HANDWRITTEN CHINESE CHARACTER RECOGNITION USING *X-Y* GRAPHS DECOMPOSITION AND TWO-DIMENSIONAL FUNCTIONAL RELATIONSHIP MODEL

By

LEE JIA CHII

A thesis submitted to the Department of Mathematical Sciences,
Faculty of Engineering and Science,
Universiti Tunku Abdul Rahman,
in partial fulfillment of the requirements for the degree of
Master of Mathematical Sciences
May 2011

ABSTRACT

HANDWRITTEN CHINESE CHARACTER RECOGNITION USING *X*-*Y* GRAPHS DECOMPOSITION AND TWO-DIMENSIONAL FUNCTIONAL RELATIONSHIP MODEL

Lee Jia Chii

This thesis is about the recognition of handwritten Chinese character based on online approach. Wide-ranging applications of handwritten Chinese character recognition (HCCR) from the advancement of automation process and telecommunication to educational purpose have made this topic become popular among research area. In practice, feature extraction and classification are two main phases in a recognition system, such that the overall performance of the recognition system depends greatly on them.

In this research, a new approach of feature extraction method for HCCR called *X*-*Y* graphs decomposition is presented. Central to the proposed method is the idea of capturing the geometrical and topological information from the trajectory of handwritten character using two unique decomposed graphs: *X*-graph and *Y*-graph. For feature size reduction, Haar wavelet is applied on the graphs. This is a new attempt of wavelet transform. Features extracted using *X*-*Y* graphs decomposition with Haar wavelet not only cover both the global and local features of the characters, but also are invariant of different writing styles. As a result, the discrimination power of the recognition system can be strengthened, especially for recognizing similar characters, deformed characters and characters with connected strokes.

For classification, a similarity measure is established via statistical technique which calculates the coefficient of determination $\left(R_p^2\right)$ for 2-dimensional unreplicated linear functional relationship (2D-ULFR) model between the trajectory pattern of input character and character in database, according to which the recognition result is determined. The principle of the proposed method makes R_p^2 very robust against size and position variation as well as character shape deformation, even without normalization. Furthermore, R_p^2 also enhances the stability and reliability of the recognition system.

Experimental results based on database of 3000 frequently used Chinese character have proved the efficiency of the new designed recognition system which is embedded with new proposed feature extraction method: *X*-*Y* graphs decomposition and new classifier: R_p^2. The most attractive advantage of this recognition system is that it still remains a promising recognition rate even without undergoing normalization: a high recognition rate of 98.2% despite of small feature size such that the dimensionality is between 64 (inclusive) and 128 (exclusive), and reduced processing time up to 75.31%, 73.05%, 58.27% and 40.69% if compared to CBDD, MD, CMF and MQDF classifiers respectively. Therefore, it is more practical and preferable for applications nowadays.

ACKNOWLEDGEMENTS

It is my pleasure to take this opportunity to thank my supervisor, Mr. Chang Yun Fah and my co-supervisor, Dr Chen Huey Voon. They have given me invaluable encouragement and useful advices throughout the duration of this research. Their guidance helps me a lot in completing my thesis successfully.

Besides, I would like to acknowledge the funding support from E-Science Grant (No: 01-02-11-SF0053) of Ministry of Science, Technology and Innovative, Malaysia. Without their support, my aim to do this research will hardly be realized.

Moreover, I am very grateful to Professor Dr. Lianwen Jin. He provides me HCH-GB1 dataset which is very helpful in verifying my experimental results obtained in this research.

Thank also my group members who are willing to share their information. Their assistance is greatly appreciated.

Last but not least, a sincere gratitude and keen appreciation to my family member for their unfailing love and support.

APPROVAL SHEET

This thesis entitled "**HANDWRITTEN CHINESE CHARACTER RECOGNITION USING *X*-*Y* GRAPHS DECOMPOSITION AND TWO-DIMENSIONAL FUNCTIONAL RELATIONSHIP MODEL**" was prepared by LEE JIA CHII and submitted as partial fulfillment of the requirements for the degree of Master of Mathematical Sciences at Universiti Tunku Abdul Rahman.

Approved by:

____

(Mr. CHANG YUN FAH)
Date: 25th May, 2011
Supervisor
Department of Mathematical Sciences
Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

____

(Dr. CHEN HUEY VOON)
Date: 25th May, 2011
Co-supervisor
Department of Mathematical Sciences
Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

FACULTY OF ENGINEERING AND SCIENCE
UNIVERSITI TUNKU ABDUL RAHMAN

Date: 25th May, 2011

PERMISSION SHEET

It is hereby certified that **LEE JIA CHII** (ID No: **09UEM02339**) has completed this thesis/dissertation entitled "HANDWRITTEN CHINESE CHARACTER RECOGNITION USING *X*-*Y* GRAPHS DECOMPOSITION AND TWO-DIMENSIONAL FUNCTIONAL RELATIONSHIP MODEL" under the supervision of Mr. Chang Yun Fah (Supervisor) from the Department of Mathematical Sciences, Faculty of Engineering and Science, and Dr. Chen Huey Voon (Co-Supervisor) from the Department of Mathematical Sciences, Faculty of Engineering and Science.

I hereby give permission to the University to upload softcopy of my thesis in pdf format into UTAR Institutional Repository, which will be made accessible to UTAR community and public.

Yours truly,

(LEE JIA CHII)

DECLARATION

I hereby declare that the dissertation is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTAR or other institutions.

Name: Lee Jia Chii

Date: 25th May, 2011

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure **Page**

LIST OF ABBREVIATIONS

AMD	Asymmetric Mahalanobis distance
ARG	Attributed relational graph
BMN	Bi-moment normalization
CBDD	City block distance with deviation
COD	Coefficient of determination
CMF	Compound Mahalanobis function
CWT	Continuous wavelet transform
DSM	Decision surface mapping
DS-CMF	Difference subspace based compound Mahalanobis function
DEF	Directional element feature
DFD	Directional feature densities
DWT	Discrete wavelet transform
DLQDF	Discriminative learning quadratic discriminant function
DP	Dynamic Programming
FT	Fourier transform
FARG	Fuzzy attributed relational graph
HCCR	Handwritten Chinese character recognition
HMM	Hidden Markov model
LVQ	Learning vector quantization
LRHMM	Left-right hidden Markov model
LBG	Linde-Buzo-Gray
LDA	Linear discriminant analysis
LN	Linear normalization
MD	Mahalanobis Distance
ML	Maximum likelihood
MLE	Maximum likelihood estimation
MCE	Minimum classification error

MD	Minimum distance
MCBA	Modified centroid-boundary alignment
MQDF	Modified quadratic discriminant function
MN	Moment normalization
MULFR	Multidimensional unreplicated linear functional relationship
MRA	Multiresolution analysis
NN	Neural Network
NLN	Nonlinear normalization
NSN	Nonlinear shape normalized
OLCCR	Online Chinese character recognition
PCHMM	Path controlled hidden Markov model
PSNR	Peak signal to noise ratio
PDA	Personal digital assistants
PNN	Probabilistic neural networks
PDF	Probability density functions
pseudo 2D	Pseudo-two-dimensional
QDF	Quadratic discriminant function
RBF	Radical basis function
RDA	Regularized discriminant analysis
SPDNN	Self-growing probabilistic decision-based neural network
STFT	Short-time Fourier transform
SVM	Support vector machine
T*XY*GD	Trajectory-based *X*-*Y* graphs decomposition
TPID	Transformation based on partial inclination detection
2D-ULFR	Two-dimensional unreplicated linear functional relationship
VQ	Vector quantization
WT	Wavelet transform

CHAPTER 1

INTRODUCTION

Character recognition is one of the main branches of pattern recognition. Research on printed Chinese character recognition has become a significant area since Casey and Nagy (1966) opened up the field. After a few years, this has extended to the problem of handwritten Chinese character recognition (HCCR) which is of vital importance in applications where handwriting is the desirable input channel, such as form filling, personal digital assistants (PDA) and computer-aided education. As mentioned in Jain and Lazzerini (1999) and Kherallah *et al*. (2008), the area of handwriting recognition can be divided into offline approach and online approach. Offline approach recognizes the handwriting in the form of an image through scanners or cameras (Srihari *et al*., 2007). In this case, only the completed character or word is available. Whereas, the online approach deals with the recognition of handwriting captured by a tablet or similar touch-sensitive device, and the digitized trace of the pen is used to recognize the character (Deepu *et al*., 2004). In this instance, the recognizer will have access to x and y coordinates as a function of time which has temporal information about how the character was formed. The research on online handwritten character recognition has been receiving a great interest since 1980s (Liu *et al*., 2004). Even in recent years, online handwriting recognition still remains as an active topic among the research community.

When the online handwritten character recognition just started, it was only limited to the recognition of English alphanumerics. A comprehensive survey of Plamondon and Srihari (2000) reviewed the handwriting recognition which mainly focused on western handwriting. At the end of the 1960s, the online handwritten character recognition in Japanese language (which includes Hiragana, Katakana, Kanji, English alphanumerics and symbols) has gradually emerged. Early works have been done in the online Japanese character recognition by Nakagawa (1990) and Wakahara *et al.* (1992). Since 1980s, the problem of online handwritten Chinese character recognition (HCCR) has received considerable attention from the research area due to its wide-ranging applications and technical challenges. Various approaches of HCCR that have been proposed over the years can be reviewed in Cheng (1998), Hao *et al.* (1997), Kato *et al.* (1996), and Wakabayashi *et al.* (1996).

In this chapter, Section 1.1 gives a fundamental idea about the overall process of the commonly used HCCR system. The feature extraction and classification process are introduced in Section 1.1.1 and Section 1.1.2 respectively. Section 1.2 explains the problem statement and motivation of this research. It is followed by the research objectives and scope of study in Section 1.3 and Section 1.4 correspondingly. This chapter concludes with the organization of the thesis.

1.1 Overview of HCCR Process

In general, the HCCR system involves three main stages: preprocessing, feature extraction and classification. Firstly, the image (for offline approach) or trajectory (for online approach) of an input character is captured by the recognition system. Individual writing variations, distortions due to erratic hand motions, and inaccuracy of digitization are particularly disruptive to character recognition. Hence, preprocessing on the original input character is needed in order to alleviate these problems and obtain a more proper form for later stages. In the case of offline approach, preprocessing may focus on the image of the input character, which includes binarization or converting the image to black and white colors, smoothing, interfering-lines removal, stroke slant or width adjustment, image enhancement and others. Whereas for online approach, preprocessing on the input character trajectory involves normalization, which standardizes the size and position of characters, and overcome the shape deformation problem. After undergoing preprocessing, it will proceed to the stage of feature extraction, which is also referred to as pattern description. In this stage, the input character is represented in a special form for classification. Lastly, the output character from database that matches with the input character is determined using classifiers. A practical HCCR system is depicted diagrammatically in Figure 1.1.

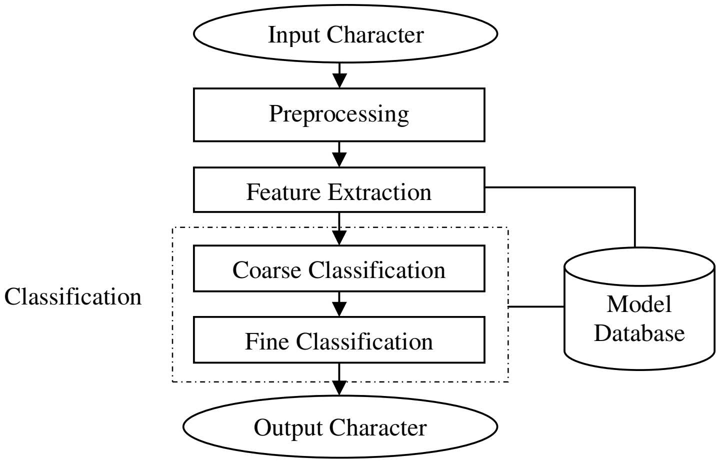


Figure 1.1: Diagram of a practical HCCR system.

1.1.1 Feature Extraction

According to Devijver and Kittler (1982), feature extraction is a problem of extracting from the raw data the information which is most relevant for classification purposes, in the sense of minimizing the within-class pattern variability while enhancing the between-class pattern variability. From the theoretical aspect, feature extraction is a method which determines an appropriate subspace of dimensionality *m*, either in a linear or a nonlinear way, in the original feature space of dimensionality *d* where $m \leq d$ (Jain *et al.*, 2000). The feature extraction methods for input character and database are of particular important since the classification techniques depend significantly on them. Some important properties of feature extraction are illustrated below:

i. System dependency - Different feature extraction methods fulfill the requirements to a varying degree, depending on the specific recognition problem and available data. A feature extraction method that proves to be successful in one character recognition system may turn out not to be very efficient in another recognition system (Devijver and Kittler, 1982).

ii. Handwriting invariant - The extracted features must be invariant to the expected distortions and variations that the characters may have in a specific application. In other words, the extracted features must remain a promising recognition rate in the recognition system, without affecting by the variation of size, position, orientation or skewed, and writing styles of the characters.

iii. Curse of dimensionality - With a limited training set, the size of features must be kept reasonably small if a statistical classifier is to be used, so that the trained recognition system is sufficiently reliable. It is recommended to use 5 to 10 times as many training patterns of each class as the dimensionality of the feature vector (Jain and Chandrasekaran, 1982).

iv. Classifier stability - The type or format of the extracted features must match the requirements of the chosen classifier. For instances, graph descriptions of the characters are well suited for structural classifiers; discrete features that may assume to be two or three distinct values only are ideal for decision trees; real-valued feature vectors are ideal for statistical classifiers. However, multiple classifiers may be used, either as a multi-stage classification scheme or as parallel classifiers, where a combination of the individual classification results decides the final classification. In this case, features of more than one type or format may be extracted from input characters.

1.1.2 Classification

Basically, classification is the final stage in most of the HCCR process. Once the proper representations for the characters have been found, classifiers can be designed using different approaches. In practice, the choice of the appropriate and efficient classifiers is a difficult problem. Similar to feature extraction, the classifier is expected to have some desired invariant properties so that a high accuracy rate can be achieved. On the other hand, to accelerate the recognition of large category set, the classification is often decomposed into coarse classification and fine classification. A coarse classification is commonly applied to first select from database a small subset of candidate classes which possibly matches the input character. After that, the input character is classified into one of these candidate classes in the fine classification stage. This two-stage classification technique has been widely adopted nowadays, instead of tree classification (Guo and Gelfand, 1992) and multistage classification (Cao *et al.*, 1995). Furthermore, for the sake of memory limited devices, the parameter complexity of the classifiers is also a significant problem that has to be taken into consideration. It must be reduced as greatly as possible, so that it can be applied to the devices.

1.2 Problem Statement and Motivation

Although many researches on HCCR have been done, the overall performance of the current HCCR system can still be improved in a large extent, especially for unconstrained handwriting, i.e. characters written in freestyle without any constraints on character size, position, stroke number and stroke order. The recognition of handwritten Chinese characters is very different from recognition of handwritten characters of other languages such as English and Arabic. There are many difficulties in recognizing handwritten Chinese characters, which are as follows.

(1) The Chinese character set is of the greatest number if compared to characters of other languages. According to Srihari *et al.* (2007), the Chinese script consists of approximately 50,000 characters, of which

only a few thousand are commonly used. Besides, due to the complicated structure of Chinese character, the complexity of features for characters has increased, i.e. each Chinese character can be transformed to a vector of nearly or more than 100 dimensionalities. Besides, a single character may have various handwritten versions. Hence, the HCCR can be referred to as a problem with a feature space of high dimensionality and data set with large amount of samples that belong to many different classes. As a result, the performance of the recognition system will be degraded in term of the system storage and speed.

(2) There exist many similarly shaped Chinese characters. Some examples of this are demonstrated in Figure 1.2. These similar characters are easy to be confused with each other by either computer or human since they often share the common radicals and have very slight shape difference only in local details. Therefore, this will increase the misclassification error rate of the recognition system.

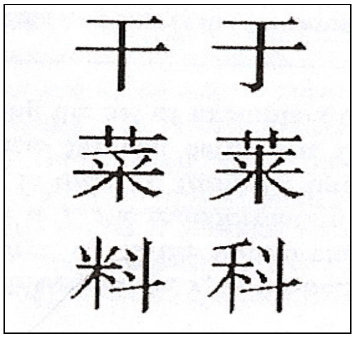
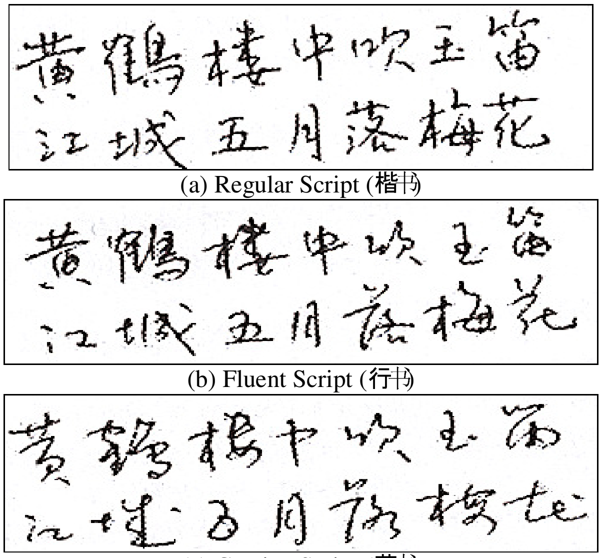


Figure 1.2: Example of six visually similar characters: "干", "于", "菜", "莱", "料" and "科".

(3) The variability of writing styles has further blurred the shape difference between similar characters. Formally, the Chinese handwritten scripts are categorized into three typical styles: regular script, fluent script and cursive script. Some instances of these three typical styles are shown in Figure 1.3 (Liu *et al*., 2004). Among them, regular script is commonly used by most of the writers and it can be recognized easily. However, for fluent and cursive script (irregular script), the handwriting is deformed seriously and some character shapes even totally differ from the standard shape. Thus, it is difficult to recognize them correctly, even by humans. Furthermore, another common problem occurs in HCCR is the variability of size and position of handwritten characters, i.e. different writers tend to write the characters in different size and position.



(a) Regular Script (楷书)

(b) Fluent Script (行书)

(c) Cursive Script (草书)

Figure 1.3: “黄鹤楼中吹玉笛 江城五月落梅花” is one of the stanzas in the famous Chinese poem “*Kanshiro in Listening and Spot on Chin Piper*” written by Li Bai. Three common types of Chinese character writing styles for this stanza are shown: (a) regular script, (b) fluent script and (c) cursive script.

(4) Stroke number and stroke order of Chinese characters are significant information for online recognition where trajectory (pen-tip movement) of the characters is captured as the raw feature. If the input character is written in a wrong stroke number or order, then this can be a bottleneck for recognition where the system cannot recognize it totally. However, some writers have no knowledge about the correct stroke number and order of Chinese characters, especially those who do not undergo a proper Chinese education.

Although many feature extraction and classification schemes have emerged over the years to cope with the above problems faced in HCCR, there are still some deficiencies in those existing schemes. The shortages of the schemes that proposed previously are revealed below.

I. Large dimensionality and parameter complexity

In feature extraction, the features extracted for each Chinese character are often in large dimensionality, especially for the feature extraction methods based on structural approach (refer to Section 2.2.1) and statistical approach (refer to Section 2.2.3). For examples, the features formed by using structural-based Delaunay triangulation (Zeng *et al*., 2006) and statistical-based 8-directional techniques (Bai and Huo, 2005) are of more than 500 dimensions. Despite of many dimensionality reduction strategies such as using Gaussian filter (Bai and Huo, 2005), the size of the features after reduction is still not small enough. Moreover, some dimensionality reduction methods will affect the recognition rate tremendously due to the loss of significant information.

Whereas, for classification, the problem of parameter complexity often occurs in the existing classifiers, including the most widely used quadratic discriminant function (refer to Section 2.4.1). For example, the modified quadratic discriminant function (MQDF) (Kimura *et al.*, 1987) requires storing the parameters such as the mean vectors, the dominant eigenvalues and eigenvectors of the covariance matrix for each class. The eigenvectors are the main factor that leads to the high parameter complexity problem as the size of the eigenvectors increase proportionally with the size of the classes. As a result, it is not practical to embed them directly into the hand-held devices. Instead, it has to undergo a compression process, but this will lengthen the processing time of the recognition system.

II. Algorithm complexity

The algorithms used are the core of the whole recognition system. It determines the overall performance of the recognition system. Since last decades, many impressive algorithms for both feature extraction and classification were presented and had been proved to be successful in achieving high accuracy rate. However, most of these previously proposed algorithms are so complicated that it will degrade the quality of the recognition system in term of speed and robustness. For examples, forming the fuzzy attributed relational graph (FARG) (Chan and Cheung, 1992) that represent each character in feature extraction and calculating the Compound Mahalanobis Function (CMF) (Suzuki *et al.*, 1997) in classification involve heavy computations. Besides, due to high complexity of the algorithm, it can only restrict to some particular systems with special functionalities since the normal systems can hardly support it.

III. Learning process

Learning, or alternatively called training is a process of estimating the appropriate parameters for feature extraction schemes and/or classifiers by using a set of character samples (called training set). The larger the training set used in the learning process, the more reliable the recognition system is. However, the learning process for huge training set can be very time-consuming and requires large memory space. Besides, it is not easy to find a stable and reliable learning algorithm that can guarantee a promising recognition rate. Some examples of feature extraction methods that need learning process are fuzzy attributed relational graph (FARG) (Chan and Cheung, 1992) and hidden Markov model (HMM) (Tokuno *et al.*, 2002; Nakai *et al.*, 2002; Hasegawa *et al.*, 2000); while the examples for classifiers are neural network (NN) classifiers (Fu and Xu, 1998; Saruta *et al.*, 1996) and support vector machine (SVM) (Burges, 1998; Dong *et al.*, 2005; Kilic *et al.*, 2008).

IV. Normalization process

To tackle the problem of size and position variation, as well as shape distortion for the input characters, normalization process is necessary as to improve the recognition rate. Linear normalization (LN) method is easy to implement and usually used in the preprocessing stage of the recognition system, but such method can solely standardize the size and position of the input character. Hence, nonlinear normalization (NLN) methods such as moment normalization (MN) (Casey, 1970), bi-moment normalization (BMN) (Liu *et al.*, 2003), modified centroid-boundary alignment (MCBA) (Liu and Marukawa, 2004) and pseudo-two-dimensional (pseudo 2D) normalization method (Horiuchi *et al.*, 1997; Liu and Marukawa, 2005) are needed, in order to handle the character shape distortion problem. Unfortunately, these NLN methods are not as simple as LN methods and they may burden the operation of the recognition system. Consequently, the speed of the recognition system will be decelerated to a great extent.

V. Storage of excessive samples

The problem of stroke number and stroke order variation has become the most concerned issues in HCCR nowadays. In order to cope with this problem, the previous recognition systems used to store many samples of different writing styles, different stroke numbers and stroke orders for each character. Obviously, this technique seems to be not very efficient since it occupied extremely large storage space. An investigation of the improved methods in solving this problem still remains as a challenge in the research area of HCCR.

The obstacles occur in HCCR as stated above has initiated the motivation of doing a research on this topic, as to improve the existing recognition system.

1.3 Objectives

The main objective of this research is to develop a new online HCCR system such that this can be divided into four subtasks as stated below.

I. To setup database

Instead of using the existing databases, a new database will be created because the existing databases are not suitable and compatible with the recognition system developed in this research. The database of this research is based on Ju Dan's modern Chinese character frequency list (Dan, 2004), which is generated from a large corpus of Chinese texts collected from online sources. The first 3000 most frequently used simplified Chinese characters in Ju Dan's modern Chinese character list are chosen as the characters in the proposed database. The Chinese characters in database are in the font style of *songti*, due to its widely used and similarity to handwritten Chinese characters.

II. To propose a new feature extraction method

The feature used to represent each character must have strong discriminative capability, so that the recognition system will be precise in recognizing every different character, especially in differentiating the similar characters. Besides, it must also be able to tolerate well with the size and position variation of the handwritten characters. At the same time, the dimension of the features has to be as small as possible for the sake of small storage space. According to all these requirements, a new feature extraction method which is efficient in term of memory space and accuracy rate will be proposed.

III. To design a new classifier

New designed classifier with improved generalization performance compared to other existing classifiers will be presented in this research. In order to make the classifier more practical to be embedded in memory limited devices, the simplicity of the algorithm for classification plays an important role, i.e. the algorithm must be easy to implement. Moreover, the parameter complexity of the classifier must also be reduced as greatly as possible and simultaneously the error rate still maintains in a promising level.

IV. To evaluate the performance of the proposed methods

To determine the efficiency of the proposed methods, performance evaluation will be implemented through experiment using the samples that have been collected. This involves comparing the recognition rate, processing time and storage space with some existing feature extraction methods and previously proposed classifiers.

1.4 Scope of Study

Over the years, the research on online handwriting recognition has evolved from being academic exercises to developing technology-driven applications. In online handwriting recognition, due to the availability of both temporal stroke information and spatial shape information, it is able to yield higher accuracy than offline recognition. Besides, it also provides good interaction and adaptation capability since the writer can correct the error or change the writing style based on the recognition result. In recent years, new modified pen input interfaces and devices have been developed for the improvement of precision of the trajectory capturing and the convenience of writing. This advancement stimulates a renewed interest in the research on online handwriting recognition. Furthermore, many applications of handwriting recognition have emerged, including text entry for form filling, message composition in mobile (Ma and Leedham, 2007), computer-aided education (Nakagawa *et al.*, 1999), personal digital assistants (PDA), handwritten document retrieval (Russel *et al.*, 2002) and so on. Hence, instead of offline handwriting recognition, online approach will be taken into account in this research.

Chinese characters are used in daily communication by over one quarter of world's population, mainly in Asia. As stated in Liu *et al.* (2004), a Chinese character is an ideography and is composed of mostly straight lines or "poly-line" strokes, i.e. strokes formed by two or more than two straight lines. Many characters contain relatively independent substructures, called radicals, and some common radicals are shared by different characters. Compared to recognizing handwritten English alphanumerics, the recognition of Chinese handwriting, which is the main scope of this research, poses special challenges due to large category set, existence of similar characters and wide variability of writing styles. In general, there are two types of Chinese character sets: (i) traditional Chinese characters and (ii) simplified Chinese characters. The examples of these two Chinese character sets are shown in Figure 1.4. If compared to simplified Chinese characters, traditional Chinese characters are less often used in handwriting nowadays due to its stroke complexity, such that some characters are composed of more than 30 strokes. Therefore, this research will only emphasize on the simplified Chinese characters.

書 說 紅 蘇 鐘 聲 儀 馬

(a)

书 说 红 苏 钟 声 仪 马

(b)

Figure 1.4: Examples of the eight Chinese characters in the form of (a) traditional Chinese and (b) Simplified Chinese.

For feature extraction, there exist two cases based on two different types of input to the system: (i) a sequence of handwritten characters such as text entry for form filling and handwritten document retrieval and (ii) an isolated character such as message composition in mobile phone and personal digital assistant (PDA). Thus, the features may be extracted from a sequence of characters or a single character. For the former case, segmentation is needed in order to segment the sequence handwritten characters into isolated character according to the temporal and shape information. Some of the well-known segmentation strategies are genetic algorithm (Wei *et al.*, 2005), metasynthetic (Liang and Shi, 2005) and heuristic merging with Dynamic Programming (DP) (Tseng and Chen, 1998). In this research, only isolated characters will be considered. Hence, segmentation is unnecessary.

Classification is one of the considerable stages in HCCR process since the performance of the recognition system depend significantly on the designed classifier. The classification approaches are wide-ranging, but basically from the structural aspect, they can be partitioned into three different groups which are radical-based, stroke-based and holistic approaches. The holistic approaches fall into pattern matching category while the radical-based and stroke-based approaches are categorized into structural analyzing category. The whole categorization of the classification techniques is shown in Figure 1.5. In recent year, pattern matching approach is preferably adopted instead of structural analyzing approach due to two main reasons. The first is that it is much easier to recognize the whole character than to recognize its parts or primitives when the character is greatly deformed. The second is that it is more reliable to provide a prototype or prototypes for each Chinese character than to create special primitive strokes or radicals when flexible adjustment to the number of characters in the database is needed. In this research, only holistic approaches will be discussed.

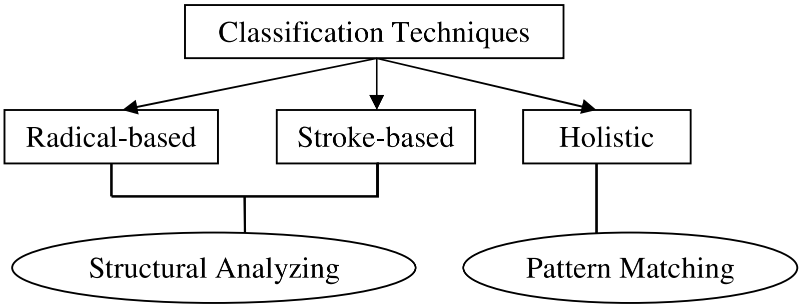


Figure 1.5: Categorization of the classification techniques.

1.5 Organization of Thesis

This thesis consists of six chapters and they are organized as follows. Chapter 2 discusses about the literature review. Before introducing the new proposed feature extraction and classification techniques, it is fundamental to have a brief idea about how the research area of Chinese character recognition grows over the decades. This can assist the readers which are not familiar with this field. By learning the previously proposed methods, the readers will also be able to make comparison between the new designed methods proposed in this research and those classical methods. Thus, they can understand easily about the improvement and advancement of these new proposed schemes. Next, the detail of the new feature extraction method and new classifier will be illustrated in Chapter 3 and 4 respectively. From these two chapters, the structure and the whole process of the recognition system will also be presented. Chapter 5 describes the setup of the experiment and reports the experimental results on recognition of handwritten Chinese characters. The efficiency of the new developed HCCR system can be validated in this chapter. Finally, conclusion will be drawn in Chapter 6.

CHAPTER 2

LITERATURE REVIEW

The historical background of HCCR will be reviewed in order to understand the basic knowledge of its evolution chronologically. Studying the classical as well as modern feature extraction and classification methods is very crucial as to obtain the key idea of the strategies used in solving the problems occurred in HCCR. This chapter is organized as follows. Section 2.1 introduces the historical background of feature extraction methods for HCCR and the detail of them will be illustrated in Section 2.2. On the other hand, Section 2.3 describes the historical background of classifiers for HCCR. The main concept and procedure of those previously proposed classifiers will be presented in Section 2.4. Lastly, the whole chapter 2 is summarized in Section 2.5.

2.1 Historical Background of Feature Extraction Methods

Feature extraction methods, also known as pattern representation schemes, of input pattern and database are of particular importance since the classification method depends largely on them. As stated in Zeng *et al.* (2006), feature extraction methods for online handwriting can be divided into 2 categories: image-based extraction and shaped-based extraction. Image-based extraction (Teredesai *et al.*, 2002) transforms online handwriting to its image form and computes features such as stroke direction (Kawamura *et al.*, 1992; Sun *et al.*, 1991) and Gabor filter-based histogram feature (Wang *et al.*, 2005) by image processing methods. Whereas, shape-based extraction works directly on the level of temporal and discrete signal set. The extracted features usually include local features (such as horizontal change, vertical change, angle change and so on, with respect to the previous and successive neighboring points) and global features (such approximate curvature, aspect ratio, linearity and so on). On the other hand, Govindan and Shivaprasad (1990) classified the feature extraction methods for Chinese character recognition into statistical and structural approach.

Chinese characters are 2-D pictographic characters and intuitively, humans recognize them by making use of their structural information. In the early of 1970's, researchers believe that structural approaches which capture the 2-D structural information of Chinese characters will result in better performance (Tappert *et al.*, 1990). The attributed relational graph (ARG) which is a powerful tool for the representation of the relational structure of a pattern, is developed in the late of 1970's (Tsai and Fu, 1979) and it has been utilized for Chinese character recognition since 1990 (Chen and Lieh, 1990; Lu *et al.*, 1991). It allows variations in stroke number and stroke order. However, the application of ARG to Chinese character recognition faces heavy computational problem due to the large sets of Chinese characters and complexity of the graph-matching algorithms. In order to save the computational time, Lu *et al.* (1991), Chen and Lieh (1990) proposed two-layer graphs to represent Chinese characters. In the first layer, nodes denote the components of a Chinese character and arcs denote the relations among these components; while in the second layer, each component of the first layer is represented by a graph such that the nodes and arcs denote the stroke and the relations among these strokes respectively. This gives several smaller graphs for each Chinese character and thus, reduces the matching time. However, another problem about how to correctly classify the strokes of a Chinese character arises. The wide variation of handwriting styles makes it very difficult to extract Chinese characters successfully and precisely. Hence, the concept of ARG has been extended to fuzzy ARG (FARG) by Chan and Cheung (1992) to handle the fuzzy attributes, but the high complexity of the algorithm and long computational time become the obstacles of this approach. In 2006, Delaunay triangulation (Zeng *et al.*, 2006) was introduced as a feature extraction scheme. It has stronger discrimination ability since it captures both the geometrical information and topological structure of the characters.

In the late 1970s, feature extraction for character recognition based on statistical approach was started (Yasuda and Fujisawa, 1979) and has attracted high attention in 1980s (Kimura *et al.*, 1987; Hamanaka *et al.*, 1993). The use of the feature vector has achieved a great success and is well commercialized in recent years due to its computational efficiency. Among the statistical based feature extraction methods, the most famous one is the direction feature (Yasuda and Fujisawa, 1979). It is widely used in offline character recognition (Kimura *et al.*, 1987; Kimura *et al.*, 1997) and is now being used in online recognition (Kawamura *et al.*, 1992; Hamanaka *et al.*, 1993; Nakagawa *et al.*, 1996). In Jaeger *et al.* (2003), it is named “direction histogram feature”, which is motivated by the fact that it describes the number of occurrences for each stroke direction. The most fundamental direction feature developed in the early years is 4-dimensional features, where 4 directions are defined naturally as vertical (|), horizontal (−), diagonal (\) and anti-diagonal (/). Besides, different ways of extracting directional features were used in previous works. For example, in Kawamura *et al.* (1992) and Nakagawa *et al.* (1996), 4-directional features were extracted directly from nonlinear shape normalized (NSN) online trajectory; while in Hamanaka *et al.* (1993) and Nakai *et al.* (2002), the features were extracted from a bitmap using an "offline approach". Since 1990s, the direction features were extended to 8-direction (Liu, 2006; Bai and Huo, 2005), 12-direction (Liu and Ding, 2005) and even 16-direction (Kimura *et al.*, 1997).

Hidden Markov Model (HMM) approaches include both the ideas of feature extraction and classification. Some researchers might view HMM as a classifier, but in our literature review, we consider HMM (for examples, in the paper of (Takahashi *et al.*, 1997), (Nakai *et al.*, 2001) and (Zheng *et al.*, 1999)) as a feature extraction method. HMM is claimed to be the most efficient way for temporal modeling. It combines both the statistical and structural techniques. Besides, it has many advantages like segmentation free and easy to train (Rabiner, 1989). In fact, HMM is a directed graph with nodes and between-node transitions measured probabilistically. It has been used in speech recognition since 1970s (Rabiner, 1989) and has been applied to online western character recognition since 1980s (Kundu and Bahl, 1988; Nag *et al.*, 1986). Only in recent years, it has been applied to Chinese character recognition. Generally, left-right HMMs (LRHMM) are used to model the sequence of points or line segments for substrokes (Shimodaira *et al.*, 2003;

Tokuno *et al.*, 2002; Nakai *et al.*, 2001), stroke, radicals (Kim *et al.*, 1997), or whole characters (Takahashi *et al.*, 1997; Hasegawa *et al.*, 2000). Such HMMs are stroke-order dependent. For character-based HMM, multiple models are often generated to handle this problem. Whereas, for substroke-based HMM, the character models can be constructed hierarchically and the stroke-order variation can be represented in a variation of network (Nakai *et al.*, 2003). To overcome the stroke-order variation problem more efficiently, a constrained ergodic HMM, namely path controlled HMM (PCHMM) was proposed by Zheng *et al.* (1999) and this approach has been proved to be very successful.

2.2 Feature Extraction Methods

In this thesis, the feature extraction methods are categorized into three approaches: structural, statistical and hybrid statistical-structural (Liu *et al.*, 2004). This categorization is illustrated in Figure 2.1. The structural representation scheme has long been dominating the online Chinese character recognition (OLCCR) technology, whereas the statistical scheme and the hybrid scheme are receiving increasing attention in recent years. The detail of each approach and some examples of it will be discussed in the next section.

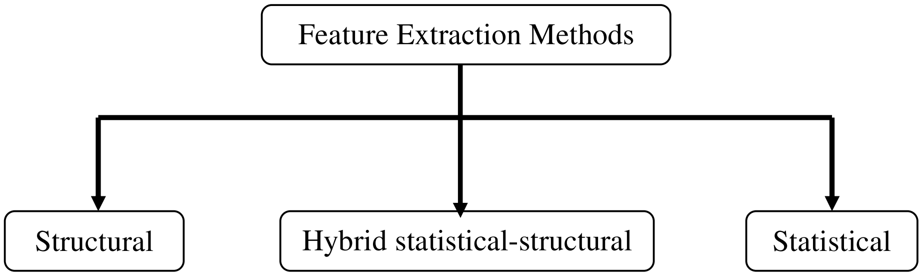


Figure 2.1: Categorization of the feature extraction models.

2.2.1 Structural Approaches

The structural representation schemes decompose the character into primitives (smaller segments) which act as the features of the character. The structural approaches can be further partitioned into five levels: sampling points, feature points or line segments, stroke codes or Hidden Markov Models (HMMs), relational and hierarchical. The hierarchy of these five levels is shown in Figure 2.2. To describe the pattern of the characters, the higher-level primitives are composed of the lower-level primitives. For example, a stroke is constructed by the line segments or sampling points, while the relation structure takes strokes as primitives.

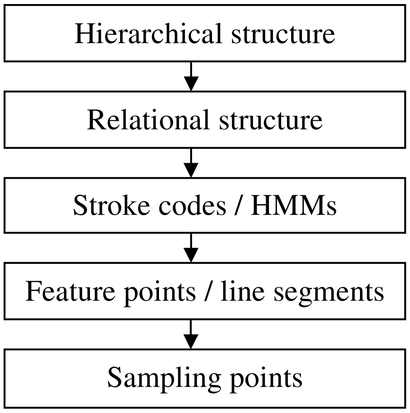
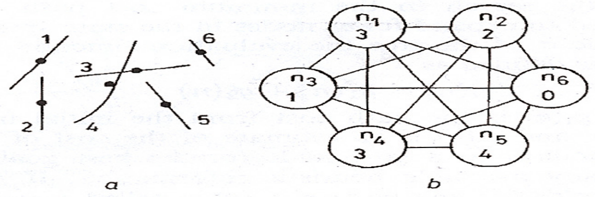


Figure 2.2: Hierarchy of structural representation schemes.

Some examples of the feature extraction method based on structural approach are described as follows.

I. Attributed Relational Graph

Attributed Relational Graph (ARG) was first used to represent the structural information of patterns by Tsai and Fu (1979). This feature extraction technique was then applied to Chinese character recognition by Lu *et al.* (1991) and Liu *et al.* (1996). To represent the complex structure of a Chinese character with an ARG, the straightaway way is that the nodes of the ARG describe the strokes of the character and the arcs describe the relations between any two different strokes. An example of the complete ARG representation for a character pattern is shown in Figure 2.3. Firstly, the geometric centre on each stroke of a character as shown in Figure 2.3(a) is determined. Secondly, the stroke types are represented by nodes n_{1-6} in a complete ARG as in Figure 2.3(b), where 0=short stroke, 1=horizontal stroke, 2=vertical stroke, 3=anti-diagonal stroke and 4=diagonal stroke. Lastly, the relations of the strokes are represented in a generalized relation matrix $R = \left[r_{ij} \right]_{6\times 6}$ as shown in Figure 2.3(c), where $r_{ij} = (a_{ij}^1, a_{ij}^2, a_{ij}^3; a_{ij}^1 = \{above/below\}$, $a_{ij}^2 = \{left/right\}, a_{ij}^2 = \{intersect/no\ intersect\})$ and $a_{ij}^1, a_{ij}^2, a_{ij}^3 \in \{0,1,2\}$. Here, $r_{ii}\ (i = 1,2,\ldots,6)$ are not defined and $a_{ij}^k = 2\ (k = 1,2,3)$ if the relation between the geometric centres of two strokes is uncertain. To tolerate with handwriting variation, the relation matrix of the Chinese characters in the database must be designed carefully.



	n_1	n_2	n_3	n_4	n_5	n_6
n_1		(1,2,0)	(2,1,0)	(2,1,0)	(1,1,0)	(2,1,0)
n_2	(0,2,0)		(2,1,0)	(2,1,0)	(2,1,0)	(0,1,0)
n_3	(2,0,0)	(2,0,0)		(2,2,1)	(1,2,0)	(2,1,0)
n_4	(2,0,0)	(2,0,0)	(2,2,1)		(2,1,0)	(0,1,0)
n_5	(0,0,0)	(2,0,0)	(0,2,0)	(2,0,0)		(0,2,0)
n_6	(2,0,0)	(1,0,0)	(2,0,0)	(1,0,0)	(1,2,0)	

c

Figure 2.3: (a) Geometric centre on each stroke of a Chinese character sample, (b) Complete ARG of character in (a), (c) Generalised relation matrix of ARG in (b).

II. Fuzzy Attributed Relational Graph

Fuzzy Attributed Relational Graph (FARG) is an improved version of ARG which has been applied to HCCR by Chan and Cheung (1992), and Zheng *et al*. (1997). Fuzzy set theory which was first introduced by Zadeh (1965) allows the gradual assessment of the membership of elements in a set. Such concepts are suitable for the characteristics of handwritten Chinese characters and it describes the uncertainty in stroke types, stroke relations and many other properties of strokes.

Figure 2.4 shows how FARG describes the structure of a Chinese character, where $\tilde{R}$ describes the relationship of its two strokes in X-axis and $\mu_{\tilde{R}}$ is the membership function of the stroke relation. FARG is able to reach a far better tradeoff between precision and robustness than ARG since the attribute set of node or arc is described with the aid of a membership function valued in the real unit interval [0, 1], instead of binary terms (says 1 and 0 only). It can also handle both stroke order variation and stroke connection problem. Furthermore, by using the fuzzy set concept, FARG is able to handle handwriting variation efficiently which is of great importance. However, learning process is needed in order to obtain the optimal attribute value, membership function, stability factor and some other parameters.

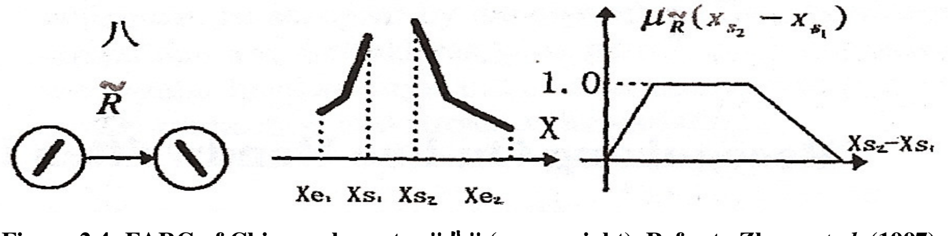


Figure 2.4: FARG of Chinese character "八" (means eight). Refer to Zheng *et al*. (1997) for more detail.

III. Delaunay Triangulation

Zeng *et al.* (2006) introduced a novel feature extraction scheme for online handwritten characters. It is based on Delaunay triangles in describing each stroke segment. As a shape context descriptor, Delaunay triangle descriptor captures the geometrical information and topological structure of the handwritten characters. It also covers both local and global features. For a constrained edge, its own information and its feature triangles on each side are combined into a feature vector. The three main components, i.e. current edge, left and right Delaunay triangles, involved in the feature vector are described below.

1. For the current directional edge $\mathbf{e}$, three local features are selected as its information: (i) Edge type t_e, (ii) Edge length l_e (iii) Edge direction θ_e.
2. For the left Delaunay triangle $\mathbf{T}_1$ of $\mathbf{e}$, five feature elements are designed: (i) Triangle type $t\Delta$, (ii) Triangle area $S\Delta$, (iii) Edge length l_2, (iv) Edge length l_3 and (v) Opposite angle θ_{op}.
3. For the right Delaunay triangles $\mathbf{T}_2$ of $\mathbf{e}$, the same five features as for the left triangle $\mathbf{T}_1$ are utilized.

An example of Delaunay triangulation is demonstrated diagrammatically in Figure 2.5 and the feature vector of *AB* is defined as follows.

$$\left[t_{AB}, |AB|, \theta_{AB}, t\Delta_{ABC}, \sqrt{(S\Delta_{ABC})}, |BC|, |CA|, \angle ACB, t\Delta_{ABD}, \sqrt{(S\Delta_{ABD})}, |BD|, |DA|, \angle ADB\right] \tag{2.1}$$

The Delaunay triangle descriptor has good discrimination power since they satisfy the desirable properties of the Delaunay triangulation, such as (i) it is unique, (ii) it can be calculated efficiently in an expected linear time for a planar region with edge constraints and (iii) noise or deformation of characters affect it locally only. As a result, higher accuracy and stability can be achieved if compared with other alternative feature combinations.

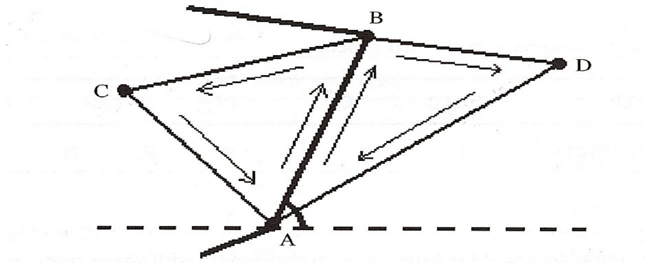


Figure 2.5: Feature extraction for the stroke AB and on each side are its Delaunay triangles ΔABC and ΔABD.

2.2.2 Statistical-Structural Approaches

In a statistical-structural representation scheme, the same structure as the traditional structural representation is taken, yet the structure elements (primitives) and/or relationships are measured probabilistically to better model the shape variation of input characters. In general, any structural model can be described probabilistically by replacing the attributes of primitives and/or relationships with probability density functions (PDFs). In Liu *et al.* (1993), the mean and variance of stroke and relationship attributes are connected to PDF representations. Besides, Gaussian PDFs have been used to illustrate the distribution of feature points and stroke attributes in Chen *et al.* (1988) and Zheng *et al.* (1999) respectively.

The examples of the feature extraction method based on statistical-structural approach are illustrated below.

I. Whole Character-based Hidden Markov Model

The theory of hidden Markov model (HMM) was initially introduced and studied by Baum and Petrie (1966). An example of HMM with probabilistic parameters is shown in Figure 2.6. Takahashi *et al.* (1997) proposed a fast discrete HMM algorithm for online handwritten character recognition. In the construction of HMM, let $O(t) := \left(V_1(t), V_2(t)\right), t = 1, \ldots, T$ be the observed output sequence, where $V_1(t)$ represents the pen up or pen down information which is already quantized and $V_2(t)$ represents the quantized angle information. An HMM of a character H, with the set of parameters $\left(\left\{a_{ij}\right\}, \left\{b_{ik}^1\right\}, \left\{b_{il}^2\right\}, \pi, N\right)$ is defined by the joint distribution as follows:

$$P\left(\left\{Q(t), O(t)\right\}_{t=1}^{T} \mid H\right) = \pi_{Q(1)} \prod_{t=1}^{T-1} a_{Q(t+1)Q(t)} \prod_{t=1}^{T} b_{Q(t)V_1(t)}^{1} b_{Q(t)V_2(t)}^{2} \quad (2.2)$$

where $Q(t) \in \left\{q_i\right\}_{i=1}^{N}$ is state, $\left\{a_{ij}\right\}$ is state transition probability, $\left\{b_{ik}^1\right\}$ and $\left\{b_{ik}^1\right\}$ are output emission probabilities, $\left\{\pi_i\right\}$ is initial state probability and N is the number of states. The set of HMM parameters is estimated by the learning process. Since HMM naturally incorporates time evolution of a system, it is suitable for online handwriting recognition if pen trajectories are utilized. However, it requires very large memory if the number of character classes and their associated models become large.

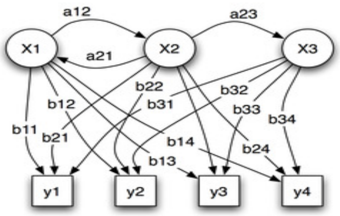


Figure 2.6: Hidden Markov model with probabilistic parameters *x* (states), *y* (possible observations), *a* (state transition probabilities) and *b* (output probabilities).

