NANYANG TECHNOLOGICAL UNIVERSITY

A MACHINE COMPATIBLE SCRIPT FOR FAST ENTRY OF TEXT USING HANDWRITING

Ma Yang

A thesis submitted to Nanyang Technological University in fulfillment of the requirements for the Degree of Doctor of Philosophy

Division of Computing Systems

School of Computer Engineering

2005

Acknowledgements

I am greatly indebted to my supervisor, Prof. Graham Leedham, who is an endless source of enthusiasm, ideas, and patience. It was him who led me into this exciting area of pattern recognition, and has offered me constant encouragement and advice throughout the last three years. I hope I have learned from him not just his broad knowledge, but his insights, inspiration, and his way of conducting research.

I gratefully acknowledge Prof. Colin Higgins and his Ph.D student Swe Myo Htwe from Nottingham University. It is a pleasure to work with them on this project. I benefited a lot from discussions with them.

Special thanks to Ms Lynn, Gao Hongliang, Lu mingchun, Feng Yunshan, Daisy Chang, Francis chan, Niang Niang Maw, Yang minhua for their time and patience to write evaluation samples for this project.

I gratefully acknowledge all my colleagues in the Center of computational intelligence for creating such a pleasant working environment. Particular thanks to Wang xiangrui, Zhu lin, Ma Li, Liu Feng and Chen Fei. It is really wonderful to work with them and talk with them on different issues. My former and present friends, Zhang Yi, Chen Jinmiao, Li Xuhong, helped lighted the atmosphere beyond the work. To all, my heartfelt thanks.

I owe deepest appreciation to my parents and my husband who have given me loads of love, happiness and encouragement in the whole journey. This thesis is dedicated to my family for their constant support over all these years.

Finally, I would like to thank Nanyang Technological University for offering me this doctoral research opportunity.

Table of Contents

List of Figures

List of Tables

Abstract

There is a wish to be able to enter text into hand-held devices at the speed of speech (around 180 words per minute). Accurate unconstrained speech recognition remains an unresolved research challenge. Commonly used handwriting methods, like block capital letters and cursive script, are relatively slow and can only achieve a fraction of the speed of spoken speech (30 words per minute would be considered fast handwriting). Only handwritten shorthand schemes, which can, with training, be used at speeds in excess of 100 words per minute, can achieve a rate close to the speed of normal speech. This research focuses on techniques of recognizing and translating two shorthand systems -- Pitman shorthand and Renqun shorthand, which represent English and Chinese respectively.

According to the review of the latest on-line handwriting recognition techniques, most techniques are so sensitive to geometric features of characters that they are not able to achieve a satisfactory recognition accuracy for the recognition of shorthand. Shape recognition techniques recently demonstrated for the recognition of Korean script are applicable for recognizing shorthand handwritings. Based on this, an overall solution to the recognition of both Pitman shorthand and Renqun shorthand is proposed. Recognition approaches of Pitman shorthand components are firstly discussed in detail. A two-stage approach (segmentation & classification) is proposed for the recognition of the consonant outlines, which is the most difficult part in Pitman notation. A template-based matching approach is proposed for the shortform classification. Hausdorff Distance is introduced to measure the similarity between the input outline and the corresponding templates. Compared with Pitman shorthand, both vocalized outlines and shortforms of Renqun shorthand are composed of rhymes and consonants and have similar shape features. Due to the inherited geometric features of Renqun shorthand from Pitman

shorthand, the two-stage approach used for Pitman shorthand recognition is tailored for recognizing both vocalized outlines and shortforms of Renqun shorthand.

In order to improve the machine ‘readability’ of shorthand systems, new writing rules are proposed to overcome the difficulties encountered with smooth junctions and boundaries with low angularity. This is achieved in Pitman shorthand by requiring a circle to be written near the smooth junction. In Renqun shorthand, the primitives are required to be written with a slant angle of approximately 45^o. Experimental results show that the machine compatible shorthand improves the recognition accuracy greatly while placing little burden, and hence reduction in writing speed, on the human side.

