Universität.
- Jain, A., & Zongker, D. (1997). Feature selection: Evaluation, application, and small sample performance. *IEEE Trans. Pattern Anal. Machine Intell.*, 19(2), 153–158.
- John, G. H., Kohavi, R., & Pfleger, K. (1994). Irrelevant features and the subset selection problem. In *Proc. 11th Int'l Conf. Machine Learning (ML94)* (pp. 121–129). Rutgers University, New Brunswick, NJ.
- Kam, M., Fielding, G., & Conn, R. (1997). Writer identification by professional document examiners. *Journal of Forensic Sciences*, 42(5), 778–786.
- Kam, M., Fielding, G., & Conn, R. (1998). Effects of monetary incentives on performance in document examination proficiency tests. *Journal of Forensic Science*, 43(5), 1000–1004.
- Kam, M., Gummadidala, K., Fielding, G., & Conn, R. (2001). Signature authentication by forensic document examiners. *Journal of Forensic Science*, 46(6), 884–888.
- Kam, M., Wetstein, J., & Conn, R. (1994). Proficiency of professional document examiners in writer identification. *Journal of Forensic Sciences*, 39(1), 5–14.
- Kavallieratou, E., Fakotakis, N., & Kokkinakis, G. (2001, December). Slant estimation algorithm for OCR system. *Pattern Recognition*, 34(12), 2515–2522.
- Kégl, B., & Krzyżak, A. (2002, January). Piecewise linear skeletonization using principal curves. *IEEE Trans. Pattern Anal. Machine Intell.*, 24(1), 59–74.
- Kim, G., & Govindaraju, V. (1997). A lexicon driven approach to handwritten word recognition for real-time applications. *IEEE Trans. Pattern Anal. Machine Intell.*, 19(4), 366–379.
- Kittler, J. (2000). A framework for classifier fusion: Is it still needed? In F. J. Ferri, J. M. I. Quereda, A. Amin, & P. Pudil (Eds.), *SSPR/SPR* (Vol. 1876, p. 45-56). Springer.
- Kittler, J., Hatef, M., Duin, R. P. W., & Matas, J. (1998, March). On combining classifiers. *IEEE Trans. Pattern Anal. Machine Intell.*, 20(3), 226–239.
- Kondo, S. (1989). A model of handwriting process and stroke structure of character-figures. In R. Plamondon, C. Y. Suen, & M. L. Simner (Eds.), *Computer recognition and human production of handwriting* (pp. 103–118). World Scientific.
- Kuncheva, L. I. (1998). On combining multiple classifiers. In *Proc. Int'l. Conf. Information Processing and Management of Uncertainty (IPMU'98)* (pp. 1890–1891). Paris, France.
- Kuncheva, L. I., & Jain, L. C. (1999). Nearest neighbor classifier: Simultaneous editing and feature selection. *Pattern Recognition Letters*, 20, 1149–1156.
- Kuncheva, L. I., Whitaker, C. J., Shipp, C. A., & Duin, R. (2000, September). Is

- independence good for combining classifiers? In *Proc. 15th Int'l Conf. Pattern Recognition* (pp. 168–171). Barcelona, Spain.
- Lam, L., Lee, S.-W., & Suen, C. Y. (1992, September). Thinning methodologies — a comprehensive survey. *IEEE Trans. Pattern Anal. Machine Intell.*, 14(9), 869–885.
- Lau, K. K., Yuen, P. C., & Tang, Y. Y. (2003, August). Recovery of writing sequence of static images of handwriting using UWM. In *Proc. 7th Int'l Conf. Document Analysis and Recognition (ICDAR'2003)* (pp. 1123–1128). Edinburgh, UK.
- Lecolinet, E., & Crettez, J. P. (1991, October). A grapheme-based segmentation technique for cursive script recognition. In *Proc. 1st Int'l Conf. Document Analysis and Recognition (ICDAR'91)* (pp. 740–748). Saint-Malo, France.
- Lee, S., Cha, S.-H., & Srihari, S. N. (2002). Combining macro and micro features for writer identification. In *Proceedings SPIE, Document Recognition and Retrieval IX* (pp. 155–166). San Jose, CA, USA.
- Leedham, C. G. (1999). *Image analysis tools for authentication and enhanced classification of handwritten script using forensic techniques* (Final Report No. RG25/95). School of Applied Science, Nanyang Technological University.
- Leedham, C. G., Chen, Y., Takru, K., Tan, J. H. N., & Mian, L. (2003, August). Comparison of some thresholding algorithms for text/background segmentation in difficult document images. In *Proc. 7th Int'l Conf. Document Analysis and Recognition (ICDAR'2003)* (pp. 859–864). Edinburgh, UK.
- Leedham, C. G., Holcombe, G., & Sagar, V. K. (1995). Image processing tools for the interactive forensic examination of questioned document. In *Proc. European Convention on Security and Detection* (pp. 225–228). Brighton, UK.
- Leedham, C. G., & Pervouchine, V. (2005, February). Validating the use of handwriting as a biometric and its forensic analysis. In U. Pal, S. K. Parui, & B. B. Chaudhuri (Eds.), *Document analysis: Proc. Int'l. Workshop on Document Analysis (IWDA'05)* (pp. 175–192). Chennai: Allied Publishers Ltd. (Invited lecture)
- Leedham, C. G., Pervouchine, V., & Tan, W. K. (2004). Quantitative letter-level extraction and analysis of features used by document examiners. *Journal of Forensic Document Examination*. (In press)
- Leedham, C. G., Pervouchine, V., Tan, W. K., & Jacob, A. (2003, November). Automatic quantitative letter-level extraction of features used by document examiners. In H. L. Teulings & A. W. A. Van Gemmert (Eds.), *Proc. 11th Conf. Int'l. Graphonomics Society (IGS2003)* (pp. 291–294). Scottsdale, AZ, USA.
- Leedham, C. G., & Sagar, V. K. (1994, July). Using forensic handwriting analysis techniques to enhance automatic handwritten script recognition and processing. In *Proceedings of the IEE European Workshop on Handwriting Analysis and*