II. Substroke-based Hidden Markov Model

Substroke-based HMM approach was proposed by Nakai *et al.* (2001). Similar to whole character HMM, substroke HMM also has a topology of left-to-right model illustrated in Figure 2.7. The pen-down models have three states for each that represents different stroke velocities, whereas pen-up models have only one state that outputs a displacement vector without self-loop probability. Let $\lambda^{(k)} = \left(A^{(k)}, B^{(k)}, \pi^{(k)}\right)$ be the set of HMM parameters of substroke *k*, where $A^{(k)} = \left\{a_{ij}^{(k)}\right\}$ are the state-transition probability distributions from state S_i to S_j , $B^{(k)} = \left\{b_j^{(k)}(o)\right\}$ are the probability distributions of observation symbols o at state S_j and $\pi^{(k)} = \left\{\pi_i^{(k)}\right\}$ are the initial state probability distributions. The observation probability distribution is defined by a Gaussian distribution given as follows.

$$b_j(o) = \frac{1}{\sqrt{(2\pi)^n \left|\mathbf{\Sigma}_j\right|}} \exp\left(-\frac{1}{2}\left(o - \boldsymbol{\mu}_j\right)^t \Sigma_j^{-1} \left(o - \boldsymbol{\mu}_j\right)\right) \qquad (2.3)$$

where $\boldsymbol{\mu}$ is mean vector and $\mathbf{\Sigma}$ is covariance matrix. The substroke HMM is superior to the whole character HMM that the memory requirement for the database and models is significantly small. Besides, the recognition speed of substroke HMM can be improved by using efficient substroke network search,

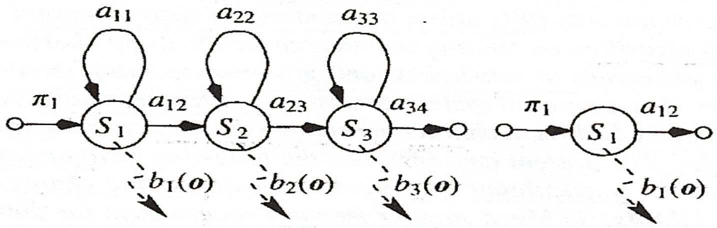


Figure 2.7: Substroke HMMs: (Left) pen down model, (Right) pen up model.

III. Path Controlled Hidden Markov Model

An improved HMM called path controlled hidden Markov model (PCHMM) was presented by Zheng *et al.* (1999). PCHMM model controls the state transition path directly with a Path Controlling Function $R(Q)$ on the state sequence space and it is denoted as

$$\lambda^{PC} = \left(\mathbf{A}, \mathbf{B}, \pi, R\right) \qquad (2.4)$$

where **A** is the state transition probability distribution matrix, **B** is the state emission density functions vector and π is the initial state distribution vector. In Zheng *et al.* (1999), the tie between every states of the PCHMM which represent each standard stroke segment of a Chinese character is fixed and an extra state describing all ligatures segments is included in the model. The main aim of this proposed method is to keep the tie while allowing different stroke orders. However, there is no topology of HMM that satisfies the above property. Hence, to tackle this problem, $R(Q)$ where $Q = q_1 q_2 \ldots q_T$ is introduced.

$$R\left(q_1 \ldots q_T\right) = \begin{cases} 0, & \exists 1 \le t_0 < t_1 < t_2 \le T \textit{ such that } S_0 \ne q_{t_0} = q_{t_0} \ne q_{t_1}, \\ 1 & \textit{otherwise} \end{cases} \qquad (2.5)$$

The training of PCHMM can be implemented with Z-algorithm as described in Zheng *et al.* (1999). The PCHMM strengthens the description ability of conventional left-right HMM (LRHMM) by using a path controlling function. Unlike LRHMM, it can tolerate well with the different stroke orders.

2.2.3 Statistical Approaches

For statistical representation approaches, the input pattern is described by feature vector, while the database contains the classification parameters, which can be estimated by standard statistical techniques. The feature vector representation of character patterns enables stroke-order and stroke-number free recognition by mapping the pattern trajectory into a 2D image and extracting so-called offline features (Hamanaka *et al.*, 1993). Therefore, in this context, various feature extraction strategies in offline character recognition can be applied to online recognition as well (Umeda, 1996; Hilderbrand and Liu, 1993).

The following shows some examples of the feature extraction method based on statistical approach.

I. Directional Feature Densities

Directional Feature Densities (DFD) is a statistical-based feature extraction method developed by Kawamura *et al.* (1992). Let P_{ij} be the *j*-th point on the *i*-th stroke and V_{ij} be the vector from P_{ij} to P_{ij+1}, that is

$$V_{ij} = P_{ij+1} - P_{ij} \tag{2.6}$$

Let e^1, e^2, e^3 and e^4 be the elementary vectors of each of 4-directions as described in Figure 2.8. The angle range $\{\theta \mid -\pi \le \theta < \pi\}$ is divided into 4 sub-ranges, Ω^1, Ω^2, Ω^3 and Ω^4. The directional feature vector

$$\mathbf{A}_{ij} = \left(\left|a_{ij}^1\right|, \left|a_{ij}^2\right|, \left|a_{ij}^3\right|, \left|a_{ij}^4\right|\right) \tag{2.7}$$

is determined by using the constraint below.

$$V_{ij} = \begin{cases} a_{ij}^1 e^1 + a_{ij}^2 e^2, a_{ij}^3 = a_{ij}^4 = 0 & if \angle V_{ij} \in \Omega^1 \\ a_{ij}^2 e^2 + a_{ij}^3 e^3, a_{ij}^4 = a_{ij}^1 = 0 & if \angle V_{ij} \in \Omega^2 \\ a_{ij}^3 e^3 + a_{ij}^4 e^4, a_{ij}^1 = a_{ij}^2 = 0 & if \angle V_{ij} \in \Omega^3 \\ a_{ij}^4 e^4 + a_{ij}^1 e^1, a_{ij}^2 = a_{ij}^3 = 0 & if \angle V_{ij} \in \Omega^4 \end{cases} \qquad (2.8)$$

An example of this is illustrated in Figure 2.9. Next, the character is divided into 15×15 square areas. Let R_{mn} be the square at row m and column n. The $4 \times 15 \times 15$ - dimensional vector $\mathbf{F}$ is defined as follows.

$$\mathbf{F} = \left\{ f_{mn}^l \mid l = 1,2,3,4; m,n = 1,2,\ldots,15 \right\},\ where\, f_{mn}^l = \sum_{P_{ij} \in R_{mn}} \left| a_{ij}^l \right| \qquad (2.9)$$

Then, $\mathbf{F}$ is condensed into a $4 \times 8 \times 8$ -dimensional vector $\mathbf{G}$ with a spatial weighted filter $\mathbf{B} = \{ b_{sr} \mid s, r = -1, 0, 1 \}$.

$$G = \left\{ g_{ij}^l \mid l = 1,2,3,4; i, j = 1,2,\ldots,8 \right\} \qquad (2.10)$$

$$g_{mn}^l = \sum_{s=-1}^{1} \sum_{r=-1}^{1} b_{sr} f_{2m-1+s,2n-1+r,}^l\ with\ f_{0n} = f_{m0} = 0 \qquad (2.11)$$

The vector $\mathbf{G}$ is known as the DFD which describes how much of each directional feature a character pattern has in each area. DFD is independent of both stroke number and stroke order, but depends only on information of the writing direction.

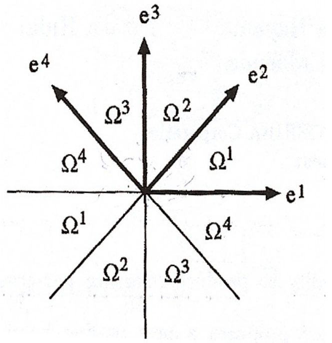


Figure 2.8: The four elementary vectors and the angle areas.

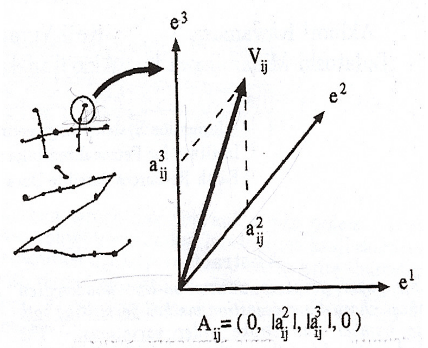


Figure 2.9: Directional feature.

II. Directional Element Feature

Directional element feature (DEF) is developed by Kato *et al.* (1999) and the feature is extracted from a bitmap using an offline approach. The operation for extracting the DEF includes three steps: contour extraction, dot orientation and vector construction. After preprocessing, contour extraction is implemented. If a white pixel adjoins a black pixel to the upward, downward, left or right direction, then the black pixel is regarded as on the contour. The feature vector is extracted from the pixels of contour. In dot-orientation, four types of line elements: vertical, horizontal, diagonal and anti-diagonal are assigned to each black pixel. For a center black pixel in a 3×3 mask, two cases are considered as illustrated in Figure 2.10.

For vector construction, an input pattern is placed in a 64×64 mesh which is first divided into 49, or 7×7 subareas of 16×16 pixels (refer to Figure 2.11). Each subarea overlaps eight pixels of the adjacent subareas. Then, each subarea is further divided into four regions: *A, B, C* and *D*. In order to reduce the negative effect caused by position variation of character image, weighting factors are defined greater at the center of each subarea and decrease towards the edge, that is 4, 3, 2, 1 for the regions *A, B, C, D* respectively. Each subarea is defined as a four-dimensional vector shown below.

$$x = \left(x_1, x_2, x_3, x_4\right) \tag{2.12}$$

where x_1, x_2, x_3, x_4 represent the element quantities of the four orientations. Each element quantity is determined as follows.

$$x_j = 4x_j^{(A)} + 3x_j^{(B)} + 2x_j^{(C)} + x_j^{(D)},\ j = 1, \ldots, 4 \tag{2.13}$$

where $x_j^{(A)}$, $x_j^{(B)}$, $x_j^{(C)}$ and $x_j^{(D)}$ denote the quantity of each element in *A, B, C* and *D* respectively. This vector is known as DEF. To achieve a better DEF, transformation based on partial inclination detection (TPID) (Kato *et al.*, 1999) can also be applied to eliminate some distortions caused by writers' habits.

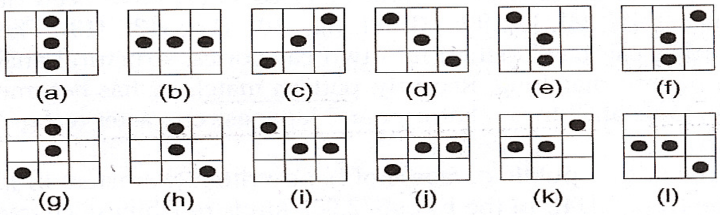


Figure 2.10: Types of connections of black pixels. One type of line element is assigned for center pixels in the cases of (a) - (d). Two types of line elements are assigned for center pixels in the cases of (e) - (l).

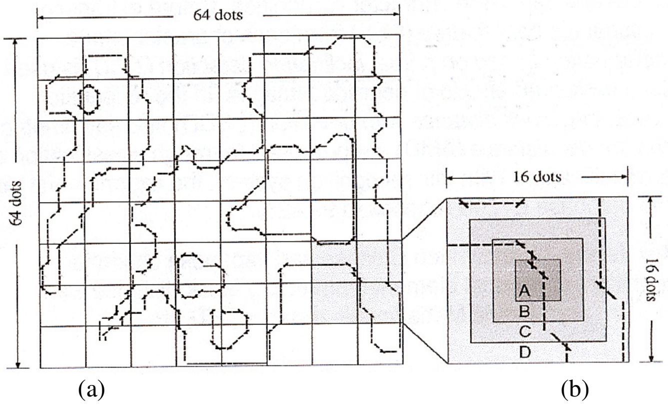


Figure 2.11: (a) Oriented-dot image of a Chinese character, (b) One Subarea of the 64×64 mesh.

III. 8-Directional Features

The feature extraction method introduced in Section 2.2.3(I) and 2.2.3(II) is based on 4-directional features. In order to improve the feature to a more detailed extent, 8-directional features are presented by Bai and Huo (2005). A comparison of 4-directional and 8-directional features is illustrated in Figure 2.12.

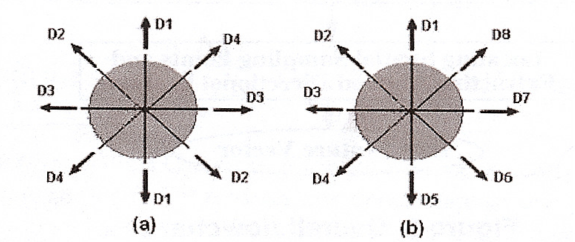


Figure 2.12: A notion of (a) 4 directions and (b) 8 directions.

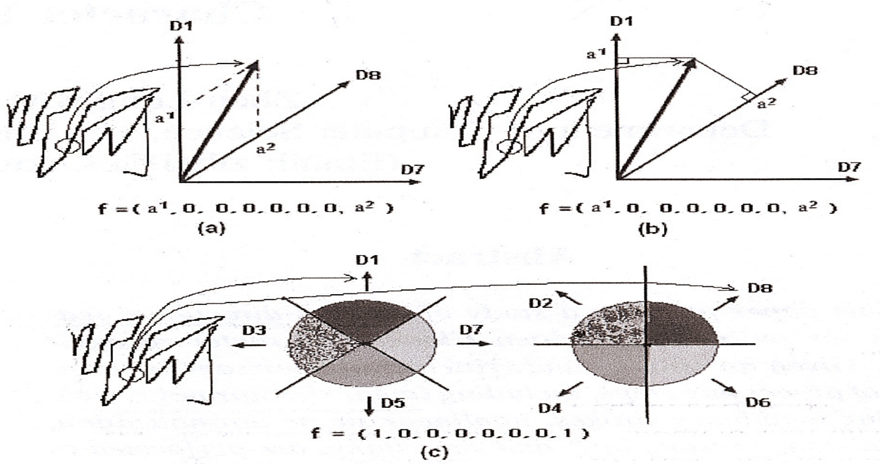

D1
D8
a¹
a²
D7
f = (a¹, 0, 0, 0, 0, 0, 0, a²)
(a)
D1
a¹
D8
a²
D7
f = (a¹, 0, 0, 0, 0, 0, 0, a²)
(b)
D1
D8
D2
D3
D7
D5
D4
D6
f = (1, 0, 0, 0, 0, 0, 0, 1)
(c)

Figure 2.13: Different ways of projecting a direction vector onto directional axes and the corresponding directional "feature values": (a) adapted from (Kawamura *et al.*, 1992); (b) adapted from (Nakagawa *et al.*, 1996); (c) adapted from (Bai and Huo, 2005).

Given a stroke point P_j, its normalized direction vector, $\vec{V}_j / \|\vec{V}_j\|$, is projected onto two 4 directional axes, as shown in Figure 2.13. One is from the directional set of $\{D_1, D_3, D_5, D_7\}$ and denoted as $d_j^1$and the other is from the set of $\{D_2, D_4, D_6, D_8\}$ and denoted as d_j^2 . An 8-dimensional feature vector can be formed with non-zero directional feature values a_j^1 and a_j^2 corresponding to d_j^1 and d_j^2 respectively. Whereas, feature values correspond to the remaining 6 directions are set as 0s. Basically, there are different ways to calculate a_j^1 and a_j^2. (refer to Bai and Huo (2005), Kawamura *et al.*, (1992), Nakagawa *et al.* (1996)). After extracting 8-directional features from all the points of a character, 8 directional pattern images

$\left\{B_d = \left[f_d\left(x,y\right)\right], x,y = 1,2,\ldots,64; d = D_1, D_2,\ldots,D_8\right\}$ can be generated as follows: set $f_{d_j^1}\left(x_j, y_j\right) = a_j^1$, $f_{d_j^2}\left(x_j, y_j\right) = a_j^2$ and set the values for all the other $f_d\left(x,y\right)$ as 0s. Each directional pattern image is then divided uniformly into 8×8 grids whose centers are treated as locations of 8×8 spatial sampling points. Finally, the $8\times8\times8 = 512$ dimensional feature vector is formed by using the nonlinear transformed features below.

$$\left\{\sqrt{F_d\left(x_i, y_j\right)}, d = D_1, D_2, \ldots, D_8; i,j = 1,2,\ldots,8\right\} \qquad (2.14)$$

Bai and Huo (2005) have proved that using 8-directional features gives better performance in recognition system than that of 4-directional features.

2.3 Historical Background of Classifiers

If compared to radical-based and stroke-based approaches, holistic approaches which recognize character as a whole without preliminary segmentation is the most popular in HCCR. For holistic approaches, the design of classifiers is varying. During the last decades, the statistical approaches (Jain *et al.*, 2000), especially classifiers based on quadratic discriminant function (QDFs) have been applied successfully to HCCR. They become the most popular methods in the literature due to their simplicity and robustness. Among them, the most widely used one is the modified quadratic discriminant function (MQDF) proposed by Kimura *et al.* (1987) and it makes a vital part in the reported high accuracy classifiers. Compared to the ordinary QDF, the MQDF reduces the computational complexity and meanwhile improves the generalization performance by replacing the eigenvalues in the minor subspace of each class with a constant. However, for memory limited hand-held devices, MQDF classifier still faces parameter complexity problem for large Chinese character set. Thus, to overcome this problem, compact MQDF was developed by Long and Jin (2008). Furthermore, other improved versions of QDF also emerged in recent years, which include discriminative learning quadratic discriminant function (DLQDF) (Liu, 2006), regularized discriminant analysis (RDA) (Kawatani, 2000) and so on.

Although the quadratic classifiers mentioned above perform fairly well in HCCR, they fail to discriminate all classes well, especially similar characters. Thus, neural network (NN) classifiers (Fu and Xu, 1998; Saruta *et al*., 1996) and support vector machine (SVM) (Burges, 1998; Dong *et al*., 2005; Kilic *et al*., 2008) which are appropriate for discriminating subset of similar characters were developed. Normally, the classifiers for subset of classes are more preferable than the all-class classifiers due to the lower separation complexity. Before discriminating the subset of similar characters, this particular subset can be determined in advance according to nearest neighbor rules or using a coarse classifier. Probabilistic decision-based NNs are designed by Fu and Xu (1998) for discriminating groups of classes divided by clustering. Each network is trained by using the samples of the class in a group. Other alternative versions of NN are based on learning vector quantization (LVQ) algorithm and decision surface mapping (DSM) algorithm proposed by Romero *et al*. (1997), and geometrical approach proposed by Wu *et al*. (2000). Dong *et al*. (2005) used SVM in HCCR to classify one class from the others. He designed SVM trained on all the samples for each class by using a fast algorithm. For SVM, the training complexity can be reduced via training with the samples of one class (positive samples) and its similar classes (negative samples) only, as done by NN classifiers in Saruta *et al*. (1996). Despite the high discrimination ability of NN and SVM classifiers in differentiating subsets of similar characters, the subsets of classes must be fixed in most of the cases.

Another well-known classification technique used to differentiate similar characters is compound Mahalanobis function (CMF) which was developed by Suzuki *et al.* (1997). This CMF can be viewed as a pairwise classifier. It calculates a complementary distance on a one-dimensional subspace (discriminant vector) for a pair of classes. The complementary distance is then used to further discriminate the similar classes. However, the decision boundary between similar classes is so complicated that using only a discriminant vector on one-dimensional subspace is insufficient. Hence, Hirayama *et al.* (2007) proposed Difference Subspace based Compound Mahalanobis Function (DS-CMF), which treats difference between two similar classes as a multi-dimensional projective space. This has improved the conventional CMF by further emphasizing the difference of similar classes and thus, achieved higher accuracy in recognition. On the other hand, asymmetric Mahalanobis distance (AMD) is another improved classifier which is under the category of Mahalanobis distance (MD) classifier.

Furthermore, from the perspective of training process, there are two different types of classifiers which are discriminative classifiers and generative (or model-based) classifiers. Discriminative approaches (Bahlmann *et al.*, 2002) deal with the problem of between class variations during training stage and try to construct decision boundaries to distinguish classes. On the other hand, generative approaches (Bahlmann and Burkhardt, 2004; Mitoma *et al.*, 2004) attempt to model the variations within a class during training process, so that these variations may be de-emphasized while computing the similarity of the test sample to that class. An example for discriminative classifier is Neural Network classifier while Dynamic Time Warping (DTW) is the example of generative classifier."

2.4 Classifiers

In this section, some classifiers based on holistic approaches which have been proposed in HCCR, as mentioned above will be discussed.

2.4.1 Quadratic Discriminant Function

Let n be the dimension of feature vector and the probability density function of n-dimensional normal distribution is

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{n/2}|\mathbf{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^t \mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\} \tag{2.15}$$

where $\mathbf{x}$ is an n-dimensional vector, $\boldsymbol{\mu}$ is the mean vector and $\mathbf{\Sigma}$ is the $n \times n$ covariance matrix. As stated in Omachi *et al.* (2000), the quadratic discriminant function (QDF) is derived from Equation (2.15) as below:

$$\begin{aligned} g(\mathbf{x}) &= (\mathbf{x}-\boldsymbol{\mu})^t \mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}) + \log|\mathbf{\Sigma}| \\ &= \sum_{i=1}^{n} \frac{\left((\mathbf{x}-\boldsymbol{\mu})^t \phi_i\right)^2}{\lambda_i} + \sum_{i=1}^{n} \log \lambda_i \end{aligned} \tag{2.16}$$

where λ_i is the ith eigenvalue of $\mathbf{\Sigma}$ (sorted in descending order) and ϕ_i is the eigenvector that corresponds to λ_i.

I. Modified Quadratic Discriminant Function

The modified quadratic discriminant function (MQDF) is originally proposed by Kimura *et al.* (1987). It is a modified version of the ordinary QDF. It reduces the complexity of QDF by replacing the minor eigenvalues of covariance matrix of each class with a constant. MQDF is defined as

$$\begin{aligned} d_i(\mathbf{x}) &= \sum_{j=1}^{K} \frac{1}{\lambda_{ij}} \left[\phi_{ij}^T (\mathbf{x} - \mu_i) \right]^2 + \sum_{j=K+1}^{D} \frac{1}{\delta_i} \left[\phi_{ij}^T (\mathbf{x} - \mu_i) \right]^2 + \sum_{j=1}^{K} \log \lambda_{ij} + (D-K) \log \delta_{\mathrm{i}} \\ &= \frac{1}{\delta_i} \left(\left\| \mathbf{x} - \mu_i \right\|^2 - \sum_{j=1}^{K} (1 - \frac{\delta_i}{\lambda_{ij}}) \left[\phi_{ij}^T (\mathbf{x} - \mu_i) \right]^2 \right) + \sum_{j=1}^{K} \log \lambda_{ij} + (D-K) \log \delta_{\mathrm{i}} \end{aligned} \tag{2.17}$$

where $\mathbf{x} = (x_1, x_{2,} x_3, \ldots, x_d)^T$ denotes a d-dimensional feature vector for input character pattern, μ_i is the mean of ith class for $i = 1, 2, \ldots, M$ such that M is the number of classes, λ_{ij} denote the descending order eigenvalues of the ith class covariance matrix Σ_i for $j = 1, \ldots, D$ such that D is the dimension of μ_i, ϕ_{ij} are the ordered eigenvectors, δ_i is constant replacing the minor eigenvalues and K denotes the number of dominant eigenvectors. The Equation (2.17) utilizes the invariance of Euclidean distance:

$$d_E(\mathbf{x}, w_i) = \left\| \mathbf{x} - \mu_i \right\|^2 = \sum_{j=1}^{D} \left[\phi_{ij}^T (\mathbf{x} - \mu_i) \right]^2 \tag{2.18}$$

$\mathbf{x}$ is assigned to ith class if $d_i(\mathbf{x})$ yields the largest numerical value. Although omitting the minor eigenvectors in MQDF can improve the performance of classification in term of storage space and speed, it still faces the parameter complexity problem for large character set recognition. Hence, it is not practical to apply it directly into the memory limited devices.

II. Compact Modified Quadratic Discriminant Function

To overcome the parameter complexity problem of MQDF mentioned in Section 2.4.1(I), Long and Jin (2008) have developed a compact MQDF which reduces the storage of the ordinary MQDF classifier by combining linear discriminant analysis (LDA) and subspace distribution sharing.

In practice, the feature vector is usually compressed to a smaller size by LDA (Liu and Ding, 2005) before using MQDF. However, it is still not small enough for most of the memory limited embedded devices. Hence, a kind of vector quantization (VQ) is applied to the classifier by Long and Jin (2008) to further compress the dominant eigenvectors of each class. For each class w_i, the dominant eigenvector matrix $\Phi_i^K = [\phi_{i1},\ldots,\phi_{iK}]$ with $\phi_{ij}\,(j=1,\ldots,K)$ is partitioned into subspace eigenvector matrix, i.e. each D-dimensional eigenvector ϕ_{ij} is equally partitioned into Q sub-vectors $\phi_{ij}^1,\phi_{ij}^2,\ldots,\phi_{ij}^Q$ with D_Q dimension, where D=D_Q×Q. Then, a general statistical model is determined for the distributions of all the sub-vectors. By using Linde-Buzo-Gray (LBG) clustering algorithm (Linde *et al.*, 1980) in subspaces of the parameters, the sub-vectors $\phi_{ij}^q\left(i=1,\ldots,M, j=1,\ldots,K, q=1,\ldots,Q\right)$ are clustered into a small set of L prototypes. Each original subspace eigenvectors is then presented by its nearest prototype. The block diagram of the recognition system using the compact MQDF classifier is illustrated in Figure 2.14. This method can be a general compressing method for any classifier.

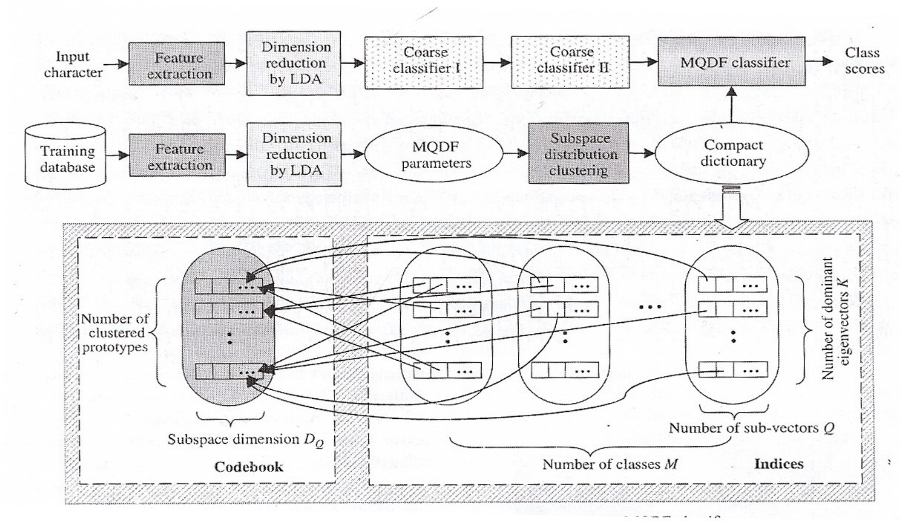


Figure 2.14: Block diagram of the recognition system using the compact MQDF classifier.

III. Discriminative Learning Quadratic Discriminant Function

Although MQDF can reduce the computational complexity and improve classification performance compared to the original QDF, it fails to perform well for classes which are not in Gaussian distribution. In order to overcome this problem, the parameters of MQDF can be optimized on training samples by optimizing the minimum classification error (MCE) criterion (Juang and Katagiri, 1992). This optimized discriminant function is known as discriminative learning QDF (DLQDF). It was applied to HCCR by Liu (2006) and obtained a promising result.

Let **x** be the input feature vector and w_i be the character class. The parameters of DLQDF, i.e. mean vector, eigenvalues and eigenvectors are updated iteratively on a training sample set to minimize the empirical loss. The empirical loss is summed up over a training sample set. To constrain the motion of parameters, a regularization term related to maximum likelihood (ML) is added to the empirical loss function:

$$L = \frac{1}{N}\sum_{n=1}^{N}\left[l_c\left(\mathbf{x}^n\right) + \alpha d_Q\left(\mathbf{x}^n, w_c\right)\right] \tag{2.19}$$

where $d_Q\left(\mathbf{x}^n, w_c\right)$ is quadratic distance between input feature vector and genuine class, $l_c\left(\mathbf{x}^n\right)$ is empirical loss, and α is regularization coefficient. In the learning process for parameters of DLQDF, eigenvalues are kept positive by transforming them into exponential functions and eigenvectors of each class are kept orthonormal by Gram-Schmidt orthonormalization. For more detail, ones can refer to Liu *et al.* (2004).

IV. Regularized Discriminant Analysis

One of the methods that have been proposed to solve QDF problems is the Regularized Discriminant Analysis (RDA) which is presented by Kawatani (2000). In fact, RDA (Friedman, 1989) is a determinant normalized QDF. In RDA, the influence of parameter estimation errors or determinant errors was reduced through regularization, i.e. normalizing the determinants of the covariance matrices. This is done by the combination of each class covariance matrix with the pooled matrix and biasing eigenvalues.

Let $\tilde{\mathbf{\Sigma}}_k\left(\beta\right)$ and $\mathbf{\Sigma}'_w$ be the covariance matrix of class k after regularization and the pooled covariance matrix respectively. Kawatani (2000) proposed $\tilde{\mathbf{\Sigma}}_k\left(\beta\right)$ to be defined as below:

$$\tilde{\mathbf{\Sigma}}_k\left(\beta\right) = \beta\mathbf{\Sigma}'_k + \left(1-\beta\right)\mathbf{\Sigma}'_w \tag{2.20}$$

Moreover,

$$\tilde{\mathbf{\Sigma}}_k\left(\beta,\gamma\right) = \tilde{\mathbf{\Sigma}}_k\left(\beta\right) + \left(1-\gamma\right)cI \tag{2.21}$$

where c is a constant. Equation (2.20) represents the combination of covariance matrix of class k with the pooled matrix while Equation (2.21) represents the biasing of eigenvalues. More detail about RDA can be found in McLachlan (1992) and Friedman (1989).

RDA gains better recognition rate due to the normalization of determinants of the covariance matrices. Besides, RDA requires less memory to achieve high accuracy. However, this method is currently an empirical method and the theoretical analysis still remains as a problem for the future.

2.4.2 Support Vector Machine

Support Vector Machine (SVM) is developed by Vapnik (1999). It operates under the principle of structural risk minimization rule. It was originally designed for binary classification (two class pattern classification) with the goal of finding the optimal hyperplane, so that the margin of separation between the negative and positive data set will be maximized. Gao *et al*., (2002) proposed the use of SVM in HCCR.

Denote $\{\mathbf{x}_i, y_i\}, i = 1, 2, \ldots, k$ to be the training samples, where $\mathbf{x}_i \in \Re^n$ is the training vector and $y_i \in \{-1, 1\}$ is its corresponding target value. With the input pattern $\mathbf{x}$, the decision function of binary classifier is

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{k} y_i \alpha_i K(\mathbf{x}, \mathbf{x}_i) + b \right) \qquad (2.22)$$

where $\operatorname{sgn}(u) = \begin{cases} 1 & for\ u>0 \\ -1 & for\ u<0 \end{cases}$, k is the number of learning patterns, b is a bias, α_i is the Lagrange multiplier of the optimization problem discussed below and $K(\mathbf{x}, \mathbf{x}_i)$ is a kernel function which defined as

$$K(\mathbf{x}, \mathbf{x}_i) = \phi(\mathbf{x}) \cdot \phi(\mathbf{x}_i) \qquad (2.23)$$

The kernel function is needed to transform the pattern into a higher dimensional space where a hyperplane can be used as the nonlinear decision boundary. Some common kernel functions include polynomial kernel, Gaussian radical basis function (RBF) kernel and sigmoid kernel which are shown in Equation (2.24), (2.25) and (2.26) respectively.

$$K(\mathbf{x}, \mathbf{x}_i) = \left(\mathbf{x} \cdot \mathbf{x}_i + 1\right)^P \qquad (2.24)$$

$$K(\mathbf{x}, \mathbf{x}_i) = \exp\left(\frac{\left\|\mathbf{x} - \mathbf{x}_i\right\|^2}{2\sigma^2} \right) \qquad (2.25)$$

$$K(\mathbf{x}, \mathbf{x}_i) = \tanh\left(a\mathbf{x} \cdot \mathbf{x}_i + r\right) \qquad (2.26)$$

In polynomial, the kernel is called linear if *p*=1 while if *p*=2, it is called quadratic kernel. For RBF kernel, the kernel width σ^2 is estimated from the variance of the sample vectors. The sigmoid kernel is equivalent to two-layer perceptron neural network. *a* can be viewed as a scaling parameter of the input pattern and *r* as a shifting parameter that controls the threshold of mapping when $a > 0$. The decision function in Equation (2.22) can be determined via training. Boser *et al*., (1992) stated that training SVM for pattern recognition problem can be formulated as the quadratic optimization problem:

$$\text{maximize: } \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\boldsymbol{\alpha}^T \mathbf{Q}\boldsymbol{\alpha} \qquad (2.27)$$

$$\text{subject to: } 0 \le \alpha_i \le C, i = 1, \ldots, l \text{ and } \sum_{i=1}^{l} y_i \alpha_i = 0$$

where $\boldsymbol{\alpha}$ is a vector of length *l* and its component α_i corresponds to a training sample $\{\mathbf{x}_i, y_i\}$, $\mathbf{Q}$ is an $l \times l$ semidefinite kernel matrix and C is a regularization parameter that controls the tolerance of classification errors in training. The training vector $\mathbf{x}_i$ whose corresponding α_i is nonzero is known as support vector. It constrains the width of the margin that separates two classes. With the support vector, the decision function and the optimal hyperplane can be obtained. Since the training kernel matrix grows proportionally with data set, training SVM on large data sets is a very time-consuming and thus a fast training algorithm (Dong *et al.*, 2005) is needed.

2.4.3 Mahalanobis Distance

The Mahalanobis Distance (MD) (Pinho *et al.*, 2007) is a standard approach used in data association for tracking features along image sequences. MD, also known as a statistical distance, is a distance which each of its components takes their variability into account when determining its distance to the corresponding centre. For two points $\mathbf{x}_i = (x_{1i}, x_{2i}, \ldots, x_{ni})$ and $\mathbf{y}_i = (y_{1i}, y_{2i}, \ldots, y_{ni})$, the MD is given as follows:

$$d_M = \sqrt{(\mathbf{x}_i - \mathbf{y}_i)^T C^{-1} (\mathbf{x}_i - \mathbf{y}_i)} \qquad (2.28)$$

where $C_{(n \times n)}$ is a non-singular covariance matrix.

I. Compound Mahalanobis Function

The Compound Mahalanobis Function (CMF) has been constructed by Suzuki *et al.* (1997) to clearly differentiate the similar characters. It is a compound discriminant function which improves the performance of the ordinary Mahalonabis Distance (MD), by projecting the difference of the class-mean feature vectors of two similar classes onto a certain subspace such that the difference between two similar classes is obviously appears.

Let Ω_1 and Ω_2 be the two similar classes. Denote the eigenvalues of the covariance matrix of class Ω_1 as $\lambda_1, \lambda_2, \cdots \lambda_d$ where $\lambda_i \geq \lambda_{i+1}, i = 1, 2, \cdots, d-1$ and the eigenvectors which corresponds to λ_i as ϕ_i. By adding the mean of eigenvalues of all class to each eigenvalue as bias b and transforming non-dominant eigenvectors $\phi_i,\ p+1 \leq i \leq d, p = 1, 2, \ldots, d-1$ to the subtraction-form of Euclidean Norm approximating $\lambda_i \square\ b(i \geq p+1)$, the CMF for class Ω_1 is defined by Suzuki *et al.* (1997) in the form shown below:

$$\begin{aligned}\mathrm{CMF}_1(\mathbf{x}, \mathbf{u}) = {} & \sum_{i=1}^{p} \frac{\left\{\phi_i^T (\mathbf{x} - \mathbf{u})\right\}^2}{\lambda_i + b} + \frac{1}{b}\left\{\left\|\mathbf{x} - \mathbf{u}\right\|^2 - \sum_{i=1}^{p}\left\{\phi_i^T (\mathbf{x} - \mathbf{u})\right\}^2\right\} \\ & + \mu\left[\sum_{i=k+1}^{p} \frac{\left\{\phi_i^T \boldsymbol{\delta}_1\right\}^2}{\lambda_i + b} + \frac{1}{b}\left\{\left\|\boldsymbol{\delta}_1\right\|^2 - \sum_{i=1}^{p}\left\{\phi_i^T \boldsymbol{\delta}_1\right\}^2\right\}\right] \qquad (2.29)\end{aligned}$$

where $\mathbf{x} = (x_1, x_2, \cdots, x_d)^T$ is an input character pattern, $\mathbf{u}$ is a class-mean vector of class Ω_1, μ is the weighting parameter and $\boldsymbol{\delta}_1$ is a projective vectors for Ω_1 as shown in the following:

$$\boldsymbol{\delta}_1 = \left\{\psi^T(\mathbf{x}-\mathbf{u})\right\}\psi\,,\ \psi = \frac{(\mathbf{u}-\mathbf{v}) - \sum_{i=1}^{k}\left\{\phi_i^T(\mathbf{u}-\mathbf{v})\right\}\phi_i}{\sqrt{\|\mathbf{u}-\mathbf{v}\|^2 - \sum_{i=1}^{k}\left\{\phi_i^T(\mathbf{u}-\mathbf{v})\right\}^2}} \tag{2.30}$$

where $\mathbf{v}$ is a mean vector of Ω_2 and ψ is a unit vector obtained by projecting the difference vector of class-mean vectors $(\mathbf{u}-\mathbf{v})$ onto a subspace constructed by $\phi_{k+1}, \phi_{k+2}, \cdots, \phi_d$ and normalizing the length to 1.

CMF, as a fine classifier, stores no extra parameter in addition to the coarse classifier. Although previous works (Liu and Ding, 2005; Suzuki *et al.*, 1997; Nakajima *et al.*, 2000) have proved outstanding recognition performance in CMF, it has two drawbacks. Firstly, discriminant vector for two classes is not optimized. Secondly, the calculation for discriminant vector during recognition is time-consuming despite no extra parameters is involved in CMF.

II. Difference Subspace based Compound Mahalanobis Function

The conventional CMF which is based on a difference vector of two class-mean vectors is still inadequate to illustrate the difference information between similar characters. Therefore, Hirayama *et al.* (2007) proposed Difference Subspace based Compound Mahalanobis Function (DS-CMF) that expands the original CMF by using the concept of “Difference Subspace”. A difference vector *d* is a difference of two particular vectors with single dimension while Difference Subspace is an extension of difference vector to multi dimension as diagramed in Figure 2.15.

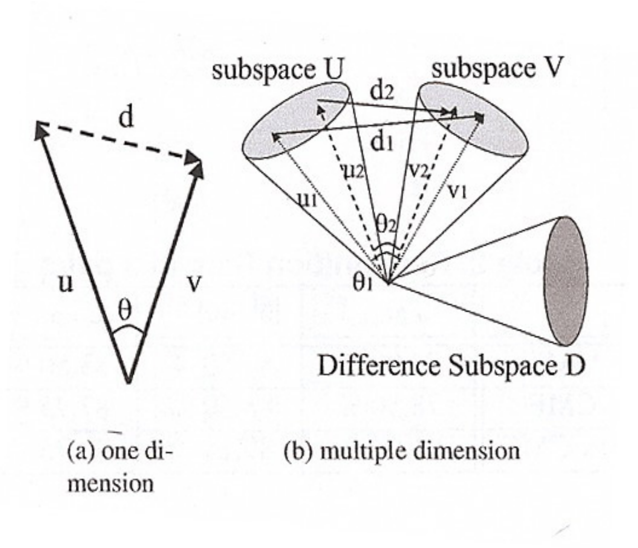


Figure 2.15: Difference subspace.

Let $\mathbf{P}$, $\mathbf{Q}$ be projection matrices of similar character classes Ω_1 and Ω_2 respectively, and $\mathbf{G}$ be the sum of these matrices as defined below:

$$\mathbf{P} = \sum_{i=1}^{N_b} \phi_i \phi_i^T \ , \quad \mathbf{Q} = \sum_{i=1}^{N_b} \varphi_i \varphi_i^T \qquad (2.31)$$

$$\mathbf{G} = \mathbf{P} + \mathbf{Q} \qquad (2.32)$$

where ϕ_i and φ_i are eigenvectors which construct the subspaces of Ω_1 and Ω_2 respectively. N_p eigenvectors which corresponds to the N_p greatest eigenvalues are common space of two similar classes whereas the remaining $N_b \times 2 - N_p$ eigenvectors which corresponds to the smallest eigenvalues are difference space. Among the $N_b \times 2$ eigenvectors $\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_{N_b \times 2-1}, \mathbf{d}_{N_b \times 2}$ of $\mathbf{G}$, Difference Subspace *D* is defined as follows:

$$D = \left(\mathbf{d}_{N_p+1}, \mathbf{d}_{N_p+2}, \ldots, \mathbf{d}_{N_b \times 2-1}, \mathbf{d}_{N_b \times 2}\right) \qquad (2.33)$$

The Mahalanobis Distance on the Difference Subspace is determined by

$$\sum_{i=N_p+1}^{N_b\times 2}\frac{\left\{\mathbf{d}_i^T\left(\mathbf{x}-\mathbf{u}\right)\right\}^2}{\lambda_i^d} \tag{2.34}$$

where $\mathbf{x}$ is an input character pattern, $\mathbf{u}$ is a class-mean vector of class Ω_1 and λ_i^d are the variances of $\mathbf{d}_i$. By making linear combination of Modified Mahalanobis Distance and correction term in Equation (2.34), DS-CMF for class Ω_1 is defined as follows:

$$\text{DS-CMF}_1(\mathbf{x},\mathbf{u})=\sum_{i=1}^{q}\frac{\left\{\phi_i^T\mathbf{x}\right\}^2}{\lambda_i+b}+\frac{1}{b}\left\{\left\|\mathbf{x}\right\|^2-\sum_{i=1}^{q}\left\{\phi_i^T\mathbf{x}\right\}^2\right\}+\upsilon\sum_{i=N_p+1}^{N_b\times 2}\frac{\left\{\mathbf{d}_i^T\left(\mathbf{x}-\mathbf{u}\right)\right\}^2}{\lambda_i^d+b^d} \tag{2.35}$$

It has proved that the Difference Subspace used in the DS-CMF is more efficient as a discriminant feature space than a projective space constructed in CMF because the difference of similar characters can be emphasized by using multi-dimensional difference vectors.

III. Asymmetric Mahalanobis Distance

Since majority of the distribution of the samples is asymmetric rather than normal, Mahalanobis distance (MD) is no longer a suitable function for it. Hence, asymmetric Mahalanobis distance (AMD) is proposed by Kato *et al.* (1999) to describe an asymmetric distribution. Denote $\mathbf{v}^1,\mathbf{v}^2,\ldots,\mathbf{v}^N$, $\mathbf{v}^i=\left(v_1^i,v_2^i,\ldots,v_n^i\right)$ be the *n*-dimensional feature vectors for the samples of a class, where *N* is the number of samples. Let $\boldsymbol{\mu}$ be the mean vector of these samples, λ_j and ϕ_j be the *j*th eigenvalue and *j*th eigenvector of the covariance matrix of this class. To illustrate the asymmetric distribution, quasi-mean $\hat{m}_j$, quasi-variance $\left(\hat{\sigma}_j^+\right)^2$ and $\left(\hat{\sigma}_j^-\right)^2$ are required. This asymmetric distribution is described in Figure 2.16. Based on this distribution, AMD is defined as follows.

$$d_{AMD}\left(\mathbf{v}\right)=\sum_{j=1}^{n}\frac{1}{\left(\hat{\sigma}_j\right)^2+b}\left(\mathbf{v}-\hat{\boldsymbol{\mu}},\phi_j\right)^2 \qquad (2.36)$$

where b is a bias, $\hat{\boldsymbol{\mu}}=\sum_{j=1}^{n}\hat{m}_j\phi_j$ and $\hat{\sigma}_j=\begin{cases}\hat{\sigma}_j^+ & if\left(\mathbf{v}-\hat{\boldsymbol{\mu}},\phi_j\right)\geq 0\\ \hat{\sigma}_j^- & otherwise\end{cases}$. Although a high recognition rate is obtained by using AMD, characters with extreme noise or blur characters can be misclassified easily.

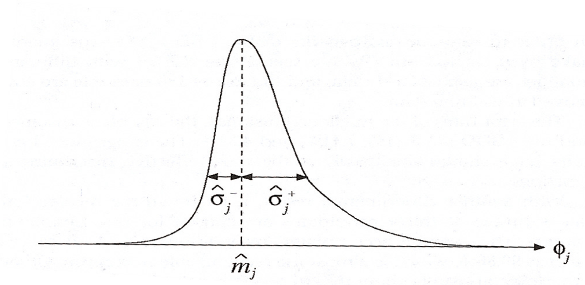


Figure 2.16: The asymmetric distribution on an axis described with quasi-mean and quasi-variance.

2.4.4 Neural Network Classifiers

Since the late 1980's, application of neural network (NN) to recognition of handprinted digits, characters and cursive handwriting have become a very famous area (Garris *et al.*, 1998). Basically, the decision function of NN is based on the weighted sum of its input and the perceptron model for two pattern classes is as follows:

$$d(\mathbf{x})=\sum_{i=1}^{n}w_ix_i+w_{n+1} \qquad (2.37)$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ is the input pattern and the coefficients $w_i, i = 1, 2, \ldots, n, n+1$ are the weights. The decision function of NN classifiers is determined via training. Figure 2.17 shows schematically the perceptron model for two pattern classes. The NN classifiers can also be extended to recognition of multiclass. The most useful characteristics of NN classifier are their ability to learn from examples, their ability to operate in parallel and their ability to perform well using data that is seriously deformed.

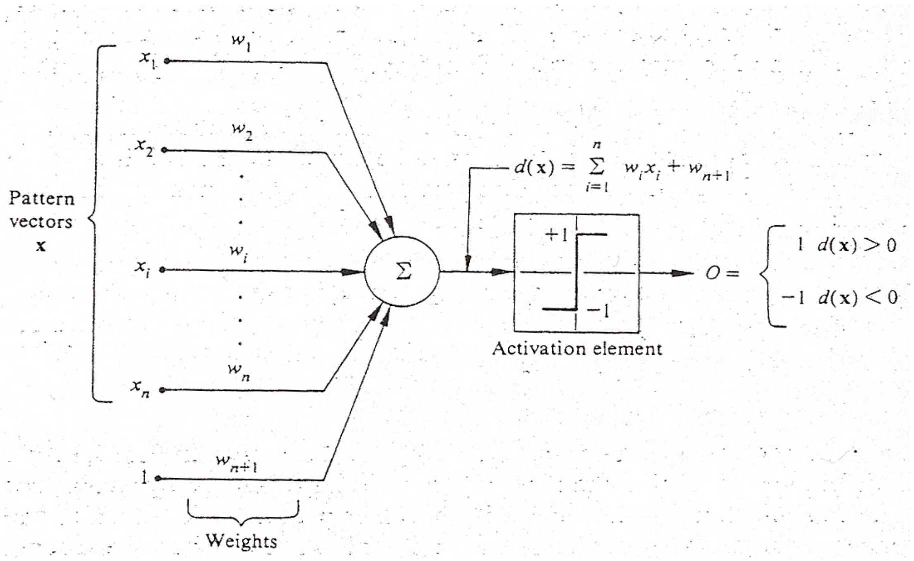

Pattern vectors x
x_1
x_2
x_i
x_n
1
w_1
w_2
w_i
w_n
w_{n+1}
Weights
Σ
$d(\mathbf{x}) = \sum_{i=1}^{n} w_i x_i + w_{n+1}$
+1
−1
Activation element
$O = \begin{cases} 1 & d(\mathbf{x}) > 0 \\ -1 & d(\mathbf{x}) < 0 \end{cases}$

Figure 2.17: The perceptron model for two pattern classes.

I. Nearest-neighborhood based Neural Networks

The two commonly used neural network classifiers based on nearest-neighborhood approach are Learning Vector Quantization (LVQ) and Decision Surface Mapping (DSM). These two neural classification schemes were applied to Chinese character recognition by Romero *et al.* (1995). The difference between them is their training algorithms that are used to determine the prototype. For both LVQ and DSM, the frequency of occurrence of a character has a strong correlation with its recognition rate. However, LVQ and DSM algorithms lack of confidence measure on the classification results, causing the generated prototype in the training process less reliable.

LVQ was developed by Kohonen (1988) and was viewed as a method of vector classification. By using a fixed number of class prototypes, firstly, the nearest prototype to the current exemplar is found. If the prototype and the exemplar are from the same class, move the prototype closer to the exemplar; if they are from different class, move the prototypes away from the exemplar. Direction of the movement is defined by the line joining the two vectors.

Geva and Sitte (1991) proposed DSM algorithm which is a refinement of LVQ algorithm. In the training process, prototypes that are located well in the class are untouched by the DSM algorithm while those located less centrally are moved by both intra-class and extra-class exemplars until they lie right on the class boundary. DSM algorithm allows the creation of new prototypes which will be located at the position of the incorrectly classified exemplar if the ratio of the distance exceeds a threshold.