Finally, critical technological commercialization issues of the shorthand recognition system are discussed from two aspects: algorithm performance and software principles. Through collaboration with another research group who are focusing on the linguistic post-processing of Pitman shorthand, a demonstration system including all the core functions of a commercial shorthand system - data collection, shape recognition, transcription and graphical result display, is implemented on a tablet PC running under Microsoft Windows XP Tablet Edition. The demonstration system shows the potential of Pitman shorthand as a means of fast text entry and facilitates the research process for further work.

CHAPTER

1

Introduction

1.1 Background

There has recently been a rapid growth in the use of small powerful handheld computers, which combine a flat panel color display with a handwriting digitizer. Such devices that include Palm, HP Jornada and the electronic tablet interface produced by Microsoft, enable both pointing and selecting, and limited handwritten text and data entry using a pen directly on the digitiser display. In addition, these pen-based computers are combined with communication channels for more interactive and powerful mobile computing applications.

A major bottleneck in the utilization of these devices is the cumbersome manipulation of textual information. Currently, speech, handwriting and keyboards are the three most widely used methods of recording textually rich information. Speech has particular advantages in some situations and can record up to 180 words per minutes (wpm). However, current speech recognition systems impose many constraints such as limits on vocabulary, syntax, speaker dependency and the speed of speech. It usually binds the user to its predefined post-processing stage to achieve satisfactory recognition accuracy. In noisy, uncontrolled environments, speech recording is neither ideal nor feasible. Thus, whilst speech input offers the greatest speed potential, its practical implementation makes it a less-than-ideal solution.

Whilst keyboards are still the predominant means of entering text into computers, small handheld devices are too small for a full-sized QWERTY keyboard. Even if the keyboard is reduced in size to fit the small handprint of a handheld device, it is still space intensive and difficult to use. Even with a full-size QWERTY keyboard and proficient operator, few can type more than 60 English wpm, a far cry from the speed of normal speech. For QWERTY soft keyboards and predictive keyboards, their text entry speeds on the limited-size screen of the hand-held devices were reported to be around 14wpm and 6wpm respectively (Lewis, 1999). Lewis also observed that the uncertain arrangement of the keys of the predictive keyboard significantly hindered performance. Although Palantype and Stenograph shorthand keyboard systems are able to record English speech with up to 180 wpm, the following prerequisites limit their usage. Firstly, a special Palantype or Stenograph keyboard should be attached to a laptop for code entry by a highly trained operator. Secondly, a computer-aided transcription system is indispensable to decode the chords back into English using a special dictionary created by the Verbatim Reporter.

For large-alphabet languages, such as Chinese and Japanese, keyboards are cumbersome. Special software is required to convert input English characters entered using a QWERTY keyboard, into Chinese and Japanese characters. Consequently, the rate of text entry relies greatly on the software properties and the experience of the operator. For example, using the Microsoft Intelligent Pinyin Input System (Note: Pinyin is a widely used system to represent the phonetic representation of Chinese words using the normal A to Z alphabet used in western languages.) embedded in the Windows Operating system, skilled operators can input Chinese words at 40-45 wpm while the input speed is considerably slower, at between 20-30 wpm, for less proficient keyboard operators. Using other input systems such as NJStar™ in which a combination of English characters can be defined freely to represent any phrase or word, 120 wpm input rate was reported by a highly trained operator. (http://www.njstar.com)

Handwriting is another popular form of communication which is effective for interaction with the recently introduced hand-held devices. Different forms of handwriting have different input speeds. Currently, most commercially available text entry systems on these devices are restricted to constrained writing of individual characters (eg the Graffiti system for the input of English text used by Palm http://www.palm.com/us/products/input/Palm_Graffiti.pdf), limited performance cursive script recognition systems (Schambach, 2003) or menu pointing to a display of a keyboard. Microsoft has also produced a pen-based version of their Windows Operating System with the facility for unconstrained handwritten cursive script input (http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/input_pen_overview.mspx). Unfortunately, the maximum speed of normal cursive handwriting, about 30-40 wpm, and writing individual printed characters, about 10 wpm, are very slow compared to speech and do not enhance a devices' potential for rapid input of a large amount of text. Whilst individual characters or small amounts of cursive script input are adequate for many purposes such as making an appointment or short note, these are inadequate for detailed note-taking or writing of short reports.