- Recognition: a European Perspective* (pp. 2/1–2/3). Brussels.
- L'Homer, E. (2000, July). Extraction of strokes in handwritten characters. *Pattern Recognition*, 33(7), 1147–1160.
- Liao, C.-W., & Huang, J. S. (1990). Stroke segmentation by Bernstein-Bezier curve fitting. *Pattern Recognition*, 23(5), 475–484.
- Lindblom, B. (1999). Document examination. In G. M. Chayko & E. D. Gulliver (Eds.), *Forensic Evidence in Canada* (2nd ed., pp. 505–525). Aurora: Canada Law Book.
- Liu, K., Huang, Y. S., & Suen, C. Y. (1997, August). Robust stroke segmentation method for handwritten Chinese character recognition. In *Proc. 4th Int'l Conf. Document Analysis and Recognition (ICDAR'97)* (pp. 211–215). Ulm, Germany.
- Liu, K., Huang, Y. S., & Suen, C. Y. (1999, October). Identification of fork points on the skeletons of handwritten Chinese characters. *IEEE Trans. Pattern Anal. Machine Intell.*, 21(10), 1095–1100.
- Maarse, F. J., Schomaker, L., & Teulings, H. L. (1988). Automatic identification of writers. In G. van der Veer & G. Mulder (Eds.), *Human-Computer Interaction: Psychonomic Aspects* (pp. 353–360). Berlin, Heidelberg, New York, London, Paris, Tokyo: Springer-Verlag.
- Marcelli, A., & De Stefano, C. (Eds.). (2005). *Proc. 12th Conf. Int'l. Graphonomics Society (IGS2005)*. University of Salerno, Salerno, Italy.
- Melikhov, K., Tian, F., Seah, H. S., Chen, Q., & Qiu, J. (2004). Frame skeleton based auto-inbetweening in computer assisted cel animation. In *Proc. 2004 int'l conf. cyberworlds*. Tokyo, Japan.
- Mestetskii, L. M., Reyer, I. A., & Sedeberg, T. W. (2002, August). Continuous approach to segmentation of handwritten text. In *Proc. 8th Int'l Workshop on Frontiers in Handwriting Recognition (IWFHR-8)* (pp. 440–445). Ontario, Canada.
- Miller, B. L., & Shaw, M. J. (1996). Genetic algorithms with dynamic niche sharing for multimodal function optimization. In *International Conference on Evolutionary Computation* (pp. 786–791).
- Nagy, G., & Zou, J. (2002, August). Interactive visual pattern recognition. In *Proc. 16th Int'l Conf. Pattern Recognition* (Vol. 2, pp. 478–481). Québec, Canada.
- Nakajima, Y., Mori, S., & Taguekami, S. (1999). Global methods for stroke segmentation. In S. W. Lee (Ed.), *Advances in Handwriting Recognition* (Vol. 34, pp. 225–234). World Scientific.
- Nakajima, Y., Mori, S., Takegami, S., & Sato, S. (1999, July). Global methods for stroke segmentation. *International Journal on Document Analysis and Recognition*, 2(1), 19–23.
- Nicolas, S., Paquet, T., & Heutte, L. (2004, October). Text line segmentation in