II. Probabilistic Neural Networks

Probabilistic Neural Networks (PNN) which have the advantages of both neural networks and statistical approaches were applied to HCCR by Romero *et al*. (1997). The PNN implementation attempts to model the actual probability distributions of classes by using mixtures of Gaussians. Different from the original NN, three sets of values are computed in PNN: the mean for the component of each class, the mixing proportions for the individual components and the final within-class covariance matrix. The decision function of classifying exemplar $\mathbf{x}$ to class *j* is defined as follows:

$$d_j(\mathbf{x}) = \sum_{i=1}^{G_j} \pi_{ij} p_{ij}(x) \qquad (2.38)$$

where

$$p_{ij}(x) = (2\pi)^{-N/2} |\boldsymbol{\Sigma}|^{-1/2} \exp\left[-\frac{1}{2} (x-\mu_{ij})^{'} \boldsymbol{\Sigma}^{-1} (x-\mu_{ij}) \right] \qquad (2.39)$$

with μ_{ij} be the *i*th component mean for class *j*, π_{ij} be its mixing proportions, $\boldsymbol{\Sigma}$ be the final within-class covariance matrix and G_j be the number of Gaussian components used to model class *j*. This decision function is implemented under assumption that the costs of choosing any of the characters incorrectly are the same. Same as the LVQ and DSM algorithms, the accuracy of the recognition system is also dependent on the character occurrence rates. For PNN, the computation of confidence measures on the classification results is available, but the number of mixtures of Gaussian distributions is fixed. Hence, it is not very efficient for the representation of Chinese character distributions since Chinese characters pose high stroke complexity.

III. Self-growing Probabilistic Decision-based Neural Network

The Self-growing Probabilistic Decision-based Neural Network (SPDNN) is a probabilistic variant of the original PNN. It was developed by Fu and Xu (1998). Instead of fixed number of mixtures of Gaussian distributions, the discriminant function of a SPDNN uses a flexible number of mixtures of Gaussian distributions for each different character.

Each subnet of an SPDNN is designed to represent one character class. Due to the flexible number of clusters in a subnet of SPDNN, the subnet discriminant functions are designed in a log-likelihood model for different handwritten characters. Given an D-dimensional input character pattern $\mathbf{x} = [x_{1,} x_{2,\ldots,} x_D]^T$, the discriminant function of the multiclass SPDNN models the log-likelihood function as shown below:

$$\begin{aligned} \phi(\mathbf{x}, \mathbf{w}_i) &= \log p(\mathbf{x} \mid w_i) \\ &= \log\left[\sum_{r_i=1}^{R_i} P\left(\Theta_{r_i} \mid w_i\right) p\left(x \mid w_i, \Theta_{r_i}\right)\right] \end{aligned} \qquad (2.40)$$

where $\mathbf{w}_i = \left\{\boldsymbol{\mu}_{r_i}, \boldsymbol{\Sigma}_{r_i}, P\left(\Theta_{r_i} \mid w_i\right), T_i\right\}$, $\boldsymbol{\mu}_{r_i} = \left[\mu_{r_i 1,} \mu_{r_i 2}, \ldots, \mu_{r_i D}\right]^T$ is the mean vector, diagonal matrix $\boldsymbol{\Sigma}_{r_i} = diag[\sigma^2_{r_i 1}, \sigma^2_{r_i 2}, \ldots, \sigma^2_{r_i D}]$ is the covariance matrix, $P\left(\Theta_{r_i} \mid w_i\right)$ denotes the prior probability of the cluster r_i, T_i is the output threshold of the subnet *i*, $p\left(x \mid w_i, \Theta_{r_i}\right)$ represents one of the Gaussian distribution that comprise $p\left(x \mid w_i\right)$ and Θ_{r_i} represents the parameter set $\left\{\boldsymbol{\mu}_{r_i}, \boldsymbol{\Sigma}_{r_i}\right\}$ for a cluster r_i in subnet *I*. By definition, $\sum_{r_i=1}^{R_i} P(\Theta_{r_i} \mid w_i) = 1$, where R_i is the number of clusters in w_i. In this case, the likelihood function $p\left(\mathbf{x} \mid w_i\right)$ for character class w_i is assumed to be a mixture of Gaussian distributions and $p\left(\mathbf{x} \mid w_i, \Theta_{r_i}\right)$ is defined as follows:

$$p\left(\mathbf{x} \mid w_i, \Theta_{r_i}\right) = \frac{1}{(2\pi)^{\frac{D}{2}} \left|\boldsymbol{\Sigma}_{r_i}\right|^{\frac{1}{2}}} \cdot \exp\left[-\frac{1}{2} \frac{\left(\mathbf{x} - \boldsymbol{\mu}_{r_i}\right)^T \left(\mathbf{x} - \boldsymbol{\mu}_{r_i}\right)}{\boldsymbol{\Sigma}_{r_i}}\right] \qquad (2.41)$$

A SPDNN with *K* subnets that are used to represent a *K*-category classification problem as presented in Fu and Xu (1998) and is illustrated in Figure 2.18. Although SPDNN is able to perform well in recognizing distorted handwritten characters, the computation complexity is high when dealing with high-dimensional features and this will decelerate the speed of the recognition system greatly.

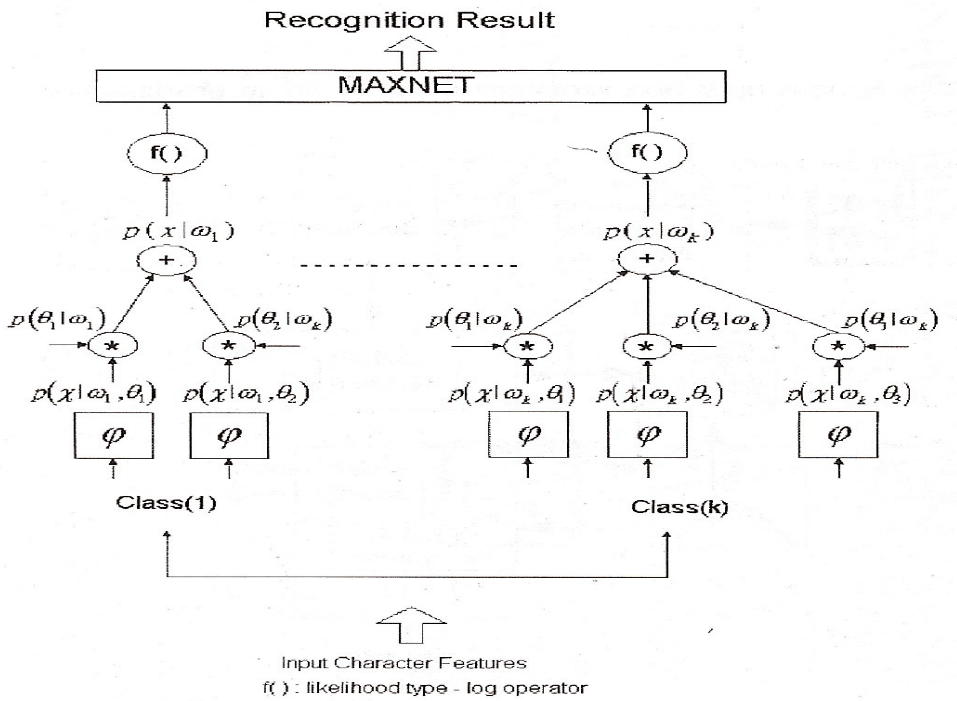


Figure 2.18: Schematic diagram of *K*-class SPDNN character recognizer.

IV. Geometric Approach based Neural Networks

Motivated by new designed neural networks proposed in Zhang and Zhang (1999), Wu *et al.* (2000) built a neural network based on geometrical approach. This network scheme has a special name called CSN network which stands for "covering with sphere neighborhoods". The CSN network is closely related to M-P neuron model which each M-P neuron corresponds to a sphere neighborhood on the sphere surface. Hence, it is necessary to understand the geometrical representation of M-P neuron model briefly in order to implement CSN algorithm. An M-P neuron is an element with n inputs and one output. The general form of its function is shown below:

$$y = \operatorname{sgn}\left(\mathbf{W}^T\mathbf{X} - \psi\right) \qquad (2.42)$$

where $\mathbf{X} = \left(x_1, x_2, \ldots, x_n\right)^T$ is an input vector, $\mathbf{W} = \left(w_1, w_2, \ldots, w_n\right)^T$ is the weight vector, ψ is the threshold and $\operatorname{sgn}(v) = 1$ if $v \geq 0$, or else $\operatorname{sgn}(v) = -1$. Without loss of generality, it can be assumed that all input vectors are restricted to an n-dimensional sphere surface S^n. $\mathbf{W}^T\mathbf{X} - \psi = 0$ can be interpreted as a hyper-plane P, so $\mathbf{W}^T\mathbf{X} - \psi > 0$ represents the intersection between S^n and the positive half-space partitioned by the hyper-plane P. This intersection is called “sphere neighborhood” as shown in Figure 2.19.

The basic idea of CNS algorithm is to transform the design of a neural network classifier to a training sample covering problem. In this case where the input vector is not covered by any sphere neighborhood, Wu *et al.* (2000) defined a membership function $\mu_C(\mathbf{X})$, where $\mathbf{X}$ is an input vector and C is a set of sphere neighborhoods. Wu *et al.* (2000) defined that $\mu_C(\mathbf{X}) = 1$ if $\mathbf{X}$ is covered by any of the sphere neighborhoods in set C, otherwise $\mu_C(\mathbf{X}) = 1/\left(\text{dist}(\mathbf{X}, C) \times M\right)$, where $\text{dist}(\mathbf{X}, C)$ is a distance function between $\mathbf{X}$ and C while M is a positive integer whose value is large enough such that $1/\left(\text{dist}(\mathbf{X}, C) \times M\right)$ is smaller than 1. The whole process of a CSN network with K-class is shown schematically in Figure 2.20. When determining an input vector $\mathbf{X}$'s class, the function of the CNS network is essentially to select the corresponding geometrical area whose value of membership function is maximum among all the regions. In order to make the efficiency of CNS network holds, the construction of sphere neighborhood and definition of the distance function must be determined optimally which these are not easy tasks.

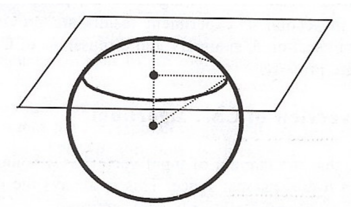


Figure 2.19: A sphere neighborhood.

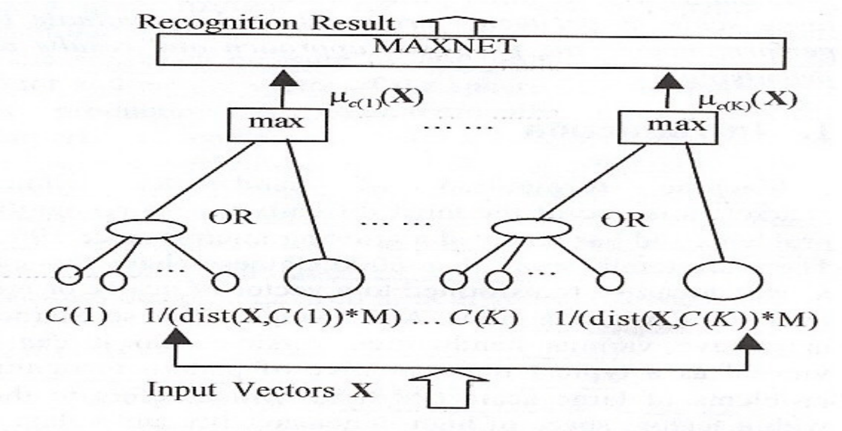

Recognition Result
MAXNET
$\mu_{c(1)}(\mathbf{X})$
max
... ...
$\mu_{c(K)}(\mathbf{X})$
max
OR
... ...
OR
...
...
$C(1)$ 1/(dist($\mathbf{X}$,C(1))*M) ... $C(K)$ 1/(dist($\mathbf{X}$,$C(K)$)*M)
Input Vectors $\mathbf{X}$

Figure 2.20: Schematic diagram of *K*-class CSN network ($C(i), 1 \le i \le K$, the *i*th set of sphere neighborhoods corresponding to the *i*th class).

2.4.5 Other Classifiers

Below are the simple classifiers based on distance measures which are commonly used in character recognition.

I. Minimum Distance Classifier

One way to determine the class membership of an unknown input character $\mathbf{a}$ is to assign it to the character class of its closest prototype. To determine the closeness, Euclidean distance is used:

$$D_i(\mathbf{a}) = \left\|\mathbf{a} - \mathbf{b}_i\right\| \text{ for a class } \mathbf{b}_i,\ i = 1, 2, \ldots, M \qquad (2.44)$$

where $\left\|\mathbf{h}\right\| = \left(\mathbf{h}^T\mathbf{h}\right)^{1/2}$ is the Euclidean norm. Without loss of generality, it is equivalent to evaluating the functions

$$d_i^{MD}(\mathbf{a}) = \mathbf{a}^T\mathbf{b}_i - \frac{1}{2}\mathbf{b}_i{}^T\mathbf{b}_i \qquad (2.45)$$

Equation (2.52) is, therefore, the discriminant function of minimum distance (MD), as mentioned in Gonzalez and Woods (1993). MD classifier was used in the application of Chinese character recognition by Senda *et al.* (1995).

II. City Block Distance with Deviation Classifier

Let D-dimensional input character to be $\mathbf{a} = \left(a_1, a_2, \ldots, a_D\right)^T$ and M character classes in database to be $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_M$, where $\mathbf{b}_i = \left(b_{i1}, b_{i2}, \ldots, b_{iD}\right)^T$ for $i = 1, 2, \ldots, M$. As stated in Kato *et al.* (1999), city block distance with deviation (CBDD) is defined as

$$d_i^{CBDD}(\mathbf{a}) = \sum_{j=1}^{D} \max\left\{0, \left|a_j - b_{ij}\right| - \theta\square s_{ij}\right\} \text{ for a class } \mathbf{b}_i \qquad (2.43)$$

where s_j denotes the standard deviation of *j*th element, and θ is a constant. The most important property of Equation (2.43) is that the variations of handwritten characters are being taken account in the city block distance measure.

III. Dynamic Time Warping Classifier

Dynamic Time Warping (DTW) (Kruskall and Liberman, 1983; Eamonn and Michael, 2001) is an algorithm for measuring similarity between two sequences which may vary in time. It is known to be useful in online handwriting recognition as proposed in Sridhar *et al.* (1999), Niels and Vuurpijl (2005). Suppose that $S = (s_1, s_2, s_3, \ldots, s_n)$ and $R^l = (r_1^{\,l}, r_2^{\,l}, r_3^{\,l}, \ldots, r_m^{\,l})$ are time series of input character and of character from class l in database respectively, where $l = 1, 2, \ldots, k$ and k is the number of character classes in database. DTW measures the similarity of these two time series (S, R^l) in term of the distance between S and R^l after they have been warped together. The value of DTW distance for each class is determined by minimizing a cumulative cost which is defined by the Euclidean distances between all matches $(s_i, r_j^{\,l})$. The character (or class) in database which gives the minimum DTW distance is the recognized character. Due to the quadratic time and space complexity $O(nm)$, standard DTW faces computing time and memory space problems, which limit its practical use especially in online handwriting recognition. To solve this problem, there are some approaches to speed up DTW as described in Vuori *et al.* (2001), Bashir and Kempf (2008).

2.5 Summary

In the whole, chapter 2 reviews the past researches on HCCR that have been done over the years. This chapter is separated into two main parts, which are (i) historical background of feature extraction methods and (ii) historical background of classifiers.

The former part describes the classical and modern feature extraction techniques that have been proposed since last few decades. Basically, the feature extraction methods can be partitioned into three different approaches: (i) structural approaches, (ii) statistical approaches and (iii) statistical-structural approaches. Structural approaches focus on the structural information of Chinese characters. It is stroke number and stroke order independent but it faces computational problem due to the large sets of Chinese characters and complexity of the graph-matching algorithms; statistical approaches which make use of the feature vector have been proved to be computational efficiency, however, some of them fail to tolerate well with character deformation problem due to their dependency on the information of writing direction; Hidden Markov Model (HMM) is the representative feature extraction method for statistical-structural approaches. It is the most efficient way for temporal modeling but the learning process has caused large time and storage space consumption. For each approach, some examples of the popular feature extraction methods are selected and illustrated in detail. They are summarized in Table 2.1.

The latter part presents the previously proposed classifiers which are based on holistic approaches. In this thesis, the classifiers are categorized into four groups: (i) quadratic discriminant function (QDF), (ii) support vector machine (SVM), (iii) Mahalanobis distance (MD) and (iv) neural network (NN) classifiers. QDF is the most famously used classification method in HCCR due its simplicity and robustness, but they fail to discriminate similar characters. Whereas, SVM, MD and NN classifiers are appropriate in differentiating the similar characters. Despite of high accuracy rate, all these classifiers cannot be applied to the hand-held devices directly due to the parameter complexity problem. Hence, simple classifiers such as minimum distance (MD) classifier and city block distance with deviation (CBDD) classifier are sometimes more preferable for the memory limited devices. Furthermore, dynamic time warping (DTW) classifier which is based on similarity match of time series is also largely used specifically for online handwriting recognition. For brief understanding, the readers can refer to the summary of these classifiers in Table 2.2.

Table 2.1: Summary of some feature extraction methods.

Year	Authors	Feature Extraction Methods	Property	Testing Samples	Recognition Rate (%)
Structural Approaches					
1996	Liu *et al.*	ARG	Represent the complex structure of a Chinese character by ARG, such that the nodes of the ARG describe the strokes of the character and the arcs describe the relations between any two different strokes.	320 Chinese characters collected from 8 writers	98.90 (for correct stroke number), 94.20 (for stroke number variations)
1997	Zheng *et al.*	FARG	Is an improved version of traditional ARG which uses fuzzy set to describe the attribute set of node or arc.	3755 Chinese characters collected from 6 writers	98.80
2006	Zeng *et al.*	Delaunay Triangulation	Associates a unique topological structure with handwritten shape using the Delaunay triangulation. It has strong geometrical information also, and covers both the local and global features.	5073 character samples	92.06
Statistical-Structural Approaches					
1997	Takahashi *et al.*	Whole Character-based HMM	Construct a discrete left-right HMM (LRHMM) model for each character, with the set of parameters: state transition probability, output emission probabilities, initial state probability and the number of states. These parameters are estimated by the learning process.	881 Kanji characters collected form 5 writers	90.00
1999	Zheng *et al.*	PCHMM	Is an improved version of the conventional LRHMM that solves the stroke order variation problem by controlling the state transition path directly with a Path Controlling Function $R(Q)$ on the state sequence space.	3755 Chinese characters collected from 7 writers	95.52

2001	Nakai *et. al.*	Substroke-based HMM	Is the extended version of the Whole Character-based Hidden Markov Model (HMM), which reduces the amount of training samples, and memory requirement for database and model for each character. Besides, the adaption to writer can also be implemented easily with a few sample characters provided by user. Furthermore, the recognition speed can be improved by using efficient substroke network search.	1016 Kanji characters collected from 49 writers	95.34
Statistical Approaches					
1992	Kawamura *et al.*	DFD	Extract feature from the online trajectory which is independent of both stroke number and stroke order, but depends only on information of the writing direction (4 directional line elements are defined: vertical, horizontal and diagonal and anti-diagonal). It uses the concept of 'density' to describe how much of each directional feature a character pattern has in each segmented area of the character image.	2965 Kanji characters collected from 10 writers.	91.78
1999	Kato *et. al.*	DEF	Extract feature from the bitmap using an offline approach. It includes three steps: contour extraction, dot orientation and vector construction.	2965 Kanji characters collected from 20 writers.	99.42
2005	Bai and Huo	8-Directional Feature	Is an extended version of 4-directional feature. The direction vector of a stroke point is projected onto two 4 directional axes, resulting in 8 directional feature. It is followed by locating spatial sampling points and extracting blurred directional feature by using Gaussian filter.	3755 Chinese characters collected from 100 writers	84.57

Remark: Kanji characters in Japanese are the same as Chinese characters.

Table 2.2: Summary of some classifiers.

Year	Authors	Classifiers	Property	Data Set	Recognition Rate (%)
Quadratic Discriminant Function (QDF)					
1987	Kimura *et al.*	MQDF	Improves recognition accuracy and reduces the complexity of QDF by replacing the minor eigenvalues of covariance matrix of each class with a constant.	*HCL2000	97.97
2000	Kawatani	RDA	Reduces the influence of parameter estimation errors in QDF through normalizing the determinants of the covariance matrices, which this is done by the combination of each class covariance matrix with the pooled matrix and by biasing eigenvalues.	2,965 Kanji characters, with 570 handwritten samples	98.08
2006	Liu	DLQDF	Is updated version of the MQDF which overcomes the non-Gaussianity of probability densities problem and optimizes the parameters of MQDF through optimizing the minimum classification error (MCE).	**ETL9B, ***CASIA	99.39, 98.43
2008	Long and Jin	Compact MQDF	Compresses the storage of the original MQDF classifier by using a method which combines the linear discriminant analysis (LDA) and subspace distribution sharing.	*HCL2000	97.74
Support Vector Machine (SVM)					
2002	Gao *et al.*	Polynomial-based SVM	Is helpful in discriminating similar characters by transforming the sample pattern into a higher dimensional space where a hyperplane can be used to do the separation. The transformation is performed by using polynomial kernel.	*HCL2000	96.55
2002	Gao *et al.*	RBF-based SVM	The transformation is performed by using Gaussian radical basis function (RBF) kernel. The kernel width is estimated from the variance of the sample vectors.	*HCL2000	96.60

2002	Gao *et al.*	Sigmoid-based SVM	The transformation is performed by using sigmoid kernel. It is equivalent to two-layer perceptron neural network.	*HCL2000	96.85
Mahalanobis Distance (MD)					
1997	Suzuki *et al.*	CMF	Improves the discriminant performance of the ordinary Mahalonabis Distance (MD), by projecting the difference of the class-mean feature vectors of two similar classes onto discriminant subspace that the difference between two similar classes is obviously appears.	**ETL9B	93.72
1999	Kato *et al.*	AMD	Is a new probability density function that can be used to describe pattern classes of asymmetric distribution.	**ETL9B	99.42
2007	Hirayama *et al.*	DS-CMF	Expands the original CMF using Difference Subspace (difference of two subspaces) and it treats difference between two similar classes as a multi-dimensional projective space.	**ETL9B	94.01
Neural Network (NN) Classifiers					
1995	Romero *et al.*	LVQ	Can be viewed as a method of vector classification and it adjusts the prototypes to approximate the density of exemplar in each class.	17,588 printed Chinese characters with different fonts	98.41
1995	Romero *et al.*	DSM	Is a refinement of LVQ algorithm and fewer classification errors will result if the prototypes are concentrated close to the class boundaries.	17,588 printed Chinese characters with different fonts	98.78
1997	Romero *et al.*	PNN	Models the actual probability distributions of classes using a fixed number of Gaussian components. The computation of confidence measures on the classification results is available.	13,984 printed Chinese characters with different fonts	97.43
1998	Fu and Xu	SPDNN	Is a probabilistic variant of the original PNN, which is in a form of a flexible number of mixtures of Gaussian distributions for each different character.	****CCL/ HCCR1	90.12

2000	Wu *et al.*	CSN	Is based on geometrical approach, which transforms the design of a neural network classifier to a training sample covering problem.	700 handwritten Chinese characters	95
Other Classifiers					
1995	Senda *et al.*	MD	Determine the class membership of an unknown input character by assigning it to the character class of its closest prototype. To determine the closeness, Euclidean distance is used.	**ETL9B	79.1
1999	Kato *et al.*	CBDD	The variations of handwritten characters are being taken account in the city block distance measure.	**ETL9B	99.42
2008	Bashir and Kempf	Reduced DTW	Is an algorithm for measuring similarity between two sequences based on time series.	11990 handwritten character samples	99.22

Remark: Kanji characters in Japanese are the same as Chinese characters.
* HCL2000 database is collected by Beijing University of Posts and Telecommunications for China 863 project, which includes 3,755 frequently used Chinese characters in GB2312-80 level 1 character set, 1000 samples per class.
** ETL9B database is collected by the Electro-Technical Laboratory of Japan, which contains handwritten samples of 2,965 Kanji characters, 200 samples per class.
*** CASIA database is collected by the Institute of Automation, Chinese Academy of Sciences, which contains handwritten samples of 3,755 Chinese characters in GB2312-80 level 1 character set, 300 samples per class.
**** CCL/HCCR1 database is collected from 2600 people, including junior high school and college students as well as employees of ERSO/ITRI, which contains more than 200 samples of 5401 frequently used Chinese characters.

CHAPTER 3

TRAJECTORY-BASED *X-Y* GRAPHS DECOMPOSITION

The objective of feature extraction is to characterize the object and then reduce the pattern space to a size appropriate for the application of pattern classification methods. In general, the important steps involved in feature extraction process are to select discriminatory features and extract these features. In this process, only the significant features necessary for the recognition process are retained such that classification can be implemented on a hugely reduced feature space. However, defining a feature vector for Chinese character recognition is not an easy task because Chinese character set is vast in size, complex in structure and contains many similar characters. Furthermore, for handwritten Chinese character recognition, it becomes more difficult due to variability of writing styles. The quality of the features has a great effect upon the performance of a Chinese character recognition system. Hence, one of the essential criteria of a good recognition system is that optimum features be selected.

In this research, building an improved handwritten Chinese character recognition (HCCR) system with more efficient overall performance by using new feature extraction and classification method is the main motivation. Different from the commonly used or previously proposed recognition systems which involve three main stages (refer to Section 1.1), the new HCCR system developed in this research only consists of two main stages, which are feature extraction and classification. This is the greatest advantage of the new designed HCCR system, which is omitting preprocessing process while preserving high accuracy rate at the same time. Definitely, it is a breakthrough in the area of character recognition.

As mentioned in Section 1.1, preprocessing is an unavoidable process for the character recognition system, so as to diminish the negative effects caused by individual writing variations and distortions, as well as inaccuracy of digitization. Hence, preprocessing process is very vital in converting the input character into a more proper pattern representation in order to achieve a high accuracy rate. For HCCR system based on online approach, normalization is the major process involved in preprocessing. The performance of HCCR system is largely dependent on character shape normalization, which aims to regulate the size, position and the shape of character patterns, so that the shape variation between the patterns of the same character can be reduced. In Srihari *et al.* (2007), it states that linear normalization (LN) which can scale the image in spite of its within structure and the nonlinear normalization (NLN) method which is based on line density equalization are popularly used now. However, some of these normalization methods are very complex in computation and thus, the speed and efficiency of the HCCR system will be degraded.

In this research, the evidence of how the new HCCR system works without undergoing normalization and attains an even improved performance simultaneously by using the new proposed feature extraction method and new designed classifier will be shown in Chapter 5. For comparison purpose, both recognition system with and without preprocessing stage will be implemented in this research. The whole process of the recognition systems with preprocessing is depicted diagrammatically in Figure 3.1.

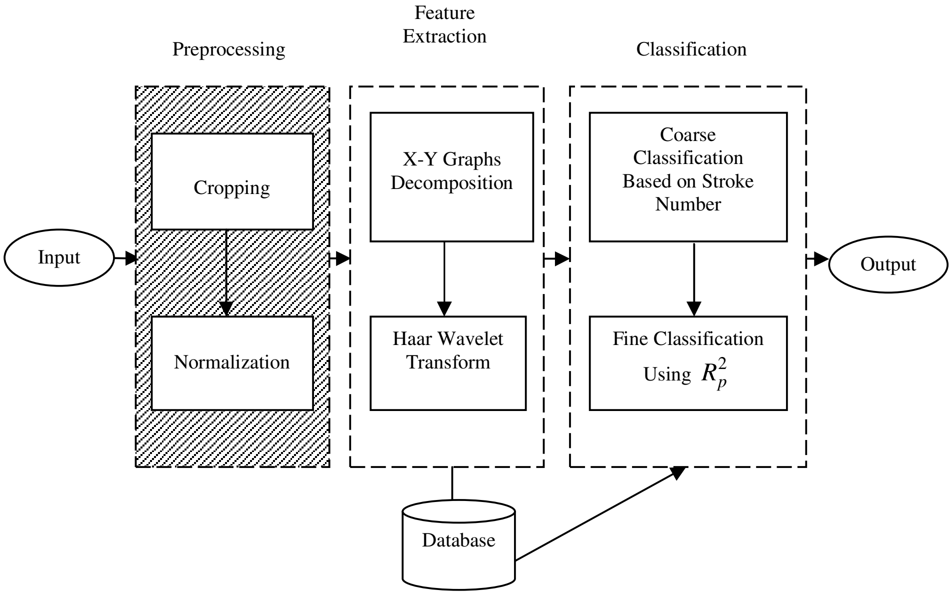


Figure 3.1: Diagram of the whole process of the recognition system with preprocessing stage.

In this chapter, a new feature extraction technique will be presented. Section 3.1 illustrates the process of preprocessing implemented in the new HCCR system. Section 3.2 explains the new feature extraction schemes which are separated into two stages: (i) trajectory-based *X*-*Y* graphs decomposition (T*XY*GD) and (ii) Haar wavelet transform. These two stages are described in Section 3.2.1 and 3.2.2 respectively. Finally, this chapter is ended with a summary. The classification process will be discussed in Chapter 4.

3.1 Preprocessing Process

The practical process of preprocessing may vary in detail for different recognition systems. In this section, the procedure of the preprocessing process applied in the new HCCR system of this research is described.

Firstly, for online handwriting recognition, the trajectory of the character is captured. The input from the digitizer corresponding to handwritten Chinese character is a sequence of points in the form of x and y coordinates, $[x_t, y_t]$ with embedded pen-up and pen-down events when multiple strokes are involved. The Wacom Intuos®3 pen tablet is used as the digitizer in this research. For each character, the strokes are resampled by using linear equi-distant resampling technique and 128 points are used to represent each stroke. Thus, a w-strokes character, for example, will have a total of $128 \times w$ points. The preprocessing stage includes two parts: (i) cropping and (ii) normalization and these are illustrated in the following.

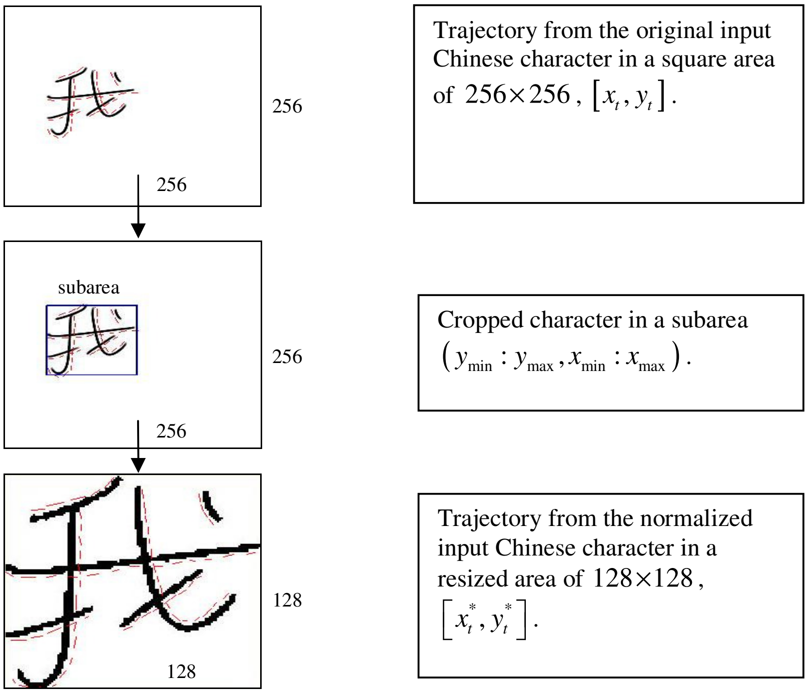
(i) **Cropping**: Given the sequence of points for an input character, the maximum x and y coordinates, and also the minimum of them are determined. Then, a particular part of the original area of 256×256, that is the subarea from row $y_{\min}$ to row $y_{\max}$ and column $x_{\min}$ to column $x_{\max}$ $(y_{\min} : y_{\max}, x_{\min} : x_{\max})$ is cropped.

(ii) **Normalization**: The sequence of points $[x_t, y_t]$, which range within the cropped subarea are normalized to the size of 128×128, as shown below.

$$x_t^* = 127\left(\frac{x_t - x_{\min}}{x_{\max} - x_{\min}}\right) + 1 \qquad (3.1)$$

$$y_t^* = 127\left(\frac{y_t - y_{\min}}{y_{\max} - y_{\min}}\right) + 1 \qquad (3.2)$$

The whole procedure of preprocessing is presented in Figure 3.2 and the effect of normalization is shown in Figure 3.3. The left side of Figure 3.3(a)-(d) displays the characters before normalization whereas the right side is the characters after undergoing normalization. Notice that the size and the position of the normalized characters are all standardized if compared to the non-normalized characters with various size and position which will affect the accuracy rate severely. Therefore, all the researchers believe that preprocessing process must be executed in every recognition system as to obtain a promising accuracy rate.



Trajectory from the original input Chinese character in a square area of 256×256, $[x_t, y_t]$.

Cropped character in a subarea $(y_{\min} : y_{\max}, x_{\min} : x_{\max})$.

Trajectory from the normalized input Chinese character in a resized area of 128×128, $[x_t^*, y_t^*]$.

Figure 3.2: Diagram of the whole preprocessing procedure for the Chinese character '我' (means I or me).

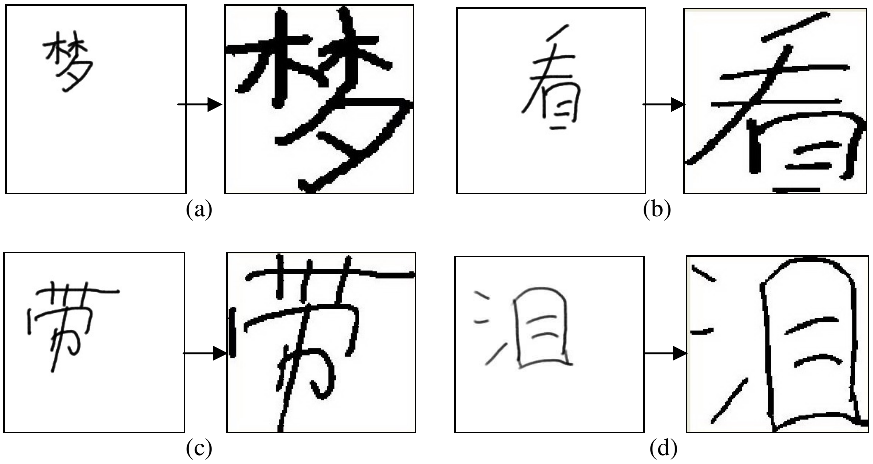


Figure 3.3: Examples of non-normalized (left) and normalized (right) Chinese character (a) '梦' (means dream), (b) '看' (means see or look), (c) '带' (means bring) and (d) '泪' (means tear).

3.2 Feature Extraction

In this research, the new feature extraction method of the online recognition system consists of two main steps: (i) trajectory-based *X*-*Y* graphs decomposition (T*XY*GD) and (ii) Haar wavelet tranform. The detail of each step will be explained in the sub-sections below.

3.2.1 Trajectory-Based *X*-*Y* Graphs Decomposition

Trajectory-based *X*-*Y* graphs decomposition (T*XY*GD) is used for feature extraction in online recognition system. For short, it is also named *X*-*Y* graphs decomposition. It is considered as a holistic approach which extracts the character as a whole without any preliminary identification of strokes. The principle of *X*-*Y* graphs decomposition is to trace the pattern of each Chinese character based on the trajectory of handwriting. Here, the trajectory of handwriting is referred to the way of writing a Chinese character based on certain stroke order. According to the points from this trajectory, *X*-graph and *Y*-graph are formed, such that these two graphs used to represent each Chinese character. The details of how the *X*-*Y* graphs decomposition works are illustrated below.

In the *X*-*Y* graphs decomposition, the sequence of normalized points $\left[x_t^*, y_t^*\right]$ (or non-normalized points $\left[x_t, y_t\right]$), where $1 \le t \le N = 128 \times w$ is transformed into two separated graph: (i) graph of *x*-coordinate versus time sequence (called *X*-graph) and (ii) graph of *y*-coordinate versus time sequence (called *Y*-graph). These two graphs are described in Figure 3.4. Notice that the pattern of the graphs depends on how the character is written. As an example, the values in the *X*-graph rise while in *Y*-graph the values remain unchanged, when the second stroke of character '我', i.e. the horizontal stroke is written. For more detail information, the *X*-*Y* graphs for some of the strokes exist in Chinese characters are shown in Appendix A1. Consequently, the feature vectors are constructed from the sequences of points in *X*-graph and *Y*-graph as $\left\{\left[x_1^*,\ldots,x_N^*\right]^T, \left[y_1^*,\ldots,y_N^*\right]^T\right\}$. The size of the feature vectors is varying with the stroke number of the character. The smaller dimensionality of feature vectors is obtained for character with lesser stroke number and vice versa. This specialty is very helpful for rough classification as the characters can be partitioned into smaller groups based on the stroke number (or the feature size) of the character before fine classification. Therefore, instead of the whole database, the fine classification is only dealing with a small subset of the database and this indeed will boost up the speed of the recognition system greatly.

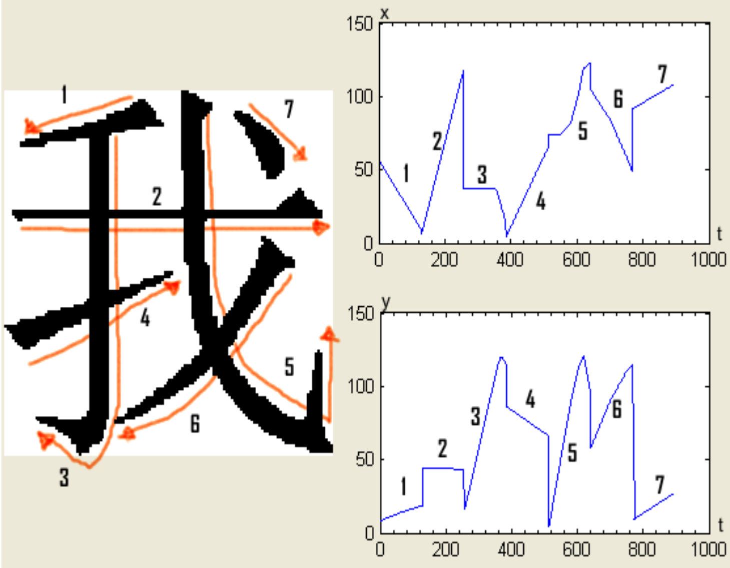
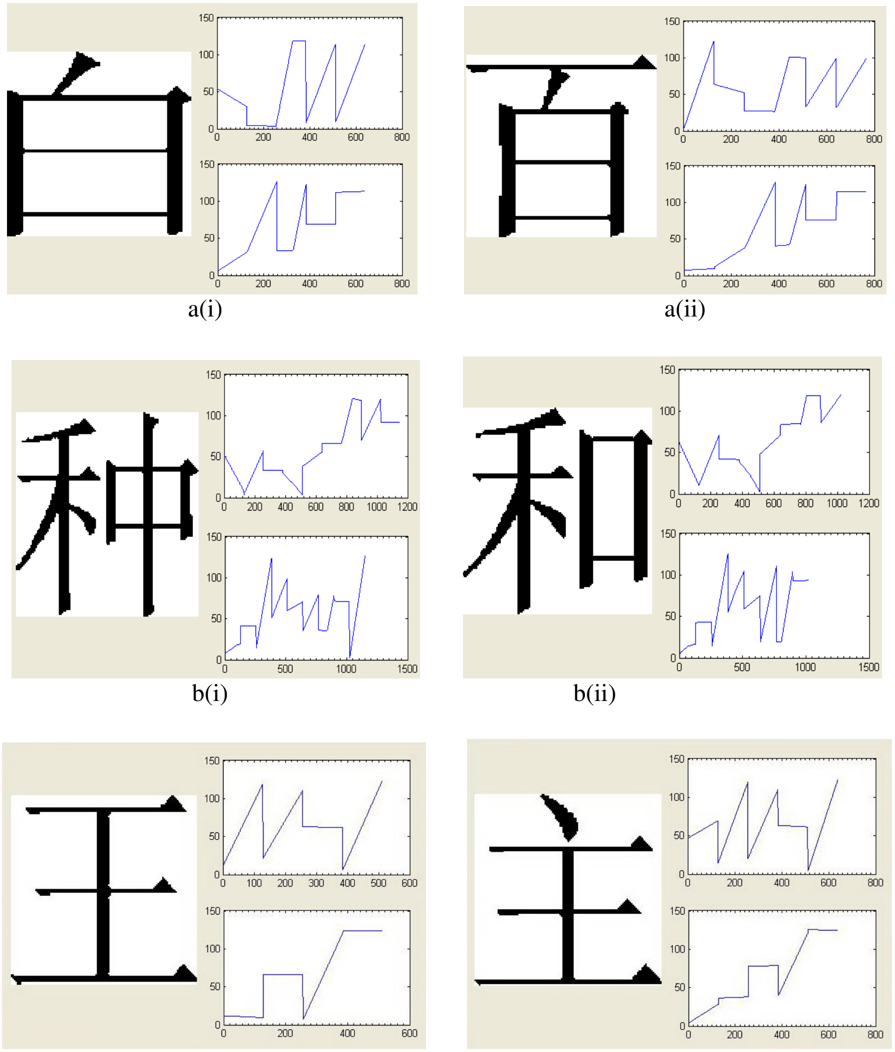


Figure 3.4: ***X*-graph (above) and *Y*-graph (below) of Chinese character '我' (means I or me).**

Rich handwriting styles and lack of consistency between temporal and spatial fields make the character shapes differ in many ways. Thus, instead of constructing the features based on the shape of the characters as in Zeng *et al.* (2006) and Shu (1996), it is more preferable to extract the features from the trajectory of the handwritten characters. As a trajectory context descriptor, *X*-*Y* graphs represent geometrical information and topological structure of a Chinese character. Each point on the *X*-*Y* graphs identifies the geometric characteristic of the strokes written and the sequence of the points represent the whole topological structure. Moreover, there are two types of features, which are local features and global features. Local features include horizontal change Δx, vertical change Δy and writing angle change $\Delta\theta$ with respect to the neighboring points; global features include approximate curvature, aspect ratio and linearity. *X*-*Y* graphs retain both the local and global features without explicit calculation. Furthermore, this new approach is designed in accordance with the following properties:

I. Uniqueness

The trajectory of each Chinese character is unique and in turn forms the unique *X*-graph and *Y*-graph. Even for similarly shaped characters, the graphs plotted are also of different shapes. Some examples are illustrated in Figure 3.5. Hence, both the *X*-graph and *Y*-graph can be claimed to contain the most essential information of Chinese characters and have strong discriminative ability.



a(i) a(ii)

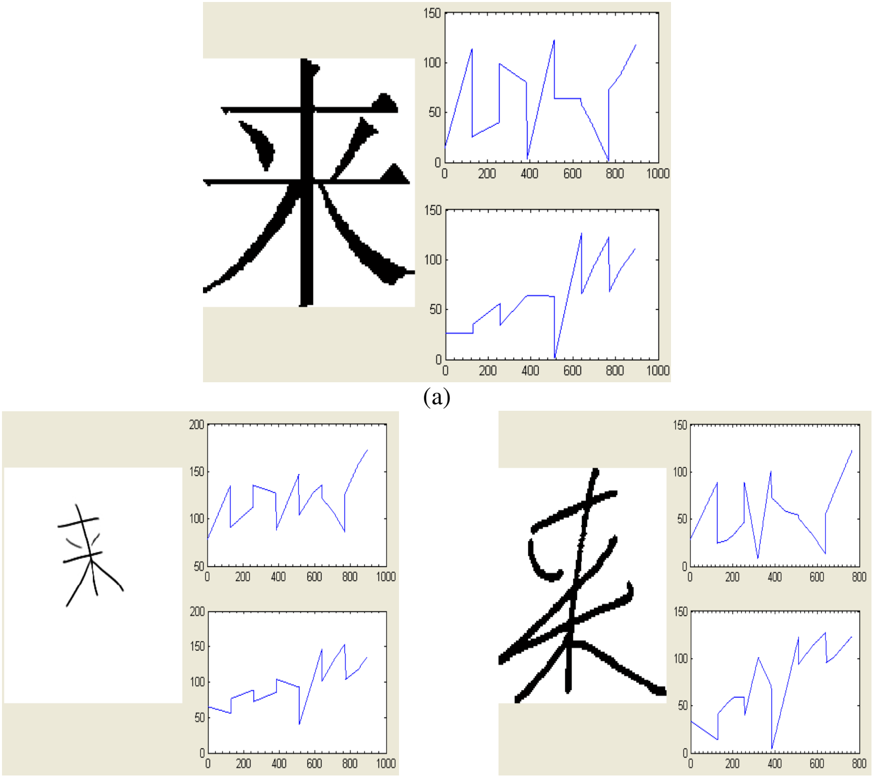
b(i) b(ii)

c(i) c(ii)

Figure 3.5: *X*-graphs (above) and *Y*-graphs (below) plotted for the similar characters: a(i) ‘白’ (means white) and a(ii) ‘百’ (means hundred); b(i) ‘种’ (means plant) and b(ii) ‘和’ (means and); c(i) ‘王’ (means king) and c(ii) ‘主’ (means host or owner).

II. Invariant of different writing styles

Different writing styles are the main problem faced in handwriting recognition. Remark that here, different writing styles refer to writing the characters in a way such that they are different in size and position, with deformed shape and slant. To address this problem, *X*-*Y* graphs decomposition is a solution since *X*-graph and *Y*-graph plotted for the Chinese characters are invariant of size and position of the written character. In other words, the pattern of both *X*-graph and *Y*-graph will generally be retained for the same characters despite of various character size and position. Besides, the preservation of graph patterns for deformed characters can also helps in improving the recognition rate. The examples of these are presented in Figure 3.6.



(a)

(b)

(c)

Figure 3.6: *X*-graphs (above) and *Y*-graphs (below) plotted for character '来 (means come). (a) Regular character in database (b) Characters written in various size and position (c) Deformed character written with connected strokes.

III. Simplicity

In *X-Y* graphs decomposition, neither complicated process nor heavy computation is involved. Decomposing the character coordinates into two separated *X-Y* graphs and obtaining the feature vectors from the corresponding graphs are the only tasks that required to be implemented. As a result, the simplicity of this approach will definitely enhance the efficiency and speed of the recognition system.

3.2.2 Haar Wavelet Transform

Many applied fields are making use of wavelets and it has become part of the toolbox of statisticians, signal and image processors, medical persons, geophysicists and so on. Besides that, wavelet based approaches become increasingly popular in pattern recognition and have been applied to character recognition in recent year (Shioyama *et al*., 1998; Huang and Huang, 2001; Zhang *et al*., 2006). Most of these wavelet approaches are applied to the character images. They utilize the theory of multiresolution analysis (MRA) to interpret the image at different resolutions and construct the feature vectors accordingly. In this research, the application of wavelet transform on the *X*-graph and *Y*-graph which is a new attempt of wavelet approach is proposed.

Among different wavelet families, Haar wavelet is the most fundamental and widely used due to its algorithm simplicity and efficiency. Besides, in each level of Haar transform, the sequence of points will be reduced to half of its size. This will lead to an easier computation than other wavelet transforms. Thus, Haar wavelet is chosen for feature size reduction, though other wavelets can also be applied in similar way.

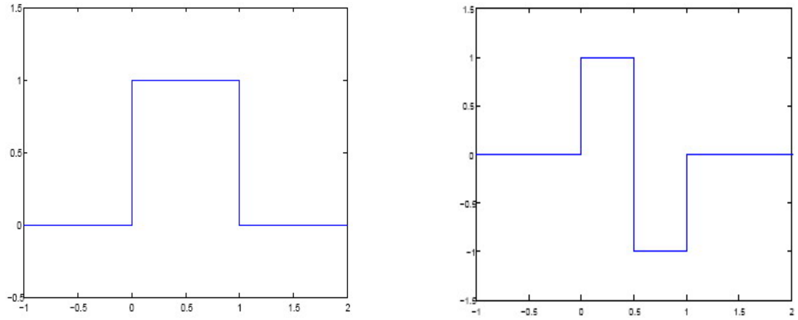
For Haar wavelet, let the father wavelet, or scaling function be

$$\varphi(t) = \begin{cases} 1 & for\ 0 < t < 1 \\ 0 & elsewhere \end{cases} \tag{3.3}$$

Then $V^0 = span\left(\{\varphi(t-k)\}_{k\in Z}\right)$ consists of piecewise constant functions with jumps only at integers. Likewise the subspace $V^j = span\left(\left\{\varphi_{j,k}\right\}_{k\in Z}\right)$ are piecewise constant functions with jumps only at the integer multiples of 2^{-j}. On the other hand, the mother wavelet is defined as

$$\psi(t) = \begin{cases} 1\ for\ 0 < t < \frac{1}{2} \\ -1\ \ for\ \frac{1}{2} \le t < 1 \\ 0\ \ elsewhere \end{cases} \tag{3.4}$$

The subspace $W^0 = span\left(\{\psi(t-k)\}_{k\in Z}\right)$ are piecewise constant functions with jumps only at half-integers, and average 0 between integers. Likewise the subspaces $W^j = span\left(\left\{\psi_{j,k}\right\}_{k\in Z}\right)$ are piecewise constant functions with jumps only at the integer multiples of $2^{-(j+1)}$, and average 0 between the integers multiples of 2^{-j}. Both the father wavelet and mother wavelet for Haar are described in Figure 3.7.