- handwritten document using a production system. In *Proc. 9th Int'l Workshop on Frontiers in Handwriting Recognition (IWFHR-9)* (pp. 245–250). Tokyo, Japan.
- Novovičová, J., Pudil, P., & Kittler, J. (1996). Divergence based feature selection for multimodal class densities. *IEEE Trans. Pattern Anal. Machine Intell.*, 28(2), 218–223.
- Nunes, C. M., Britto Jr., A. S. de, Kaestner, C. A. A., & Sabourin, R. (2004, October). An optimized hill climbing algorithm for feature subset selection: Evaluation on handwritten character recognition. In *Proc. 9th Int'l Workshop on Frontiers in Handwriting Recognition (IWFHR-9)* (pp. 365–370). Tokyo, Japan.
- Oliveira, L. S., & Sabourin, R. (2003). A methodology for feature selection using multiobjective genetic algorithms for handwritten digit string recognition. *Int'l Journal on Pattern Recognition and Artificial Intelligence*, 17(6), 903–929.
- Oliveira, L. S., Sabourin, R., Bortolozzi, F., & Suen, C. Y. (2002, August). Feature selection using multi-objective genetic algorithms for handwriting digit recognition. In *Proc. 16th Int'l Conf. Pattern Recognition* (pp. 568–571). Québec, Canada.
- Otsu, N. (1979). A threshold selection method from gray-level histogram. *IEEE Trans. Syst., Man, Cybern.*, 9, 62–66.
- Pankanti, S., Prabhakar, S., & Jain, A. K. (2002). On the individuality of fingerprints. *IEEE Trans. Pattern Anal. Machine Intell.*, 24(8), 1010–1025.
- Parekh, R., Yang, J., & Honavar, V. (2000). Constructive neural-network learning algorithms for pattern classification. *IEEE Trans. Neural Networks*, 11(2), 436–451.
- Pavlidis, T. (2003). 36 years on the pattern recognition front. *Pattern Recognition Letters*, 24, 1–7. (Lecture given at ICPR'2000 in Barcelona, Spain, on the occasion of receiving the K.S. Fu prize)
- Pervouchine, V., & Leedham, C. G. (2004). The validity and usefulness of forensic document examiner features for automatic writer identification. *Pattern Recognition*, 21–39.
- Pervouchine, V., Leedham, C. G., & Melikhov, K. (2005a, March). Handwritten character skeletonisation for forensic document analysis. In *Proc. 20th Annual ACM Symposium on Applied Computing* (pp. 754–758). Santa Fe, NM, USA.
- Pervouchine, V., Leedham, C. G., & Melikhov, K. (2005b, Aug.-Sep.). Three-stage handwriting stroke extraction method with hidden loop recovery. In *Proc. 8th Int'l Conf. Document Analysis and Recognition (ICDAR'2005)* (pp. 000–000). Seoul, Korea.
- Pikaso Write-On handwriting comparison software.* (n.d.).  
<http://www.pikaso.com/write-on/main.htm>.

- Plamondon, R. (1989). A handwriting model based on differential geometry. In R. Plamondon, C. Y. Suen, & M. L. Simner (Eds.), *Computer Recognition and Human Production of Handwriting* (pp. 179–192). World Scientific.
- Plamondon, R., & Lorette, G. (1989). Automatic signature verification and writer identification — the state of the art. *Pattern Recognition*, 22(2), 107–131.
- Plamondon, R., & Maarse, F. J. (1989, September/October). An evaluation of motor models of handwriting. *IEEE Trans. Syst., Man, Cybern.*, 19(5).
- Plamondon, R., & Privitera, C. M. (1999, January). The segmentation of cursive handwriting: An approach based on off-line recovery of the motor-temporal information. *IEEE Trans. Image Processing*, 8(1), 80–91.
- Plamondon, R., Suen, C. Y., Bourdeau, M., & Barrière, C. (1994). Methodologies for evaluating thinning algorithms for character recognition. In C. Y. Suen & P. S. P. Wang (Eds.), *Thinning methodologies for pattern recognition* (Vol. 8, pp. 283–306). Singapore, New Jersey, London, Hong Kong: World Scientific.
- Plamondon, R., Suen, C. Y., & Simner, M. L. (Eds.). (1989). *Computer recognition and human production of handwriting*. World Scientific.
- Poddar, A. (1997). *Detecting the slant of handwriting using hough transform* (Final Year Project Report). School of Applied Science, Nanyang Technological University.
- Pottmann, H., & Hofer, M. (2002, May). Geometry of the squared distance function to curves and surfaces. In H.-C. Hege & K. Pilthier (Eds.), *Visualization and mathematics* (Vol. III, pp. 223–244). Berlin, Germany: Springer. (Proc. VisMath 2002)
- Qiao, Y., & Yasuhara, M. (2004, October). Recovering dynamic information from static handwritten images. In *Proc. 9th Int'l Workshop on Frontiers in Handwriting Recognition (IWFHR-9)* (pp. 118–123). Tokyo, Japan.
- Rattarangsi, A., & Chin, R. T. (1992, April). Scale-based detection of corners of planar curves. *IEEE Trans. Pattern Anal. Machine Intell.*, 14(4), 430–449.
- Robertson, E. W. (1991). *Fundamentals of document examination*. Nelson-Hall.
- Roli, F., & Giacinto, G. (2002). Design of multiple classifier systems. In H. Bunke & A. Kandel (Eds.), *Hybrid methods in pattern recognition* (Vol. 47, pp. 199–226). World Scientific.
- Sagar, V. K., Chong, S. W., Leedham, C. G., & Solihin, Y. (1996). Slant manipulation and character segmentation for forensic document examination. In *TENCON'96 proceedings. IEEE region ten conference: Digital signal processing applications* (Vol. 2, pp. 933–938). Perth, WA, Australia.
- Said, H. E. S., Peake, G. S., Tan, T. N., & Baker, K. D. (1998). Writer identification from non-uniformly skewed handwriting images. In *Proceedings of the 9th british machine vision conference* (pp. 479–487). University of Southampton, UK.

- Said, H. E. S., Tan, T. N., & Baker, K. D. (2000). Personal identification based on handwriting. *Pattern Recognition*, *33*, 149–160.
- Sauvola, J., & Pietikäinen, M. (2000). Adaptive document image binarization. *Pattern Recognition*, *33*, 225–236.
- Savakis, A. E. (1998, October). Adaptive document image thresholding using foreground and background clustering. In *Proc. Int'l. Conf. Image Processing (ICIP'98)* (Vol. 3, pp. 785–789). Chicago, IL, USA.
- Schaffer, J. D. (1985). Multiple objective optimization with vector evaluated genetic algorithms. In *Proc. 1st Int'l. Conf. Genetic Algorithms* (pp. 93–100). Mahwah, NJ, USA: Lawrence Erlbaum Associates.
- Schomaker, L., Bulacu, M., & Erp, M. van. (1995, August). Sparse-parametric writer identification using heterogeneous feature groups. In *Proc. 3rd Int'l Conf. Document Analysis and Recognition (ICDAR'95)* (pp. 545–548). Montreal, Canada.
- Schomaker, L., Thomassen, A. J. W. M., & Teulings, H. L. (1989). A computational model of cursive handwriting. In R. Plamondon, C. Y. Suen, & M. L. Simner (Eds.), *Computer Recognition and Human Production of Handwriting* (pp. 153–177). World Scientific.
- Schrümann, J. (1996). *Pattern classification: A unified view of statistical and neural approaches*. New York / Chichester / Brisbane / Toronto / Singapore: John Wiley & Sons.
- Setiono, R., & Liu, H. (1997). Neural-network feature selector. *IEEE Trans. Neural Networks*, *8*(3), 654–662.
- Shah-Hosseini, H., & Safabakhsh, R. (2002, October). Automatic multilevel thresholding for image segmentation by the growing time adaptive self-organizing map. *IEEE Trans. Pattern Anal. Machine Intell.*, *24*(10), 1388–1393.
- Sharkley, A. J. C., Sharkley, N. E., Gerecke, U., & Chandroth, G. O. (2000). The “test and select” approach to ensemble combination. In *Proceedings of First International Workshop on Multiple Classifier Systems*. Cagliari, Italy.
- Shi, Z., & Govindaraju, V. (2003, August). Skew detection for complex document images using fuzzy runlength. In *Proc. 7th Int'l Conf. Document Analysis and Recognition (ICDAR'2003)* (pp. 715–719). Edinburgh, UK.
- Simner, M. L., Leedham, C. G., & Thomassen, A. J. W. M. (Eds.). (1996). *Handwriting and drawing research: Basic and applied issues*. Amsterdam, The Netherlands: IOS Press.
- Sita, J., Found, B., & Rogers, D. (2001). Forensic handwriting examiners' expertise for signature comparison. *Journal of Forensic Sciences*, *47*, 1117–1124.
- Solihin, Y. (1997). *A toolset of image processing algorithms for forensic document examination* (Master of Applied Science Thesis). School of Applied Science, Nanyang Technological University.