(a) (b)

Figure 3.7: Plot of (a) father wavelet $\varphi(t)$, and (b) mother wavelet $\psi(t)$ for Haar.

In practical, Haar wavelet transform is utilized as DWT and applied to discrete signal which is expressed in the form

$$\mathbf{f} = \left(f_1, f_2, \ldots, f_N \right) \tag{3.5}$$

where N is a positive even integer, representing the length of $\mathbf{f}$ with $f_1 = g(t_1), f_2 = g(t_2), \ldots, f_N = g(t_N)$. Haar transform decomposes a discrete signal $\mathbf{f}$ into two subsignal of half its length. One subsignal is trend or approximation coefficient and the other subsignal is fluctuation or detailed coefficient. The approximation coefficient and detailed coefficient for Haar are defined in the following.

$$a_m = \frac{f_{2m-1} + f_{2m}}{\sqrt{2}} \tag{3.6}$$

$$d_m = \frac{f_{2m-1} - f_{2m}}{\sqrt{2}} \tag{3.7}$$

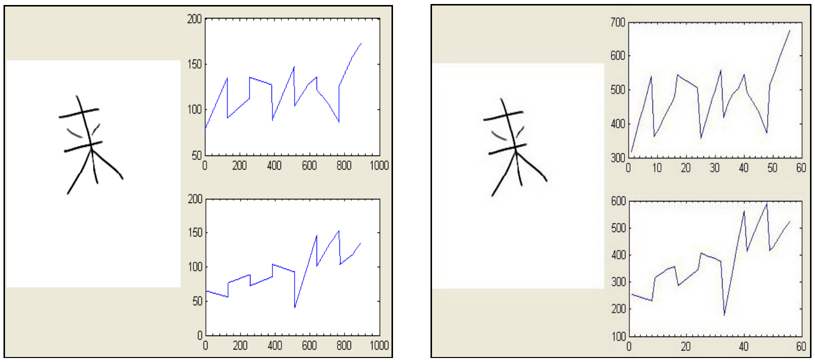
for $m = 1, 2, 3, \ldots, \dfrac{N}{2}$.

In this research, a new attempt of wavelet transform which is applied on graph, instead of image is presented. Haar wavelet reduces the size of the feature vector obtained from *X*-graph and *Y*-graph by converting the input feature vector into new sequence of points $\mathbf{a}_j = \left[a_{xj}, a_{yj} \right]$ and $\mathbf{d}_j = \left[d_{xj}, d_{yj} \right]$, $1 \le j \le D$, $2^5 \le D < 2^6$, which are the approximation and detailed coefficient respectively. Both coefficients are defined below.

$$a_{xj} = \frac{x^*_{2j-1} + x^*_{2j}}{\sqrt{2}}, \; a_{yj} = \frac{y^*_{2j-1} + y^*_{2j}}{\sqrt{2}} \tag{3.8}$$

$$d_{xj} = \frac{x^*_{2j-1} - x^*_{2j}}{\sqrt{2}}, \; d_{yj} = \frac{y^*_{2j-1} - y^*_{2j}}{\sqrt{2}} \tag{3.9}$$

Remark that Haar Transform is performed in several stages or levels until the size of the extracted feature is in the range of $\left[2^5, 2^6 \right)$. The constraint with the range of the reduced features dimensionality is set to be $2^5 \le D < 2^6$, but not $2^4 \le D < 2^5$, $2^6 \le D < 2^7$ and so on because $2^5 \le D < 2^6$ gives the best tradeoff between recognition rate and dimensionality of features from observation. In the proposed recognition system, only approximation coefficient $\mathbf{a}_j$ is considered as new extracted feature used for classification. Whereas, $\mathbf{d}_j$ is assumed as the error or variation between input character and character in database. Similarly, $\mathbf{b}_j = \left[b_{xj}, b_{yj} \right]$ and $\mathbf{e}_j = \left[e_{xj}, e_{yj} \right]$ represent the approximation and detailed coefficient of the character in database. It has been proved that the extracted features still retain the important information of the characters after size reduction by Haar wavelet transform. The example is demonstrated in Figure 3.8. Notice that the graph patterns are still preserved after undergoing Haar wavelet transform.



(a) (b)

Figure 3.8: *X*-graphs (above) and *Y*-graphs (below) plotted for character ‘来’ (means come). Left shows the graphs before Haar wavelet transform while right shows th e graphs after Haar wavelet transform.

3.3 Summary

This chapter mainly discusses about new feature extraction method and new approach of wavelet transform proposed for feature vector acquisition in this research. It begins with the overview of new developed HCCR system, such that the preprocessing process can be omitted. Trajectory-based *X*-*Y* graphs decomposition (T*XY*GD) or for short, *X*-*Y* graphs decomposition is a new idea for feature extraction in online recognition system, with the principle of tracing the pattern of each Chinese character based on the trajectory of handwriting. As a trajectory context descriptor, *X*-*Y* graphs represent geometrical information and topological structure of a Chinese character. Besides, they also cover both local and global features of the character. The uniqueness, invariant of different writing styles and simplicity property of *X*-*Y* graphs has strengthened the discrimination power and accelerated the speed of the recognition system. For feature size reduction, new attempt of Haar wavelet transform is presented. It is applied on the graphs instead of images and it preserves the significant information of the character after size reduction. Therefore, *X*-*Y* graphs decomposition with Haar wavelet transform can be claimed as an efficient method for feature extraction in HCCR and this can be validated from the experimental result in Chapter 5.

CHAPTER 4

TWO-DIMENSIONAL FUNCTIONAL CLASSIFIER

The feature vectors obtained from *X*-*Y* graphs decomposition with Haar wavelet transform will proceed to classification stage for matching. Compared to numerals and alphabets, the number of Chinese character set is extremely large. Hence, in order to speed up the recognition system, the classification process is separated into two stages: coarse classification and fine classification. In the proposed HCCR system, the coarse classification is based on stroke number; whereas the coefficient of determination (COD) for two-dimensional unreplicated linear functional relationship (2D-ULFR) model will be the new designed fine classifier or similarity measure.

This chapter will discuss about the whole classification procedure of the new HCCR system in detail. Section 4.1 describes the coarse classification while fine classification will be illustrated in Section 4.2. Besides, some important properties of the new proposed classifier are also included. Section 4.3 validates the normality assumption of 2D-ULFR model and lastly, a summary of this chapter is presented in Section 4.4.

4.1 Coarse Classification

In the proposed HCCR system, every different single Chinese character is assumed to be a class in the database and each class consists of only one sample. Firstly, in coarse classification, the Chinese characters of the same number of stroke as that of input character are selected from the large amount of classes in database to be the candidates for fine classification. The number of stroke, *w*, is determined as follows.

$$w = \frac{N}{128} \qquad (4.1)$$

where *N* is the length of feature vector before Haar transform (refer to Section 3.1). Example of the candidate lists chosen for some input characters by using this coarse classification method is demonstrated in Table 4.1. Compared to implementing fine classification on the whole database, dealing with only the subset of database chosen by coarse classification can save the processing time to a great extent. Consequently, these selected candidates are further classified by fine classifiers as described in the next section.

Table 4.1: Example of the candidate lists for three input Chinese characters with different number of stroke.

Input Characters	**Candidate Lists**
了 (2 Strokes)	了 人 力 十 又 二 入 几 九 八 七 厂 刀 乃 丁 卜
间 (7 Strokes)	我 这 来 时 你 作 里 没 还 进 但 把 两 间 应 体 利 身 位 声 抖 肝 纽 纹 豆 岗 吞 宏 肚 扭 坛 员 走 条 系 更 别 何 报 克 形 社 听 却 即 完 住 告 求 张 识 沃 歼 劫 闷 串 妥 址 妖 妨 汪 尿 运 步 改 每 极 快 证 近 远 兵 连 花 况 技 际 究 找 吧 李 医 芬 町 肠 忌 纲 壳 吻 扮 卵 泄 杆 局 希 投 陈 足 护 志 严 批 围 攻 苏 低 诉 男 助 坐 否 状 初 吟 肖 驳 吵 矣 佐 扯 孝 冻 陀 亨 陆 劳 财 纳 层 冷 村 判 余 灵 角 沉 坚 免 怀 乱 抗 佛 块 岛 吼 甫 巫 抄 狄 罕 姊 坑 钉 坠 屁 评 弟 伯 坏 丽 良 序 沙 困 县 均 附 呀 饭 含 补 疗 材 词 君 吩 辰 亩 坟 轩 龟 诈 纺 兑 纱 杖 园 宋 抓 束 纸 私 弄 忘 杨 床 迎 努 谷 库 弃 针 纯 忍 折 吴 坎 邱 芦 秃 旱 伽 芯 坊 驴 佣 鸣 秀 狂 伸 麦 阻 纷 违 译 汽 犹 纵 彻 役 玛 妙 杜 尾 伴 启 估 甸 灿 妒 芙 岚 诏 灶 纬 杉 尬 沦 泛 拒 呆 穷 灾 劲 迟 吹 鸡 诊 抢 返 忧 辛 励 扰 驱 闲 拟 贡 沪 旷 芽 汹 佑 妓 删 伺 坝 杏 冶 呈 戒 饮 芳 垂 帐 沟 扶 寿 吾

	邮 邻 赤 沈 抛 牢 吨 抚 苍 抑 庇 吱 邵 呕 灼 呐 伶 庐 扼 汰
瞬 (17 Strokes)	藏 戴 翼 繁 瞧 魏 臂 擦 癌 赢 糟 爵 霞 骤 瞪 瞬 鞭 徽 螺 黛 燥 藉 霜 蹈 朦 簇 礁 豁 磷 襄

4.2 Fine Classification

For fine classification, a classifier or similarity measure is established via statistical technique. It calculates the coefficient of determination (COD) for two-dimensional unreplicated linear functional relationship (2D-ULFR) model between the trajectory pattern of input character and character in database, according to which the recognition result is determined. 2D-ULFR model is used for the classification instead of other conventional regression models such as simple linear regression model, since the assumption of the conventional linear regression models that the explanatory variable can be measured exactly may not be realistic in the proposed recognition system. In this research, only the approximation coefficients of Haar wavelet transform are used as feature vector to represent the database and input characters. The COD for 2D-ULFR model then measures the similarity between these two set of feature vectors whereas their detailed coefficients are served as the error terms in the model. Hence, 2D-ULFR model is more preferable and appropriate to be adopted in the proposed recognition system. Furthermore, by utilizing 2D-ULFR model, more variation on handwriting is allowed, and this helps in achieving higher recognition rate even without undergoing normalization process.

4.2.1 Two-Dimensional Unreplicated Linear Functional Relationship Model

Two-dimensional unreplicated linear functional relationship (2D-ULFR) model is a special case of multidimensional unreplicated linear functional relationship (MULFR) model developed by Chang *et al.* (2009). The COD of MULFR was then used to measure the quality of JPEG compressed image in Chang *et al.* (2009). Until now, it has not been applied to character recognition yet. In 2D-ULFR, the dimension of the variables is set to two ($p = 2$). Instead of MULFR, 2D-ULFR is applied in the proposed recognition system since the extracted features are two dimensional vectors composed by *x*-coordinates and *y*-coordinates.

Consider the extracted feature for the input character trajectory, $\mathbf{a}$, where $\mathbf{a}_j = [a_{xj}, a_{yj}]$, $1 \le j \le N$, $2^5 \le N < 2^6$ and the extracted feature for trajectory of character in database, $\mathbf{b}$, where $\mathbf{b}_j = [b_{xj}, b_{yj}]$. Note that a_{xj}, a_{yj}, b_{xj} and b_{yj} are the approximation coefficients obtained from Haar wavelet transform for smoothing purpose and hence it will create errors. Suppose that $\mathbf{a}$ and $\mathbf{b}$ are observed with errors $\boldsymbol{\delta}_j = \left[\delta_{xj}, \delta_{yj}\right]$ and $\boldsymbol{\varepsilon}_j = \left[\varepsilon_{xj}, \varepsilon_{yj}\right]$ respectively, where $\delta_{xj}, \delta_{yj}, \varepsilon_{xj}$ and ε_{yj} are referred to the detailed coefficients of the Haar wavelet transform. In other words, $\boldsymbol{\delta}_j = \mathbf{d}_j$ and $\boldsymbol{\varepsilon}_j = \mathbf{e}_j$. Then, 2D-ULFR is defined as follow:

$$\mathbf{a}_j = \mathbf{A}_j + \boldsymbol{\delta}_j \tag{4.2}$$

$$\mathbf{b}_j = \mathbf{B}_j + \boldsymbol{\varepsilon}_j \tag{4.3}$$

where $\mathbf{A}_j = \left[A_{xj}, A_{yj}\right]$ and $\mathbf{B}_j = \left[B_{xj}, B_{yj}\right]$ are two linearly related unobservable true values of $\mathbf{a}_j$ and $\mathbf{b}_j$ such that

$$\mathbf{B}_j = \boldsymbol{\alpha} + \beta \mathbf{A}_j \qquad (4.4)$$

where $\boldsymbol{\alpha} = \left[\alpha_1, \alpha_2\right]$ is the intercept vector and β is the slope of the functional model. Assuming both error vectors are mutually and independently normally distributed with

(i) $E\left(\boldsymbol{\delta}_j\right) = \mathbf{0} = E\left(\boldsymbol{\varepsilon}_j\right)$

(ii) $Var\left(\delta_{xj}\right) = Var\left(\delta_{yj}\right) = \sigma^2$ and $Var\left(\varepsilon_{xj}\right) = Var\left(\varepsilon_{yj}\right) = \tau^2$ for

$\forall j;\ 1 \le j \le N,\ 2^5 \le N < 2^6$

(iii) $Cov\left(\delta_{xi}, \delta_{xj}\right) = Cov\left(\delta_{yi}, \delta_{yj}\right) = 0 = Cov\left(\varepsilon_{xi}, \varepsilon_{xj}\right) = Cov\left(\varepsilon_{yi}, \varepsilon_{yj}\right)$

for $\forall i \ne j;\ 1 \le i, j \le N,\ 2^5 \le N < 2^6$

$Cov\left(\delta_{xj}, \delta_{yj}\right) = 0 = Cov\left(\varepsilon_{xj}, \varepsilon_{yj}\right)$ for $\forall j;\ 1 \le j \le N,\ 2^5 \le N < 2^6$

$Cov\left(\delta_{xi}, \varepsilon_{yj}\right) = 0$ for $\forall i \ne j;\ 1 \le i, j \le N,\ 2^5 \le N < 2^6$

That is $\boldsymbol{\delta}_j \square IND\left(\mathbf{0}, \boldsymbol{\Omega}_{22}\right)$ and $\boldsymbol{\varepsilon}_j \square IND\left(\mathbf{0}, \boldsymbol{\Omega}_{11}\right)$, where $\boldsymbol{\Omega}_{11} = \begin{pmatrix} \tau^2 & 0 \\ 0 & \tau^2 \end{pmatrix} = \tau^2 \mathbf{I}$,

$\boldsymbol{\Omega}_{22} = \begin{pmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{pmatrix} = \sigma^2 \mathbf{I}$ and let $\mathbf{v}_j = \begin{pmatrix} \boldsymbol{\varepsilon}_j \\ \boldsymbol{\delta}_j \end{pmatrix}$, then $Cov\left(\mathbf{v}_j, \mathbf{v}_j\right) = \boldsymbol{\Omega} = \begin{pmatrix} \boldsymbol{\Omega}_{11} & \boldsymbol{\Omega}_{12} \\ \boldsymbol{\Omega}_{21} & \boldsymbol{\Omega}_{22} \end{pmatrix}$ are diagonal variance-covariance matrices with $\boldsymbol{\Omega}_{12} = \boldsymbol{\Omega}_{21} = \mathbf{0}$, $\boldsymbol{\Omega}_{11}$ and $\boldsymbol{\Omega}_{22}$ are positive definite.

4.2.2 Estimation of Parameters

Given the 2D-ULFR model with single slope defined by Equations (4.2), (4.3) and (4.4), the maximum likelihood estimators of $\boldsymbol{\alpha}$, β, $\mathbf{A}_j$ and σ^2 are

$$\hat{\boldsymbol{\alpha}} = \overline{\mathbf{b}} - \hat{\beta}\overline{\mathbf{a}}$$

$$\hat{\beta} = \frac{(S_{\mathbf{bb}} - \lambda S_{\mathbf{aa}}) + \sqrt{(S_{\mathbf{bb}} - \lambda S_{\mathbf{aa}})^2 + 4\lambda S_{\mathbf{ab}}^2}}{2S_{\mathbf{ab}}}$$

$$\hat{\mathbf{A}}_j = \frac{\lambda \mathbf{a}_j + \hat{\beta}\left(\mathbf{b}_j - \hat{\boldsymbol{\alpha}}\right)}{\lambda + \hat{\beta}^2}$$

and

$$\hat{\sigma}^2 = \frac{1}{D-2}\left\{\sum_{j=1}^{D}\left(\mathbf{a}_j - \hat{\mathbf{A}}_j\right)'\left(\mathbf{a}_j - \hat{\mathbf{A}}_j\right) + \frac{1}{\lambda}\sum_{j=1}^{D}\left(\mathbf{b}_j - \hat{\boldsymbol{\alpha}} - \hat{\beta}\hat{\mathbf{A}}_j\right)'\left(\mathbf{b}_j - \hat{\boldsymbol{\alpha}} - \hat{\beta}\hat{\mathbf{A}}_j\right)\right\}$$

where λ is the ratio of error variances, and $S_{\mathbf{aa}} = \sum_{j=1}^{D}\mathbf{a}'_j\mathbf{a}_j - D\overline{\mathbf{a}}'\overline{\mathbf{a}}$, $S_{\mathbf{bb}} = \sum_{j=1}^{D}\mathbf{b}'_j\mathbf{b}_j - D\overline{\mathbf{b}}'\overline{\mathbf{b}}$ and $S_{\mathbf{ab}} = \sum_{j=1}^{D}\mathbf{a}'_j\mathbf{b}_j - D\overline{\mathbf{a}}'\overline{\mathbf{b}}$. The estimation of these four parameters is derived below (Chang *et al.*, 2009).

It starts with the joint probability density function of $\boldsymbol{\delta}_j$ and $\boldsymbol{\varepsilon}_j$

$$f\left(\mathbf{a}_j, \mathbf{b}_j\right) = \frac{1}{\left(\sqrt{2\pi}\right)^r |\boldsymbol{\Omega}|^{1/2}} \exp\left[-\frac{1}{2}\left\{\begin{pmatrix}\mathbf{b}_j - E\left(\mathbf{b}_j\right) \\ \mathbf{a}_j - E\left(\mathbf{a}_j\right)\end{pmatrix}' \boldsymbol{\Omega}^{-1} \begin{pmatrix}\mathbf{b}_j - E\left(\mathbf{b}_j\right) \\ \mathbf{a}_j - E\left(\mathbf{a}_j\right)\end{pmatrix}\right\}\right]$$

$$= \frac{1}{(2\pi)^{r/2}\left|\boldsymbol{\Omega}\right|^{1/2}} \exp\left[-\frac{1}{2}\left\{\left[\left(\mathbf{b}_j - \mathbf{B}_j\right)' \quad \left(\mathbf{a}_j - \mathbf{A}_j\right)'\right]\boldsymbol{\Omega}^{-1}\begin{pmatrix}\mathbf{b}_j - \mathbf{B}_j \\ \mathbf{a}_j - \mathbf{A}_j\end{pmatrix}\right\}\right] \tag{4.5}$$

where $r = 2p = 2(2) = 4$, $E\left(\mathbf{a}_j\right) = E\left(\mathbf{A}_j + \boldsymbol{\delta}_j\right) = \mathbf{A}_j$ and $E\left(\mathbf{b}_j\right) = E\left(\mathbf{B}_j + \boldsymbol{\varepsilon}_j\right) = \mathbf{B}_j$.

The likelihood function for Equation (4.5) is

$$L = \prod_{j=1}^{N} f\left(\mathbf{a}_j, \mathbf{b}_j\right) = \prod_{j=1}^{N} \frac{1}{(2\pi)^{r/2}\left|\boldsymbol{\Omega}\right|^{1/2}} \exp\left[-\frac{1}{2}\left\{\left[\left(\mathbf{b}_j - \mathbf{B}_j\right)' \quad \left(\mathbf{a}_j - \mathbf{A}_j\right)'\right]\boldsymbol{\Omega}^{-1}\begin{pmatrix}\mathbf{b}_j - \mathbf{B}_j \\ \mathbf{a}_j - \mathbf{A}_j\end{pmatrix}\right\}\right]$$

$$= \frac{1}{(2\pi)^{rN/2}\left|\boldsymbol{\Omega}\right|^{N/2}} \exp\left[-\frac{1}{2}\left\{\sum_{j=1}^{N}\left(\left(\mathbf{b}_j - \mathbf{B}_j\right)' \boldsymbol{\Omega}_{11}^{-1}\left(\mathbf{b}_j - \mathbf{B}_j\right) + \left(\mathbf{a}_j - \mathbf{A}_j\right)' \boldsymbol{\Omega}_{22}^{-1}\left(\mathbf{a}_j - \mathbf{A}_j\right)\right)\right\}\right]$$

$$= \frac{1}{\mathrm{K}\left|\boldsymbol{\Omega}\right|^{N/2}} \exp\left\{-\frac{1}{2}\sum_{j=1}^{N}\left[\left(\mathbf{a}_j - \mathbf{A}_j\right)' \boldsymbol{\Omega}_{22}^{-1}\left(\mathbf{a}_j - \mathbf{A}_j\right)\right.\right.$$

$$\left.\left. + \left(\mathbf{b}_j - \boldsymbol{\alpha} - \beta\mathbf{A}_j\right)' \boldsymbol{\Omega}_{11}^{-1}\left(\mathbf{b}_j - \boldsymbol{\alpha} - \beta\mathbf{A}_j\right)\right]\right\}$$

where $\mathrm{K} = (2\pi)^{rN/2}$ and the log-likelihood function is

$$L^* = \ln L$$

$$= -\ln \mathrm{K} - \frac{N}{2}\ln\left|\boldsymbol{\Omega}\right| - \frac{1}{2}\sum_{j=1}^{N}\left[\left(\mathbf{a}_j - \mathbf{A}_j\right)' \boldsymbol{\Omega}_{22}^{-1}\left(\mathbf{a}_j - \mathbf{A}_j\right)\right.$$

$$\left. + \left(\mathbf{b}_j - \boldsymbol{\alpha} - \beta\mathbf{A}_j\right)' \boldsymbol{\Omega}_{11}^{-1}\left(\mathbf{b}_j - \boldsymbol{\alpha} - \beta\mathbf{A}_j\right)\right] \tag{4.6}$$

To overcome the unbounded problem of Equation (4.6), an additional constraint followed Kendall and Stuart (1979) will be suggested, i.e.:

$$\mathbf{\Omega}_{11} = \lambda\mathbf{\Omega}_{22} \quad \Leftrightarrow \quad \mathbf{\Omega}_{11}^{-1} = \frac{1}{\lambda}\mathbf{\Omega}_{22}^{-1} \quad \Leftrightarrow \quad \tau^2 = \lambda\sigma^2$$

where the ratio of error variances λ is a known constant. In this case, Equation (4.6) becomes

$$\begin{aligned} L^* &= -\ln \mathrm{K} - \frac{N}{2}\ln \lambda^2 \left|\mathbf{\Omega}_{22}\right|^2 \\ &\quad -\frac{1}{2}\sum_{j=1}^{N}\left[\left(\mathbf{a}_j - \mathbf{A}_j\right)' \mathbf{\Omega}_{22}^{-1}\left(\mathbf{a}_j - \mathbf{A}_j\right) + \frac{1}{\lambda}\left(\mathbf{b}_j - \boldsymbol{\alpha} - \beta\mathbf{A}_j\right)' \mathbf{\Omega}_{22}^{-1}\left(\mathbf{b}_j - \boldsymbol{\alpha} - \beta\mathbf{A}_j\right)\right] \\ &= -\ln \mathrm{K} - \frac{N}{2}\ln \lambda^2 - N\ln\left|\mathbf{\Omega}_{22}\right| \\ &\quad -\frac{1}{2}\sum_{j=1}^{N}\left[\left(\mathbf{a}_j - \mathbf{A}_j\right)' \mathbf{\Omega}_{22}^{-1}\left(\mathbf{a}_j - \mathbf{A}_j\right) + \frac{1}{\lambda}\left(\mathbf{b}_j - \boldsymbol{\alpha} - \beta\mathbf{A}_j\right)' \mathbf{\Omega}_{22}^{-1}\left(\mathbf{b}_j - \boldsymbol{\alpha} - \beta\mathbf{A}_j\right)\right] \end{aligned} \tag{4.7}$$

where $\left|\mathbf{\Omega}\right| = \left|\begin{pmatrix}\mathbf{\Omega}_{11} & \mathbf{\Omega}_{12} \\ \mathbf{\Omega}_{21} & \mathbf{\Omega}_{22}\end{pmatrix}\right| = \left|\mathbf{\Omega}_{11}\mathbf{\Omega}_{22}\right| = \left|\lambda\mathbf{\Omega}_{22}\mathbf{\Omega}_{22}\right| = \lambda^2\left|\mathbf{\Omega}_{22}\right|^2$.

There are $\left(Np + p + 2\right) = 2N + 4$ parameters to be estimated, which are $\mathbf{A}_j$, $\boldsymbol{\alpha}$, β, and σ^2.

From the vector derivative formula for quadratic matrix equation evaluating to a scalar,

$$\begin{aligned} \frac{\partial L^*}{\partial\left(\beta\mathbf{A}_j - \mathbf{b}_j\right)} &= -\frac{1}{2\lambda}\sum_{j=1}^{N}\left\{\left(\mathbf{\Omega}_{22}^{-1}\right) + \left(\mathbf{\Omega}_{22}^{-1}\right)'\right\}\left[\left(\beta\mathbf{A}_j - \mathbf{b}_j\right) + \boldsymbol{\alpha}\right] \\ &= -\frac{1}{2\lambda}\sum_{j=1}^{N} 2\mathbf{\Omega}_{22}^{-1}\left[\left(\beta\mathbf{A}_j - \mathbf{b}_j\right) + \boldsymbol{\alpha}\right] \qquad \left(\because \left(\mathbf{\Omega}_{22}^{-1}\right)' = \mathbf{\Omega}_{22}^{-1}\right) \end{aligned}$$

$$= -\frac{1}{\lambda}\sum_{j=1}^{N}\mathbf{\Omega}_{22}^{-1}\left[\beta\mathbf{A}_j - \mathbf{b}_j + \boldsymbol{\alpha}\right]$$

and the tangent vector to curve $\left(\beta\mathbf{A}_j - \mathbf{b}_j\right): \mathbb{R} \to \mathbb{R}^D$ is $\frac{\partial\left(\beta\mathbf{A}_j - \mathbf{b}_j\right)}{\partial\beta} = \mathbf{A}_j$.

By using the Chain rule,

$$\frac{\partial L^*}{\partial\beta} = \frac{\partial L^*}{\partial\left(\beta\mathbf{A}_j - \mathbf{b}_j\right)'}\frac{\partial\left(\beta\mathbf{A}_j - \mathbf{b}_j\right)}{\partial\beta}$$

$$= -\frac{1}{\lambda}\sum_{j=1}^{N}\left(\beta\mathbf{A}_j - \mathbf{b}_j + \boldsymbol{\alpha}\right)'\mathbf{\Omega}_{22}^{-1}\mathbf{A}_j$$

$$= \frac{1}{\lambda}\sum_{j=1}^{N}\left(\mathbf{b}_j - \boldsymbol{\alpha} - \beta\mathbf{A}_j\right)'\mathbf{\Omega}_{22}^{-1}\mathbf{A}_j$$

Therefore, differentiate Equation (4.7) with respect to β and set the result equal to zero $\left(\frac{\partial L^*}{\partial\beta} = 0\right)$, yields

$$\sum_{j=1}^{N}\frac{1}{\sigma^2}\left(\mathbf{b}_j - \boldsymbol{\alpha} - \beta\mathbf{A}_j\right)'\mathbf{I}\left(\mathbf{A}_j\right) = 0 \qquad \left(\because \mathbf{\Omega}_{22} = \sigma^2\mathbf{I}\right)$$

$$\sum_{j=1}^{N}\mathbf{b}_j'\mathbf{A}_j - \boldsymbol{\alpha}'\sum_{j=1}^{N}\mathbf{A}_j - \beta\sum_{j=1}^{N}\mathbf{A}_j'\mathbf{A}_j = 0$$

$$\therefore \hat{\beta} = \frac{\sum_{j=1}^{N}\mathbf{b}_j'\hat{\mathbf{A}}_j - \hat{\boldsymbol{\alpha}}'\sum_{j=1}^{N}\hat{\mathbf{A}}_j}{\sum_{j=1}^{N}\hat{\mathbf{A}}_j'\hat{\mathbf{A}}_j} \qquad (4.8)$$

Similarly, differential Equation (4.7) with respect $\boldsymbol{\alpha}$, $\mathbf{A}_j$ and σ give the following results

$$\frac{\partial L^*}{\partial \mathbf{A}_j} = -\frac{1}{2}\left\{-2\boldsymbol{\Omega}_{22}^{-1}\left(\mathbf{a}_j - \mathbf{A}_j\right) + \frac{2}{\lambda}\boldsymbol{\Omega}_{22}^{-1}\left(\mathbf{b}_j - \boldsymbol{\alpha} - \beta\mathbf{A}_j\right)\left(-\beta\right)\right\} = \mathbf{0}$$

$$\left(\mathbf{a}_j - \mathbf{A}_j\right) + \frac{1}{\lambda}\beta\left(\mathbf{b}_j - \boldsymbol{\alpha} - \beta\mathbf{A}_j\right) = \mathbf{0}$$

$$-\left(\lambda + \beta^2\right)\mathbf{A}_j + \left(\lambda\mathbf{a}_j + \beta\mathbf{b}_j - \beta\boldsymbol{\alpha}\right) = \mathbf{0}$$

$$\therefore \hat{\mathbf{A}}_j = \frac{\lambda\mathbf{a}_j + \hat{\beta}\left(\mathbf{b}_j - \hat{\boldsymbol{\alpha}}\right)}{\lambda + \hat{\beta}^2} \tag{4.9}$$

$$\frac{\partial L^*}{\partial \boldsymbol{\alpha}} = -\frac{1}{2}\sum_{j=1}^{N}\frac{-2}{\lambda}\boldsymbol{\Omega}_{22}^{-1}\left(\mathbf{b}_j - \boldsymbol{\alpha} - \beta\mathbf{A}_j\right) = \mathbf{0}$$

$\sum_{j=1}^{N}\left(\mathbf{b}_j - \boldsymbol{\alpha} - \beta\mathbf{A}_j\right) = \mathbf{0}$ $\because \boldsymbol{\Omega}_{22}$ is positive definite, diagonal and $\boldsymbol{\Omega}_{22} = \sigma^2\mathbf{I}$

$$\sum_{j=1}^{N}\mathbf{b}_j - N\boldsymbol{\alpha} - \beta\sum_{j=1}^{N}\mathbf{A}_j = \mathbf{0}$$

$$\therefore \hat{\boldsymbol{\alpha}} = \frac{1}{N}\sum_{j=1}^{N}\mathbf{b}_j - \hat{\beta}\frac{1}{N}\sum_{j=1}^{N}\hat{\mathbf{A}}_j$$

Substitute Equation (4.9) yields

$$\hat{\boldsymbol{\alpha}} = \frac{1}{N}\sum_{j=1}^{N}\mathbf{b}_j - \hat{\beta}\frac{1}{N}\sum_{j=1}^{N}\left[\frac{\lambda\mathbf{a}_j + \hat{\beta}\left(\mathbf{b}_j - \hat{\boldsymbol{\alpha}}\right)}{\lambda + \hat{\beta}^2}\right]$$

$$\hat{\boldsymbol{\alpha}} = \overline{\mathbf{b}} - \frac{1}{\lambda + \hat{\beta}^2}\left(\lambda\hat{\beta}\overline{\mathbf{a}} + \hat{\beta}^2\overline{\mathbf{b}} - \hat{\beta}^2\hat{\boldsymbol{\alpha}}\right)$$

$$\hat{\boldsymbol{\alpha}} - \frac{\hat{\beta}^2\hat{\boldsymbol{\alpha}}}{\lambda + \hat{\beta}^2} = \left(1 - \frac{\hat{\beta}^2}{\lambda + \hat{\beta}^2}\right)\overline{\mathbf{b}} - \frac{\lambda\hat{\beta}\overline{\mathbf{a}}}{\lambda + \hat{\beta}^2}$$

$$\hat{\boldsymbol{\alpha}} = \overline{\mathbf{b}} - \hat{\beta}\overline{\mathbf{a}} \tag{4.10}$$

where $\overline{\mathbf{b}} = \left[\overline{b}_1\ \overline{b}_2\right]'$ and $\overline{\mathbf{a}} = \left[\overline{a}_1\ \overline{a}_2\right]'$.

$$\frac{\partial L^{*}}{\partial \sigma} = -\frac{2N}{\sigma} + \sigma^{-3}\left\{\sum_{j=1}^{N}\left(\mathbf{a}_j - \mathbf{A}_j\right)'\left(\mathbf{a}_j - \mathbf{A}_j\right) + \frac{1}{\lambda}\sum_{j=1}^{N}\left(\mathbf{b}_j - \boldsymbol{\alpha} - \beta\mathbf{A}_j\right)'\left(\mathbf{b}_j - \boldsymbol{\alpha} - \beta\mathbf{A}_j\right)\right\} = 0$$

$$\frac{2N}{\sigma} = \frac{1}{\sigma^{3}}\left\{\sum_{j=1}^{N}\left(\mathbf{a}_j - \mathbf{A}_j\right)'\left(\mathbf{a}_j - \mathbf{A}_j\right) + \frac{1}{\lambda}\sum_{j=1}^{N}\left(\mathbf{b}_j - \boldsymbol{\alpha} - \beta\mathbf{A}_j\right)'\left(\mathbf{b}_j - \boldsymbol{\alpha} - \beta\mathbf{A}_j\right)\right\}$$

$$\therefore \hat{\sigma}^{2} = \frac{1}{2N}\left\{\sum_{j=1}^{N}\left(\mathbf{a}_j - \hat{\mathbf{A}}_j\right)'\left(\mathbf{a}_j - \hat{\mathbf{A}}_j\right) + \frac{1}{\lambda}\sum_{j=1}^{N}\left(\mathbf{b}_j - \hat{\boldsymbol{\alpha}} - \hat{\beta}\hat{\mathbf{A}}_j\right)'\left(\mathbf{b}_j - \hat{\boldsymbol{\alpha}} - \hat{\beta}\hat{\mathbf{A}}_j\right)\right\}$$

(4.11)

Since $\hat{\sigma}^2$ is a bias estimator of σ^2 (Kendall and Stuart, 1979), Equation (4.11) is multiplied by $\frac{2N}{N-2}$ yields the consistent estimator

$$\therefore \hat{\sigma}^{2} = \frac{1}{N-2}\left\{\sum_{j=1}^{N}\left(\mathbf{a}_j - \hat{\mathbf{A}}_j\right)'\left(\mathbf{a}_j - \hat{\mathbf{A}}_j\right) + \frac{1}{\lambda}\sum_{j=1}^{N}\left(\mathbf{b}_j - \hat{\boldsymbol{\alpha}} - \hat{\beta}\hat{\mathbf{A}}_j\right)'\left(\mathbf{b}_j - \hat{\boldsymbol{\alpha}} - \hat{\beta}\hat{\mathbf{A}}_j\right)\right\}$$

(4.12)

Substitute Equations (4.9) and (4.10) into Equation (4.8) yields

$$\therefore \hat{\beta} = \frac{\sum_{j=1}^{N}\mathbf{b}_j'\left(\frac{\lambda\mathbf{a}_j + \hat{\beta}\left(\mathbf{b}_j - \hat{\boldsymbol{\alpha}}\right)}{\lambda + \hat{\beta}^2}\right) - \hat{\boldsymbol{\alpha}}'\sum_{j=1}^{N}\left(\frac{\lambda\mathbf{a}_j + \hat{\beta}\left(\mathbf{b}_j - \hat{\boldsymbol{\alpha}}\right)}{\lambda + \hat{\beta}^2}\right)}{\sum_{j=1}^{N}\left(\frac{\lambda\mathbf{a}_j + \hat{\beta}\left(\mathbf{b}_j - \hat{\boldsymbol{\alpha}}\right)}{\lambda + \hat{\beta}^2}\right)'\left(\frac{\lambda\mathbf{a}_j + \hat{\beta}\left(\mathbf{b}_j - \hat{\boldsymbol{\alpha}}\right)}{\lambda + \hat{\beta}^2}\right)}$$

$$= \frac{\left(\lambda + \hat{\beta}^2\right)\left\{\sum_{j=1}^{N}\left(\lambda\mathbf{a}_j'\mathbf{b}_j + \hat{\beta}\mathbf{b}_j'\mathbf{b}_j - \hat{\beta}\hat{\boldsymbol{\alpha}}'\mathbf{b}_j\right) - \lambda\hat{\boldsymbol{\alpha}}'\sum_{j=1}^{N}\mathbf{a}_j - \hat{\beta}\hat{\boldsymbol{\alpha}}'\sum_{j=1}^{N}\mathbf{b}_j + N\hat{\beta}\hat{\boldsymbol{\alpha}}'\hat{\boldsymbol{\alpha}}\right\}}{\sum_{j=1}^{N}\left(\lambda^2\mathbf{a}_j'\mathbf{a}_j + 2\lambda\hat{\beta}\mathbf{a}_j'\mathbf{b}_j - 2\hat{\beta}\hat{\boldsymbol{\alpha}}'\mathbf{a}_j - 2\hat{\beta}^2\hat{\boldsymbol{\alpha}}'\mathbf{b}_j + \hat{\beta}^2\mathbf{b}_j'\mathbf{b}_j + \hat{\beta}^2\hat{\boldsymbol{\alpha}}'\hat{\boldsymbol{\alpha}}\right)}$$

$$=\frac{\left(\lambda+\hat{\beta}^2\right)\left\{\lambda\sum_{j=1}^{N}\mathbf{a}_j'\mathbf{b}_j+\hat{\beta}\sum_{j=1}^{N}\mathbf{b}_j'\mathbf{b}_j-2N\hat{\beta}\hat{\boldsymbol{\alpha}}'\overline{\mathbf{b}}-\lambda N\hat{\boldsymbol{\alpha}}'\overline{\mathbf{a}}+N\hat{\beta}\hat{\boldsymbol{\alpha}}'\hat{\boldsymbol{\alpha}}\right\}}{\lambda^2\sum_{j=1}^{N}\mathbf{a}_j'\mathbf{a}_j+2\lambda\hat{\beta}\sum_{j=1}^{N}\mathbf{a}_j'\mathbf{b}_j-2\lambda\hat{\beta}\hat{\boldsymbol{\alpha}}'\sum_{j=1}^{N}\mathbf{a}_j-2\hat{\beta}^2\hat{\boldsymbol{\alpha}}'\sum_{j=1}^{N}\mathbf{b}_j+\hat{\beta}^2\sum_{j=1}^{N}\mathbf{b}_j'\mathbf{b}_j+N\hat{\beta}^2\hat{\boldsymbol{\alpha}}'\hat{\boldsymbol{\alpha}}}$$

$$=\frac{\left(\lambda+\hat{\beta}^2\right)\left\{\lambda\sum_{j=1}^{N}\mathbf{a}_j'\mathbf{b}_j+\hat{\beta}\sum_{j=1}^{N}\mathbf{b}_j'\mathbf{b}_j-2N\hat{\beta}\left(\overline{\mathbf{b}}-\hat{\beta}\overline{\mathbf{a}}\right)'\overline{\mathbf{b}}-\lambda N\left(\overline{\mathbf{b}}-\hat{\beta}\overline{\mathbf{a}}\right)'\overline{\mathbf{a}}+N\hat{\beta}\left(\overline{\mathbf{b}}-\hat{\beta}\overline{\mathbf{a}}\right)'\left(\overline{\mathbf{b}}-\hat{\beta}\overline{\mathbf{a}}\right)\right\}}{\lambda^2\sum_{j=1}^{N}\mathbf{a}_j'\mathbf{a}_j+2\lambda\hat{\beta}\sum_{j=1}^{N}\mathbf{a}_j'\mathbf{b}_j-2N\lambda\hat{\beta}\left(\overline{\mathbf{b}}-\hat{\beta}\overline{\mathbf{a}}\right)'\overline{\mathbf{a}}_j-2N\hat{\beta}^2\left(\overline{\mathbf{b}}-\hat{\beta}\overline{\mathbf{a}}\right)'\overline{\mathbf{b}}_j+\hat{\beta}^2\sum_{j=1}^{N}\mathbf{b}_j'\mathbf{b}_j+N\hat{\beta}^2\left(\overline{\mathbf{b}}-\hat{\beta}\overline{\mathbf{a}}\right)'\left(\overline{\mathbf{b}}-\hat{\beta}\overline{\mathbf{a}}\right)}$$

$$=\frac{\left(\lambda+\hat{\beta}^2\right)\left\{\lambda\left(\sum_{j=1}^{N}\mathbf{a}_j'\mathbf{b}_j-N\overline{\mathbf{a}}'\overline{\mathbf{b}}\right)+\hat{\beta}\left(\sum_{j=1}^{N}\mathbf{b}_j'\mathbf{b}_j-N\overline{\mathbf{b}}'\overline{\mathbf{b}}\right)+\lambda N\hat{\beta}\overline{\mathbf{a}}'\overline{\mathbf{a}}+N\hat{\beta}^3\overline{\mathbf{a}}'\overline{\mathbf{a}}\right\}}{\lambda^2\sum_{j=1}^{N}\mathbf{a}_j'\mathbf{a}_j+2\lambda\hat{\beta}\left(\sum_{j=1}^{N}\mathbf{a}_j'\mathbf{b}_j-N\overline{\mathbf{a}}'\overline{\mathbf{b}}\right)+\hat{\beta}^2\left(\sum_{j=1}^{N}\mathbf{b}_j'\mathbf{b}_j-N\overline{\mathbf{b}}'\overline{\mathbf{b}}\right)+2N\lambda\hat{\beta}^2\overline{\mathbf{a}}'\overline{\mathbf{a}}+N\hat{\beta}^4\overline{\mathbf{a}}'\overline{\mathbf{a}}}$$

$$=\frac{\left(\lambda+\hat{\beta}^2\right)\left\{\lambda S_{\mathbf{ab}}+\hat{\beta}S_{\mathbf{bb}}+\lambda N\hat{\beta}\overline{\mathbf{a}}'\overline{\mathbf{a}}+N\hat{\beta}^3\overline{\mathbf{a}}'\overline{\mathbf{a}}\right\}}{\lambda^2\sum_{j=1}^{N}\mathbf{a}_j'\mathbf{a}_j+2\lambda\hat{\beta}S_{\mathbf{ab}}+\hat{\beta}^2S_{\mathbf{bb}}+2N\lambda\hat{\beta}^2\overline{\mathbf{a}}'\overline{\mathbf{a}}+N\hat{\beta}^4\overline{\mathbf{a}}'\overline{\mathbf{a}}}$$

where $S_{\mathbf{aa}}=\sum_{j=1}^{N}\mathbf{a}_j'\mathbf{a}_j-N\overline{\mathbf{a}}'\overline{\mathbf{a}}$, $S_{\mathbf{bb}}=\sum_{j=1}^{N}\mathbf{b}_j'\mathbf{b}_j-N\overline{\mathbf{b}}'\overline{\mathbf{b}}$ and $S_{\mathbf{ab}}=\sum_{j=1}^{N}\mathbf{a}_j'\mathbf{b}_j-N\overline{\mathbf{a}}'\overline{\mathbf{b}}$.

This implies that

$$\lambda^2\hat{\beta}\sum_{j=1}^{N}\mathbf{a}_j'\mathbf{a}_j+2\lambda\hat{\beta}^2S_{\mathbf{ab}}+\hat{\beta}^3S_{\mathbf{bb}}+2N\lambda\hat{\beta}^3\overline{\mathbf{a}}'\overline{\mathbf{a}}+N\hat{\beta}^5\overline{\mathbf{a}}'\overline{\mathbf{a}}$$
$$=\lambda^2S_{\mathbf{ab}}+\lambda\hat{\beta}S_{\mathbf{bb}}+\lambda^2N\hat{\beta}\overline{\mathbf{a}}'\overline{\mathbf{a}}+\lambda N\hat{\beta}^3\overline{\mathbf{a}}'\overline{\mathbf{a}}+\lambda\hat{\beta}^2S_{\mathbf{ab}}+\hat{\beta}^3S_{\mathbf{bb}}+\lambda N\hat{\beta}^3\overline{\mathbf{a}}'\overline{\mathbf{a}}+N\hat{\beta}^5\overline{\mathbf{a}}'\overline{\mathbf{a}}$$

$$\Rightarrow\lambda^2\hat{\beta}\sum_{j=1}^{N}\mathbf{a}_j'\mathbf{a}_j-\lambda^2N\hat{\beta}\overline{\mathbf{a}}'\overline{\mathbf{a}}+\lambda\hat{\beta}^2S_{\mathbf{ab}}-\lambda^2S_{\mathbf{ab}}-\lambda\hat{\beta}S_{\mathbf{bb}}=0$$

$$\hat{\beta}^2S_{\mathbf{ab}}+\hat{\beta}\left(\lambda S_{\mathbf{aa}}-S_{\mathbf{bb}}\right)-\lambda S_{\mathbf{ab}}=0 \qquad (4.13)$$

Solving the quadratic Equation (4.13) yields

$$\hat{\beta} = \frac{-\left(\lambda S_{\mathbf{aa}} - S_{\mathbf{bb}}\right) \pm \sqrt{\left(\lambda S_{\mathbf{aa}} - S_{\mathbf{bb}}\right)^2 + 4\lambda S_{\mathbf{ab}}^2}}{2S_{\mathbf{ab}}}$$

$$= \frac{\left(S_{\mathbf{bb}} - \lambda S_{\mathbf{aa}}\right) \pm \sqrt{\left(S_{\mathbf{bb}} - \lambda S_{\mathbf{aa}}\right)^2 + 4\lambda S_{\mathbf{ab}}^2}}{2S_{\mathbf{ab}}}$$

$$\therefore \hat{\beta} = \frac{\left(S_{\mathbf{bb}} - \lambda S_{\mathbf{aa}}\right) + \sqrt{\left(S_{\mathbf{bb}} - \lambda S_{\mathbf{aa}}\right)^2 + 4\lambda S_{\mathbf{ab}}^2}}{2S_{\mathbf{ab}}} \qquad (4.14)$$

The positive sign is used in Equation (4.14) because it gives a maximum to the likelihood function in Equation (4.7) as shown below. From the previous result,

$$\frac{\partial L^*}{\partial \beta} = \frac{1}{\lambda}\sum_{j=1}^{N}\left(\mathbf{b}_j - \boldsymbol{\alpha} - \beta \mathbf{A}_j\right)' \boldsymbol{\Omega}_{22}^{-1}\mathbf{A}_j = \frac{1}{\lambda}\left(\sum \mathbf{B}_j' \boldsymbol{\Omega}_{22}^{-1}\mathbf{A}_j - \boldsymbol{\alpha}' \sum \boldsymbol{\Omega}_{22}^{-1}\mathbf{A}_j - \beta \sum \mathbf{A}_j' \boldsymbol{\Omega}_{22}^{-1}\mathbf{A}_j\right)$$

and the second order derivative yields

$$\frac{\partial^2 L^*}{\partial \beta^2} = \frac{-1}{\lambda}\sum \mathbf{A}_j' \boldsymbol{\Omega}_{22}^{-1}\mathbf{A}_j = \frac{-1}{\lambda\sigma^2}\sum \mathbf{A}_j' \mathbf{I} \mathbf{A}_j = \frac{-1}{\lambda\sigma^2}\sum \mathbf{A}_j' \mathbf{A}_j$$
$$\left(\because \boldsymbol{\Omega}_{22} = \sigma^2 \mathbf{I}\right)$$

Since $\sum \mathbf{A}_j' \mathbf{A}_j > 0$ (practically $\mathbf{A} \neq \mathbf{0}$) and $\lambda > 0$, this implies that $\frac{\partial^2 L^*}{\partial \beta^2} < 0$.

The $\hat{\beta}$ s are local maximum points. Now, let

$$\hat{\beta} = \frac{\left(S_{\mathbf{bb}} - \lambda S_{\mathbf{aa}}\right) \pm \sqrt{\left(S_{\mathbf{bb}} - \lambda S_{\mathbf{aa}}\right)^2 + 4\lambda S_{\mathbf{ab}}^2}}{2S_{\mathbf{ab}}} = \frac{\Delta}{2S_{\mathbf{ab}}}$$

Furthermore, Chang *et al.* (2009) shown that $\Delta = 2\hat{\beta}S_{\mathbf{ab}} \geq 0$ must be non-negative and therefore the positive square root must always be taken.

4.2.3 Coefficient of Determination for 2D-ULFR Model

In statistic, coefficient of determination (COD) is defined as the proportion of variability in a dependent variable explained by the independent variable(s) of the regression model (Montgomery and Peck, 1992). In the context of this research, character in database will be the dependent variable while input character is the independent variable. In other words, the value of COD is assumed to be the similarity measure between the character in database and input character. Given an input character, the characters in database with largest value of COD will be the final result. The formula for the COD of 2D-ULFR model is given as

$$R_p^2 = \frac{SS_R}{S_{\mathbf{bb}}} = \frac{\hat{\beta} S_{\mathbf{ab}}}{S_{\mathbf{bb}}} \tag{4.15}$$

where the ratio of the error variances be known and equals to one $(\lambda = 1)$. The derivation of Equation (4.15) is demonstrated below (Chang *et al.*, 2009).

Re-write the Equations (4.2), (4.3) and (4.4) as

$$\mathbf{b}_j = \boldsymbol{\alpha} + \beta \mathbf{A}_j + \boldsymbol{\varepsilon}_j = \boldsymbol{\alpha} + \beta \mathbf{a}_j + \left(\boldsymbol{\varepsilon}_j - \beta \boldsymbol{\delta}_j\right) = \boldsymbol{\alpha} + \beta \mathbf{a}_j + \mathbf{V}_j \tag{4.16}$$

where the errors of the model

$$\mathbf{V}_j = \boldsymbol{\varepsilon}_j - \beta \boldsymbol{\delta}_j = \mathbf{b}_j - \boldsymbol{\alpha} - \beta \mathbf{a}_j,\ 1 \le j \le N,\ 2^5 \le N < 2^6 \tag{4.17}$$

is a normally distributed 2-dimensional random variable with

$$E\left(\mathbf{V}_j\right) = E\left(\boldsymbol{\varepsilon}_j - \beta \boldsymbol{\delta}_j\right) = E\left(\boldsymbol{\varepsilon}_j\right) - \beta E\left(\boldsymbol{\delta}_j\right) = \mathbf{0} \quad \left(\because E\left(\boldsymbol{\varepsilon}_j\right) = E\left(\boldsymbol{\delta}_j\right) = 0\right)$$

and $$Var\left(\mathbf{V}_j\right) = Var\left(\boldsymbol{\varepsilon}_j - \beta\boldsymbol{\delta}_j\right)$$

$$= Var\left(\boldsymbol{\varepsilon}_j\right) + \beta^2 Var\left(\boldsymbol{\delta}_j\right) - 2Cov\left(\boldsymbol{\varepsilon}_j, \beta\boldsymbol{\delta}_j\right)$$

$$= \boldsymbol{\Omega}_{11} + \beta^2\boldsymbol{\Omega}_{22} \qquad \left(\because Cov\left(\boldsymbol{\varepsilon}_j, \boldsymbol{\delta}_j\right) = \boldsymbol{\Omega}_{12} = \mathbf{0}\right)$$

If $\hat{\boldsymbol{\alpha}}$ and $\hat{\beta}$ are estimators of $\boldsymbol{\alpha}$ and β, respectively, then from Equation (4.17),

$$\hat{\mathbf{V}}_j = \mathbf{b}_j - \hat{\mathbf{b}}_j = \mathbf{b}_j - \hat{\boldsymbol{\alpha}} - \hat{\beta}\mathbf{a}_j,\ 1 \le j \le N,\ 2^5 \le N < 2^6$$

is the residual of the model. Since

$$Var\left(\hat{\mathbf{V}}_j\right) = Var\left(\boldsymbol{\varepsilon}_j - \hat{\beta}\boldsymbol{\delta}_j\right)$$

$$= Var\left(\boldsymbol{\varepsilon}_j\right) + \hat{\beta}^2 Var\left(\boldsymbol{\delta}_j\right) - 2Cov\left(\boldsymbol{\varepsilon}_j, \hat{\beta}\boldsymbol{\delta}_j\right)$$

$$= \boldsymbol{\Omega}_{11} + \hat{\beta}^2\boldsymbol{\Omega}_{22}$$

$$\because Var\left(\boldsymbol{\varepsilon}_j\right) = \boldsymbol{\Omega}_{11},\ Var\left(\boldsymbol{\delta}_j\right) = \boldsymbol{\Omega}_{22},\ Cov\left(\boldsymbol{\varepsilon}_j, \hat{\beta}\boldsymbol{\delta}_j\right) = \mathbf{0}$$

$$= \lambda\boldsymbol{\Omega}_{22} + \hat{\beta}^2\boldsymbol{\Omega}_{22} \qquad \because \boldsymbol{\Omega}_{11} = \lambda\boldsymbol{\Omega}_{22}$$

$$= \left(\lambda + \hat{\beta}^2\right)\boldsymbol{\Omega}_{22}$$

The residual sum of squares is divided by $\left(\lambda + \hat{\beta}^2\right)$ yields

$$SS_E = \frac{1}{\lambda + \hat{\beta}^2}\sum \hat{\mathbf{V}}_j^2 = \frac{1}{\lambda + \hat{\beta}^2}\sum\left(\mathbf{b}_j - \hat{\boldsymbol{\alpha}} - \hat{\beta}\mathbf{a}_j\right)^2$$

$$=\frac{1}{\lambda+\hat{\beta}^2}\left(\sum \mathbf{b}_j'\mathbf{b}_j-2\hat{\boldsymbol{\alpha}}'\sum \mathbf{b}_j-2\hat{\beta}\sum \mathbf{a}_j'\mathbf{b}_j+2\hat{\beta}\hat{\boldsymbol{\alpha}}'\sum \mathbf{a}_j+N\hat{\boldsymbol{\alpha}}'\hat{\boldsymbol{\alpha}}+\hat{\beta}^2\sum \mathbf{a}_j'\mathbf{a}_j\right)$$

$$\begin{aligned}=\frac{1}{\lambda+\hat{\beta}^2}\Bigg(&\sum \mathbf{b}_j'\mathbf{b}_j-2N\left(\overline{\mathbf{b}}-\hat{\beta}\overline{\mathbf{a}}\right)'\overline{\mathbf{b}}-2\hat{\beta}\sum \mathbf{a}_j'\mathbf{b}_j+2N\hat{\beta}\left(\overline{\mathbf{b}}-\hat{\beta}\overline{\mathbf{a}}\right)'\overline{\mathbf{a}}\\ &+N\left(\overline{\mathbf{b}}-\hat{\beta}\overline{\mathbf{a}}\right)'\left(\overline{\mathbf{b}}-\hat{\beta}\overline{\mathbf{a}}\right)+\hat{\beta}^2\sum \mathbf{a}_j'\mathbf{a}_j\Bigg)\end{aligned}$$

$$=\frac{1}{\lambda+\hat{\beta}^2}\left(\left[\sum \mathbf{b}_j'\mathbf{b}_j-N\overline{\mathbf{b}}'\overline{\mathbf{b}}\right]-2\hat{\beta}\left[\sum \mathbf{a}_j'\mathbf{b}_j-N\overline{\mathbf{a}}'\overline{\mathbf{b}}\right]+\hat{\beta}^2\left[\sum \mathbf{a}_j'\mathbf{a}_j-N\overline{\mathbf{a}}'\overline{\mathbf{a}}\right]\right)$$

$$=\frac{S_{\mathbf{bb}}-2\hat{\beta}S_{\mathbf{ab}}+\hat{\beta}^2S_{\mathbf{aa}}}{\lambda+\hat{\beta}^2}$$

Only the case $\lambda=1$ that is when $\boldsymbol{\Omega}_{11}=\boldsymbol{\Omega}_{22}$ is considered. For those cases when $\lambda\neq1$, it can always be reduced to the case of $\lambda=1$ as the ULFR illustrated in Kendall and Stuart (1979). Hence,

$$SS_E=\frac{S_{\mathbf{bb}}-2\hat{\beta}S_{\mathbf{ab}}+\hat{\beta}^2S_{\mathbf{aa}}}{1+\hat{\beta}^2} \tag{4.18}$$

Then, the COD can be defined as

$$R_p^2=\frac{SS_R}{S_{\mathbf{bb}}}=1-\frac{SS_E}{S_{\mathbf{bb}}}=\frac{S_{\mathbf{bb}}-SS_E}{S_{\mathbf{bb}}} \tag{4.19}$$

For the case $\lambda=1$, Equation (4.19) becomes

$$R_p^2=\frac{\hat{\beta}S_{\mathbf{ab}}}{S_{\mathbf{bb}}} \tag{4.20}$$

Proof: $\dfrac{SS_R}{S_{\mathbf{bb}}} = \dfrac{\hat{\beta} S_{\mathbf{ab}}}{S_{\mathbf{bb}}} \Leftrightarrow SS_R = \hat{\beta} S_{\mathbf{ab}}$ is needed to be shown.

By definition, $SS_R = S_{\mathbf{bb}} - SS_E$

$$= S_{\mathbf{bb}} - \left(\frac{S_{\mathbf{bb}} - 2\hat{\beta} S_{\mathbf{ab}} + \hat{\beta}^2 S_{\mathbf{aa}}}{1+\hat{\beta}^2} \right)$$

$$= \frac{\left(S_{\mathbf{bb}} + \hat{\beta}^2 S_{\mathbf{bb}}\right) - \left(S_{\mathbf{bb}} - 2\hat{\beta} S_{\mathbf{ab}} + \hat{\beta}^2 S_{\mathbf{aa}}\right)}{1+\hat{\beta}^2}$$

$$= \frac{\hat{\beta}^2 S_{\mathbf{bb}} + 2\hat{\beta} S_{\mathbf{ab}} - \hat{\beta}^2 S_{\mathbf{aa}}}{1+\hat{\beta}^2}$$

$$= \frac{\hat{\beta}^2 \left(S_{\mathbf{bb}} - S_{\mathbf{aa}}\right) + 2\hat{\beta} S_{\mathbf{ab}}}{1+\hat{\beta}^2} \tag{4.21}$$

From Equation (4.14) and $\lambda = 1$,

$$S_{\mathbf{ab}} \hat{\beta}^2 = \left(S_{\mathbf{bb}} - S_{\mathbf{aa}}\right)\hat{\beta} + S_{\mathbf{ab}} \tag{4.22}$$

Substitute Equation (4.22) into Equation (4.21) yields

$$SS_R = \frac{\hat{\beta}\left\{\left[\left(S_{\mathbf{bb}} - S_{\mathbf{aa}}\right)\hat{\beta} + S_{\mathbf{ab}}\right] + S_{\mathbf{ab}}\right\}}{1+\hat{\beta}^2}$$

$$= \frac{\hat{\beta}\left\{\hat{\beta}^2 S_{\mathbf{ab}} + S_{\mathbf{ab}}\right\}}{1+\hat{\beta}^2}$$

$$= \frac{\hat{\beta} S_{\mathbf{ab}} \left(\hat{\beta}^2 + 1\right)}{1+\hat{\beta}^2}$$

$$= \hat{\beta} S_{\mathbf{ab}}$$

4.2.4 Properties of Coefficient of Determination when $\lambda = 1$

Some properties of the coefficient of determination (COD) of 2D-ULFR model, say R_p^2 when $\lambda = 1$ are described in the following (Chang *et al.*, 2009).

Property 4.1: Boundedness

To date, there is no standard range for a similarity or dissimilarity measure between characters. For example, peak signal to noise ratio (PSNR) (Glenn, 1996) is defined on $[0, \infty)$. Whereas, Wang *et al.* (2004) defined a good similarity measure on the range $(-\infty, 1]$, but Van de Weken *et al.* (2002) defined it on $[0,1]$. Among others, $[0,1]$ is the most commonly and well accepted dynamic range for a good quality measure. The proposed measure R_p^2 is also defined on $[0,1]$ and this can be easily shown from the regression sum of squares

$$0 \le SS_R = S_{\mathbf{bb}} - SS_E \le S_{\mathbf{bb}}$$

$$0 \le \frac{SS_R}{S_{\mathbf{bb}}} \le \frac{S_{\mathbf{bb}}}{S_{\mathbf{bb}}} = 1$$

$$\therefore 0 \le R_p^2 \le 1 \qquad (4.22)$$

Property 4.2: Reflexive

Note from Equations (4.14) and (4.19) that when the input character and character in database are identical or $\mathbf{a} = \mathbf{b}$, this implies $R_p^2 = 1$. On the other hand, $R_p^2 = 0$ indicates that the two characters are dislike. Hence, it is undoubted that R_p^2 decreases when the degree of similarity between the two characters decreases. The examples of R_p^2 values between some input characters and characters in database are shown in Appendix A2.

Property 4.3: Non-symmetry

Given $S_{\mathbf{bb}} = kS_{\mathbf{aa}}$ where $k > 0$. Let R_p^2 and $\tilde{R}_p^2$ be the coefficient of determination for 2D-ULFR model with $\lambda = 1$ as defined by $\mathbf{B}_j = \boldsymbol{\alpha} + \beta \mathbf{A}_j$ and $\mathbf{A}_j = \boldsymbol{\alpha}^* + \beta^* \mathbf{B}_j$, respectively. Then,

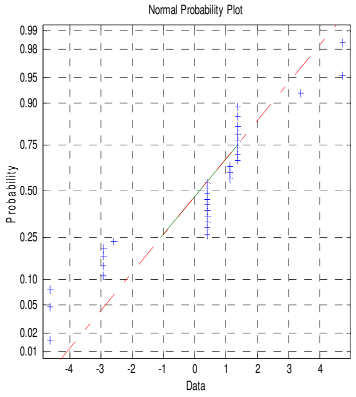
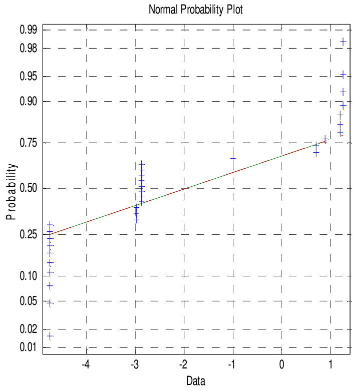
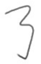
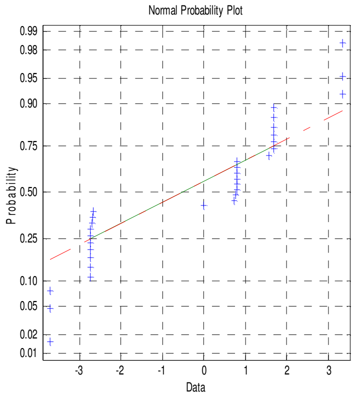
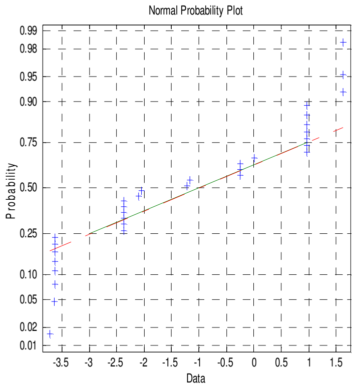
$$R_p^2 = \frac{1}{k}\left(\tilde{R}_p^2 - 1\right) + 1 \qquad (4.23)$$

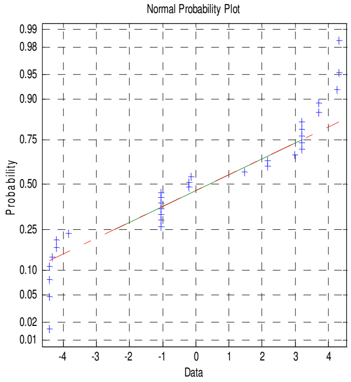
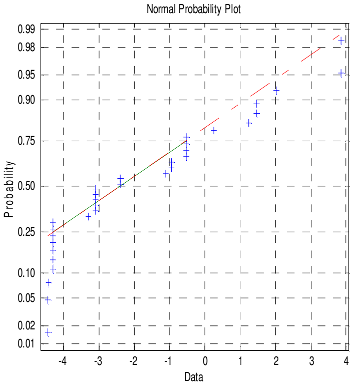
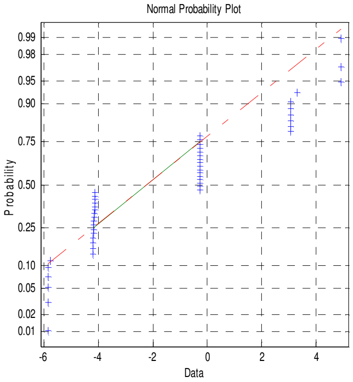
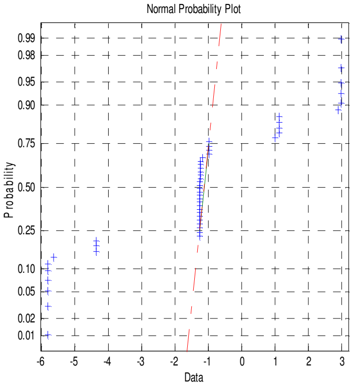
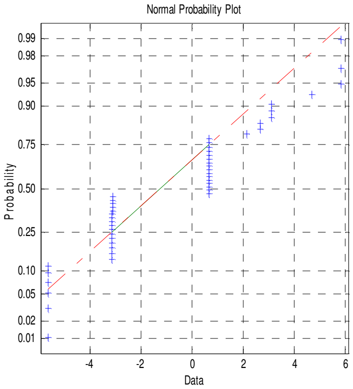
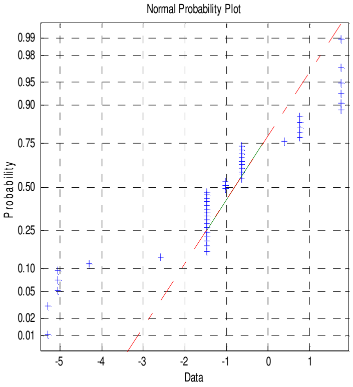
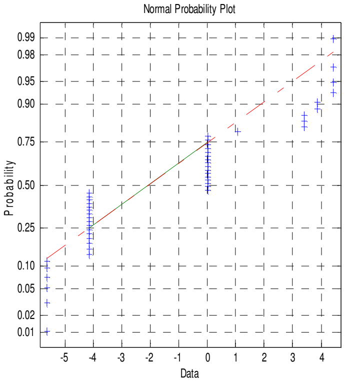
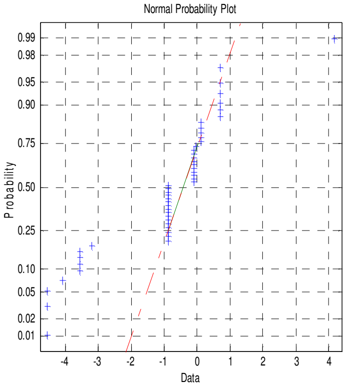
This indicates that R_p^2 is non-symmetry when $k \neq 1$. The proof of this property is provided in Chang *et al.* (2009). In order to solve this problem, the character in database or the input character, whichever has larger variance is assumed to be $\mathbf{A}$, and the other character be $\mathbf{B}$. With this arrangement, the full range $0 \leq R_P^2 \leq 1$ will be granted. Otherwise, Equation (4.23) provides a simple way to convert $\tilde{R}_p^2$ to R_p^2.

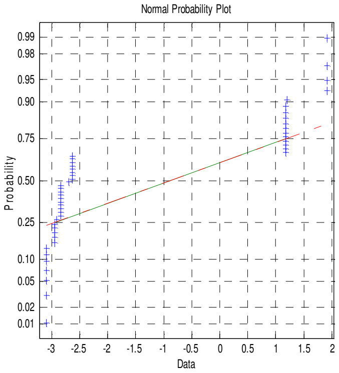
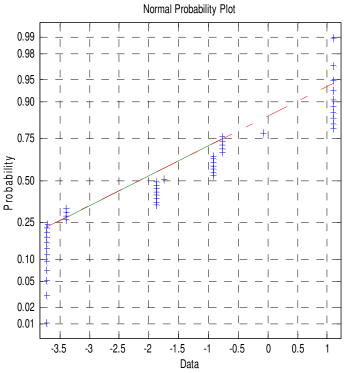
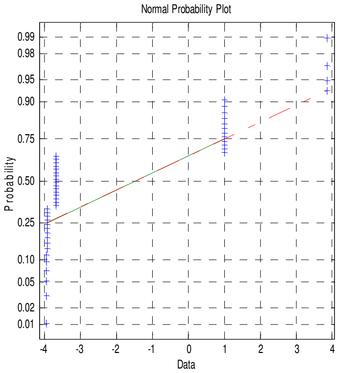
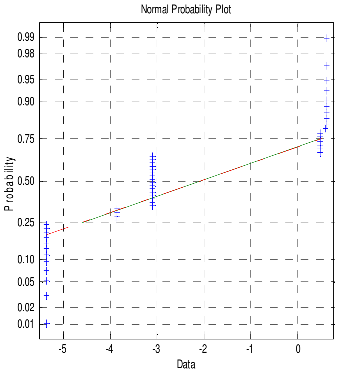
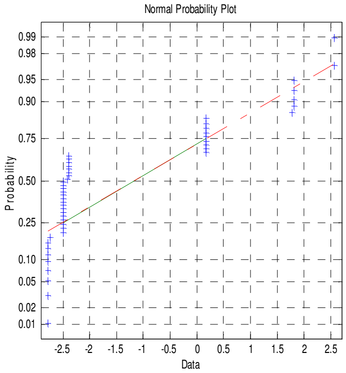
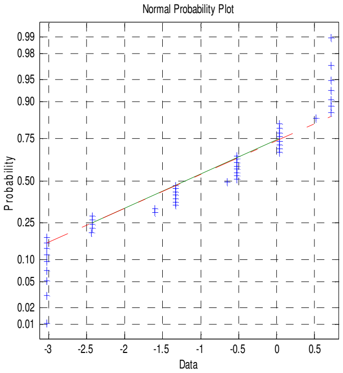
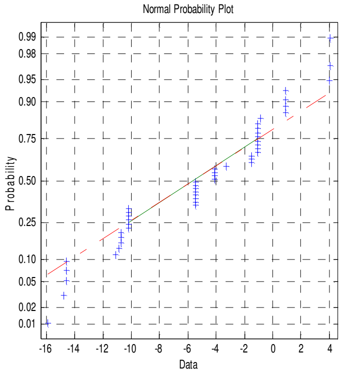
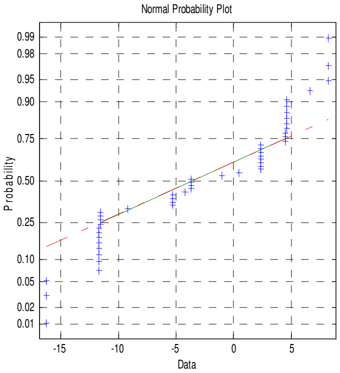
4.3 Validation of Normality Assumption

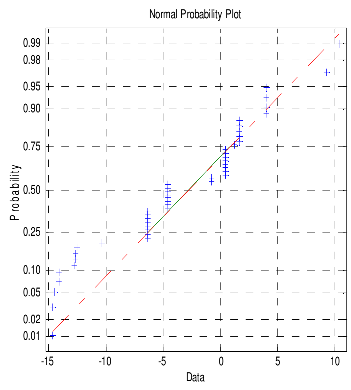
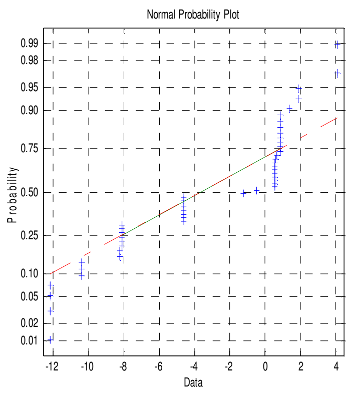
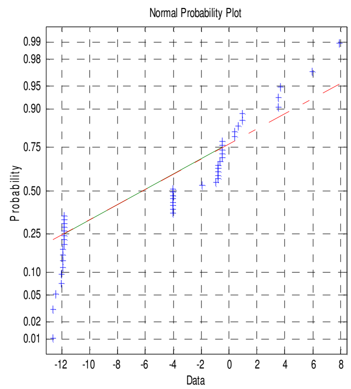
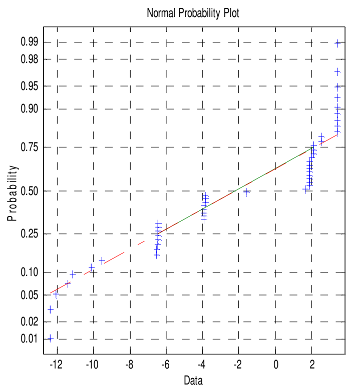
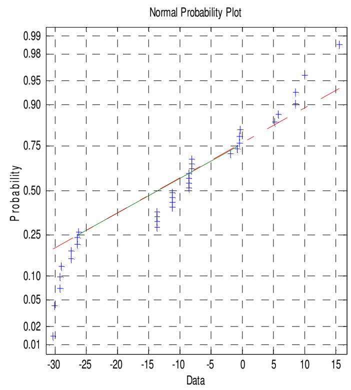
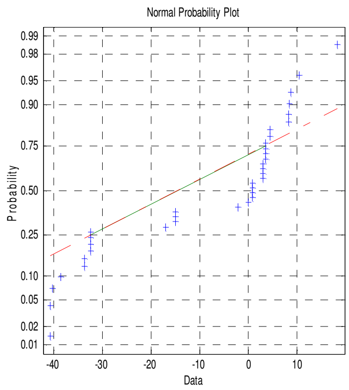
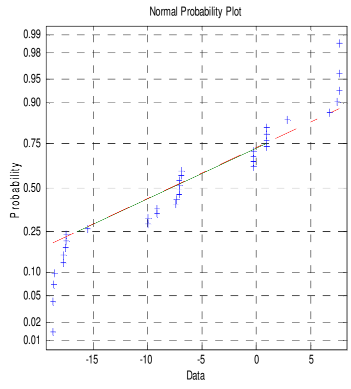
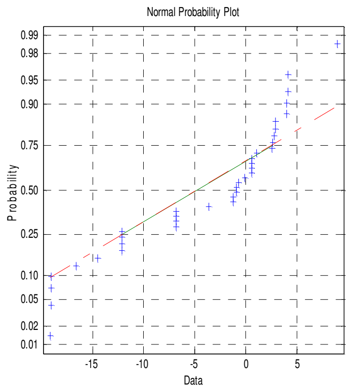
The 2D-ULFR model stated in Equations (4.2), (4.3) and (4.4) assumes both errors $\boldsymbol{\delta}$ and $\boldsymbol{\varepsilon}$ are normally distributed. In the application of the proposed HCCR system, detailed coefficients of Haar wavelet transform are considered as the error terms in 2D-ULFR model. In order to validate the proposed recognition system satisfy normality assumption of the model, normal probability plot of some Chinese characters, each with three different writing styles are generated and demonstrated in Table 4.2. The plots on the left are for the detailed coefficients of *X*-graphs while right shows the plots for detailed coefficients of *Y*-graphs.

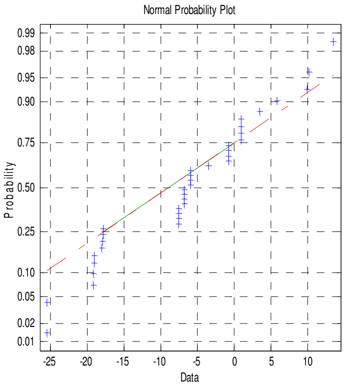
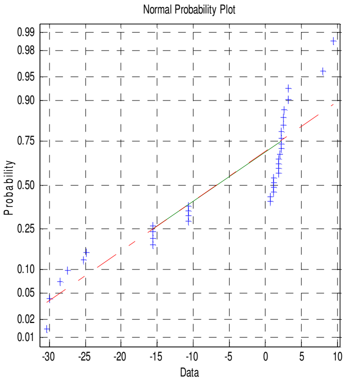
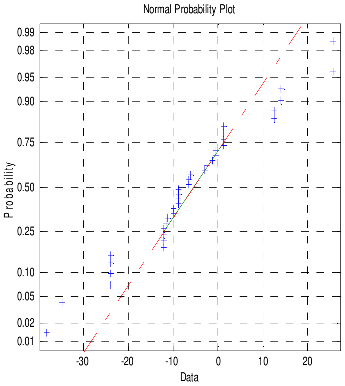
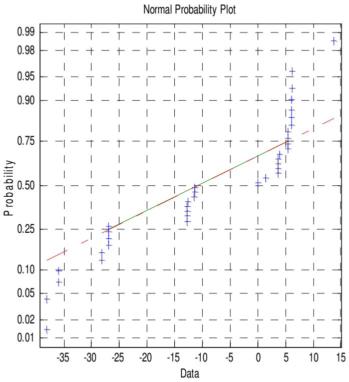
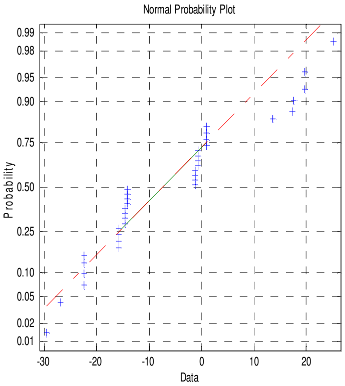
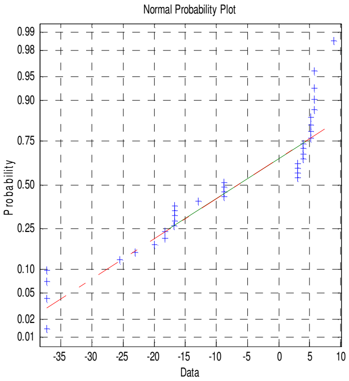
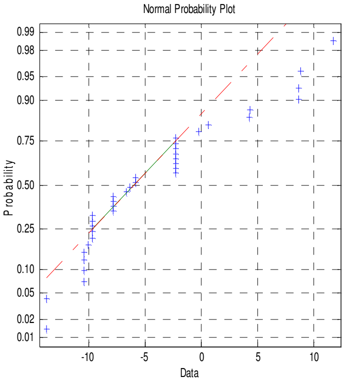
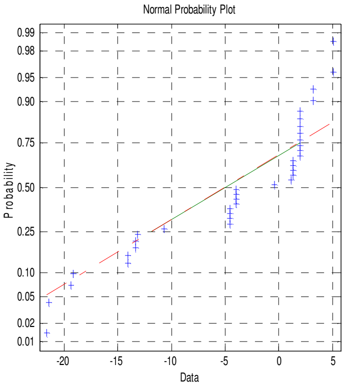
Table 4.2: Normal probability plots of Chinese character '了', '之', '大', '军', '厘', '信', '魏', '陵' and '薛. Each character is of three different writing styles.

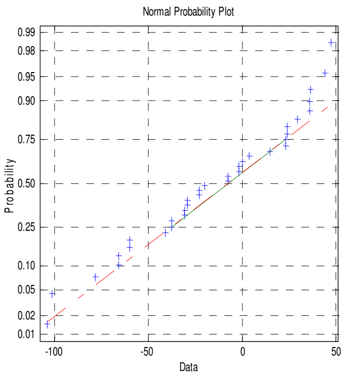
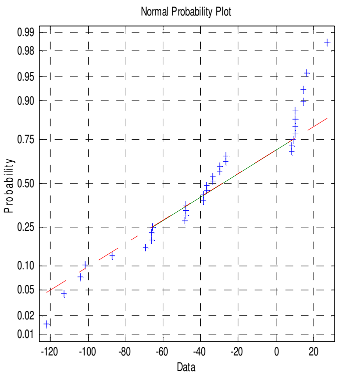
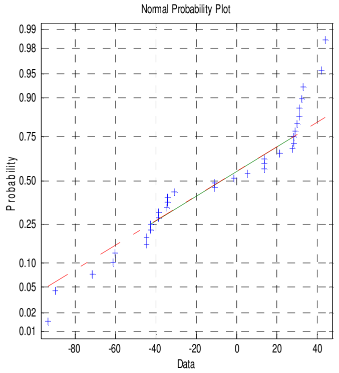
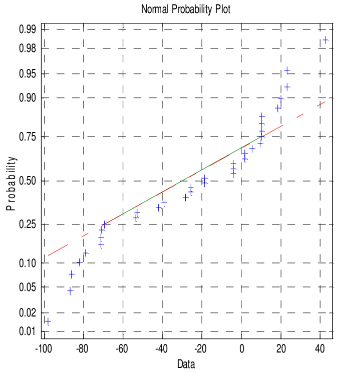
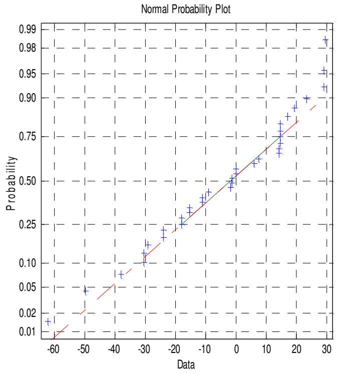
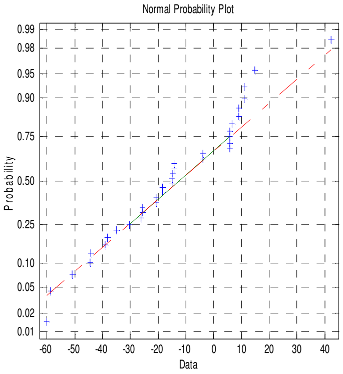
了		
		
		

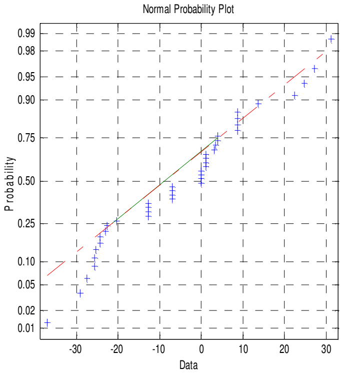
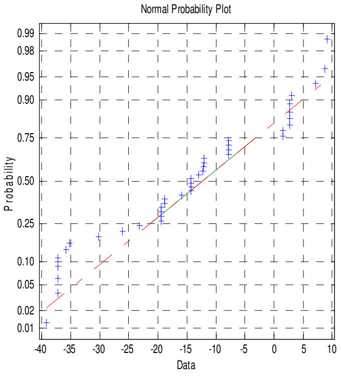
了	Normal Probability Plot 0.99 0.98 0.95 0.90 0.75 0.50 0.25 0.10 0.05 0.02 0.01 Probability -4 -3 -2 -1 0 1 2 3 4 Data	Normal Probability Plot 0.99 0.98 0.95 0.90 0.75 0.50 0.25 0.10 0.05 0.02 0.01 Probability -4 -3 -2 -1 0 1 2 3 4 Data
之		
之	Normal Probability Plot 0.99 0.98 0.95 0.90 0.75 0.50 0.25 0.10 0.05 0.02 0.01 Probability -6 -4 -2 0 2 4 Data	Normal Probability Plot 0.99 0.98 0.95 0.90 0.75 0.50 0.25 0.10 0.05 0.02 0.01 Probability -6 -5 -4 -3 -2 -1 0 1 2 3 Data
之	Normal Probability Plot 0.99 0.98 0.95 0.90 0.75 0.50 0.25 0.10 0.05 0.02 0.01 Probability -4 -2 0 2 4 6 Data	Normal Probability Plot 0.99 0.98 0.95 0.90 0.75 0.50 0.25 0.10 0.05 0.02 0.01 Probability -5 -4 -3 -2 -1 0 1 Data
之	Normal Probability Plot 0.99 0.98 0.95 0.90 0.75 0.50 0.25 0.10 0.05 0.02 0.01 Probability -5 -4 -3 -2 -1 0 1 2 3 4 Data	Normal Probability Plot 0.99 0.98 0.95 0.90 0.75 0.50 0.25 0.10 0.05 0.02 0.01 Probability -4 -3 -2 -1 0 1 2 3 4 Data

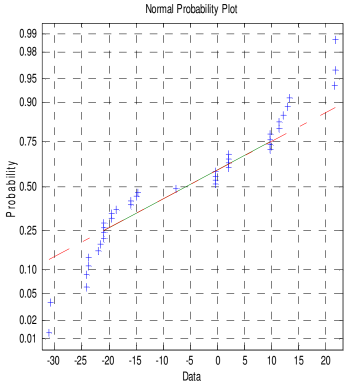
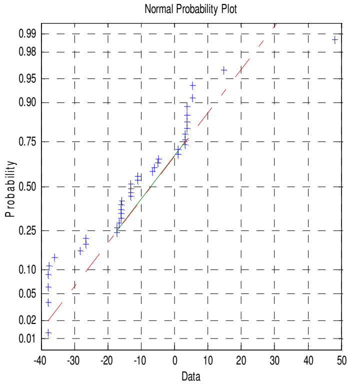
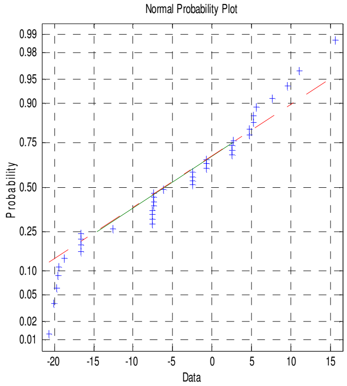
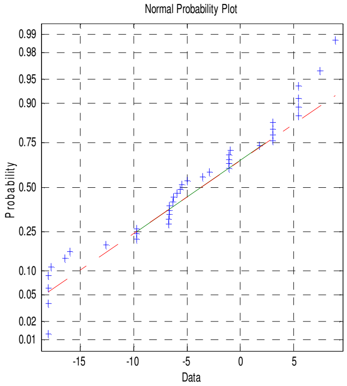
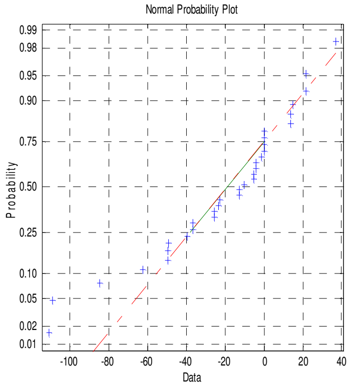
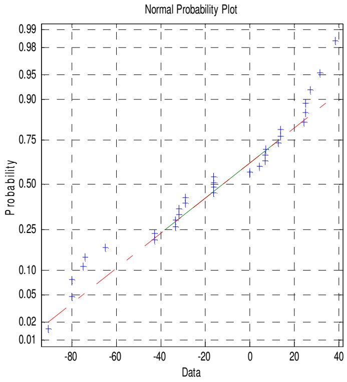
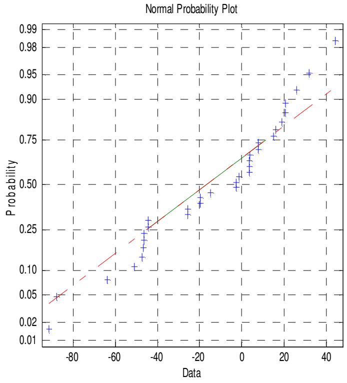
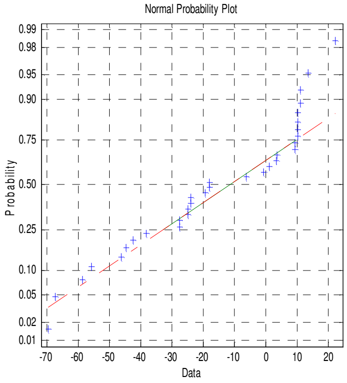
<table>
<tr><th colspan="3">大</th></tr>
<tr><td>大</td><td>
Normal Probability Plot
Probability
0.99
0.98
0.95
0.90
0.75
0.50
0.25
0.10
0.05
0.02
0.01
-3
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
Data</td><td>
Normal Probability Plot
Probability
0.99
0.98
0.95
0.90
0.75
0.50
0.25
0.10
0.05
0.02
0.01
-3.5
-3
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
Data</td></tr>
<tr><td>大</td><td>
Normal Probability Plot
Probability
0.99
0.98
0.95
0.90
0.75
0.50
0.25
0.10
0.05
0.02
0.01
-4
-3
-2
-1
0
1
2
3
4
Data</td><td>
Normal Probability Plot
Probability
0.99
0.98
0.95
0.90
0.75
0.50
0.25
0.10
0.05
0.02
0.01
-5
-4
-3
-2
-1
0
Data</td></tr>
<tr><td>大</td><td>
Normal Probability Plot
Probability
0.99
0.98
0.95
0.90
0.75
0.50
0.25
0.10
0.05
0.02
0.01
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
Data</td><td>
Normal Probability Plot
Probability
0.99
0.98
0.95
0.90
0.75
0.50
0.25
0.10
0.05
0.02
0.01
-3
-2.5
-2
-1.5
-1
-0.5
0
0.5
Data</td></tr>
<tr><th colspan="3">军</th></tr>
<tr><td>军</td><td>
Normal Probability Plot
Probability
0.99
0.98
0.95
0.90
0.75
0.50
0.25
0.10
0.05
0.02
0.01
-16
-14
-12
-10
-8
-6
-4
-2
0
2
4
Data</td><td>
Normal Probability Plot
Probability
0.99
0.98
0.95
0.90
0.75
0.50
0.25
0.10
0.05
0.02
0.01
-15
-10
-5
0
5
Data</td></tr>
</table>

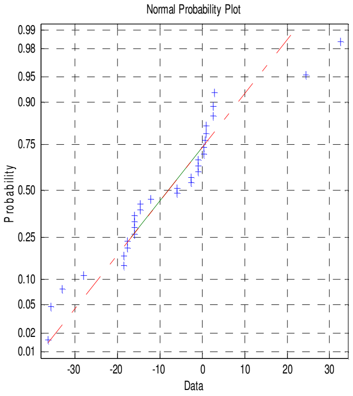
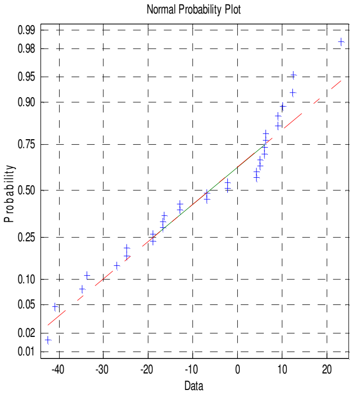
军	Normal Probability Plot Probability Data	Normal Probability Plot Probability Data
军	Normal Probability Plot Probability Data	Normal Probability Plot Probability Data
	厘	
厘	Normal Probability Plot Probability Data	Normal Probability Plot Probability Data
厘	Normal Probability Plot Probability Data	Normal Probability Plot Probability Data

厘	Normal Probability Plot 0.99 0.98 0.95 0.90 0.75 0.50 0.25 0.10 0.05 0.02 0.01 Probability -25 -20 -15 -10 -5 0 5 10 Data	Normal Probability Plot 0.99 0.98 0.95 0.90 0.75 0.50 0.25 0.10 0.05 0.02 0.01 Probability -30 -25 -20 -15 -10 -5 0 5 10 Data
	信	
信	Normal Probability Plot 0.99 0.98 0.95 0.90 0.75 0.50 0.25 0.10 0.05 0.02 0.01 Probability -30 -20 -10 0 10 20 Data	Normal Probability Plot 0.99 0.98 0.95 0.90 0.75 0.50 0.25 0.10 0.05 0.02 0.01 Probability -35 -30 -25 -20 -15 -10 -5 0 5 10 15 Data
信	Normal Probability Plot 0.99 0.98 0.95 0.90 0.75 0.50 0.25 0.10 0.05 0.02 0.01 Probability -30 -20 -10 0 10 20 Data	Normal Probability Plot 0.99 0.98 0.95 0.90 0.75 0.50 0.25 0.10 0.05 0.02 0.01 Probability -35 -30 -25 -20 -15 -10 -5 0 5 10 Data
信	Normal Probability Plot 0.99 0.98 0.95 0.90 0.75 0.50 0.25 0.10 0.05 0.02 0.01 Probability -10 -5 0 5 10 Data	Normal Probability Plot 0.99 0.98 0.95 0.90 0.75 0.50 0.25 0.10 0.05 0.02 0.01 Probability -20 -15 -10 -5 0 5 Data

魏		
魏	Normal Probability Plot Probability 0.99 0.98 0.95 0.90 0.75 0.50 0.25 0.10 0.05 0.02 0.01 -100 -50 0 50 Data	Normal Probability Plot Probability 0.99 0.98 0.95 0.90 0.75 0.50 0.25 0.10 0.05 0.02 0.01 -120 -100 -80 -60 -40 -20 0 20 Data
魏	Normal Probability Plot Probability 0.99 0.98 0.95 0.90 0.75 0.50 0.25 0.10 0.05 0.02 0.01 -80 -60 -40 -20 0 20 40 Data	Normal Probability Plot Probability 0.99 0.98 0.95 0.90 0.75 0.50 0.25 0.10 0.05 0.02 0.01 -100 -80 -60 -40 -20 0 20 40 Data
魏	Normal Probability Plot Probability 0.99 0.98 0.95 0.90 0.75 0.50 0.25 0.10 0.05 0.02 0.01 -60 -50 -40 -30 -20 -10 0 10 20 30 Data	Normal Probability Plot Probability 0.99 0.98 0.95 0.90 0.75 0.50 0.25 0.10 0.05 0.02 0.01 -60 -50 -40 -30 -20 -10 0 10 20 30 40 Data

陵		
陵	Normal Probability Plot Probability 0.99 0.98 0.95 0.90 0.75 0.50 0.25 0.10 0.05 0.02 0.01 -30 -20 -10 0 10 20 30 Data	Normal Probability Plot Probability 0.99 0.98 0.95 0.90 0.75 0.50 0.25 0.10 0.05 0.02 0.01 -40 -35 -30 -25 -20 -15 -10 -5 0 5 10 Data

陵	 Normal Probability Plot Probability Data 0.99 0.98 0.95 0.90 0.75 0.50 0.25 0.10 0.05 0.02 0.01 -30 -25 -20 -15 -10 -5 0 5 10 15 20	 Normal Probability Plot Probability Data 0.99 0.98 0.95 0.90 0.75 0.50 0.25 0.10 0.05 0.02 0.01 -40 -30 -20 -10 0 10 20 30 40 50
陵	 Normal Probability Plot Probability Data 0.99 0.98 0.95 0.90 0.75 0.50 0.25 0.10 0.05 0.02 0.01 -20 -15 -10 -5 0 5 10 15	 Normal Probability Plot Probability Data 0.99 0.98 0.95 0.90 0.75 0.50 0.25 0.10 0.05 0.02 0.01 -15 -10 -5 0 5
	薛	
薛	 Normal Probability Plot Probability Data 0.99 0.98 0.95 0.90 0.75 0.50 0.25 0.10 0.05 0.02 0.01 -100 -80 -60 -40 -20 0 20 40	 Normal Probability Plot Probability Data 0.99 0.98 0.95 0.90 0.75 0.50 0.25 0.10 0.05 0.02 0.01 -80 -60 -40 -20 0 20 40
薛	 Normal Probability Plot Probability Data 0.99 0.98 0.95 0.90 0.75 0.50 0.25 0.10 0.05 0.02 0.01 -80 -60 -40 -20 0 20 40	 Normal Probability Plot Probability Data 0.99 0.98 0.95 0.90 0.75 0.50 0.25 0.10 0.05 0.02 0.01 -70 -60 -50 -40 -30 -20 -10 0 10 20

In general, the normal probability plot can be explained by the following algorithm. Assume that $\mathbf{e}=\left[e_1,e_2,\ldots,e_n\right]$ be an n-dimensional error.

Step 1: Sort the elements of the error in ascending order. Denote the sorted error to be $\mathbf{e}_{sorted}=\left[e_{[1]},e_{[2]},\ldots,e_{[n]}\right]$, where $e_{[1]}\le e_{[2]}\le\ldots\le e_{[n]}$.

Step 2: Determine the percentile P_i for $\forall e_{[i]},\ 1\le i\le n$ as defined as below.

$$P_i=\frac{i-0.5}{n}\times 100 \qquad (4.24)$$

Step 3: Generate the normal probability plot such that P_i versus $e_{[i]},\ 1\le i\le n$.

In a normal probability plot, if all the data points fall near the line, an assumption of normality is reasonable. However, it can be observed from Table 4.2 that the points fall away from the line for some Chinese characters. Hence, this indicates that the detailed coefficients for some of the characters do not follow normal distribution and this has violated the assumption of the model. To solve this problem, a transformation of a non-normal distributed variable to a normal distribution can be performed. Power transformation, or known as Box-Cox transformation introduced by Box and Cox (1964), and simple quadratic method used by Pooi (2003) can be applied to transform the data to normal distribution.