- Solihin, Y., & Leedham, C. G. (1999). Integral ratio: A new class of global thresholding techniques for handwriting images. *IEEE Trans. Pattern Anal. Machine Intell.*, 21(8), 761–768.
- Somol, P., Pudil, P., Novovičová, J., & Paclík, P. (1999). Adaptive floating search methods in feature selection. *Pattern Recognition Letters*, 20, 1157–1163.
- Srihari, S. N. (1988). Feature extraction for locating address blocks on mail pieces. In *From pixels to features* (pp. 22–27). Elsevier Science Pub. Co.
- Srihari, S. N. (1993). Recognition of handwritten and machine-printed text for postal address interpretation. *Pattern Recognition Letters*, 14, 291–302.
- Srihari, S. N. (2003). Quantifying handwriting individuality. In *Proceedings 55th Annual American Academy of Forensic Sciences*. Chicago, IL, USA.
- Srihari, S. N., & Cha, S.-H. (2001, September). Apriori algorithm for sub-classification analysis of handwriting. In *Proc. 6th Int'l Conf. Document Analysis and Recognition (ICDAR'2001)* (pp. 1022–1025). Seattle, USA.
- Srihari, S. N., Cha, S.-H., Arora, H., & Lee, S. (2001a). *Handwriting identification: Research to study validity of individuality of handwriting and develop computer-assisted procedures for comparing handwriting* (Technical Report No. CEDAR-TR-01-1). NY, USA: SUNY at Buffalo.
- Srihari, S. N., Cha, S.-H., Arora, H., & Lee, S. (2001b, September). Individuality of handwriting: A validation study. In *Proc. 6th Int'l Conf. Document Analysis and Recognition (ICDAR'2001)* (pp. 106–109). Seattle, USA.
- Srihari, S. N., Cha, S.-H., Arora, H., & Lee, S. (2002). Individuality of handwriting. *Journal of Forensic Sciences*, 47(4), 1–17.
- Srihari, S. N., Cha, S.-H., & Lee, S. (2001, September). Establishing handwriting individuality using pattern recognition techniques. In *Proc. 6th Int'l Conf. Document Analysis and Recognition (ICDAR'2001)* (pp. 1195–1204). Seattle, USA.
- Srihari, S. N., Huang, C., & Srinivasan, H. (2005, January). A search engine for handwritten documents. In *Proc. SPIE Conf. Document Recognition and Retrieval XII* (p. 66-75). San Jose, CA, USA.
- Srihari, S. N., Huang, C., Srinivasan, H., & Shah, V. (2005). Biometric and forensic aspects of digital document processing. In B. B. Chaudhuri (Ed.), *Digital document processing*. Springer.
- Srihari, S. N., & Leedham, C. G. (2003, November). A survey of computer methods in forensic document examination. In H. L. Teulings & A. W. A. Van Gemmert (Eds.), *Proc. 11th Conf. Int'l. Graphonomics Society (IGS2003)* (pp. 278–281). Scottsdale, AZ, USA.
- Srihari, S. N., & Shi, Z. (2004, January). Forensic handwritten document retrieval system. In *Proc. Document Image Analysis for Digital Libraries* (pp. 188–194).