Box-Cox transformation is a transformation which might be used to convert a general set of k observations into a set of k independent observations from a normal distribution with constant variance. As stated in original paper (Box and Cox, 1964), the transformed observation, y^*, is related to the original observation, y, by

$$y^* = \begin{cases} \dfrac{y^\lambda - 1}{\lambda}, & \text{if } \lambda \neq 0; \\ \log\ y, & \text{if } \lambda = 0. \end{cases} \tag{4.25}$$

This transformation involves a parameter λ. To estimate the value of λ, Box and Cox (1964) considered two approaches which are maximum likelihood estimation (MLE) and Bayesian method. MLE is commonly used since it is conceptually easy and the likelihood function is easy to compute in this case. On the other hand, in the Bayesian approach, it must make sure that the model is fully identifiable, i.e. all the variables can be estimated.

For simple quadratic method (Pooi, 2003), it is capable of generating a wide range of unimodal distributions which are skewed or having thin waists with known skewness and kurtosis values. Let e be a random error with the standard normal distribution and $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \gamma_3)$ a vector of constant values. Consider the following non-linear function of e:

$$\varepsilon = \begin{cases} \gamma_1 e + \gamma_2 \left(e^2 - \dfrac{1+\gamma_3}{2} \right) & , e \geq 0 \\ \gamma_1 e + \gamma_2 \left(\gamma_3 e^2 - \dfrac{1+\gamma_3}{2} \right) & , e < 0 \end{cases} \qquad (4.26)$$

where the constants $\gamma_t;\ t = 1,2,3$ are such that for small value $q > 0$, ε is a one-to-one function of e when $|e|$ is within the $100(1-q)\%$ point Z_q of the standard normal distribution. The reverse of Equation (4.26) provides a useful and simple method for the proposed model to handle the non-normality problem in character recognition applications when the error terms are known. For example, one can reverse the transformation from Laplacian distribution to normal distribution by choosing $\gamma = (0.015819, 0.568667, -1.000010)$ before the 2D-ULFR model is applied (see Figure 4.1).

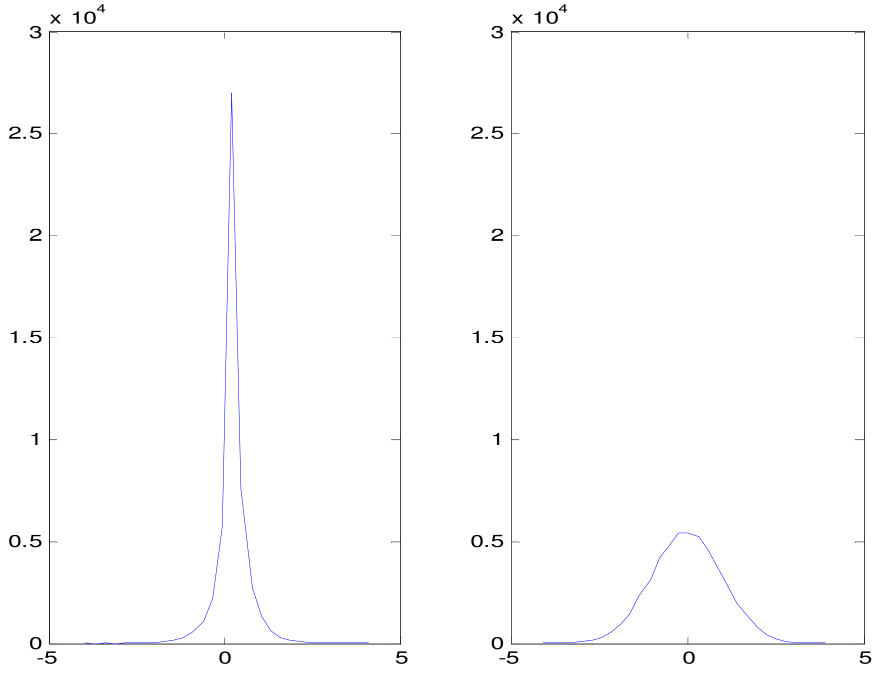


Figure 4.1: Example of transformation from Laplace distribution (left) to Normal distribution (right) using the reverse of Equation (4.26).

4.4 Summary

This chapter mainly illustrates the classification process of the HCCR system in this research. The classification process is divided into two sub-stages which are coarse classification and fine classification, as to accelerate the speed of the recognition system. It begins with the selection of a small subset of relevant candidates from the whole database by coarse classification, such that these candidates have the same number of stroke with that of input character. The approximation coefficients of these relevant candidates which are obtained from Haar wavelet transform will be used in fine classification stage.

For fine classification, R_p^2 (i.e. the COD of 2D-ULFR model) is the new proposed classifier. It is applied as a similarity measure with three important properties: (i) boundedness, (ii) reflexive and (iii) non-symmetry. This is the first attempt in making R_p^2 as a classifier in character recognition. The principle of the proposed method makes R_p^2 very robust against size and position variation as well as character shape deformation, even without undergoing normalization. Although the detailed coefficients of the Haar wavelet transform which act as the errors terms in 2D-ULFR model do not fulfill the normality assumption of the 2D-ULFR model, the results are not affected. It is because the non-normal distributed variable can always be transformed to a normal distribution by using Box-Cox transformation or simple quadratic method before the 2D-ULFR model is applied. The efficiency of this proposed method can be studied from the experimental results shown in next chapter.

CHAPTER 5

APPLICATION IN HANDWRITTEN CHINESE CHARACTER RECOGNITION SYSTEM

The feature extraction method and classifier proposed in Chapter 3 and Chapter 4 respectively will be implemented in the new proposed HCCR system and its performance will be evaluated in this chapter. Section 5.1 describes the experimental setup of this research which involves creation of new database and collection of testing samples. Next, it is followed by the experimental results obtained by comparing the previously and new proposed feature extraction methods and classifiers. The experimental results will be analyzed from three aspects, which are recognition rate, processing time and storage space in Section 5.3, 5.4 and 5.5 respectively. Finally, this chapter is summarized in Section 5.6.

5.1 Experimental Setup

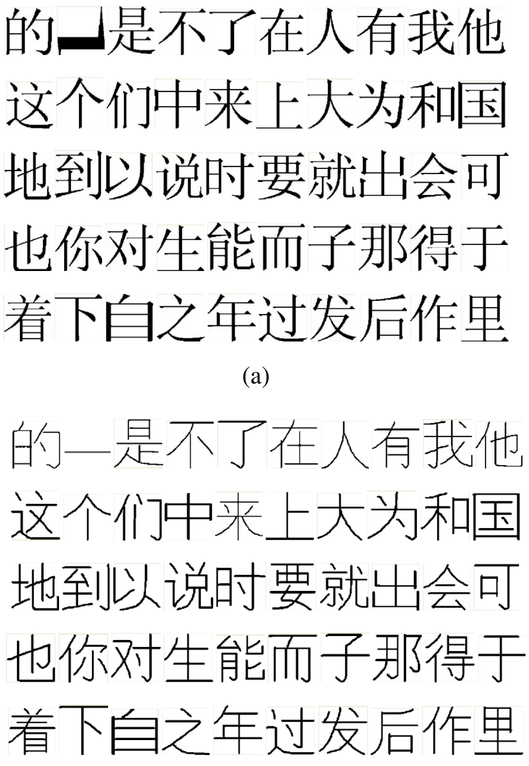
The experiments for evaluating the efficiency of the new proposed feature extraction and classification method start with the setup of database. HCL2000, ETL9B and CASIA are the most commonly used database in Chinese character recognition. Their details are illustrated in Table 5.1. Note that the number of classes is referred to the number of different characters; whereas the number of samples per class is referred to the samples collected from different writers for each character (or class). HCL2000 database has been used in Long and Jin (2008), Liu and Ding (2005); ETL9B database is applied in Dong *et al*. (2005), Gao and Liu (2008); CASIA database is used in Gao and Liu (2008). From Table 5.1, it can be noticed that the existing databases used to store many samples of different writing styles for each character, in order to cope with the problem of handwriting variation from different writers. Obviously, this technique seems to be not very efficient for some classifiers which keep all the training samples to do the classification such as *k*-NN since it occupied extremely large storage space. Different from those existing databases, only a single sample is needed to be stored for each character in the database of the recognition system in this research. It is because no training process is needed here. Besides, the proposed methods are able to handle successfully the characters with various writing styles, even the characters with severe deformed shape. Therefore, the memory space can be saved significantly since the storage of excessive samples is unnecessary. This is more practical for the situation where training samples are kept for classification but only limited storage space is allowed.

Table 5.1: Three commonly used database for Chinese character recognition and a new created database for this research.

Database	Origin	Number of classes	Number of samples per class
HCL2000	Collected by Beijing University of Posts and Telecommunications for China 863 project.	3755	1000
ETL9B	Collected by Electro-Technical Laboratory (ETL) of Japan	2965	200
CASIA	Collected by Institute of Automation of Chinese Academy of Sciences	3755	300
CL2009	Based on Ju Dan's modern Chinese character frequency list (Dan, 2004)	3000	1

In this research, instead of using the existing databases, a new database with only a single sample for each character will be created, namely CL2009.

This new created database is based on Ju Dan's modern Chinese character frequency list (Dan, 2004) which is generated from a large corpus of Chinese texts collected from online sources. The 3000 most frequently used simplified Chinese characters, i.e. the first 3000 characters in Ju Dan's modern Chinese character frequency list will be the characters in the new created database (refer to Appendix A3). Firstly, these 3000 Chinese characters are written by a particular writer in *songti*, due to its widely used and similarity to most of the handwritten Chinese characters. Then, these characters are stored in database after being cropped and normalized to the size of 128×128, as explained in Section 3.1. Examples of 50 Chinese characters in *songti* typed style and *songti* written style are demonstrated in Figure 5.1. The *songti* written style Chinese characters are the characters stored in CL2009 database.

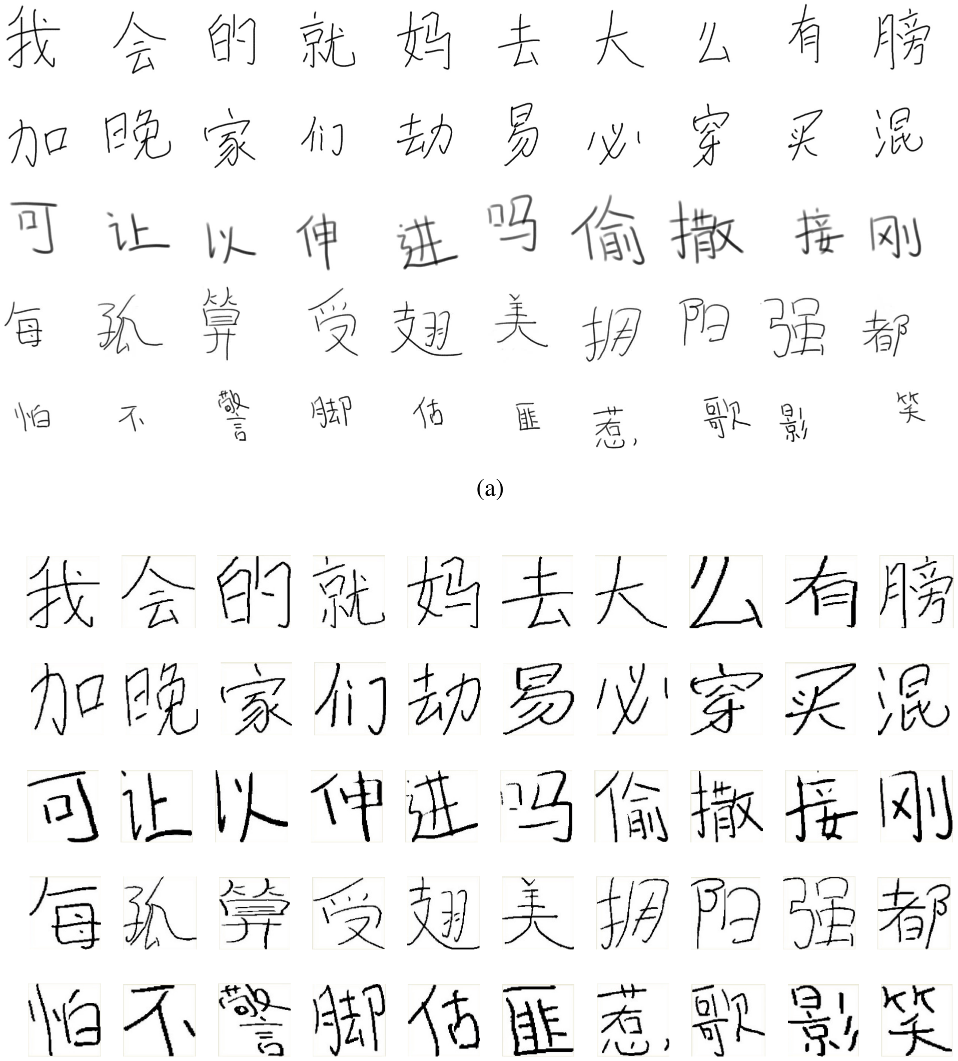


(a)

(b)

Figure 5.1: Examples of 50 normalized Chinese characters in (a) *songti* typed style and (b) *songti* written style.

For testing, the samples of Chinese character are collected from 10 different writers, 100 character samples from each writer (i.e. there are 100×10 testing samples in total). These testing samples are selected from the short Chinese passages and some examples of these testing samples are demonstrated in Figure 5.2. The writers are requested to write with stroke number and stroke order restriction.



(a)

(b)

Figure 5.2: Examples of 50 testing samples: (a) without normalization process and (b) with normalization process.

5.2 Experimental Results

In order to evaluate the performance of the proposed feature extraction method and the new designed classifier, experiment is carried out. It is done for two cases: (i) with normalization and (ii) without normalization process. The efficiency of the proposed recognition system will be judged from the three different perspectives: (i) recognition rate, (ii) processing time and (iii) storage space, which are illustrated in the following sections.

5.3 Recognition Rate

Recognition rate is the measurement of accuracy for recognizing or matching the input character with character in database correctly. It is also known as accuracy rate. In order to validate the new designed recognition system, accuracy is the major element that has to be concerned about. For comparison purpose, city block distance with deviation (CBDD), minimum distance (MD), modified quadratic discriminant function (MQDF) and compound Mahalanobis function (CMF) classifier (refer to Section 2.4.5(II), 2.4.5(I), 2.4.1(I) and 2.4.3(I) respectively) are selected. They are applied to the proposed recognition system and using the same database i.e. CL2009. Some parameters used in these classifier need to be determined in advance. Various values are examined and the optimum values are as follows.

(I) ***CBDD***: (a) $\theta = 1.2$, as stated in Kato *et al.* (1999).

(II) ***MQDF***: (a) $K = 1$ since there is only one sample for each character in database and hence it results in only one dominant eigenvalues.

(b) $\delta_i = 1.3641$, such that it is made class-independent and equals to the average of all eigenvalues of all classes. Note that as stated in Long and Jin (2008), the performance of classifier is superior when setting the constant class-independent rather than class-dependent.

(III) ***CMF***: (a) $p, K = 1$(refer to Section 5.3(II)).

(b) $Q = 1.3641$, which is average of all eigenvalues of all classes.

(c) $\mu = 2.8$, as stated in Suzuki *et al.* (1997).

The recognition rate for CBDD, MD, MQDF, CMF classifier and the proposed classifier (R_p^2) is studied in Table 5.2 and 5.3. Moreover, both these tables also show the standard deviation of the recognition rate for the testing samples. The experimental results are analyzed from two aspects: (i) based on different writers, i.e. each writer will write 100 different characters (refer to Table 5.2) and (ii) based on different characters, i.e. each character is written by 10 different writers (refer to Table 5.3). These two results imply the discrimination power of the recognition system from two perspectives. Table 5.2 analyzes how well the recognition system can recognize the variety of Chinese characters, from simple to complicated one; while Table 5.3 reveals how well it can handle the same characters with different writing styles which will lead to character size and position variation, as well as character shape deformation. The examples of 30 different characters from one of the writers and a character from 10 different writers are demonstrated in Figure 5.3 and Figure 5.4 respectively.

Table 5.2: Experimental results for 10 different writers: (a) with normalization and (b) without normalization.

(a)

Classifiers		CBDD	MD	MQDF	CMF	R_p^2
Recognition rate (%) with normalization	**Writer A**	96	97	97	97	100
	Writer B	98	98	100	99	100
	Writer C	98	99	98	97	97
	Writer D	98	99	98	99	97
	Writer E	92	96	96	95	99
	Writer F	92	91	93	93	91
	Writer G	90	94	94	93	95
	Writer H	97	99	100	99	100
	Writer I	97	96	97	96	98
	Writer J	98	100	100	99	100
Average		95.6	96.9	97.3	96.7	97.7
Standard Deviation		3.0623	2.7669	2.4518	2.4060	2.9078

(b)

Classifiers		CBDD	MD	MQDF	CMF	R_p^2
Recognition rate (%) without normalization	**Writer A**	6	27	89	72	100
	Writer B	8	23	80	69	100
	Writer C	3	13	72	43	96
	Writer D	8	20	74	65	99
	Writer E	8	22	79	68	99
	Writer F	9	18	64	41	92
	Writer G	5	13	35	20	97
	Writer H	0	17	64	51	100
	Writer I	0	17	53	50	99
	Writer J	39	50	66	61	100
Average		8.6	22	67.6	54	98.2
Standard Deviation		11.1774	10.7600	15.2985	16.2822	2.5734

Table 5.3: Experimental results for 20 different Chinese characters: (a) with normalization and (b) without normalization.

(a)

Classifiers		CBDD	MD	MQDF	CMF	R_p^2
Recognition rate (%) with normalization	**的**	100	100	100	100	100
	秦	100	100	100	100	100
	说	100	100	100	100	100
	我	100	100	100	100	100
	小	100	100	100	100	100
	有	100	100	100	100	100
	这	100	100	100	100	100
	魏	100	100	100	100	100
	劫	20	30	30	30	10
	了	100	100	100	100	100
	信	100	100	100	100	100
	国	100	100	100	100	100
	陵	100	100	100	100	100
	赵	80	90	100	90	90
	无	100	90	80	70	90
	是	100	100	100	100	100
	到	100	100	100	100	100
	来	70	100	100	90	100
	君	100	100	100	100	100
	军	100	100	100	100	100
Average		93.5	95.5	95.5	94	94.5
Standard Deviation		1.8994	1.5720	1.6051	1.6670	2.0125

(b)

Classifiers		CBDD	MD	MQDF	CMF	R_p^2
Recognition rate (%) without normalization	的	0	0	70	30	100
	秦	70	100	100	100	100
	说	0	0	70	60	100
	我	0	10	100	100	100
	小	0	0	100	100	100
	有	0	0	100	100	100
	这	0	0	50	50	100
	魏	0	100	100	100	100
	劫	0	0	40	20	10
	了	100	100	80	80	100
	信	10	20	30	20	100
	国	0	0	0	0	100
	陵	0	10	40	40	100
	赵	0	0	100	100	100
	无	0	0	0	0	90
	是	0	0	70	50	100
	到	0	0	70	80	100
	来	10	10	100	90	100
	君	0	10	80	80	100
	军	0	20	0	0	100
Average		9.5	19	65	60	95
Standard Deviation		2.6453	3.5526	3.5909	3.7836	2.0131

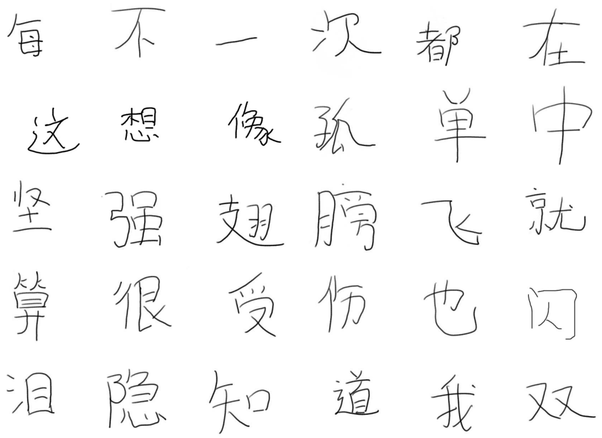


Figure 5.3: The sample of 30 Chinese characters written by one of the writers.

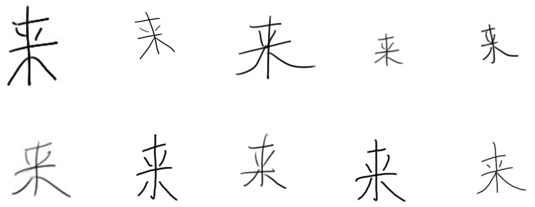


Figure 5.4: Chinese character '来' (means come) written by 10 different writers.

From the two tables above, it can be noticed that there is a huge difference between the recognition rate of the input characters with normalization and without normalization for CBDD, MD, MQDF and CMF classifier. For both the cross-writers approach and cross-characters approach, these four classifiers give satisfactory average recognition rates of above 93% in the case of with normalization. However, on the average, they result in extremely worse recognition rates such that only 8.6%, 22%, 67.6%, 54% (for cross-writers approach) and 9.5%, 19%, 65%, 60% (for cross-characters approach) are obtained for CBDD, MD, MQDF and CMF classifier respectively, when the input characters do not undergo normalization. On the other hand, by using R_p^2 classifier, the recognition rate of 100% is achieved for some writers in the case of with and without normalization as shown in Table 5.2. Although R_p^2 classifier results in the lowest recognition rate of 97%, 97% and 91% among all the classifiers for writer C, writer D and writer F in the normalization case, the results still remain promising with 96%, 99% and 92% respectively if compared to other classifiers which give terrible results in the case of without normalization. In the aspect of different characters as in Table 5.3, almost all the characters achieve 100 % recognition rate for R_p^2 classifier, except character '劫', '赵' and '无'. These characters are misclassified easily especially character '劫', such that only 10% recognition rates are obtained by R_p^2 classifier in both the case of with and without normalization. These misclassification cases will be discussed in Section 5.3.2. On the average, by using R_p^2 classifier, the recognition rates have increased 0.5% for input characters without normalization if compared to the results for normalized characters, i.e. from recognition rate of 97.7% to 98.2% and 94.5% to 95% for both cross-writers approach and cross-characters approach respectively.

Besides, the results of top-1, top-3 and top-5 recognized characters by using the five classifiers mentioned above are also analyzed for the case of with and without normalization (refer to Figure 5.5 and Figure 5.6). In the case of undergoing normalization, the CMF classifier attains an recognition rate of 99.8% and the four remaining classifiers result in 99.9% for the top-5 recognition result; whereas in the case of without normalization, the top-5 recognition result for R_p^2 classifier still remains unchanged, but only 18.7%, 55%, 90.3% and 73.7% are obtained by CBDD, MD, MQDF and CMF classifier respectively. In other words, many of the input characters are out of the top-5 recognized character list for the four existing classifiers.

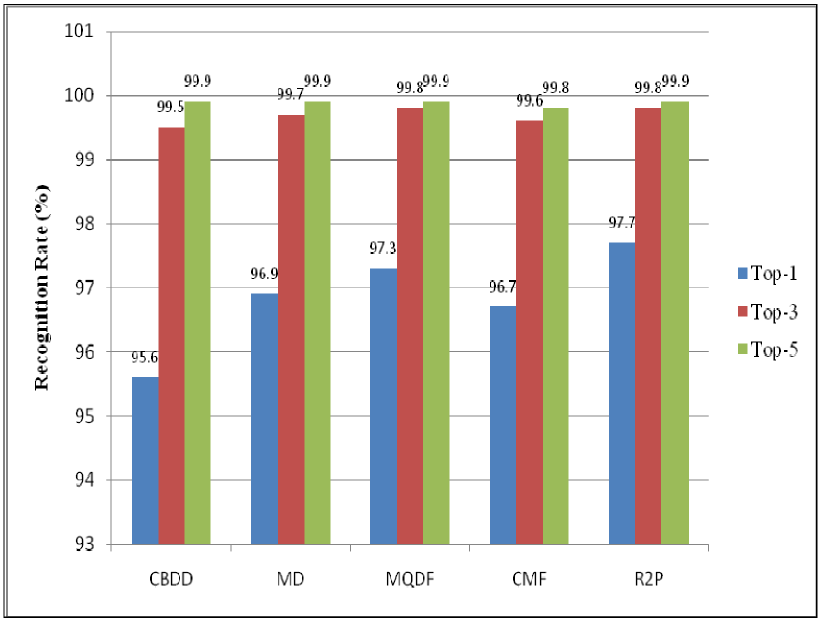


Figure 5.5: Results of top-1, top-3 and top-5 recognized characters by using CBDD, MD, MQDF, CMF and R_p^2 classifier in the case of with normalization.

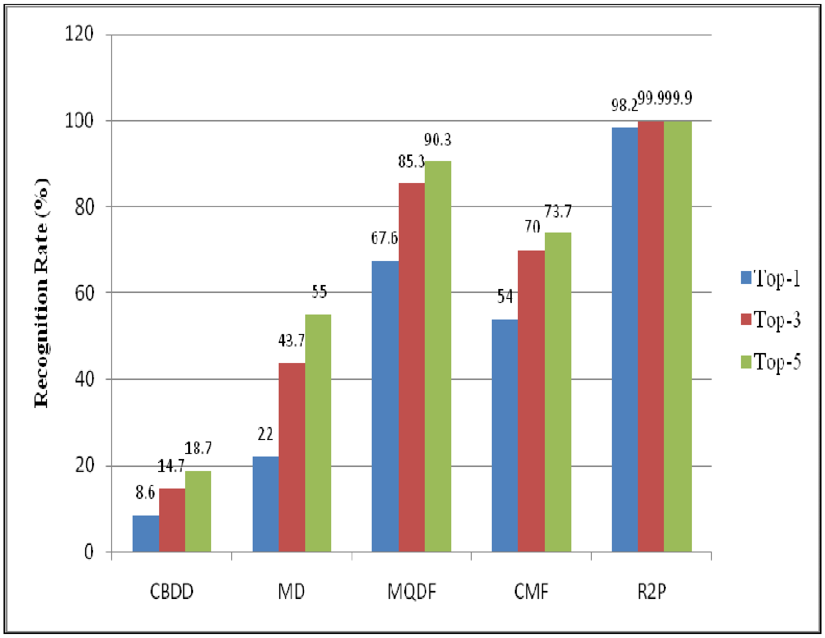


Figure 5.6: Results of top-1, top-3 and top-5 recognized characters by using CBDD, MD, MQDF, CMF and R_p^2 classifier in the case of without normalization.

All the results from Table 5.2, Table 5.3, Figure 5.5 and Figure 5.6 indicates that R_p^2 classifier has stronger discriminative ability in distinguishing Chinese characters despite of their shape complexity and the existence of many similar characters. In addition, it is invariant of different writing styles, and thus it can tolerate well with the character size and position variation, even without undergoing any normalization process. Moreover, it is also able to cope with the problem of character shape deformation, especially for the characters with distorted strokes. For the sake of more ideal recognition system, instead of displaying only the single final recognized character, the top-5 recognition result can be shown as choices for the users. This is the beneficial function that has been embedded in most of the present hand-held devices nowadays.

Furthermore, in the case of without normalization, R_p^2 classifier shows the smallest values of standard deviation, which are 2.5734 and 2.0131 for both the cross-writers approach and cross-characters approach respectively (refer to Table 5.2 and 5.3). This implies that R_p^2 classifier operates in a more stable and reliable manner than CBDD, MD, MQDF and CMF classifier in the no normalization case. Even though the standard deviations obtained by using R_p^2 classifier in the normalization case are not the smallest if compared to the other four existing classifiers, but the values of 2.9078 and 2.0125 (for cross-writers approach and cross-characters respectively) are still reasonably small.

Besides comparing the recognition rates of different classifiers, recognition rates of different feature extraction methods are also analyzed in this paper. Table 5.4 shows the recognition rates of the three existing feature extraction methods studied from Liu *et al.* (1996), Takahashi *et al.* (1997) and Kawamura *et al.* (1992) respectively, and the current methods in this paper. In this comparison, minimum distance (MD) classifier is applied. It can be proved that the proposed method is superior with higher recognition rate, since it has stronger discriminative ability when dealing with similar and deformed characters.

Table 5.4: Recognition rates for four different feature extraction methods.

Method	Recognition Rate (%)
Attributed Relational Graph (ARG)	94.20
Whole Character-Based Hidden Markov Model (HMM)	90.00
Directional Feature Densities (DFD)	91.78
***X*-*Y* Graphs Decomposition with Haar Wavelet**	95.5

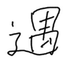
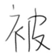
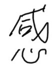
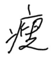
5.3.1 Incorrect Stroke Number

The proposed recognition system is stroke number and stroke order dependent. Hence, in collecting testing samples, all the writers are requested to write with correct stroke number and stroke order as stated in Section 5.1. However, this will be a hindrance when it comes to the situation where the writers have zero knowledge about the correct stroke number and stroke order of Chinese characters, especially those who do not go through a proper Chinese education. Besides, some writers also tend to write in cursive way or with connected strokes. This type of handwriting is difficult to be matched accurately by the proposed recognition system. Some examples of Chinese character with easily mistaken stroke number and with cursive handwriting are demonstrated in Table 5.5 and Table 5.6 respectively.

Table 5.5: Some examples of Chinese character with easily mistaken stroke number.

Chinese Characters	Correct Stroke number	Wrong Stroke number
了	2	1
这	7	6
出	5	3
之	3	2
后	6	5
成	6	5
已	3	2
被	10	8, 9
斯	12	11
感	13	12
张	7	5
改	7	5
病	10	9
考	6	5
革	9	8

Table 5.6: Some examples of cursive handwriting, and their correct as well as wrong stroke number.

Chinese Characters with Cursive Handwriting	Correct Stroke number	Wrong Stroke number
	2	1
	12	11
	10	8
	13	11
	13	10

平	5	4
放	8	6
就	12	10
动	6	5
烂	9	7
到	8	6
起	10	7
都	10	8
来	7	4
是	9	8

In order to tackle this problem, just like the previous database, extra samples with varying stroke number are added into the new created database (CL2009). Although this method will definitely increase the size of the database, it is still considered small after the increment of extra samples, if compared to those existing database such as HCL2000 and ETL9B. In this case, another 120 samples which have no restriction on stroke number are collected from one writer for testing. This testing is done for the cross-writers approach and the case of without normalization only. The recognition rate is shown in Table 5.7. Notice that if compared to the results in Table 5.2, there is an improvement of 2.2%, 8% and 1% in the recognition rate for CBDD, MD and R_p^2 classifier respectively whereas the recognition rate for MQDF and CMF classifier shows a slight decline of 0.1% and 4% respectively. By using this method, R_p^2 classifier is able to attain an excellent recognition rate up to 99.2% which is the largest among the five classifiers. Hence, this implies that storing extra samples of varying stroke number can solve the stroke number variation problem faced by the proposed recognition system successfully, with the size of the database still remains practically small.

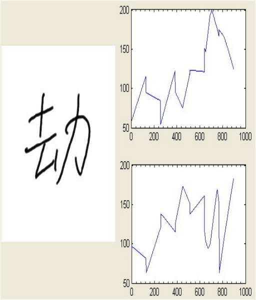
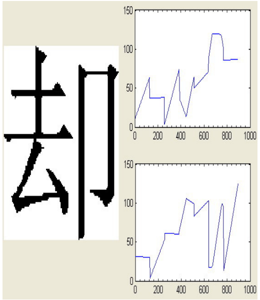
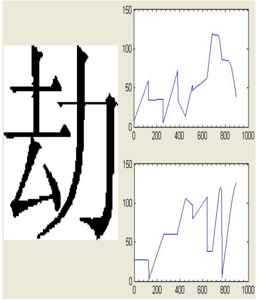
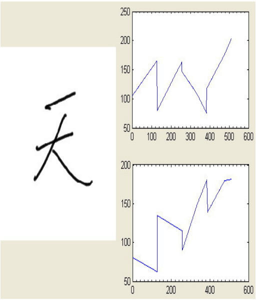
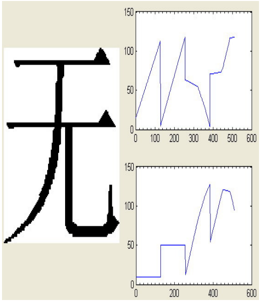
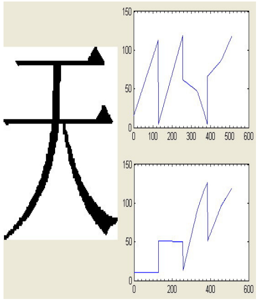
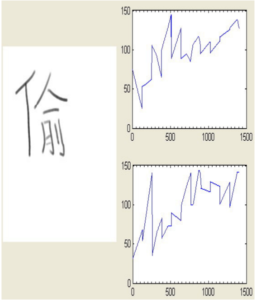
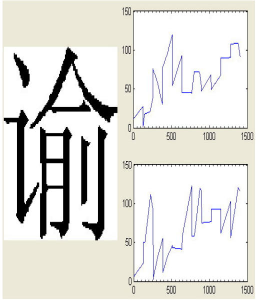
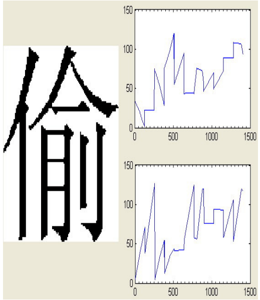
Table 5.7: Experimental results for testing samples of random stroke number without normalization.

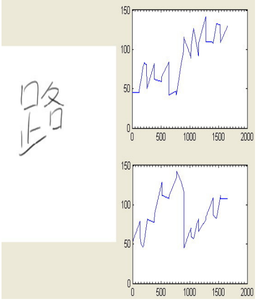
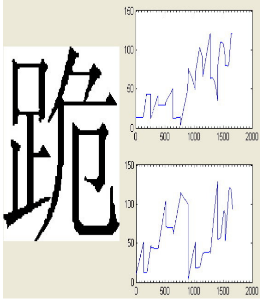
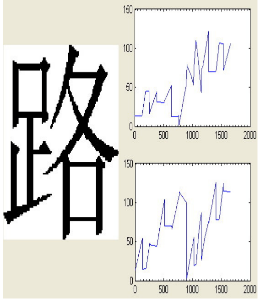
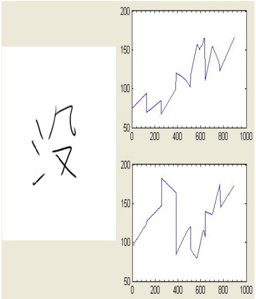
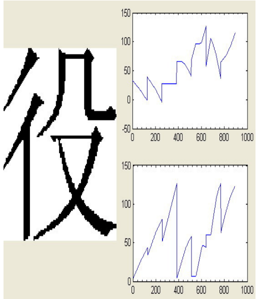
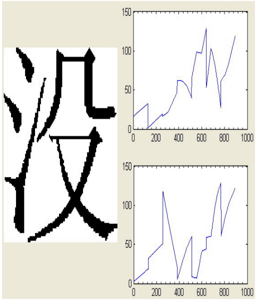
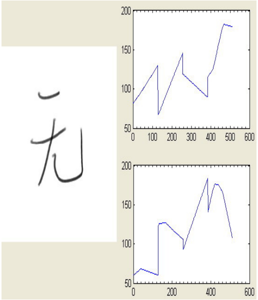
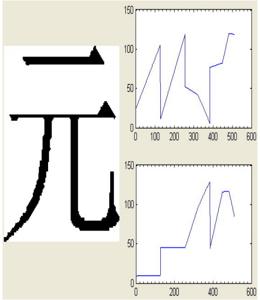
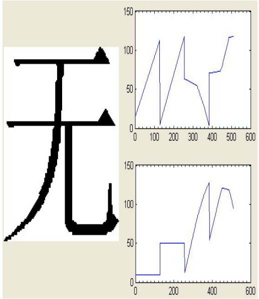
Classifiers	**CBDD**	**MD**	**MQDF**	**CMF**	R_p^2
Recognition rate (%)	10.8	30	67.5	50	99.2

5.3.2 Misclassification Cases

In spite of high recognition rate, there is still some misclassification errors occurred in the new designed recognition system in this research. Majority of the misclassification errors for those previously proposed recognition system are caused by the size, position (if no normalization process is executed) and deformed shape of the input characters. However, these will not be the problems in the case of the proposed recognition system since it is invariant of different writing styles. The factor of misclassification errors encountered in the proposed recognition system is mainly because of the similar characters which often share the common radicals and have slight difference in shape only. Although it has been mentioned in Section 3.2.1(I) that the *X*-*Y* graphs formed for the similar characters are unique, there are also cases where the *X*-*Y* graphs are nearly the same, especially for the similar characters with similar way of writing (in the aspect of stroke order) and with same number of stroke. Some misclassification cases of the proposed recognition system are illustrated in Table 5.8. Due to the great similarity between the *X*-*Y* graphs for the pairs of the similar characters, it is easy to confuse with each other and thus hard to be differentiated by the R_p^2 classifier.

Table 5.8: Some misclassification cases of the proposed recognition system. The graphs above are *X*-graphs and the graphs below are *Y*-graphs.

Input Characters	Output Characters in Database	Exact Characters in Database
		
		
		

From Table 5.8, it can be noticed that the pattern of the *X*-*Y* graphs for each pair of characters in column two and three are almost the same. Since the pattern of the graphs depends significantly on the trajectory of handwriting, a slight difference in the Chinese character written from the original one will affect the pattern of the graphs. As a result, the graphs pattern of a particular input character will be closer to the graphs pattern of its similar character in database than that of the actual or correct character. For sure, this input character will be misclassified by the classifier consequently. As an example (refer to Table 5.8), the Chinese character '劫' (means rob) is mismatched with '却' (means but/yet) since both these characters look the same in structure, except the radical on the right hand side. Moreover, they are same in stroke number and also the way of writing. Hence, minor distortion of the strokes written can affect the matching result easily.

5.4 Processing Time

The speed of the recognition system is one of the important factors for an efficient recognition system, such that reduction of processing time is necessary. In this research, the testing platform is on a Dell Vostro 1400 N-Series notebook of Intel(R) Core(TM)2 Duo Processor T5470 and 1GB (2×512 MB) 667MHz Dual Channel DDR2 SDRAM. The timing analysis of the proposed recognition system is illustrated in Table 5.9 which includes the average processing time for feature extraction, coarse classification and fine classification.

Table 5.9: Result of timing analysis for the proposed recognition system.

	Average processing time (millisecond (ms) per character)
Feature extraction	0.913
Coarse classification	1.176
Fine classification	0.957
Total	3.046

5.4.1 Efficiency of the Feature Extraction Methods

For comparison with the proposed feature extraction method, some representative traditional feature extraction methods such as Attributed Relational Graph (ARG), whole character-based Hidden Markov Model (HMM) and Directional Feature Densities (DFD) as mentioned in Section 2.2.1(I), 2.2.2(I) and 2.2.3(I) respectively are selected. The detail of the processing times for the four feature extraction schemes are described in Table 5.10. It is time consuming to compute the exact processing time since different sets of database has to be constructed for each different feature extraction methods and it is costly to obtain those databases from external institutions. Hence, due to time and cost constraint, the processing times are analyzed from the aspect of steps involved only. From Table 5.10, it can be observed that complicated procedures are involved in ARG, whole character-based HMM and DFD. Whereas, the proposed *X*-*Y* graphs decomposition with Haar wavelet requires simple algorithms, such that only defining feature vectors from graphs and implementing Haar wavelet transform are necessary. Obviously, the proposed feature extraction method thus results in the least processing time.

Table 5.10: Processing time for four different feature extraction methods.

Methods	Steps Involved
Attributed Relational Graph (ARG)	Step 1: Perform strokes identification Step 2: Fit the strokes with straight lines Step 3: Determine geometric centres for each stroke Step 4: Construct complete ARG with nodes and arcs Step 5: Convert the ARG to generalized relation matrix
Whole Character-Based Hidden Markov Model (HMM)	Step 1: Estimates parameters, such as state transition probabilities, output emission probabilities and initial state probability through learning process which is time-consuming. Step 2: Determine HMM of the character
Directional Feature Densities (DFD)	Step 1: Define vectors for each consecutive points on the strokes Step 2: Compute directional feature vectors Step 3: Define vector for square areas Step 4: Perform dimension condensation
***X*-*Y* Graphs Decomposition with Haar Wavelet**	Step 1: Define feature vectors from *X*-*Y* graphs Step 2: Implement Haar wavelet transform

5.4.2 Comparison between Different Classifiers

In this section, the comparison of the processing time between different classifiers is discussed. Figure 5.7 which reveals the processing time for the recognition of characters with varying stroke numbers by using R_p^2, MD, CBDD, MQDF and CMF classifier are presented. Notice that characters with stroke number between 6 and 12 consume longer processing time; while for characters with less than 6 strokes and more than 12 strokes, the processing time is shorter. This is because most of the Chinese characters are in the range of 6 to 12 strokes, so more candidates of that range are chosen for fine classification. Consequently, more processing time is needed. Besides, Figure 5.7 also shows that the recognition system with R_p^2 classifier results in the least processing time, followed by MQDF, CMF, MD and CBDD classifier. Due to the algorithm simplicity of R_p^2 classifier, the processing time can be reduced up to 75.31%, 73.05%, 58.27% and 40.69% if compared to CBDD, MD, CMF and MQDF classifier respectively (refer to Table 5.11). Besides, by omitting the preprocessing stage, the speed of the recognition system can be further accelerated. As a result, the overall performance of the recognition system can be improved tremendously.

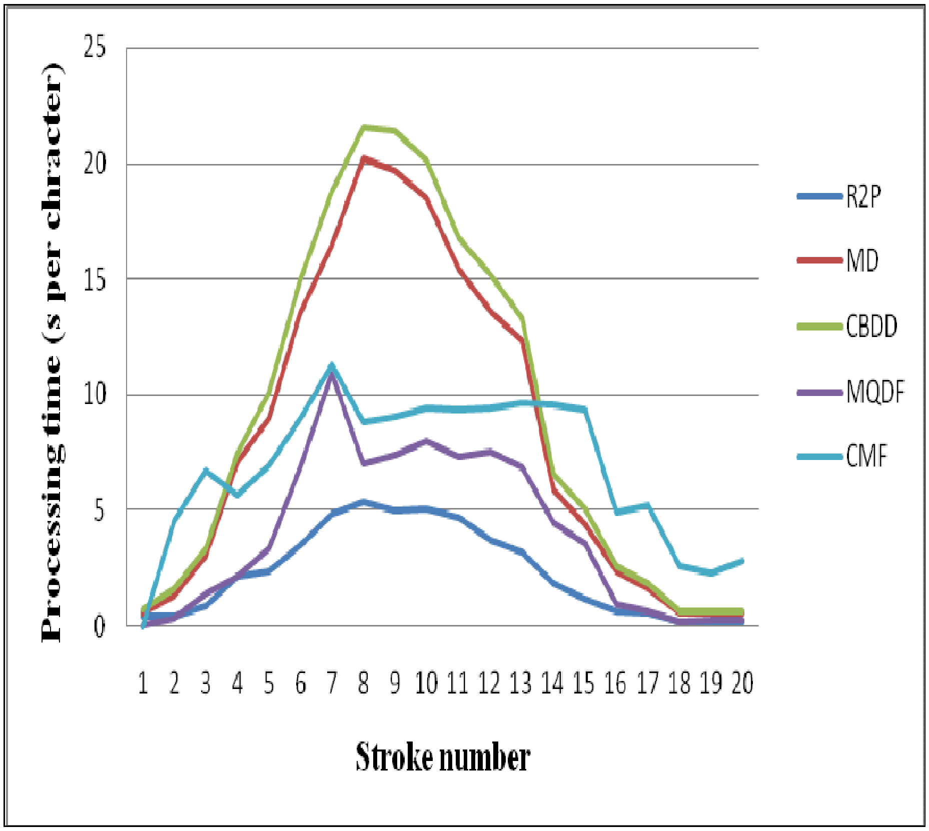


Figure 5.7: Processing time for recognizing characters with varying stroke numbers by using R_p^2, MD, CBDD, MQDF and CMF classifier.

Table 5.11: Reduced time rates in comparing the algorithm of R_p^2 with CBDD, MD, MQDF and CMF classifier.

Classifiers		**CBDD**	**MD**	**MQDF**	**CMF**
Reduced time rate (%) using R_p^2	*With preprocessing*	74.57	72.28	40.36	58.09
	Without preprocessing	75.31	73.05	40.69	58.27

5.5 Storage Space

The size of the extracted features and the complexity of parameters are two main factors that determine the storage space required for the recognition system. For commercial purpose, it is ideal to have a small storage requirement, so that it can be directly embedded into the hand-held devices.

5.5.1 Feature Size

In order to compare the size of the extracted features, the existing feature extraction schemes as being used for comparison in Section 5.4.1, which include ARG, whole character-based HMM and DFD are chosen. From Table 5.12, it implies that feature extraction using *X*-*Y* graphs decomposition with Haar wavelet which is applied in the new designed recognition system gives the smallest feature size, such that the dimension is between 64 (inclusive) and 128 (exclusive). Compared to other three existing feature extraction methods with dimensionality of more than 200, *X*-*Y* graphs decomposition with Haar wavelet is more efficient in term of storage space. Furthermore, due to the ability of *X*-*Y* graphs decomposition in tackling the problem of different writing styles, the memory space can be further saved vastly since the storage of excessive samples in database and learning process are unnecessary. Therefore, the proposed feature extraction method in this research is the most practical and appropriate for memory limited devices.

Table 5.12: Feature sizes for four different feature extraction methods.

Methods	Feature Size (Dimension)
Attributed Relational Graph (ARG)	$(stroke\ number)^2$, increase of stroke number will increase the feature size massively. For example, the dimension of characters with 22 strokes is $22^2 = 484$.
Whole Character-Based Hidden Markov Model (HMM)	Sum of the size of parameters $\{a_{ij}\},\{b_{ik}^1\},\{b_{il}^2\},\{\pi_i\}$ and *N*, where $\{a_{ij}\},\{b_{ik}^1\}$ and $\{b_{il}^2\}$ are matrices, $\{\pi_i\}$ is vector and *N* is scalar.
Directional Feature Densities (DFD)	$8\times8\times4 = 256$
***X*-*Y* Graphs Decomposition with Haar Wavelet**	Between $2^5\times2 = 64$ (inclusive) and $2^6\times2 = 128$ (exclusive)

5.5.2 Parameter Complexity

In real application for hand-held devices, majority of the existing classifiers such as support vector machine (SVM), neural network (NN) and modified quadratic discriminant function (MQDF) classifier as mentioned in Section 2.4.2, 2.4.4 and 2.4.1(I) respectively have parameter complexity problem for large character set recognition despite of their high accuracy rates. For example, the MQDF classifier needs to store the parameters such as mean vectors, eigenvalues and eigenvectors of the covariance matrix for each class. As stated in Long and Jin (2008), suppose that *D*, *K* and *M* are the parameter size of the mean vectors, eigenvalues and the number of class respectively, about 293 MB storage space is required under a system setup of *D*=512, *K*=40 and *M*=3755. As the memory size of the hand-held devices is usually limited nowadays, it is not practical to embed those existing classifiers with high parameter complexity directly into these devices. Thus at present, compressing techniques for reducing the parameter size are necessary in order to embed those high accuracy classifiers into the memory limited devices. On the other hand, only mean vectors are needed to be stored for R_p^2 classifier and this occupies about 53.5 MB for *M*=3000. Larger memory size will definitely causes the increase of products' cost. In view of this, the simplicity of R_p^2 classifier makes it more preferable for real world applications since it is more efficient in term of storage space if compared to other existing classifiers. Moreover, it can be directly applied into the memory limited embedded systems without any compression process.

5.6 Verification of the Experimental Results

In order to verify the experimental results based on the new created database (CL2009) with 3000 characters and 1000 testing samples obtained in this chapter, additional experiment on HCH-GB1 dataset is conducted. HCH-GB1 dataset is one of the 11 datasets in SCUT-COUCH2009 database which is developed by Jin *et al.* (2010). This database is built to facilitate the research of unconstrained online Chinese handwriting recognition and is publicly available for usage in research community. In HCH-GB1 dataset, there are 3755 characters written by 188 writers and this makes up a total number of 705940 for the character samples. All these samples were collected using PDAs (Personal Digit Assistant) and smart phones with touch screens.

In the additional experiment, 20% of the HCH-GB1 dataset is used as the new testing samples while CL2009 remains as the database of the recognition system. Since this research only concerns about the matter of recognition system without normalization, experiment regarding the case with normalization is not carried out here. The recognition rate without normalization of the five classifiers (i.e. CBDD, MD, MQDF, CMF and R_p^2) in this additional experiment are shown in Table 5.13. The testing samples are categorized into two different datasets: (i) dataset without stroke number restriction and (ii) dataset with stroke number restriction. Table 5.13(a) demonstrates the recognition rates for dataset (i) while Table 5.13(b) shows the results for dataset (ii).

Table 5.13: Recognition rates without normalization based on HCH-GB1 dataset: (a) without stroke number restriction and (b) with stroke number restriction.