- Palo Alto, CA, USA.
- Srihari, S. N., Tomai, C. I., Zhang, B., & Lee, S. (2003, August). Individuality of numerals. In *Proc. 7th Int'l Conf. Document Analysis and Recognition (ICDAR'2003)* (pp. 1096–1100). Edinburgh, UK.
- Srihari, S. N., Zhang, B., Tomai, C., Lee, S.-J., & Shi, Z. (2003). A system for handwriting matching and recognition. In *Proc. Symp. Document Image Understanding Technology (SDIUT 03)*. Greenbelt, MD, USA.
- Srikantan, G., Lam, S. W., & Srihari, S. N. (1996, July). Gradient-based contour encoding for character recognition. *Pattern Recognition*, 29(7), 1147–1160.
- Steinke, K. (1981). Recognition of writers by handwriting images. *Pattern Recognition*, 14(1–6), 357–364.
- Su, Y.-M., & Wang, J.-M. (2003, March). A novel stroke extraction method for Chinese characters using Gabor filters. *Pattern Recognition*, 36(3), 635–647.
- Suen, C. Y., & Wang, P. S. P. (Eds.). (1994). *Thinning methodologies for pattern recognition* (Vol. 8). Singapore, New Jersey, London, Hong Kong: World Scientific.
- Syswerda, G. (1989). Uniform crossover in genetic algorithms. In *Proc. 3rd Int'l Conf. Genetic Algorithms* (pp. 2–9). George Mason University, USA: Morgan Kaufmann.
- Takru, K., & Leedham, C. G. (2002, August). Separation of touching and overlapping words in adjacent lines of handwritten text. In *Proc. 8th Int'l Workshop on Frontiers in Handwriting Recognition (IWFHR-8)* (pp. 496–501). Ontario, Canada.
- Tang, Y. Y., Tao, Y., & Lam, E. C. M. (2002, May). New method for feature extraction based on fractal behavior. *Pattern Recognition*, 35(5), 1071–1081.
- Tang, Y. Y., & You, X. (2003, September). Skeletonization of ribbon-like shapes based on a new wavelet function. *IEEE Trans. Pattern Anal. Machine Intell.*, 25(9), 1118–1133.
- Teulings, H. L., & Van Gemmert, A. W. A. (Eds.). (2003). *Proc. 11th Conf. Int'l. Graphonomics Society (IGS2003)*. Scottsdale, AZ, USA.
- Tomai, C. I., Zhang, B., & Srihari, S. N. (2004, August). Discriminatory power of handwritten words for writer recognition. In *Proc. 17th Int'l Conf. Pattern Recognition* (pp. 638–641). Cambridge, UK.
- United States v. Starzecpyzel*. (1995). 880 F. Supp. 1027, 1046 (S.D.N.Y. 1995).
- Vafaie, H., & De Jong, K. (1992, November). Genetic algorithms as a tool for feature selection in machine learning. In *Proc. 4th Int'l Conf. Tools with Artificial Intelligence (TAI'92)* (pp. 200–203). Arlington, VA: IEEE Computer Society Press.
- Van Erp, M., Vuurpijl, L. G., Franke, K., & Schomaker, L. (2003, November). The

- WANDA measurement tool for forensic document analysis. In H. L. Teulings & A. W. A. Van Gemmert (Eds.), *Proc. 11th Conf. Int'l. Graphonomics Society (IGS2003)* (pp. 282–285). Scottsdale, AZ, USA.
- Van Gemmert, A. W. A., Teulings, H. L., & Stelmach, G. E. (1999). Parkinson's disease and the control of size and speed of handwriting. *Neuropsychologia*, *37*, 685–694.
- Van Roon, D., Steenbergen, B., & Meulenbroek, R. G. (2005, June). Effects of removing visual information of the moving limb on accuracy control in cerebral palsy. In A. Marcelli & C. De Stefano (Eds.), *Proc. 12th Conf. Int'l. Graphonomics Society (IGS2005)* (pp. 271–275). University of Salerno, Salerno, Italy.
- Vinciarelli, A. (2002, July). A survey on off-line cursive word recognition. *Pattern Recognition*, *35*(7), 1433–1446.
- Wang, W., Pottmann, H., & Liu, Y. (2004). *Fitting B-spline curves to point clouds by squared distance minimization* (Technical report No. TR-2004-11). Hong Kong: The University of Hong Kong.
- Weiss, S. M., & Kulikowski, C. A. (1991). *Computer systems that learn: Classification and prediction methods from statistics, neural nets, machine learning, and expert systems*. Morgan Kaufmann.
- Wenjin, L., & Dori, D. (1998, April). Incremental arc segmentation algorithm and its evaluation. *IEEE Trans. Pattern Anal. Machine Intell.*, *20*(4), 424–431.
- Worring, M., & Smeulders, A. W. M. (1993, November). Digital curvature estimation. *CVGIP: Image Understanding*, *58*(3), 366–382.
- Xu, L., Krzyżak, A., & Suen, C. Y. (1992, May). Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Trans. Syst., Man, Cybern.*, *22*(3), 418–435.
- Yang, J., & Honavar, V. (1998a). Feature subset selection using a genetic algorithm. In H. Liu & H. Motoda (Eds.), *Feature extraction, construction and selection* (pp. 117–136). Kluwer Academic Publishers.
- Yang, J., & Honavar, V. (1998b, March/April). Feature subset selection using a genetic algorithm. *IEEE Intell. Syst.*, *13*, 44–49.
- Yang, J., Parekh, R., & Honavar, V. (1997). *Distal: An inter-pattern distance-based constructive learning algorithm* (Technical Report No. ISU-CS-TR 97-06). Department of Computer Science, Iowa State University. (Also appeared in Proc. Int'l. Conf. Neural Networks, IEEE, Piscataway, N.J, 1998)
- Zhang, B., & Srihari, S. N. (2003a, August). Analysis of handwriting individuality using word features. In *Proc. 7th Int'l Conf. Document Analysis and Recognition (ICDAR'2003)* (pp. 1142–1146). Edinburgh, UK.
- Zhang, B., & Srihari, S. N. (2003b). Binary vector dissimilarity measures for handwriting identification. In *Proceedings of Document Recognition and Retrieval X*

- (pp. 28–38). Santa Clara, CA, USA.
- Zhang, B., & Srihari, S. N. (2003c, January). On binary dissimilarity measures. In *Proc. Int'l. Conf. Computer Vision, Pattern Recognition and Image Processing*. Cary, North Carolina, USA.
- Zhang, B., & Srihari, S. N. (2004, April). Fast k-nearest neighbor classification using cluster-based trees. *IEEE Trans. Pattern Anal. Machine Intell.*, *26*(4), 525–528.
- Zhang, B., Srihari, S. N., & Lee, S. (2003, August). Individuality of handwritten characters. In *Proc. 7th Int'l Conf. Document Analysis and Recognition (ICDAR'2003)* (pp. 1086–1090). Edinburgh, UK.
- Zhang, T. Y., & Suen, C. Y. (1984). A fast parallel algorithm for thinning digital patterns. *Comm. ACM*, *27*(3), 236–239.
- Zou, J. J., Chang, H.-H., & Yan, H. (2001, October). Shape skeletonization by identifying discrete local symmetries. *Pattern Recognition*, *34*(10), 1895–1905.

# Appendix A

## Characters and grapheme extracted from CEDAR letter

### A.1 Character “d” (10 occurrences)

*From*

*Jim Elder*

*829 Loop Street, Apt 300*

*Allentown, New York 14707*

*To*

*Dr. Bob Grant*

*602 Queensberry Parkway*

*Omar, West Virginia 25638*

*We were referredd to you by Xena Cohen at the University Medical Center. This is regarding my friendd, Kate Zack.*

*It all started around six months ago while attending the “Rubeq” Jazz Concert. Organizing such an event is no picnic, andd as President of the Alumni Association, a co-sponsor of the event, Kate was overworkedd. But she enjoyed her job, andd did what was required of her with great zeal andd enthusiasm.*

*However, the extra hours affectedd her health; halfway through the show she passed out. We rushed her to the hospital, andd several questions, x-rays and bloodd tests later, were told it was just exhaustion.*

*Kate’s been in very bad health since. Could you kindly take a look at the results andd give your opinion?*

*Thank you!*

*Jim*

## A.2 Character “y” (8 occurrences)

*From*

*Jim Elder*

*829 Loop Street, Apt 300*

*Allentown, New York 14707*

*To*

*Dr. Bob Grant*

*602 Queensberryy Parkwayy*

*Omar, West Virginia 25638*

*We were referred to you by Xena Cohen at the Universityy Medical Center. This is regarding my friend, Kate Zack.*

*It all started around six months ago while attending the “Rubeq” Jazz Concert. Organizing such an event is no picnic, and as President of the Alumni Association, a co-sponsor of the event, Kate was overworked. But she enjoyed her job, and did what was required of her with great zeal and enthusiasm.*

*However, the extra hours affected her health; halfway through the show she passed out. We rushed her to the hospital, and several questions, x-rays and blood tests later, were told it was just exhaustion.*

*Kate’s been in very bad health since. Could you kindly take a look at the results and give your opinion?*

*Thank you!*

*Jim*

### A.3 Character “f” (8 occurrences)

*From*

*Jim Elder*

*829 Loop Street, Apt 300*

*Allentown, New York 14707*

*To*

*Dr. Bob Grant*

*602 Queensberry Parkway*

*Omar, West Virginia 25638*

*We were referred to you by Xena Cohen at the University Medical Center. This is regarding my friend, Kate Zack.*

*It all started around six months ago while attending the “Rubeq” Jazz Concert. Organizing such an event is no picnic, and as President of the Alumni Association, a co-sponsor of the event, Kate was overworked. But she enjoyed her job, and did what was required of her with great zeal and enthusiasm.*

*However, the extra hours affected her health; halfway through the show she passed out. We rushed her to the hospital, and several questions, x-rays and blood tests later, were told it was just exhaustion.*

*Kate’s been in very bad health since. Could you kindly take a look at the results and give your opinion?*

*Thank you!*

*Jim*

## A.4 Grapheme “th” and character “t” (9 occurrences)

*From*

*Jim Elder*

*829 Loop Street, Apt 300*

*Allentown, New York 14707*

*To*

*Dr. Bob Grant*

*602 Queensberry Parkway*

*Omar, West Virginia 25638*

*We were referred to you by Xena Cohen at **the** University Medical Center. This is regarding my friend, Kate Zack.*

*It all started around six months ago while attending **the** “Rubeq” Jazz Concert. Organizing such an event is no picnic, and as President of **the** Alumni Association, a co-sponsor of **the** event, Kate was overworked. But she enjoyed her job, and did what was required of her with great zeal and enthusiasm.*