(a)

Classifiers	**CBDD**	**MD**	**MQDF**	**CMF**	R_p^2
Recognition rate (%) without normalization	1.5	2.2	4.4	3.8	12.8

(b)

Classifiers	**CBDD**	**MD**	**MQDF**	**CMF**	R_p^2
Recognition rate (%) without normalization	8.9	11.1	22.1	19.2	74.2

As stated in Jin *et al.* (2010), the benchmark recognition rate of HCH-GB1 is 95.27% and MQDF classifier is the benchmark recognizer. The algorithms of the above MQDF classifier and of this benchmark classifier are the same. The only difference is that they are implemented in the platform with different conditions. The benchmark classifier is treated with normalization while the above MQDF classifier is treated without normalization. Notice that this normalization process is described in Chapter 3 (see Section 3.1). This is the main reason which leads to the tremendous dropping of recognition rates to 4.4% and 22.1% for our MQDF classifier as shown in Table 5.13.

On the other hand, most of the testing samples from the HCH-GB1 dataset are of incorrect stroke number and stroke order. However, the proposed recognition system is stroke number and stroke order dependent. Hence, this results in great degradation of R_p^2 classifier in term of recognition rate when using HCH-GB1 dataset as testing samples, i.e. a worse result of 12.8% for the case of without stroke number restriction (refer to Table 5.13(a)). Whereas, R_p^2 classifier achieves a higher recognition rate of 74.2% (refer to Table 5.13(b)) when the evaluation of recognition system is restricted to character samples with correct stroke number only. In the additional experiment, although R_p^2 classifier obtains a lower recognition rate if compared to the results in Table 5.2(b) and 5.3(b) which give the recognition rate of 98.2% and 95% respectively, R_p^2 still achieves the highest recognition rate among the classifiers. For CBDD, MD, MQDF and CMF classifier, even worse recognition rates of below 23% are obtained from both Table 5.13(a) and 5.13(b) because they fail to recognize the cursive handwritings with severe shape deformation which occupy a large portion in HCH-GB1 dataset. This indicates that they perform more terribly and are more inappropriate in recognizing characters with incorrect stroke number and stroke order if compared to R_p^2.

Since there is no effect on the results of processing time (in Table 5.9) and storage space (in Table 5.12) by using different testing samples, verification of these two results are unnecessary. On the whole, this section is therefore sufficient to verify the experimental results obtained in Section 5.3, 5.4 and 5.5.

5.7 Summary

This chapter shows that the new HCCR system designed in this research performs better than other previously proposed character recognition systems. Instead of using the existing database, a new database (i.e. CL2009) is created, such that only a single sample is needed to be stored for each character. By using CL2009 database, the occupancy of memory space can hence be reduced significantly since the whole database is kept to do the classification for the proposed recognition system. The new HCCR system associated with *X*-*Y* graphs decomposition, Haar wavelet transform and R_p^2 classifier is proved to be very efficient from the experimental results, in term of recognition rate, processing time and storage space. Verification of the experimental results has been done by conducting an additional experiment on HCH-GB1 dataset.

CHAPTER 6

CONCLUSIONS

As stated in Section 1.3, the key objective of this research is to develop a new online handwritten Chinese character recognition (HCCR) system in order to improve the performance of the existing recognition system. In accomplishing this objective, five main achievements are attained. The explanation of these five main achievements will be the initial focus of this chapter. The problem encountered in this research and some limitation of the proposed recognition system will be discussed in Section 6.2, followed by future works and some publications in Section 6.3 and 6.4 respectively.

6.1 Achievements

The five main achievements acquired in this research are summarized below.

I. Create new database

Majority of the researchers in the area of character recognition apply the existing databases into their proposed recognition systems directly for comparison study between their methods and others' methods. The three commonly used databases for Chinese character recognition are HCL2000, ETL9B and CASIA database. Due to training purpose, these existing databases use to store many samples per class. In this research, a new recognition system without training process is designed and hence excessive samples are not needed here. Instead of using the existing databases, a new database (i.e. CL2009) with only a single sample per class is created. The aim of having this new idea is to show the powerfulness of our recognition system such that it still gives a promising result even if there is no training process and no excessive samples is stored in the database. Since only a single sample is stored for each class in database, the sample chosen must be standard and most representative of that particular class. Thus, all the characters in CL2009 database are standardized to the font style of *songti.* The CL2009 database composed of 3000 most frequently used simplified Chinese characters which are based on Ju Dan's modern Chinese character frequency list. From experiments, it can be proved that the collaboration of the proposed feature extraction method and new designed classifier makes the recognition system with CL2009 database works out well. Furthermore, it is advantageous to use CL2009 database because it can help the proposed recognition system which keeps the whole database in the system for classification to reduce the consumption of memory space significantly.

II. Propose new feature extraction method

Making use of the rich geometrical and topological characteristic of the handwritten character trajectory, the new pattern descriptor proposed in this research, *X*-*Y* graphs decomposition, is simple but powerful in extracting informative features. This is indeed a new idea of feature extraction that represents Chinese character by using simple *X*-graph and *Y*-graph. Both of these graphs are based on how the character is written (or is referred to as trajectory). Besides, these *X*-graph and *Y*-graph also cover both the global and local features of the characters and so enhance the discrimination power. The important properties of this new proposed feature extraction method include uniqueness, invariant of different writing styles and simplicity, which have improved the performance of the recognition system to a great extent.

In *X*-*Y* graphs decomposition, these *X*-graph and *Y*-graph are unique for each Chinese character, even for the case of similar characters. Hence, the recognition system will be more distinguishable between similar characters and the misclassification errors can be decreased consequently. In addition, they are invariant of different writing styles such that it is dealing well not only for regular handwritings, but also natural writings with shape deformation. The pattern of both the *X*-graph and *Y*-graph is still preserved even though for severe deformed characters. Therefore, *X*-*Y* graphs decomposition is so discriminative that only a single sample needed to be stored for each Chinese character in database, rather than storing many excessive samples collected from different writers, as mentioned in Section 6.1(I). The strong discrimination power of *X*-*Y* graphs decomposition can be proved from the experimental results shown in Table 5.4 which compares the recognition rates of different feature extraction methods. It indicates that *X*-*Y* graphs decomposition with higher recognition rates is superior to other feature extraction methods. On the other hand, due to small dimensionality of features (refer to Table 5.12) and simplicity of algorithm (refer to Table 5.10), the performance of the system can be further boosted up in term of memory space and speed.

III. Propose new application of Haar wavelet transform

Wavelet based approaches are becoming increasingly popular in pattern recognition and have recently been applied to character recognition. Most of these wavelet approaches are applied to the character images. They utilize the theory of multiresolution analysis (MRA) to interpret the image at different resolutions and construct the feature vectors accordingly. In this research, the application of wavelet transform on the *X*-graph and *Y*-graph which is a new attempt of wavelet approach is presented.

Among different wavelet families, Haar wavelet is chosen for feature size reduction due to its algorithm simplicity and efficiency. It reduces the size of the feature vector by converting the feature vectors into two new sequences of points which are the approximation and detailed coefficients of Haar transform, but only approximation coefficients are considered for the computation in classification stage. In each level of Haar transform, the sequence of points will be reduced to half of its size and this will lead to an easier computation than other wavelet transforms. In spite of the vast size reduction by Haar transform, the extracted features still retain the important information of the characters without affecting the recognition rate.

IV. Propose new designed classifier

For classification, a new novel classifier, that is the coefficient of determination (COD) for 2-dimensional unreplicated linear functional relationship (2D-ULFR) model is developed. For short, it is called R_p^2 classifier. It is based on the similarity measure via advanced statistically technique, between two character patterns which are constructed from the trajectory of the input character and the character in database. The most attractive advantage of this new designed classifier is that normalization which is an unavoidable process for all the previously proposed classifiers can be omitted for R_p^2 classifier. The idea of adding errors to both response variable and explanatory variable allows more variation on input handwritings. Thus, R_p^2 classifier has higher discrimination ability if compared to other existing classifiers since it can tolerate well with the problem of size and position variation of input characters, as well as the character deformation even without implementing normalization process. Besides, due the simplicity of the algorithm, the recognition system can be speeded up with reduced processing time of 75.31%, 73.05%, 58.27% and 40.69% if compared to CBDD, MD, CMF and MQDF classifiers respectively. Moreover, by excluding the processing time of normalization, the speed of the recognition system can be further accelerated. On the other hand, R_p^2 classifier also enhances the stability and reliability of the recognition system. Therefore, it is undoubted that R_p^2 classifier is more powerful and more appropriate to be adopted in the character recognition system.

V. Improve performance of existing recognition systems

The new HCCR system proposed in this research is so efficient that it has successfully overcome the following obstacles occur in most of the existing recognition systems, as stated in Section 1.2:

(i) Large dimensionality and parameter complexity – Compared to other representative traditional feature extraction methods such as ARG, whole character-based HMM and DFD with dimensionality of more than 200, the new proposed feature extraction method, i.e. *X*-*Y* graphs decomposition with Haar wavelet results in the smallest feature size of dimension between 64 (inclusive) and 128 (exclusive). On the other hand, unlike those classifiers which face parameter complexity problem, including SVM, NN and MQDF classifier, the parameter size of the R_p^2 classifier is small enough to be embedded directly into the memory limited devices without any compression process.

(ii) Algorithm complexity – In spite of high recognition rate, the algorithm used in some existing character recognition systems is complex and this will lead to a degradation of the quality in term of speed. On the contrary, neither complicated nor heavy computation is involved in the proposed HCCR system. The simplicity of *X*-*Y* graphs decomposition with Haar wavelet and R_p^2 classifier has enhanced the performance of the recognition system such that the processing time is reduced greatly.

(iii) Learning process – Majority of the previously proposed recognition systems require learning process in order to estimate the appropriate parameters for the embedded feature extraction schemes and/or classifiers. However, large training set is a necessity for a reliable recognition system and this will lead to large consumption of memory space. Besides, it is also time-consuming. It is good that no learning process is needed for the proposed recognition system and hence, the problems mentioned above can be avoided.

(iv) Normalization process – High recognition rate is not the only requirement for a good quality recognition system, but simple algorithm which can boost up the speed of the recognition system is also an important criterion and cannot be neglected. Throughout many years, researchers tried hard to accelerate their recognition system by reducing the complexity of the feature extraction schemes and classifiers. Nonetheless, they never think of solving this problem from the perspective of normalization (or preprocessing) and until now, no one has tried to reduce the processing time by omitting normalization process. They believe that normalization is an unavoidable process for a recognition system and without normalization, it is impossible to obtain a promising recognition rate. However, the new recognition system with *X*-*Y* graphs decomposition, Haar wavelet and R_p^2 classifier has made this possible. It has been proved in Section 5.3 that the recognition system still remains a high recognition rate, even without undergoing normalization process since it is invariant of different size and position of the input characters.

(v) Storage of excessive samples – In order to tackle the problem of different writing styles, especially the characters with serious deformed shape, the existing databases which include HCL2000, ETL9B and CASIA database used to store many samples collected from different writers for each character. Consequently, this will occupy extremely large storage space. Since the proposed HCCR system is invariant of different writing styles, the storage of excessive sample in CL2009 database is not necessary and hence the storage space can be saved significantly.

On the whole, the efficiency of the new designed recognition system is studied from the experimental results based on the database with 3000 frequently used simplified Chinese characters and 1000 testing samples collected from 10 different writers. With the combination of X-Y graphs decomposition, Haar wavelet transform and R_p^2 classifier, the recognition system achieves a promising recognition rate up to 98.2% even though with small dimensionality, reduced processing time and no normalization is implemented. As a conclusion, these new proposed ideas give a great improvement in the performance of the recognition system in term of recognition rate, storage space and processing time. Undoubtedly, it has provided a new inspiration for this research area.

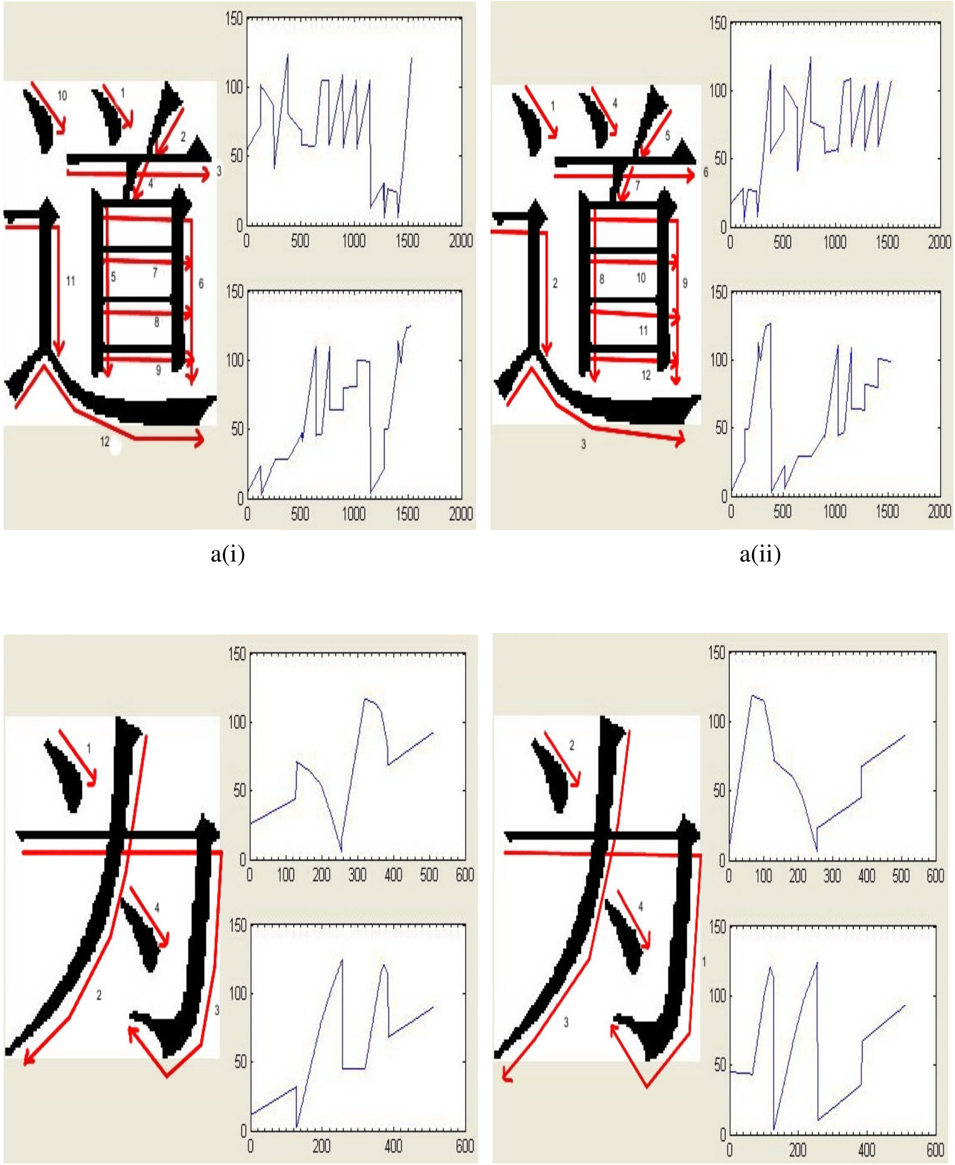
6.2 Problems Encountered and Limitations

Throughout the research, some problems are encountered and they are listed in the following:

(i) In order to make the experimental results more accurate, it is better to apply the existing methods into the proposed recognition system for comparison. However, applying others' feature extraction methods is not easy since different types of features need to be stored for different feature extraction methods. In other words, reconstruction of database is required which is very time-consuming. Hence, no existing feature extraction method is applied into the proposed recognition system to compare with *X-Y* graphs decomposition.

(ii) Due to time constraint, only 3000 characters are stored in the new created database even though the Chinese character set consists of approximately 50,000 characters, and only 1000 testing samples are collected to generate the experimental results.

The limitation of the proposed recognition system is that it is stroke number and stroke order dependent. In this research, the problem of stroke number variation is solved by adding extra samples with varying stroke number into the database (refer to Section 5.3.1). This will not affect the storage space of the recognition system much since only the characters with easily mistaken stroke number are selected. For stroke order variation problem, incorrect stroke order will change the shape of the graphs severely as illustrated in Figure 6.1. This has become a bottleneck since the recognition rate will be degraded to a great extent. In fact, this problem can be overcome by using the same solution as for stroke number variation problem. However, this will definitely increase the consumption of storage space significantly since there are many possibilities of stroke order for each character.



a(i)

a(ii)

b(i)

b(ii)

Figure 6.1: Examples of Chinese characters with correct (left) and incorrect (right) stroke order: (a) character ‘通’ (means go through or to connect) and (b) character ‘为’ (means act as or because of).

6.3 Future Works

There are some issues that require further studies:

(i) It is highly advantageous if the stroke order variation problem can be solved; so that the recognition system will be more user-friendly and more convenient in the sense that it is applicable to all the users including those have no knowledge about the stroke order of Chinese characters. Rearranging the pattern of the graphs will allow the stroke order variation, but it is absolutely a difficult task which will be left for further investigation.

(ii) Due to the flexibility of R_p^2, modification can be made in the future by adding the dimensions for extra information or adjusting the descriptor, in order to further improve the performance of the recognition system.

(iii) Instead of HCCR, R_p^2 classifier can also be applied to the problem of printed Chinese character recognition which is considered as an offline approach. The initial work of this is discussed in Chen *et al*. (2010) (refer to Section 6.4[3]). More improvements for the application of R_p^2 classifier in offline approach still can be made in the future.

6.4 Publications

The following is the list of publications at the time of thesis submission. More publications resulting from this research are planned for submission.

[1] Lee, J.C., Fong, T.J. and Chang, Y.F. (2009). Feature extraction for handwritten Chinese character recognition using *X-Y* graphs decomposition and Haar wavelet. *International Conference on Signal and Image Processing Applications 2009 (ICSIPA09*), 18-19 Nov, Kuala Lumpur.

[2] Chang, Y.F., Lee, J.C., Tong, W.L. and Gan, F.S. (2009). A new classifiers for handwritten Chinese character recognition using 2-dimensional functional relationship model. *Proceedings of 2009 IEEE International Conference on Intelligent Computing and Intelligence System (ICSI 2009)*, 20-22 Nov, Shanghai. vol. 4, no. 4, pp. 1-4.

[3] Chen, H.V., Lee, J.C. and Ng, W.S. (2010). Feature extraction on printed Chinese character recognition using array reduction. *International Conference on Mathematics, Statistics and Scientific Computing (ICMSSC 2010)*, 24-26 Feb, Penang. [Accepted]

REFERENCES

Bahlmann C., Haasdonk B. and Burkhardt H. (2002). On-line handwriting recognition with support vector machines – a kernel approach. *International Workshop on Frontiers in Handwriting Recognition (IWFHR)*. pp. 49-54.

Bahlmann C. and Burkhardt H. (2004). The writer independent online handwriting recognition system frog on hand and cluster generative statistical dynamic time warping. *IEEE Transactions on Pattern Analysis Machine Intelligence*. vol. 26, pp. 3.

Bai, Z.L. and Huo, Q. (2005). A study on the use of 8-directional features for online handwritten Chinese character recognition. *Proceedings of the 8th International Conference on Document Analysis and Recognition*. pp. 262-266.

Baum, L.E. and Petrie, T. (1966). Statistical inference for probabilistic functions of finite state Markov chains. *Ann. Math. Stat*. vol. 37, pp. 1554-1563.

Boser, B.E., Guyon, I.M. and Vapnik, V.N. (1992). A training algorithm for optimal margin classifiers. *Proc. 5th Ann. ACM Workshop Computational Learning Theory*. pp. 144-152.

Bashir, M. and Kempf, J. (2008). Reduced dynamic time warping for handwriting recognition based on multi-dimensional time series of a novel pen device. *World Academy of Science, Engineering and Technology*. vol. 45.

Box, G.E.P. and Cox, D.R. (1964). An analysis of transformations. *Journal of the Royal Statistical Society*. Series B26(2), pp 211-252.

Burges, C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*. vol. 2, pp.1-43.

Cao, J., Ahmadi, M. and Shridhar, M. (1995). Recognition of handwritten numerals with multiple feature and multistage classifier. *Pattern Recognition*. vol. 28, no. 2, pp. 153-160.10

Casey, R. and Nagy, G. (1966). Recognition of printed Chinese characters. *IEEE Trans. Electronic Computers*. vol. 15, no. 1, pp. 91-101.

Casey, R.G. (1970). Moment normalization of handprinted character, *IBM J. Res. Develop*. vol. 14, pp.548-557.

Chan, K.P. and Cheung, Y.S. (1992). Fuzzy-attribute graph with application to Chinese character recognition. *IEEE Trans. System Man, and Cybernetics*. vol. 22, no. 1, pp. 153-160.

Chang, Y.F., Abu Bakar, S.A.R., Rijal, O.M. (2009). Multidimensional unreplicated linear functional relationship model with single slope and its coefficient of determination. *University of Malaya: Technical Report*. no. 1/2009, pp. 1-21.

Chen, K.J., Li, K.K., Chang, Y.L. (1988). A system for on-line recognition of Chinese characters, Int'l J. *Pattern Recognition and Artificial Intelligence*. vol. 2, pp. 139-148.

Chen, L.H. and Lieh, J.R. (1990). Handwritten character recognition using a 2-Layer random graph model by relaxation matching. *Pattern Recognition*. vol. 23, pp. 1189-1205.

Cheng, F.H. (1998). Multi-stroke relaxation matching method for handwritten Chinese character recognition. *Pattern Recognition*. vol. 31, no. 4, pp. 401-410.

Dan, J. (2004). *Modern Chinese Character Frequency List*. URL: http://lingua.mtsu.edu/chinesecomputing/statistics/char/list.php?Which=MO. Accessed on 15th November 2008.

Deepu, V., Sriganesh, M. and Ramakrishnan, A.G. (2004). Principle component anaysis for online handwritten character recognition. *Pattern Recognition Society*. vol.2, pp. 327-330.

Devijver, P.A. and Kittler, J. (1982). Pattern recognition: a statistical approach. London: Prentice-Hall.

Dong, J.X., Krzyzak, A. and Suen, C,Y. (2005). Fast SVM training algorithm with decomposition on very large data sets. *IEEE Trans. Pattern Analysis and Machine Intelligence*. vol. 27, no. 4. pp. 603-618.

Dong, J.X., Krzyzak, A. and Suen, C.Y. (2005). An improved handwritten Chinese character recognition system Using support vector machine. *Pattern Recognition Letter*. vol. 26, no. 12, pp. 1849-1856.

Eamonn, J.K. and Michael, J.P. (2001). Derivative dynamic time warping. *In Proc. of the 1st SIAM Int. Conf. on Data Mining (SDM).*

Friedman, J.H. (1989). Regularized discriminant analysis. *J. Americ. Statist. Assoc*. vol. 84, pp. 165-175.

Fu, H.C. and Xu, Y.Y. (1998). Multilinguistic handwritten character recognition by bayesian decision-based neural networks. *IEEE Trans. Signal Process*. vol. 46, no. 10, pp. 2781-2789.

Gao, T.F., Liu, C.L. (2008). High accuracy handwritten Chinese character recognition using LDA-based compound distances. *Pattern Recognition*. vol. 41, no. 11, pp. 3442-3451.

Gao, X., Jin, L.W., Yin, J.X. and Huang, J.C. (2002). SVM-based handwritten Chinese character recognition. *Chinese Journal of Electronics*. vol. 30, no. 5, pp. 651-654.

Garris, M.D., Wilson, C.L. and Blue, J.L. (1998). Neural network-based systems for handprint OCR applications. *IEEE Trans. Image Processing*. vol. 7, no. 8, pp. 1097-1112.

Geva, S. and Sitte, J. (1991). Adaptive nearest neighbor pattern classification. *IEEE Trans. Neural Networks*. vol. 2, no. 2, pp.318-322.

Glenn, J.B. (1996). Mean square error. *AMP Journal of Technology*. vol. 5, pp. 31-36.

Gonzalez, R.C., Woods, R.E. (1993). *Digital Image Processing*. New York: Addison-Wesley Publishing Co. pp. 580-583.

Govindan, V.K. and Shivaprasad, A.P. (1990). Character recognition–a review. *Pattern Recognition*. vol. 23, pp. 671-683.

Guo, H. and Gelfand, S.B. (1992). Classification trees with neural network feature extraction*.* *IEEE Trans. Neural Networks*. vol. 3, no. 6, pp. 923-933.

Hamanaka, M., Yamada, K. and Tsukumo, J. (1993). On-line Japanese character recognition experiments by an off-line method based on normalization-cooperated feature extraction. *Proceeding of the 3rd International Conference on Document Analysis and Recognition*. pp. 204-207.

Hao, H.W., Xiao, X.H. and Dai, R.W. (1997). Handwritten Chinese character recognition by metasynthetic approach. *Pattern Recognition*. vol. 30, no. 8, pp. 1321-1328.

Hasegawa, T., Yasuda, H. and Matsumoto, T. (2000). Fast discrete HMM algorithm for on-line handwriting recognition. *Proceedings of the 15th International Conference on Pattern Recognition*. vol. 4, pp. 535-538.

Hilderbrand, T.H. and Liu, W. (1993). Optical recognition of Chinese characters: advances since 1980. *Pattern Recognition*. vol. 26, no. 2, pp. 205-225.

Hirayama, J., Nakayama, H. and Kato, N. (2007). A classifier of similar characters using compound Mahalanobis function based on difference subspace. *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*. vol. 1, pp. 432,436.

Horiuchi, T., Haruki, R., Yamada, H. and Yamamoto, K. (1997). Two-dimensional extension of nonlinear normalization method using line density for character recognition. *Proceedings of the 4th International Conference on Document Analysis and Recognition*, Ulm, Germany. pp. 511-514.

Huang, L. and Huang, X. (2001). Multiresolution recognition of offline handwritten Chinese characters with wavelet transform. *Proceedings of the 6th International Conference on Document Analysis and Recognition*. pp. 631-634.

Jaeger, S., Liu, C.L. and Nakagawa, M. (2003). The state of the art in Japanese on-line handwriting recognition compared to techniques in western handwriting recognition. *Int'l J. Conf. Document Analysis and Recognition*. vol. 6, no. 2, pp. 75-88.

Jain, A.K. and Chandrasekaran, B. (1982). Dimensionality and sample size considerations in pattern recognition practice. In: Krishnaiah P.R. and Kanal, L.N. (Eds.), *Handbook of Statistics*, vol. 2. (pp. 835-855). North-Holland, Amsterdam.

Jain, A.K., Duin, R.P.W. and Mao, J. (2000). Statistical pattern recognition: a review. *IEEE Trans. Pattern Anal. Mach. Intell.* vol. 22, pp. 4-37.

Jain, L.C. and Lazzerini, B. (1999). An introduction to handwritten character and words recognition. In: *Knowledge-Based Intelligent Techniques in Character Recognition*. (pp. 1-16). Florida, U.S.A: CRC Press LLC.

Jin, L.W., Gao, Y., Liu, G., Li, Y.Y., Ding, K. (2010). SCUT-COUCH2009----A Comprehensive Online Unconstrained Chinese Handwriting Database and Benchmark Evaluation. *To appear in International Journal of Document Analysis and Recognition.*

Juang, B.H. and Katagiri, S. (1992). Discriminative learning for minimum error classification. *IEEE Trans. Signal Processing*. vol. 40, no. 12, pp. 3043-3054.

Kato, N., Abe, M. and Nemoto, Y. (1996). A handwritten character recognition system using modified Mahalanobis distance. *Trans. IEICE*. vol. J79-D-II, no. 1, pp. 45-52.

Kato, N., Suzuki, M., Omachi, S.I., Aso, H. and Nemoto, Y. (1999). A handwritten character recognition system using directional element feature and asymmetric Mahalanobis distance. *IEEE Trans. Pattern Anal. Mach. Intell.* vol 21, pp.258-262.

Kawamura, A., Yura, K., Hayama, T., Hidai, Y., Minamikawa, T., Tanaka, A. and Masuda, S. (1992). On-line recognition of freely handwritten Japanese characters using directional feature densities. *Proceedings of the 11th International Conference on Pattern Recognition.* vol. 2, pp. 183-186.

Kendall, M.G., Stuart, A. (1979). *The Advanced Theory of Statistics, Vol. 2.* London: Griffin.

Kherallah, M., Haddad, L., Alimi, A.M. and Mitiche, A. (2008). On-line handwritten digit recognition based on trajectory and velocity modeling. *Pattern Recognition Letters.* vol. 29, pp. 580-594.

Kilic, N., Gorgel, P., Ucan, O.N. and Kala, A. (2008). Multifont Ottoman character recognition using support vector machine. *3rd International Symposium on Communications, Control and Signal Processing (ISCCSP 2008)*. pp. 328-333.

Kim, H.J., Kim, K.H., Kim, S.K. and Lee, F.T.P. (1997). On-line recognition of handwritten Chinese characters based on hidden Markov models. *Pattern Recognition*. vol. 30, no. 9, pp. 1489-1499.

Kimura, F., Takashina, K., Tsuruoka, S. and Miyake, Y. (1987). Modified quadratic discriminant functions and its application to Chinese character recognition. *IEEE Trans. Pattern Anal. Mach. Intell*. vol. 9, no. 1, pp. 149-153.

Kimura, F., Wakabayashi, T., Tsuruoka, S. and Mayake, Y. (1997). Improvement of handwritten Japanese character recognition using weigthed direction code histogram. *Pattern Recognition*. vol. 30, no. 8, pp. 1329-1337.

Kohonen, T. (1988). Self-organization and Assciative Memory, 2nd ed. In: *Springer Series in Information Sciences*. (pp. 199-202). New York: Springer-Verlag.

Kruskall, J.B. and Liberman, M. (1983). The symmetric time warping algorithm: from continuous to discrete. In time warps, string edits and macromolecules: the theory and practice of sequence comparison. *Addison-Wesley*. pp. 125-161.

Kundu, A. and Bahl, P. (1988). Recognition of handwritten script: a hidden Markov model based approach. *Proc. ICASSP'88*. vol. 2, pp. 928-931.

Liang, Z. and Shi, P. (2005). A metasynthetic approach for segmenting handwritten Chinese character strings. *Pattern Recognition Letters*. vol. 26, pp. 1498-1511.

Liu, C.L. (2006). High accuracy handwritten Chinese character recognition using quadratic classifiers with discriminative feature extraction. *The 18th International Conference on Pattern Recognition (ICPR'06)*. vol. 2, pp. 942-945.

Liu, C.L. and Marukawa, K. (2004). Global shape normalization for handwritten Chinese character recognition: a new method. *Proceedings of the 9th International Workshop on Frontiers of Handwriting Recognition*, Tokyo, Japan. pp. 300-305.

Liu, C.L. and Marukawa, K. (2005). Pseudo two-dimensional shape normalization methods for handwritten Chinese character recognition. *Pattern Recognition*. vol. 38, no. 12, pp. 2242-2255.

Liu, C.L., Jaeger, S. and Nakagawa, M. (2004). Online recognition of Chinese characters: the-state-of-the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. vol. 26, no. 2, pp. 198-213.

Liu, C.L., Sako, H. and Fujisawa, H. (2003). Handwritten Chinese character recognition: alternatives to nonlinear normalization. *Proceedings of the 7th International Conference on Document Analysis and Recognition*, Edinburgh, Scotland. pp. 524-528.

Liu, C.L., Sako, H. and Fujisawa, H. (2004). Discriminative learning quadratic discriminant function for handwriting recognition. *IEEE Trans. Neural Networks*. vol. 15, no. 2, pp. 430-444.

Liu, H. and Ding, X. (2005). Handwritten character recognition using gradient feature and quadratic classifier with multiple discrimination schemes, *Proceedings of the 8th ICDAR, Seoul, Korea*. pp. 19-23.

Liu, J.Z., Cham, W.K. and Chang, M.M.Y. (1996). Online Chinese character recognition using attributed relational graph matching. *IEE Proc. Vision Image Signal Processing*. vol. 143, no, 2, pp. 125-131.

Liu, Y.J., Zhang, L.Q. and Tai, J.W. (1993). A new approach to on-line handwriting Chinese character recognition. *Proceedings of the 2nd International Conference on Document Analysis and Recognition.* pp. 192-195.

Long, T. and Jin, L.W. (2008). Building compact MQDF classifier for large character set recognition by subspace distribution sharing. *Pattern Recognition*. vol. 41, pp. 2916-2925.

Lu, S.W., Ren, Y. and Suen, C.Y. (1991). Hierarchical attributed graph representation and recognition of handwritten Chinese characters. *Pattern Recognition*. vol. 24, pp. 617-632.

Ma, Y. and Leedham, G. (2007). On-line recognition of handwritten Renqun shorthand for fast mobile Chinese text entry. *Pattern Recognition Letters*. vol. 28, pp. 873-883.

McLachlan, G.J. (1992). Discriminant analysis and statistical pattern recognition. In: *Wiley Series in Probability and Statistics*. New York: John Willy & Sons, Inc.

Mitoma, H., Uchida S. and Sakoe, H. (2004). Online character recognition using eigen-deformation. *9th International Workshop on Frontiers in Handwriting Recognition.*

Montgomery, D.C., Peck, E.A. (1992). *Introduction to Linear Regression Analysis,2nd Edition*. New York: John Wiley & Sons.

Nag, R., Wong, K.H. and Fallside, F. (1986). Script recognition using hidden Markov models. *Proc. ICASSP'86*. vol. 3, pp. 2071-2074.

Nakagawa, M. (1990). Non-keyboard input of Japanese text - on-line recognition of handwritten characters as the most hopeful approach. *J. Information Processing*. vol. 13, no. 1, pp. 15-34.

Nakagawa, M., Akiyama, K., Oguni, T. and Kato, N. (1999). Handwriting-based user interfaces employing on-line handwriting recognition. In: Lee, S.W. (Eds.), *advances in Handwriting Recognition.* (pp. 578-587). World Scientific.

Nakagawa, M., Akiyama, K., Tu, L.V., Homma, A. and Kigashiyama, T. (1996). Robust and highly customizable recognition of on-Line handwritten Japanese characters. *Proceeding of the 13th International Conference on Pattern Recognition.* vol. 3, pp. 269-273.

Nakai, M., Akira, N., Shimodaira, H. and Sagayama, S. (2001). Substroke approach to HMM-based on-line Kanji handwriting recognition. *Proceedings of the 6th International Conference on Document Analysis and Recognition.* pp. 491-495.

Nakai, M., Shimodaira, H. and Sagayama, S. (2003). Generation of hierarchical dictionary for stroke-order free Kanji handwriting recognition based on substroke HMM. *Proceedings of the 7th International Conference on Document Analysis and Recognition*. vol. 1, pp. 514-518.

Nakai, M., Sudo, T., Shimodaira, H. and Sagayama, S. (2002). Pen pressure features for writer-independent on-line handwriting recognition based on substroke HMM. *Proceedings of the 16th International Conference on Pattern Recognition*. vol. 3, pp. 220-223.

Nakajima, T., Wakabayashi, T., Kimura, F. and Miyake, Y. (2000). Accuracy improvement by compound discriminant functions for resembling character recognition. *Trans. IEICE Jpn. J83-D-II*. vol. 2, pp. 623-633.

Niels, R. and Vuurpijl, L. (2005). Using dynamic time warping for intuitive handwriting recognition. *In Advances in Graphonomics, Proceedings of the 12th Conference of the International Graphonomics Society (IGS2005), Eds. A. Marcellli and C. De Stefano*. pp. 217-221.

Omachi, S., Sun, F. and Aso, H. (2000). A new approximation method of the quadratic discriminant function. *Proc. SSPR & SPR 2000*. vol. 1876, pp. 601-610.

Pinho, R.R., Tavares, J.M.R.S. and Correia, M.V. (2007). Efficient approximation of the Mahalanobis distance for tracking with the Kalman filter. *International Journal of Simulation Modeling*. vol.6, no. 2, pp. 84-92.

Plamondon, R. and Srihari, S.N. (2000). On-line and off-line handwriting recognition: a comprehensive survey. *IEEE Trans. Pattern Analysis and Machine Intelligence*. vol. 22, no. 1, pp. 63-82.

Pooi, A.H. (2003). Effects of non-normality on confidence intervals in linear models. *University of Malaya: Technical Report*. no. 6/2003.

Rabiner, L.R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*. vol. 77, no. 2, pp. 257-286.

Romero, R., Berger, R., Thibadeau, R. and Touretsky, D. (1995). Neural network classifiers for optical Chinese character recognition. *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval*. pp. 385-398.

Romero, R., Touretzky, D. and Thibadeau, R. (1997). Optical Chinese character recognition using probabilistic neural networks. *Pattern Recognition*. vol. 8, pp. 1279-1292.

Russel, G., Perrone, M.P., Chee, Y. and Ziq, A. (2002). Handwritten document retrieval. *Proc. Eigth Int'l Workshop Frontiers in Handwriting Recognition*. pp. 233-238.

Saruta, K., Kota, N., Abe, M. and Nemoto, Y. (1996). High accuracy recognition of ETL9B using exclusive learning neural network-II (ELNET-II). *IEICE Trans. Inf. Syst*. vol. 79-D, no. 5, pp. 516-521.

Senda, S., Minoh, M. and Katsuo, I. (1995). A fast algorithm for the minimum distance classifier and its application to Kanji character recognition. *Third International Conference on Document Analysis and Recognition (ICDAR'95).* vol. 1, pp. 283-286.

Shimodaira, H., Sudo, T., Nakai, M. and Sagayama, S. (2003). On-line overlaid-handwriting recognition based on substroke HMMs. *Proceedings of the 7th International Conference on Document Analysis and Recognition*. vol. 2, pp.1043.

Shioyama, T., Wu, H.Y. and Nojima, T. (1998). Recognition algorithm based on wavelet transform for handprinted Chinese characters. *Proceedings of the 14th International Conference on Pattern Recognition*. vol. 1, pp. 229-232.

Shu, H. (1996). *On-line handwriting recognition using hidden Markov models*. Master's Thesis, Massachusetts Institute of Technology, USA.

Sridhar, M., Mandalapu, D. and Patel, M. (1999). Active-DTW : A generative classifier that combines elastic matching with active shape modeling for online handwritten character recognition. *International Conference on Frontiers in Handwriting Recognition*. vol. 99, no. 7, pp. 1–100.

Srihari, S.N., Yang, X.S. and Ball, G.R. (2007). Offline Chinese handwriting recognition: a survey. *Front. Comput. Sci. China,* vol. 1, no. 2, pp. 137-155.

Sun, N., Tabara, T., Aso, H. and Kimura, M. (1991). Printed character recognition using directional element feature. *Trans. IECE Japan*. vol J74-DII, no. 3, pp. 330-339.

Suzuki, M., Ohmachi, S., Kato, N., Aso, H. and Nemoto, Y. (1997). A discrimination method of similar characters using compound Mahalanobis function. *Trans. IEICE Jpn J80-D-II*. vol. 10, pp. 2752-2760.

Takahashi, K., Yasuda, H. and Matsumoto, T. (1997). A fast HMM algorithm for on-line handwritten character recognition. *Proceedings of the 4th International Conference on Document Analysis and Recognition.* pp. 369-375.

Tappert, C.C., Suen, C.Y. and Wakahara, T. (1990). The state of the art in on-line handwriting recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*. vol. 12, no. 8, pp. 787-808.

Teredesai, A., Ratzlaff, E., Subrahmonia, J. and Govindaraju, V. (2002). On-line digit recognition using off-line features. *Indian Conference on Computer Vision, Graphics and Image Processing,* SAC, Ahmedabad, India.

Tokuno J., Inami, N., Matsuda, S., Nakai, M. Shimodaira, H. and Sagayama, S. (2002). Context-Dependent Substroke Model for HMM-Based On-Line Handwriting Recognition. *Proceedings of the 8th International Workshop Frontiers in Handwriting Recognition*. pp. 78-83.

Tsai, W.H. and Fu, K.S. (1979). Error-correcting isomorphisms of attributed relational graphs for pattern analysis. *IEEE Trans. Systems, Man and Cybernetics*. vol. 9, no. 12, pp. 757-768.

Tseng, L.Y. and Chen, R.C. (1998). Segmentation handwritten Chinese characters based on heuristic merging of stroke bounding boxes and dynamic programming. *Pattern Recognition Letters*. vol. 19, pp. 963-973.

Umeda, M. (1996). Advances in recognition methods for handwritten Kanji characters. *IEICE Trans. Information and System*. vol. 79-D, no. 5, pp. 401-410.

Van der Weken, D., Nachtegael, M., Kerre, E.E. (2002). Image quality evaluation. *Proceedings of 6th International Conference on Signal Processing*. vol. 1, pp. 711 – 714.

Vapnik, V. (1999). An overview of statistical learning theory. *IEEE Tran. Neural Network*. vol. 10, pp. 989-999.

Vuori, V., Laaksonen J., Oja. E. and Kangas, J. (2001). Speeding up on-line recognition of handwritten characters by pruning the prototype set. *In Proc. of (ICDAR'01)*. pp. 501-505.

Wakabayashi, T., Deng, Y., Tsuruoka, S., Kimura, F. and Miyake, Y. (1996). Accuracy improvement by nonlinear normalization and feature compression in handwritten Chinese character recognition. *Trans. IEICE*. vol. J79-D-II, no. 5, pp.765-774.

Wakahara, T., Murase, H. and Odaka, K. (1992). On-line handwriting recognition. *Proc. IEEE*. vol. 80, no. 7, pp. 1181-1194.

Wang, X.W., Ding, X.Q. and Liu, C.S. (2005). Gabor filters-based feature extraction for character recognition. *Pattern Recognition*. vol. 38, no. 3, pp. 369-379.

Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*. vol. 13, no. 4, pp. 600-612.

Wei, X., Ma, S. and Jin, Y. (2005). Segmentation of connected Chinese characters based on genetic algorithm. *Proceedings of the 9th International Conference on Document Analysis and Recognition*. vol. 1, pp. 645-649.

Wu, M., Zhang, B. and Zhang, L. (2000). A neural network based classifier for handwritten Chinese character recognition. *Proceedings of the 15th International Conference on Pattern Recognition (ICPR'00)*. vol. 2, pp. 561-564.

Yasuda, M. and Fujisawa, H. (1979). An improvement of correlation method for character recognition. *Trans. IEICE Japan*. vol. J62-D, no. 3, pp. 217-224.

Zadeh, L.A. (1965). Fuzzy Sets. *Inform. Contr*. vol. 8, pp. 338-353.

Zeng, W., Meng, X.X., Yang, C.L. and Huang, L. (2006). Feature extraction for online handwritten characters using Delaunay triangulation. *Computers & Graphics*. vol. 30, pp. 779-786.

Zhang, L. and Zhang, B. (1999). A geometrical representation of McCulloch-Pitts neural model and its applications. *IEEE Trans. Neural Networks*. vol. 10, pp. 925-929.

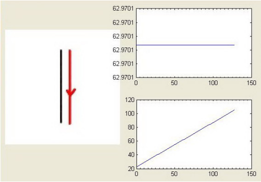
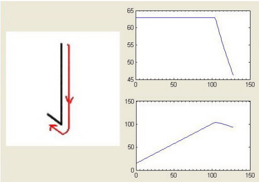
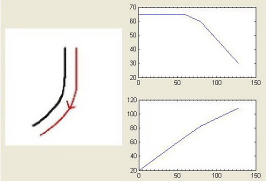
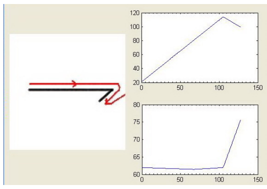
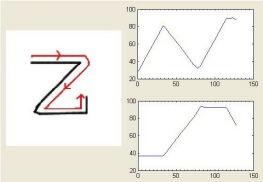
Zhang, W., Tang, Y.Y. and Xue, Y. (2006). Handwritten character recognition using combined gradient and wavelet feature. *2006 International Conference on Computational Intelligence and Security*. vol. 1 pp. 662-667.

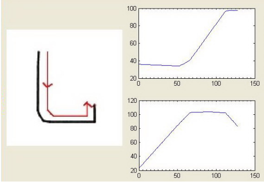
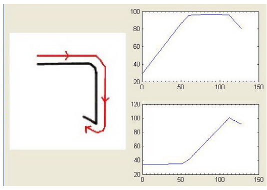
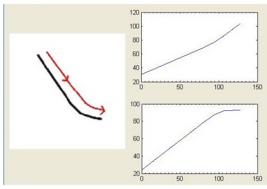
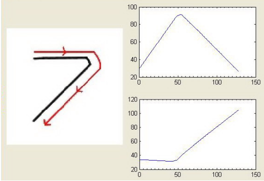
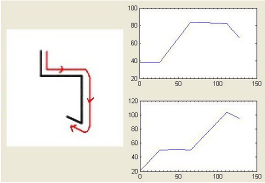
Zheng, J. Ding, X. and Wu, Y. (1997). Recognizing on-line handwritten Chinese character via FARG matching. *Proceedings of the 4th International Conference on Document Analysis and Recognition*. vol. 2, pp. 621-624.

Zheng, J., Ding, X., Wu, Y. and Lu, Z. (1999). Spatio-temporal unified model for on-line handwritten Chinese character recognition. *Proceedings of the 5th International Conference on Document Analysis and Recognition.* pp. 649-652.

Appendix A: *X-Y* GRAPHS DECOMPOSITION AND R_p^2 CLASSIFIER

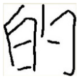
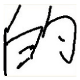
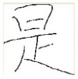
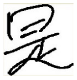
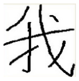
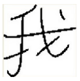
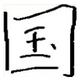
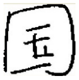
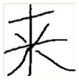
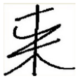
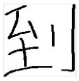
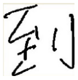
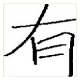
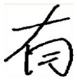
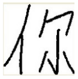
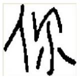
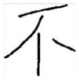
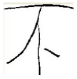
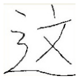
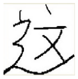
Appendix A1: *X*-graph (above) and *Y*-graph (below) Formed for Some of the Strokes Exist in Chinese Characters

No.	*X-Y* graphs formed for the strokes
1.	
2.	
3.	
4.	
5.	

No.	*X-Y* graphs formed for the strokes
6.	
7.	
8.	
9.	
10.	