*However, **the** extra hours affected her health; halfway **through** **the** show she passed out. We rushed her to **the** hospital, and several questions, x-rays and blood tests later, were told it was just exhaustion.*

*Kate’s been in very bad health since. Could you kindly take a look at **the** results and give your opinion?*

*Thank you!*

*Jim*

## Appendix B

### Example of a handwritten CEDAR letter

from

Jim Elder

829 Loop Street, Apt 300

Allentown, New York 14707

To Dr. Bob Grant

602 Queensberry Parkway

Omari, West Virginia 25638

We were referred to you by Xena Cohen at the University Medical Center. This is regarding my friend, Kate Zack.

It all started around six months ago while Alumni Association a co-sponsor of the event, Kate was overworked. And did what was required of her <sup>with</sup> great zeal and enthusiasm.

However, the extra hrs. affected her health; halfway through the show she passed out. She was rushed to the hospital, took xrays were told it was just exhaustion,

Kate's been in very bad health since. Could you kindly take a look at the results + give us your opinion?

Thank you!

Jim

# Appendix C

## List of Publications

1. Philip Jonathan Sutano, Graham Leedham, and Vladimir Pervouchine. “Study of the consistency of some of the discriminatory features used by document examiners in the analysis of handwritten letter ‘a’”. In *Proceedings of the 7th International Conference on Document Analysis and Recognition (ICDAR)*, Edinburgh, UK, Aug. 3–6, 2003. pp. 1091–1095.
2. Graham Leedham, Vladimir Pervouchine, Wei Kei Tan, and Arun Jacob. “Automatic Quantitative Letter-level Extraction of Features Used by Document Examiners.” In *Proceedings of 11th Conference of the International Graphonomics Society (IGS)*, Scottsdale, AZ, USA, Nov. 2–5, 2003. pp. 291–294.
3. Graham Leedham, Vladimir Pervouchine, Wei Kei Tan, and Arun Jacob. “Assessment of the Stability and Usefulness of Some Handwriting Features Used by Document Examiners to Identify Authorship.” In *Proceedings of 11th Conference of the International Graphonomics Society (IGS)*, Scottsdale, AZ, USA, Nov. 2–5, 2003. pp. 316–319.
4. Graham Leedham, Vladimir Pervouchine, and Tan Wei Kei. “Quantitative Letter-level Extraction and Analysis of Features Used by Document Examiners.” *Journal of Forensic Document Examination*, vol. 16, 2005. In press.
5. Vladimir Pervouchine and Graham Leedham. “The Validity and Usefulness of Forensic Document Examiner Features for Automatic Writer Identification.” Submitted to *Pattern Recognition*. Under review.
6. Vladimir Pervouchine, Graham Leedham, and Konstantin Melikhov. “Handwritten Character Skeletonisation for Forensic Document Analysis.” In *Proceedings of the 20th ACM Symposium on Applied Computing (ACM SAC)*, vol. 1, pp. 754–758, Santa Fe, NM, USA, Mar. 13–17, 2005.
7. Graham Leedham and Vladimir Pervouchine. “Validating the Use of Handwriting as a Biometric and its Forensic Analysis.” Invited lecture at the *International Workshop on Document Analysis (IWDA-2005)*, Indian Statistical Institute, Calcutta, 8–11 Mar. 2005. Published in “Document Analysis”, Eds. Umapada Pal, Swapan K. Parui and B. B. Chaudhuri, Allied Publishing Ltd,

- Chennai, pp. 175–192, ISBN 81-7784-9, 2005.
8. Graham Leedham, Vladimir Pervouchine, and Konstantin Melikhov. “Three-stage Handwriting Stroke Extraction Method with Hidden Loop Recovery.” In *Proceedings of 8th International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1, Seoul, Korea, Aug.-Sep. 2005. pp. 307–311.
  9. Vladimir Pervouchine and Graham Leedham. “Document Examiner Feature Extraction: Thinned vs. Skeletonised Handwriting Images.” In *Proc. IEEE TENCON’05*, Melbourne, Australia, 21–24 Nov. 2005.
  10. Vladimir Pervouchine and Graham Leedham. “Extraction and Analysis of Document Examiner Features from Vector Skeletons of Grapheme ‘th’.” In *Horst Bunke, A. Lawrence Spitz (Eds.): Document Analysis Systems VII, 7th International Workshop (DAS 2006)*, Nelson, New Zealand, February 13-15, 2006, pp. 196–207. Published in *Lecture Notes in Computer Science 3872* Springer ISBN 3-540-32140-3 2006.