Appendix A2: Examples of R_p^2 Values between Some Input Characters and Database Characters

Case 1: Input characters with normalization process

No.	Input characters	Database Characters	R_p^2 values
1.	 (i)  (ii)		(i) 0.9799 (ii) 0.9446
2.	 (i)  (ii)		(i) 0.9611 (ii) 0.9393
3.	 (i)  (ii)		(i) 0.9644 (ii) 0.9622
4.	 (i)  (ii)		(i) 0.9650 (ii) 0.9564
5.	 (i)  (ii)		(i) 0.9457 (ii) 0.8649
6.	 (i)  (ii)		(i) 0.9750 (ii) 0.9675
7.	 (i)  (ii)		(i) 0.9647 (ii) 0.9569
8.	 (i)  (ii)		(i) 0.9553 (ii) 0.9530
9.	 (i)  (ii)		(i) 0.9818 (ii) 0.9517
10.	 (i)  (ii)		(i) 0.9874 (ii) 0.9720

Case 2: Input characters without normalization process

No.	Input characters	Database Characters	R_p^2 values
1.	的 的 (i) (ii)	的	(i) 0.9810 (ii) 0.9437
2.	是 是 (i) (ii)	是	(i) 0.9611 (ii) 0.9507
3.	我 我 (i) (ii)	我	(i) 0.9788 (ii) 0.9607
4.	国 国 (i) (ii)	国	(i) 0.9784 (ii) 0.9586
5.	来 来 (i) (ii)	来	(i) 0.9203 (ii) 0.8425
6.	到 到 (i) (ii)	到	(i) 0.9633 (ii) 0.9623
7.	有 有 (i) (ii)	有	(i) 0.9443 (ii) 0.9429
8.	你 你 (i) (ii)	你	(i) 0.9764 (ii) 0.9642
9.	不 不 (i) (ii)	不	(i) 0.9738 (ii) 0.9513
10.	这 这 (i) (ii)	这	(i) 0.9795 (ii) 0.9773

Appendix A3: 3000 Frequently Used Simplified Chinese Characters Stored in the New Created Database

No.	Character
1	的
2	一
3	是
4	不
5	了
6	在
7	人
8	有
9	我
10	他
11	这
12	个
13	们
14	中
15	来
16	上
17	大
18	为
19	和
20	国
21	地
22	到
23	以
24	说
25	时
26	要
27	就
28	出
29	会
30	可
31	也
32	你
33	对
34	生
35	能
36	而
37	子
38	那
39	得
40	于
41	着
42	下
43	自
44	之
45	年
46	过
47	发
48	后
49	作
50	里
51	用
52	道
53	行
54	所
55	然
56	家
57	种
58	事
59	成
60	方
61	多
62	经
63	么
64	去
65	法
66	学
67	如
68	都
69	同
70	现
71	当
72	没
73	动
74	面
75	起
76	看
77	定
78	天
79	分
80	还
81	进
82	好
83	小
84	部
85	其
86	些
87	主
88	样
89	理
90	心
91	她
92	本
93	前
94	开
95	但
96	因
97	只
98	从
99	想
100	实
101	日
102	军
103	者
104	意
105	无
106	力
107	它
108	与
109	长
110	把
111	机
112	十
113	民
114	第
115	公
116	此
117	已
118	工
119	使
120	情
121	明
122	性
123	知
124	全
125	三
126	又
127	关
128	点
129	正
130	业
131	外
132	将
133	两
134	高
135	间
136	由
137	问
138	很
139	最
140	重
141	并
142	物
143	手
144	应
145	战
146	向
147	头
148	文
149	体
150	政
151	美
152	相
153	见
154	被
155	利
156	什
157	二
158	等
159	产
160	或
161	新
162	己
163	制
164	身
165	果
166	加
167	西
168	斯
169	月
170	话
171	合
172	回
173	特
174	代
175	内
176	信
177	表
178	化
179	老
180	给
181	世
182	位
183	次
184	度

No.	Character
185	门
186	任
187	常
188	先
189	海
190	通
191	教
192	儿
193	原
194	东
195	声
196	提
197	立
198	及
199	比
200	员
201	解
202	水
203	名
204	真
205	论
206	处
207	走
208	义
209	各
210	入
211	几
212	口
213	认
214	条
215	平
216	系
217	气
218	题
219	活
220	尔
221	更
222	别
223	打
224	女
225	变
226	四
227	神
228	总
229	何
230	电
231	数
232	安
233	少
234	报

No.	Character
235	才
236	结
237	反
238	受
239	目
240	太
241	量
242	再
243	感
244	建
245	务
246	做
247	接
248	必
249	场
250	件
251	计
252	管
253	期
254	市
255	直
256	德
257	资
258	命
259	山
260	金
261	指
262	克
263	许
264	统
265	区
266	保
267	至
268	队
269	形
270	社
271	便
272	空
273	决
274	治
275	展
276	马
277	科
278	司
279	五
280	基
281	眼
282	书
283	非
284	则

No.	Character
285	听
286	白
287	却
288	界
289	达
290	光
291	放
292	强
293	即
294	像
295	难
296	且
297	权
298	思
299	王
300	象
301	完
302	设
303	式
304	色
305	路
306	记
307	南
308	品
309	住
310	告
311	类
312	求
313	据
314	程
315	北
316	边
317	死
318	张
319	该
320	交
321	规
322	万
323	取
324	拉
325	格
326	望
327	觉
328	术
329	领
330	共
331	确
332	传
333	师
334	观

No.	Character
335	清
336	今
337	切
338	院
339	让
340	识
341	候
342	带
343	导
344	争
345	运
346	笑
347	飞
348	风
349	步
350	改
351	收
352	根
353	干
354	造
355	言
356	联
357	持
358	组
359	每
360	济
361	车
362	亲
363	极
364	林
365	服
366	快
367	办
368	议
369	往
370	元
371	英
372	士
373	证
374	近
375	失
376	转
377	夫
378	令
379	准
380	布
381	始
382	怎
383	呢
384	存

No.	Character
385	未
386	远
387	叫
388	台
389	单
390	影
391	具
392	罗
393	字
394	爱
395	击
396	流
397	备
398	兵
399	连
400	调
401	深
402	商
403	算
404	质
405	团
406	集
407	百
408	需
409	价
410	花
411	党
412	华
413	城
414	石
415	级
416	整
417	府
418	离
419	况
420	亚
421	请
422	技
423	际
424	约
425	示
426	复
427	病
428	息
429	究
430	线
431	似
432	官
433	火
434	断

No.	Character
435	精
436	满
437	支
438	视
439	消
440	越
441	器
442	容
443	照
444	须
445	九
446	增
447	研
448	写
449	称
450	企
451	八
452	功
453	吗
454	包
455	片
456	史
457	委
458	乎
459	查
460	轻
461	易
462	早
463	曾
464	除
465	农
466	找
467	装
468	广
469	显
470	吧
471	阿
472	李
473	标
474	谈
475	吃
476	图
477	念
478	六
479	引
480	历
481	首
482	医
483	局
484	突

No.	Character
485	专
486	费
487	号
488	尽
489	另
490	周
491	较
492	注
493	语
494	仅
495	考
496	落
497	青
498	随
499	选
500	列
501	武
502	红
503	响
504	虽
505	推
506	势
507	参
508	希
509	古
510	众
511	构
512	房
513	半
514	节
515	土
516	投
517	某
518	案
519	黑
520	维
521	革
522	划
523	敌
524	致
525	陈
526	律
527	足
528	态
529	护
530	七
531	兴
532	派
533	孩
534	验

No.	Character
535	责
536	营
537	星
538	够
539	章
540	音
541	跟
542	志
543	底
544	站
545	严
546	巴
547	例
548	防
549	族
550	供
551	效
552	续
553	施
554	留
555	讲
556	型
557	料
558	终
559	答
560	紧
561	黄
562	绝
563	奇
564	察
565	母
566	京
567	段
568	依
569	批
570	群
571	项
572	故
573	按
574	河
575	米
576	围
577	江
578	织
579	害
580	斗
581	双
582	境
583	客
584	纪

No.	Character
585	采
586	举
587	杀
588	攻
589	父
590	苏
591	密
592	低
593	朝
594	友
595	诉
596	止
597	细
598	愿
599	千
600	值
601	仍
602	男
603	钱
604	破
605	网
606	热
607	助
608	倒
609	育
610	属
611	坐
612	帝
613	限
614	船
615	脸
616	职
617	速
618	刻
619	乐
620	否
621	刚
622	威
623	毛
624	状
625	率
626	甚
627	独
628	球
629	般
630	普
631	怕
632	弹
633	校
634	苦

No.	Character
635	创
636	假
637	久
638	错
639	承
640	印
641	晚
642	兰
643	试
644	股
645	拿
646	脑
647	预
648	谁
649	益
650	阳
651	若
652	哪
653	微
654	尼
655	继
656	送
657	急
658	血
659	惊
660	伤
661	素
662	药
663	适
664	波
665	夜
666	省
667	初
668	喜
669	卫
670	源
671	食
672	险
673	待
674	述
675	陆
676	习
677	置
678	居
679	劳
680	财
681	环
682	排
683	福
684	纳

No.	Character
685	欢
686	雷
687	警
688	获
689	模
690	充
691	负
692	云
693	停
694	木
695	游
696	龙
697	树
698	疑
699	层
700	冷
701	洲
702	冲
703	射
704	略
705	范
706	竟
707	句
708	室
709	异
710	激
711	汉
712	村
713	哈
714	策
715	演
716	简
717	卡
718	罪
719	判
720	担
721	州
722	静
723	退
724	既
725	衣
726	您
727	宗
728	积
729	余
730	痛
731	检
732	差
733	富
734	灵

No.	Character
735	协
736	角
737	占
738	配
739	征
740	修
741	皮
742	挥
743	胜
744	降
745	阶
746	审
747	沉
748	坚
749	善
750	妈
751	刘
752	读
753	啊
754	超
755	免
756	压
757	银
758	买
759	皇
760	养
761	伊
762	怀
763	执
764	副
765	乱
766	抗
767	犯
768	追
769	帮
770	宣
771	佛
772	岁
773	航
774	优
775	怪
776	香
777	著
778	田
779	铁
780	控
781	税
782	左
783	右
784	份

No.	Character
785	穿
786	艺
787	背
788	阵
789	草
790	脚
791	概
792	恶
793	块
794	顿
795	敢
796	守
797	酒
798	岛
799	托
800	央
801	户
802	烈
803	洋
804	哥
805	索
806	胡
807	款
808	靠
809	评
810	版
811	宝
812	座
813	释
814	景
815	顾
816	弟
817	登
818	货
819	互
820	付
821	伯
822	慢
823	欧
824	换
825	闻
826	危
827	忙
828	核
829	暗
830	姐
831	介
832	坏
833	讨
834	丽

No.	Character
835	良
836	序
837	升
838	监
839	临
840	亮
841	露
842	永
843	呼
844	味
845	野
846	架
847	域
848	沙
849	掉
850	括
851	舰
852	鱼
853	杂
854	误
855	湾
856	吉
857	减
858	编
859	楚
860	肯
861	测
862	败
863	屋
864	跑
865	梦
866	散
867	温
868	困
869	剑
870	渐
871	封
872	救
873	贵
874	枪
875	缺
876	楼
877	县
878	尚
879	毫
880	移
881	娘
882	朋
883	画
884	班

No.	Character
885	智
886	亦
887	耳
888	恩
889	短
890	掌
891	恐
892	遗
893	固
894	席
895	松
896	秘
897	谢
898	鲁
899	遇
900	康
901	虑
902	幸
903	均
904	销
905	钟
906	诗
907	藏
908	赶
909	剧
910	票
911	损
912	忽
913	巨
914	炮
915	旧
916	端
917	探
918	湖
919	录
920	叶
921	春
922	乡
923	附
924	吸
925	予
926	礼
927	港
928	雨
929	呀
930	板
931	庭
932	妇
933	归
934	睛

No.	Character
935	饭
936	额
937	含
938	顺
939	输
940	摇
941	招
942	婚
943	脱
944	补
945	谓
946	督
947	毒
948	油
949	疗
950	旅
951	泽
952	材
953	灭
954	逐
955	莫
956	笔
957	亡
958	鲜
959	词
960	圣
961	择
962	寻
963	厂
964	睡
965	博
966	勒
967	烟
968	授
969	诺
970	伦
971	岸
972	奥
973	唐
974	卖
975	俄
976	炸
977	载
978	洛
979	健
980	堂
981	旁
982	宫
983	喝
984	借

No.	Character
985	君
986	禁
987	阴
988	园
989	谋
990	宋
991	避
992	抓
993	荣
994	姑
995	孙
996	逃
997	牙
998	束
999	跳
1000	顶
1001	玉
1002	镇
1003	雪
1004	午
1005	练
1006	迫
1007	爷
1008	篇
1009	肉
1010	嘴
1011	馆
1012	遍
1013	凡
1014	础
1015	洞
1016	卷
1017	坦
1018	牛
1019	宁
1020	纸
1021	诸
1022	训
1023	私
1024	庄
1025	祖
1026	丝
1027	翻
1028	暴
1029	森
1030	塔
1031	默
1032	握
1033	戏
1034	隐
1035	熟
1036	骨
1037	访
1038	弱
1039	蒙
1040	歌
1041	店
1042	鬼
1043	软
1044	典
1045	欲
1046	萨
1047	伙
1048	遭
1049	盘
1050	爸
1051	扩
1052	盖
1053	弄
1054	雄
1055	稳
1056	忘
1057	亿
1058	刺
1059	拥
1060	徒
1061	姆
1062	杨
1063	齐
1064	赛
1065	趣
1066	曲
1067	刀
1068	床
1069	迎
1070	冰
1071	虚
1072	玩
1073	析
1074	窗
1075	醒
1076	妻
1077	透
1078	购
1079	替
1080	塞
1081	努
1082	休
1083	虎
1084	扬
1085	途
1086	侵
1087	刑
1088	绿
1089	兄
1090	迅
1091	套
1092	贸
1093	毕
1094	唯
1095	谷
1096	轮
1097	库
1098	迹
1099	尤
1100	竞
1101	街
1102	促
1103	延
1104	震
1105	弃
1106	甲
1107	伟
1108	麻
1109	川
1110	申
1111	缓
1112	潜
1113	闪
1114	售
1115	灯
1116	针
1117	哲
1118	络
1119	抵
1120	朱
1121	埃
1122	抱
1123	鼓
1124	植
1125	纯
1126	夏
1127	忍
1128	页
1129	杰
1130	筑
1131	折
1132	郑
1133	贝
1134	尊
1135	吴
1136	秀
1137	混
1138	臣
1139	雅
1140	振
1141	染
1142	盛
1143	怒
1144	舞
1145	圆
1146	搞
1147	狂
1148	措
1149	姓
1150	残
1151	秋
1152	培
1153	迷
1154	诚
1155	宽
1156	宇
1157	猛
1158	摆
1159	梅
1160	毁
1161	伸
1162	摩
1163	盟
1164	末
1165	乃
1166	悲
1167	拍
1168	丁
1169	赵
1170	硬
1171	麦
1172	蒋
1173	操
1174	耶
1175	阻
1176	订
1177	彩
1178	抽
1179	赞
1180	魔
1181	纷
1182	沿
1183	喊
1184	违

No.	Character
1185	妹
1186	浪
1187	汇
1188	币
1189	丰
1190	蓝
1191	殊
1192	献
1193	桌
1194	啦
1195	瓦
1196	莱
1197	援
1198	译
1199	夺
1200	汽
1201	烧
1202	距
1203	裁
1204	偏
1205	符
1206	勇
1207	触
1208	课
1209	敬
1210	哭
1211	懂
1212	墙
1213	袭
1214	召
1215	罚
1216	侠
1217	厅
1218	拜
1219	巧
1220	侧
1221	韩
1222	冒
1223	债
1224	曼
1225	融
1226	惯
1227	享
1228	戴
1229	童
1230	犹
1231	乘
1232	挂
1233	奖
1234	绍

No.	Character
1235	厚
1236	纵
1237	障
1238	讯
1239	涉
1240	彻
1241	刊
1242	丈
1243	爆
1244	乌
1245	役
1246	描
1247	洗
1248	玛
1249	患
1250	妙
1251	镜
1252	唱
1253	烦
1254	签
1255	仙
1256	彼
1257	弗
1258	症
1259	仿
1260	倾
1261	牌
1262	陷
1263	鸟
1264	轰
1265	咱
1266	菜
1267	闭
1268	奋
1269	庆
1270	撤
1271	泪
1272	茶
1273	疾
1274	缘
1275	播
1276	朗
1277	杜
1278	奶
1279	季
1280	丹
1281	狗
1282	尾
1283	仪
1284	偷

No.	Character
1285	奔
1286	珠
1287	虫
1288	驻
1289	孔
1290	宜
1291	艾
1292	桥
1293	淡
1294	翼
1295	恨
1296	繁
1297	寒
1298	伴
1299	叹
1300	旦
1301	愈
1302	潮
1303	粮
1304	缩
1305	罢
1306	聚
1307	径
1308	恰
1309	挑
1310	袋
1311	灰
1312	捕
1313	徐
1314	珍
1315	幕
1316	映
1317	裂
1318	泰
1319	隔
1320	启
1321	尖
1322	忠
1323	累
1324	炎
1325	暂
1326	估
1327	泛
1328	荒
1329	偿
1330	横
1331	拒
1332	瑞
1333	忆
1334	孤

No.	Character
1335	鼻
1336	闹
1337	羊
1338	呆
1339	厉
1340	衡
1341	胞
1342	零
1343	穷
1344	舍
1345	码
1346	赫
1347	婆
1348	魂
1349	灾
1350	洪
1351	腿
1352	胆
1353	津
1354	俗
1355	辩
1356	胸
1357	晓
1358	劲
1359	贫
1360	仁
1361	偶
1362	辑
1363	邦
1364	恢
1365	赖
1366	圈
1367	摸
1368	仰
1369	润
1370	堆
1371	碰
1372	艇
1373	稍
1374	迟
1375	辆
1376	废
1377	净
1378	凶
1379	署
1380	壁
1381	御
1382	奉
1383	旋
1384	冬

No.	Character
1385	矿
1386	抬
1387	蛋
1388	晨
1389	伏
1390	吹
1391	鸡
1392	倍
1393	糊
1394	秦
1395	盾
1396	杯
1397	租
1398	骑
1399	乏
1400	隆
1401	诊
1402	奴
1403	摄
1404	丧
1405	污
1406	渡
1407	旗
1408	甘
1409	耐
1410	凭
1411	扎
1412	抢
1413	绪
1414	粗
1415	肩
1416	梁
1417	幻
1418	菲
1419	皆
1420	碎
1421	宙
1422	叔
1423	岩
1424	荡
1425	综
1426	爬
1427	荷
1428	悉
1429	蒂
1430	返
1431	井
1432	壮
1433	薄
1434	悄
1435	扫
1436	敏
1437	碍
1438	殖
1439	详
1440	迪
1441	矛
1442	霍
1443	允
1444	幅
1445	撒
1446	剩
1447	凯
1448	颗
1449	骂
1450	赏
1451	液
1452	番
1453	箱
1454	贴
1455	漫
1456	酸
1457	郎
1458	腰
1459	舒
1460	眉
1461	忧
1462	浮
1463	辛
1464	恋
1465	餐
1466	吓
1467	挺
1468	励
1469	辞
1470	艘
1471	键
1472	伍
1473	峰
1474	尺
1475	昨
1476	黎
1477	辈
1478	贯
1479	侦
1480	滑
1481	券
1482	崇
1483	扰
1484	宪
1485	绕
1486	趋
1487	慈
1488	乔
1489	阅
1490	汗
1491	枝
1492	拖
1493	墨
1494	胁
1495	插
1496	箭
1497	腊
1498	粉
1499	泥
1500	氏
1501	彭
1502	拔
1503	骗
1504	凤
1505	慧
1506	媒
1507	佩
1508	愤
1509	扑
1510	龄
1511	驱
1512	惜
1513	豪
1514	掩
1515	兼
1516	跃
1517	尸
1518	肃
1519	帕
1520	驶
1521	堡
1522	届
1523	欣
1524	惠
1525	册
1526	储
1527	飘
1528	桑
1529	闲
1530	惨
1531	洁
1532	踪
1533	勃
1534	宾
1535	频
1536	仇
1537	磨
1538	递
1539	邪
1540	撞
1541	拟
1542	滚
1543	奏
1544	巡
1545	颜
1546	剂
1547	绩
1548	贡
1549	疯
1550	坡
1551	瞧
1552	截
1553	燃
1554	焦
1555	殿
1556	伪
1557	柳
1558	锁
1559	逼
1560	颇
1561	昏
1562	劝
1563	呈
1564	搜
1565	勤
1566	戒
1567	驾
1568	漂
1569	饮
1570	曹
1571	朵
1572	仔
1573	柔
1574	俩
1575	孟
1576	腐
1577	幼
1578	践
1579	籍
1580	牧
1581	凉
1582	牲
1583	佳
1584	娜

No.	Character
1585	浓
1586	芳
1587	稿
1588	竹
1589	腹
1590	跌
1591	逻
1592	垂
1593	遵
1594	脉
1595	貌
1596	柏
1597	狱
1598	猜
1599	怜
1600	惑
1601	陶
1602	兽
1603	帐
1604	饰
1605	贷
1606	昌
1607	叙
1608	躺
1609	钢
1610	沟
1611	寄
1612	扶
1613	铺
1614	邓
1615	寿
1616	惧
1617	询
1618	汤
1619	盗
1620	肥
1621	尝
1622	匆
1623	辉
1624	奈
1625	扣
1626	廷
1627	澳
1628	嘛
1629	董
1630	迁
1631	凝
1632	慰
1633	厌
1634	脏

No.	Character
1635	腾
1636	幽
1637	怨
1638	鞋
1639	丢
1640	埋
1641	泉
1642	涌
1643	辖
1644	躲
1645	晋
1646	紫
1647	艰
1648	魏
1649	吾
1650	慌
1651	祝
1652	邮
1653	吐
1654	狠
1655	鉴
1656	曰
1657	械
1658	咬
1659	邻
1660	赤
1661	挤
1662	弯
1663	椅
1664	陪
1665	割
1666	揭
1667	韦
1668	悟
1669	聪
1670	雾
1671	锋
1672	梯
1673	猫
1674	祥
1675	阔
1676	誉
1677	筹
1678	丛
1679	牵
1680	鸣
1681	沈
1682	阁
1683	穆
1684	屈

No.	Character
1685	旨
1686	袖
1687	猎
1688	臂
1689	蛇
1690	贺
1691	柱
1692	抛
1693	鼠
1694	瑟
1695	戈
1696	牢
1697	逊
1698	迈
1699	欺
1700	吨
1701	琴
1702	衰
1703	瓶
1704	恼
1705	燕
1706	仲
1707	诱
1708	狼
1709	池
1710	疼
1711	卢
1712	仗
1713	冠
1714	粒
1715	遥
1716	吕
1717	玄
1718	尘
1719	冯
1720	抚
1721	浅
1722	敦
1723	纠
1724	钻
1725	晶
1726	岂
1727	峡
1728	苍
1729	喷
1730	耗
1731	凌
1732	敲
1733	菌
1734	赔

No.	Character
1735	涂
1736	粹
1737	扁
1738	亏
1739	寂
1740	煤
1741	熊
1742	恭
1743	湿
1744	循
1745	暖
1746	糖
1747	赋
1748	抑
1749	秩
1750	帽
1751	哀
1752	宿
1753	踏
1754	烂
1755	袁
1756	侯
1757	抖
1758	夹
1759	昆
1760	肝
1761	擦
1762	猪
1763	炼
1764	恒
1765	慎
1766	搬
1767	纽
1768	纹
1769	玻
1770	渔
1771	磁
1772	铜
1773	齿
1774	跨
1775	押
1776	怖
1777	漠
1778	疲
1779	叛
1780	遣
1781	兹
1782	祭
1783	醉
1784	拳

No.	Character
1785	弥
1786	斜
1787	档
1788	稀
1789	捷
1790	肤
1791	疫
1792	肿
1793	豆
1794	削
1795	岗
1796	晃
1797	吞
1798	宏
1799	癌
1800	肚
1801	隶
1802	履
1803	涨
1804	耀
1805	扭
1806	坛
1807	拨
1808	沃
1809	绘
1810	伐
1811	堪
1812	仆
1813	郭
1814	牺
1815	歼
1816	墓
1817	雇
1818	廉
1819	契
1820	拼
1821	惩
1822	捉
1823	覆
1824	刷
1825	劫
1826	嫌
1827	瓜
1828	歇
1829	雕
1830	闷
1831	乳
1832	串
1833	娃
1834	缴

No.	Character
1835	唤
1836	赢
1837	莲
1838	霸
1839	桃
1840	妥
1841	瘦
1842	搭
1843	赴
1844	岳
1845	嘉
1846	舱
1847	俊
1848	址
1849	庞
1850	耕
1851	锐
1852	缝
1853	悔
1854	邀
1855	玲
1856	惟
1857	斥
1858	宅
1859	添
1860	挖
1861	呵
1862	讼
1863	氧
1864	浩
1865	羽
1866	斤
1867	酷
1868	掠
1869	妖
1870	祸
1871	侍
1872	乙
1873	妨
1874	贪
1875	挣
1876	汪
1877	尿
1878	莉
1879	悬
1880	唇
1881	翰
1882	仓
1883	轨
1884	枚

No.	Character
1885	盐
1886	览
1887	傅
1888	帅
1889	庙
1890	芬
1891	屏
1892	寺
1893	胖
1894	璃
1895	愚
1896	滴
1897	疏
1898	萧
1899	姿
1900	颤
1901	丑
1902	劣
1903	柯
1904	寸
1905	扔
1906	盯
1907	辱
1908	匹
1909	俱
1910	辨
1911	饿
1912	蜂
1913	哦
1914	腔
1915	郁
1916	溃
1917	谨
1918	糟
1919	葛
1920	苗
1921	肠
1922	忌
1923	溜
1924	鸿
1925	爵
1926	鹏
1927	鹰
1928	笼
1929	丘
1930	桂
1931	滋
1932	聊
1933	挡
1934	纲

No.	Character
1935	肌
1936	茨
1937	壳
1938	痕
1939	碗
1940	穴
1941	膀
1942	卓
1943	贤
1944	卧
1945	膜
1946	毅
1947	锦
1948	欠
1949	哩
1950	函
1951	茫
1952	昂
1953	薛
1954	皱
1955	夸
1956	豫
1957	胃
1958	舌
1959	剥
1960	傲
1961	拾
1962	窝
1963	睁
1964	携
1965	陵
1966	哼
1967	棉
1968	晴
1969	铃
1970	填
1971	饲
1972	渴
1973	吻
1974	扮
1975	逆
1976	脆
1977	喘
1978	罩
1979	卜
1980	炉
1981	柴
1982	愉
1983	绳
1984	胎

No.	Character
1985	蓄
1986	眠
1987	竭
1988	喂
1989	傻
1990	慕
1991	浑
1992	奸
1993	扇
1994	柜
1995	悦
1996	拦
1997	诞
1998	饱
1999	乾
2000	泡
2001	贼
2002	亭
2003	夕
2004	爹
2005	酬
2006	儒
2007	姻
2008	卵
2009	氛
2010	泄
2011	杆
2012	挨
2013	僧
2014	蜜
2015	吟
2016	猩
2017	遂
2018	狭
2019	肖
2020	甜
2021	霞
2022	驳
2023	裕
2024	顽
2025	於
2026	摘
2027	矮
2028	秒
2029	卿
2030	畜
2031	咽
2032	披
2033	辅
2034	勾

No.	Character
2035	盆
2036	疆
2037	赌
2038	塑
2039	畏
2040	吵
2041	囊
2042	嗯
2043	泊
2044	肺
2045	骤
2046	缠
2047	冈
2048	羞
2049	瞪
2050	吊
2051	贾
2052	漏
2053	斑
2054	涛
2055	悠
2056	鹿
2057	俘
2058	锡
2059	卑
2060	葬
2061	铭
2062	滩
2063	嫁
2064	催
2065	璇
2066	翅
2067	盒
2068	蛮
2069	矣
2070	潘
2071	歧
2072	赐
2073	鲍
2074	锅
2075	廊
2076	拆
2077	灌
2078	勉
2079	盲
2080	宰
2081	佐
2082	啥
2083	胀
2084	扯

No.	Character
2085	禧
2086	辽
2087	抹
2088	筒
2089	棋
2090	裤
2091	唉
2092	朴
2093	咐
2094	孕
2095	誓
2096	喉
2097	妄
2098	拘
2099	链
2100	驰
2101	栏
2102	逝
2103	窃
2104	艳
2105	臭
2106	纤
2107	玑
2108	棵
2109	趁
2110	匠
2111	盈
2112	翁
2113	愁
2114	瞬
2115	婴
2116	孝
2117	颈
2118	倘
2119	浙
2120	谅
2121	蔽
2122	畅
2123	赠
2124	妮
2125	莎
2126	尉
2127	冻
2128	跪
2129	闯
2130	葡
2131	後
2132	厨
2133	鸭
2134	颠

No.	Character
2135	遮
2136	谊
2137	圳
2138	吁
2139	仑
2140	辟
2141	瘤
2142	嫂
2143	陀
2144	框
2145	谭
2146	亨
2147	钦
2148	庸
2149	歉
2150	芝
2151	吼
2152	甫
2153	衫
2154	摊
2155	宴
2156	嘱
2157	衷
2158	娇
2159	陕
2160	矩
2161	浦
2162	讶
2163	耸
2164	裸
2165	碧
2166	摧
2167	薪
2168	淋
2169	耻
2170	胶
2171	屠
2172	鹅
2173	饥
2174	盼
2175	脖
2176	虹
2177	翠
2178	崩
2179	账
2180	萍
2181	逢
2182	赚
2183	撑
2184	翔

No.	Character
2185	倡
2186	绵
2187	猴
2188	枯
2189	巫
2190	昭
2191	怔
2192	渊
2193	凑
2194	溪
2195	蠢
2196	禅
2197	阐
2198	旺
2199	寓
2200	藤
2201	匪
2202	伞
2203	碑
2204	挪
2205	琼
2206	脂
2207	谎
2208	慨
2209	菩
2210	萄
2211	狮
2212	掘
2213	抄
2214	岭
2215	晕
2216	逮
2217	砍
2218	掏
2219	狄
2220	晰
2221	罕
2222	挽
2223	脾
2224	舟
2225	痴
2226	蔡
2227	剪
2228	脊
2229	弓
2230	懒
2231	叉
2232	拐
2233	喃
2234	僚

No.	Character
2235	捐
2236	姊
2237	骚
2238	拓
2239	歪
2240	粘
2241	柄
2242	坑
2243	陌
2244	窄
2245	湘
2246	兆
2247	崖
2248	骄
2249	刹
2250	鞭
2251	芒
2252	筋
2253	聘
2254	钩
2255	棍
2256	嚷
2257	腺
2258	弦
2259	焰
2260	耍
2261	俯
2262	厘
2263	愣
2264	厦
2265	恳
2266	饶
2267	钉
2268	寡
2269	憾
2270	摔
2271	叠
2272	惹
2273	喻
2274	谱
2275	愧
2276	煌
2277	徽
2278	溶
2279	坠
2280	煞
2281	巾
2282	滥
2283	洒
2284	堵

No.	Character
2285	瓷
2286	咒
2287	姨
2288	棒
2289	郡
2290	浴
2291	媚
2292	稣
2293	淮
2294	哎
2295	屁
2296	漆
2297	淫
2298	巢
2299	吩
2300	撰
2301	啸
2302	滞
2303	玫
2304	硕
2305	钓
2306	蝶
2307	膝
2308	姚
2309	茂
2310	躯
2311	吏
2312	猿
2313	寨
2314	恕
2315	渠
2316	戚
2317	辰
2318	舶
2319	颁
2320	惶
2321	狐
2322	讽
2323	笨
2324	袍
2325	嘲
2326	啡
2327	泼
2328	衔
2329	倦
2330	涵
2331	雀
2332	旬
2333	僵
2334	撕

No.	Character
2335	肢
2336	垄
2337	夷
2338	逸
2339	茅
2340	侨
2341	舆
2342	窑
2343	涅
2344	蒲
2345	谦
2346	杭
2347	噢
2348	弊
2349	勋
2350	刮
2351	郊
2352	凄
2353	捧
2354	浸
2355	砖
2356	鼎
2357	篮
2358	蒸
2359	饼
2360	亩
2361	肾
2362	陡
2363	爪
2364	兔
2365	殷
2366	贞
2367	荐
2368	哑
2369	炭
2370	坟
2371	眨
2372	搏
2373	咳
2374	拢
2375	舅
2376	昧
2377	擅
2378	爽
2379	咖
2380	搁
2381	禄
2382	雌
2383	哨
2384	巩

No.	Character
2385	绢
2386	螺
2387	裹
2388	昔
2389	轩
2390	谬
2391	谍
2392	龟
2393	媳
2394	姜
2395	瞎
2396	冤
2397	鸦
2398	蓬
2399	巷
2400	琳
2401	栽
2402	沾
2403	诈
2404	斋
2405	瞒
2406	彪
2407	厄
2408	咨
2409	纺
2410	罐
2411	桶
2412	壤
2413	糕
2414	颂
2415	膨
2416	谐
2417	垒
2418	咕
2419	隙
2420	辣
2421	绑
2422	宠
2423	嘿
2424	兑
2425	霉
2426	挫
2427	稽
2428	辐
2429	乞
2430	纱
2431	裙
2432	嘻
2433	哇
2434	绣

No.	Character
2435	杖
2436	塘
2437	衍
2438	轴
2439	攀
2440	膊
2441	譬
2442	斌
2443	祈
2444	踢
2445	肆
2446	坎
2447	轿
2448	棚
2449	泣
2450	屡
2451	躁
2452	邱
2453	凰
2454	溢
2455	椎
2456	砸
2457	趟
2458	帘
2459	帆
2460	栖
2461	窜
2462	丸
2463	斩
2464	堤
2465	塌
2466	贩
2467	厢
2468	掀
2469	喀
2470	乖
2471	谜
2472	捏
2473	阎
2474	滨
2475	虏
2476	匙
2477	芦
2478	苹
2479	卸
2480	沼
2481	钥
2482	株
2483	祷
2484	剖

No.	Character
2485	熙
2486	哗
2487	劈
2488	怯
2489	棠
2490	胳
2491	桩
2492	瑰
2493	娱
2494	娶
2495	沫
2496	嗓
2497	蹲
2498	焚
2499	淘
2500	嫩
2501	韵
2502	衬
2503	匈
2504	钧
2505	竖
2506	峻
2507	豹
2508	捞
2509	菊
2510	鄙
2511	魄
2512	兜
2513	哄
2514	颖
2515	镑
2516	屑
2517	蚁
2518	壶
2519	怡
2520	渗
2521	秃
2522	迦
2523	旱
2524	哟
2525	咸
2526	焉
2527	谴
2528	宛
2529	稻
2530	铸
2531	锻
2532	伽
2533	詹
2534	毙

No.	Character
2535	恍
2536	贬
2537	烛
2538	骇
2539	芯
2540	汁
2541	桓
2542	坊
2543	驴
2544	朽
2545	靖
2546	佣
2547	汝
2548	碌
2549	迄
2550	冀
2551	荆
2552	崔
2553	雁
2554	绅
2555	珊
2556	榜
2557	诵
2558	傍
2559	彦
2560	醇
2561	笛
2562	禽
2563	勿
2564	娟
2565	瞄
2566	幢
2567	寇
2568	睹
2569	贿
2570	踩
2571	霆
2572	呜
2573	拱
2574	妃
2575	蔑
2576	谕
2577	缚
2578	诡
2579	篷
2580	淹
2581	腕
2582	煮
2583	倩
2584	卒

No.	Character
2585	勘
2586	馨
2587	逗
2588	甸
2589	贱
2590	炒
2591	灿
2592	敞
2593	蜡
2594	囚
2595	栗
2596	辜
2597	垫
2598	妒
2599	魁
2600	谣
2601	寞
2602	蜀
2603	甩
2604	涯
2605	枕
2606	丐
2607	泳
2608	奎
2609	泌
2610	逾
2611	叮
2612	黛
2613	燥
2614	掷
2615	藉
2616	枢
2617	憎
2618	鲸
2619	弘
2620	倚
2621	侮
2622	藩
2623	拂
2624	鹤
2625	蚀
2626	浆
2627	芙
2628	垃
2629	烤
2630	晒
2631	霜
2632	剿
2633	蕴
2634	圾
2635	绸
2636	屿
2637	氢
2638	驼
2639	妆
2640	捆
2641	铅
2642	逛
2643	淑
2644	榴
2645	丙
2646	痒
2647	钞
2648	蹄
2649	犬
2650	躬
2651	昼
2652	藻
2653	蛛
2654	褐
2655	颊
2656	奠
2657	募
2658	耽
2659	蹈
2660	陋
2661	侣
2662	魅
2663	岚
2664	侄
2665	虐
2666	堕
2667	陛
2668	莹
2669	荫
2670	狡
2671	阀
2672	绞
2673	膏
2674	垮
2675	茎
2676	缅
2677	喇
2678	绒
2679	搅
2680	凳
2681	梭
2682	丫
2683	姬
2684	诏
2685	钮
2686	棺
2687	耿
2688	缔
2689	懈
2690	嫉
2691	灶
2692	匀
2693	嗣
2694	鸽
2695	澡
2696	凿
2697	纬
2698	沸
2699	畴
2700	刃
2701	遏
2702	烁
2703	嗅
2704	叭
2705	熬
2706	瞥
2707	骸
2708	奢
2709	拙
2710	栋
2711	毯
2712	桐
2713	砂
2714	莽
2715	泻
2716	坪
2717	梳
2718	杉
2719	晤
2720	稚
2721	蔬
2722	蝇
2723	捣
2724	顷
2725	麽
2726	尴
2727	镖
2728	诧
2729	尬
2730	硫
2731	嚼
2732	羡
2733	沦
2734	沪
2735	旷
2736	彬
2737	芽
2738	狸
2739	冥
2740	碳
2741	咧
2742	惕
2743	暑
2744	咯
2745	萝
2746	汹
2747	腥
2748	窥
2749	俺
2750	潭
2751	崎
2752	麟
2753	捡
2754	拯
2755	厥
2756	澄
2757	萎
2758	哉
2759	涡
2760	滔
2761	暇
2762	溯
2763	鳞
2764	酿
2765	茵
2766	愕
2767	瞅
2768	暮
2769	衙
2770	诫
2771	斧
2772	兮
2773	焕
2774	棕
2775	佑
2776	嘶
2777	妓
2778	喧
2779	蓉
2780	删
2781	樱
2782	伺
2783	嗡
2784	娥

No.	Character
2785	梢
2786	坝
2787	蚕
2788	敷
2789	澜
2790	杏
2791	绥
2792	冶
2793	庇
2794	挠
2795	搂
2796	倏
2797	聂
2798	婉
2799	噪
2800	稼
2801	鳍
2802	菱
2803	盏
2804	匿
2805	吱
2806	寝
2807	揽
2808	髓
2809	秉
2810	哺
2811	矢
2812	啪
2813	帜
2814	邵
2815	嗽
2816	挟
2817	缸
2818	揉
2819	腻
2820	驯
2821	缆
2822	晌
2823	瘫
2824	贮
2825	觅
2826	朦
2827	僻
2828	隋
2829	蔓
2830	咋
2831	嵌
2832	虔
2833	畔
2834	琐
2835	碟
2836	涩
2837	胧
2838	嘟
2839	蹦
2840	冢
2841	浏
2842	裔
2843	襟
2844	叨
2845	诀
2846	旭
2847	虾
2848	簿
2849	啤
2850	擒
2851	枣
2852	嘎
2853	苑
2854	牟
2855	呕
2856	骆
2857	凸
2858	熄
2859	兀
2860	喔
2861	裳
2862	凹
2863	赎
2864	屯
2865	膛
2866	浇
2867	灼
2868	裘
2869	砰
2870	棘
2871	橡
2872	碱
2873	聋
2874	姥
2875	瑜
2876	毋
2877	娅
2878	沮
2879	萌
2880	俏
2881	黯
2882	撇
2883	粟
2884	粪
2885	尹
2886	苟
2887	癫
2888	蚂
2889	禹
2890	廖
2891	俭
2892	帖
2893	煎
2894	缕
2895	窦
2896	簇
2897	棱
2898	叩
2899	呐
2900	瑶
2901	墅
2902	莺
2903	烫
2904	蛙
2905	歹
2906	伶
2907	葱
2908	哮
2909	眩
2910	坤
2911	廓
2912	讳
2913	啼
2914	乍
2915	瓣
2916	矫
2917	跋
2918	枉
2919	梗
2920	厕
2921	琢
2922	讥
2923	釉
2924	窟
2925	敛
2926	轼
2927	庐
2928	胚
2929	呻
2930	绰
2931	扼
2932	懿
2933	炯
2934	竿
2935	慷
2936	虞
2937	锤
2938	栓
2939	桨
2940	蚊
2941	磅
2942	孽
2943	惭
2944	戳
2945	禀
2946	鄂
2947	馈
2948	垣
2949	溅
2950	咚
2951	钙
2952	礁
2953	彰
2954	豁
2955	眯
2956	磷
2957	雯
2958	墟
2959	迂
2960	瞻
2961	颅
2962	琉
2963	悼
2964	蝴
2965	拣
2966	渺
2967	眷
2968	悯
2969	汰
2970	慑
2971	婶
2972	斐
2973	嘘
2974	镶
2975	炕
2976	宦
2977	趴
2978	绷
2979	窘
2980	襄
2981	珀
2982	嚣
2983	拚
2984	酌

No.	Character
2985	浊
2986	毓
2987	撼
2988	嗜
2989	扛
2990	峭
2991	磕
2992	翘
2993	槽
2994	淌
2995	栅
2996	颓
2997	熏
2998	瑛
2999	颐
3000	忖

Appendix B: MATLAB PROGRAMMING

Appendix B1: Preprocessing

```
%  Preprocessing  includes  binarization,  cropping  &
  normalization


% ==> Binarization: Change image to black & white
BinIm=~im2bw(x);

% ==> Cropping
[mRow,nCol]=size(BinIm);
i=1;j=1;k=1;p=1;
sumRowFront=0;sumRowLast=0;sumColLeft=0;sumColRight=0;
while sumRowFront==0 ||sumRowLast==0 || sumColLeft==0 ||
sumColRight==0
    sumRowFront=sum(BinIm(i,:));
    sumRowLast=sum(BinIm(mRow-j+1,:));
    sumColLeft=sum(BinIm(:,k));
    sumColRight=sum(BinIm(:,nCol-p+1));
    if sumRowFront==0
    i=i+1;
    elseif sumRowLast==0
    j=j+1;
    elseif sumColLeft==0
    k=k+1;
    elseif sumColRight==0
    p=p+1;
    end
end
disp(i);
LastCoor=mRow-j+1;
disp(LastCoor);
disp(k);
RightCoor=nCol-p+1;
disp(RightCoor);

% ==> Normalization
width=RightCoor-k;
height=LastCoor-i;
J=~imcrop(BinIm,[k,i,width,height]);
imshow(imresize(J,[128 128]));
```

Appendix B2: Feature Extraction

```
% Feature extraction includes X-Y graphs decomposition &
  Haar wavelet transform


% ==> X-Y graphs decomposition
flag=1;
[cx1,cy1,c1]=improfile(2^7);
cx=cx1; cy=cy1; c=c1;
if flag==1
save('data1.mat', 'cx','cy', 'c','-v6');
else
load('data1.mat', 'cx', 'cy','c');
cx=[cx; cx1];
cy=[cy;cy1];
c=[c;c1];
save('data1.mat', 'cx','cy', 'c','-v6');
end
flag=flag+1;

% ==> Haar wavelet transform
load('data1.mat', 'cx', 'cy');
sizeCx=size(cx,1);
numLoop = floor(log10(sizeCx)/log10(2));
level=numLoop-5; %Set to between 32 =2^5 size and 64=2^6
for lev=1:level
if lev==1
[a11,d11]=dwt(cx,'haar');
[a12,d12]=dwt(cy,'haar');
save('data2.mat','a11','d11','a12','d12','-v6');
else
load ('data2.mat','a11','d11','a12','d12');
[a11,d11]=dwt(a11,'haar');
[a12,d12]=dwt(a12,'haar');
save('data2.mat','a11','d11','a12','d12','-v6');
end
end
load ('data2.mat','a11','d11','a12','d12');
sizeOfa11=size(a11,1);
if sizeOfa11==64
level = level+1;
[a11,d11]=dwt(a11,'haar');
[a12,d12]=dwt(a12,'haar');
save('data2.mat','a11','d11','a12','d12','-v6');
end
```

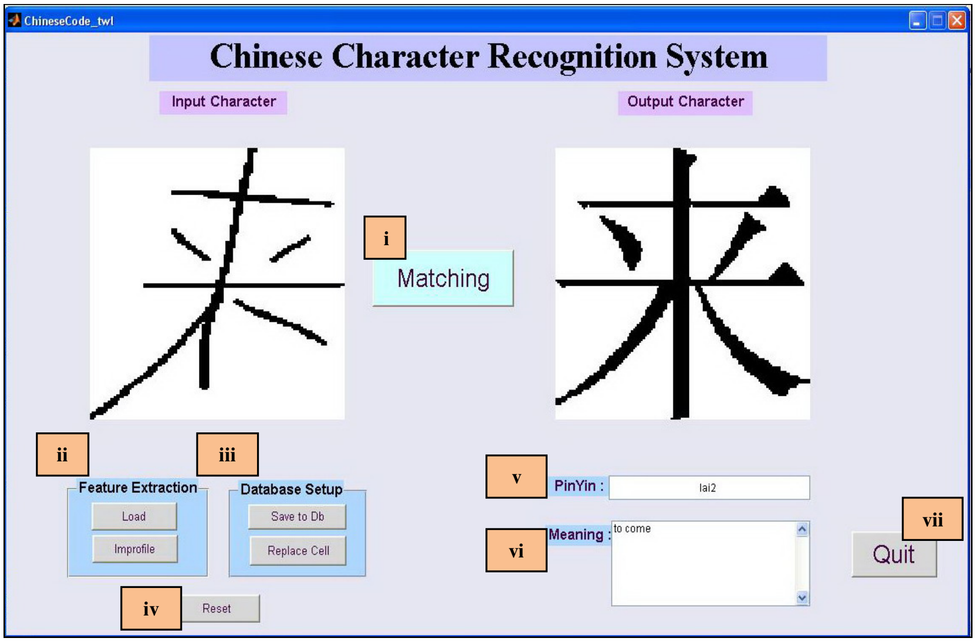
Appendix B3: Classification

```
% Classification includes coarse classification and fine
  classification


% ==> Coarse classification
flag=1;
load database.mat; %Load characters in database
load('data1.mat', 'cx', 'cy','c'); %Load input character
numStroke=(size(cx,1))/128;
[a sizeFeature] = size(feature(1, :)); %To get the index
                                        of characters in
                                        database
for i=1:sizeFeature
NumOfStroke = feature(i).stroke;
NumOfStroke = str2double(NumOfStroke);
i1=i;
tf = isequal(numStroke, NumOfStroke);
if tf ==1 %To get from database all the characters of
          same stroke as input character
if flag==1
save('data4.mat','i1','-v6'); %i1==Index of the character
else
load('data4.mat','i1');
i1=[i1; i];
save('data4.mat','i1','-v6');
end
flag=flag+1;
end
end

% ==> Fine classification
load('data4.mat','i1');
sizPossChar=size(i1,1);
if sizPossChar==0
errordlg('!!!No Match Found1!!','Matching');
else
flag2=1;
load ('data2.mat','a11','d11','a12','d12');
X=[a11,a12]; %Assign approximation coefficients of input
             character to X
for k=1:sizPossChar
num1=i1(k,1);
Y=deal(feature(num1).trend); %Assign approximation
                             coefficients of database
                             character to Y
R2M1=2D_ULFR(Y,X); %Calculate COD of 2D-ULFR between
                   input character and database character
R2M=R2M1;
num=num1;
if flag2==1
save('data5.mat','num','R2M','-v6');
else
```

```
load('data5.mat','num','R2M');
R2M=[R2M; R2M1];
num=[num; num1];
save('data5.mat','num','R2M','-v6');
end
flag2=flag2+1;
end
load('data5.mat','num','R2M');
maxR2M=max(R2M); %Determine the largest R2M values
sizR2M=size(R2M);
for p=1:sizR2M
val=R2M(p,1);
if val==maxR2M
save('data7.mat','p','-v6');
end
end
load('data7.mat','p');
dbPos=i1(p,1);  %Assign the index of matched database
              character to dbPos
axes(handles.axes4);
%Display the image, pronunciation and meaning of the
matched database character
sImg=imread(feature(dbPos).image);
set(handles.edit5,'String',feature(dbPos).pinyin);
set(handles.edit6,'String',feature(dbPos).meaning);
end
```

Appendix B4: MATLAB GUI for the Proposed HCCR System

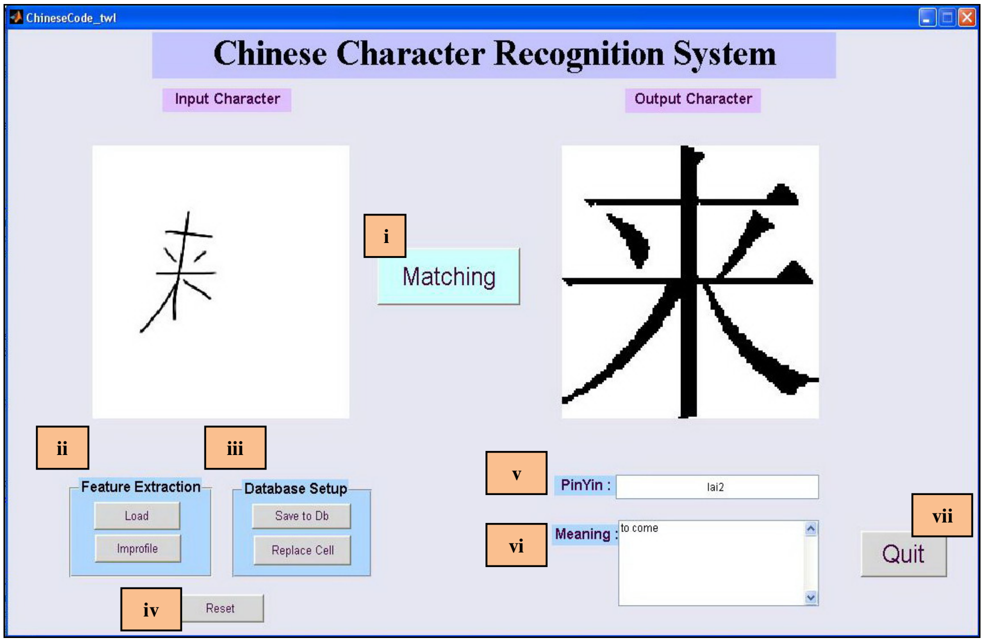
Case 1: Testing for input character ‘来’ (means come) with normalization. [Left: Input character, Right: Output character]



PinYin : lai2

Meaning : to come

Case 2: Testing for input character ‘来’ (means come) without normalization [Left: Input character, Right: Output character]



PinYin : lai2

Meaning : to come

Desciption of MATLAB GUI for the Proposed HCCR System

(i) "Matching" button – Match the input character to database character by using R_p^2 classifier.

(ii) Feature extraction:

"Load" button – Load input character image, i.e. the image of handwritten Chinese character collected from different writers.

"Improfile" button – Capture the *x*-coordinates and *y*-coordinates of each written stroke, define feature vector and perform Haar wavelet transform.

(iii) Database Setup:

"Save to Db" button – Save the features of the chinese character to the new created database (CL2009).

"Replace Cell" button – Replace or correct the character features in database if some mistakes are made in the database setup process.

(iv) "Reset" button – Reset the GUI or the recognition system to default.

(v) Pinyin – Display the pronunciation of the Chinese character.

(vi) Meaning – Display the meaning if the Chinese character.

(vii) "Quit" button – Quit the GUI or close the recognition system.