Compared with the slow speed methods of QWERTY keyboards and normal handwriting, handwritten shorthand, which can enable recording at over 100 wpm, offers the potential to significantly enhance the capability of pen-computing. If a simultaneous transcription system based on handwritten shorthand is produced, it would lead to a verbatim transcript at over 100 wpm. Figure 1.1 compares the input speed of different input approaches. For both English and Chinese, handwritten shorthand are considerably faster than other handwriting and keyboard systems and is closer to the typical 160-180 wpm of normal speech.

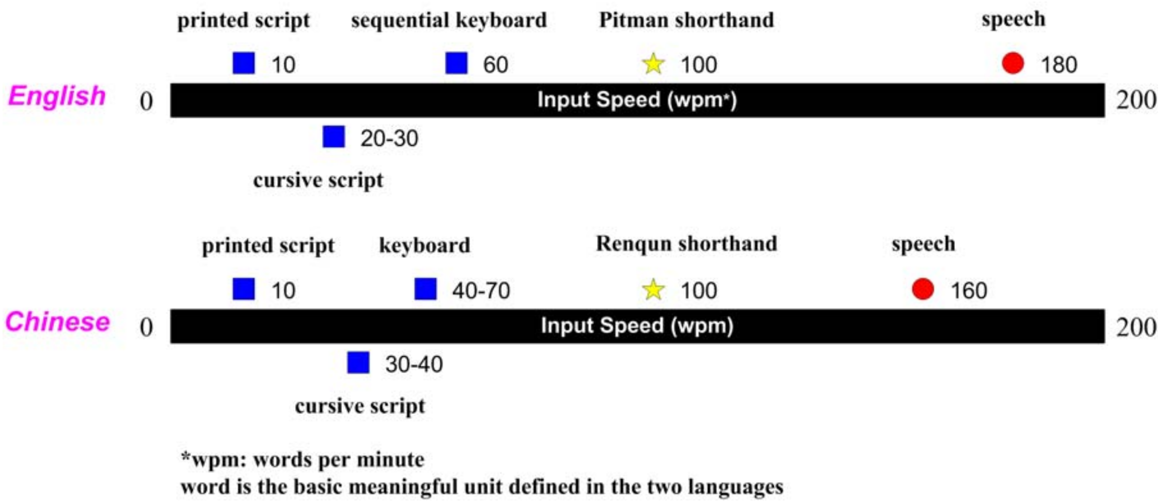


Figure 1.1: Comparison of different text entry methods for English and Chinese

1.2 Brief introduction to Pitman and Renqun shorthand

The major criteria for choosing handwritten shorthand script as the means of on-line input is its popularity and its suitability for machine recognition. Currently there are many different shorthand systems in use. The most popular ones include: Pitman shorthand and Gregg shorthand for rapid English text entry, Renqun shorthand and Yawei shorthand for fast Chinese text entry. Pitman shorthand has been widely used in over 75 countries for over 150 years. Compared with other English shorthand scripts, Pitman notation exhibits better 'machinography' (Leedham & Qiao, 1992) which makes it close to the ideal shorthand notation for machine recognition. The machine readability of Gregg shorthand is not as good as that of Pitman notation. As shown in Figure 1.2, the fundamental components of the Gregg shorthand (vowels) are complex and it is difficult to separate vowels and consonants in the same outline. Therefore, Pitman shorthand is selected in this study for the detailed investigation. Compared with Yawei shorthand, Renqun shorthand is much easier to learn and has more geometric similarity to Pitman shorthand. Therefore, it was selected as the second research topic in this project.

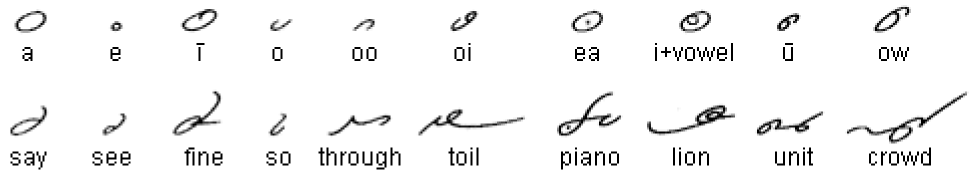


Figure 1.2: vowels in Gregg shorthand

In Pitman shorthand, words are represented by their sounds rather than their spelling. There are two types of Pitman shorthand outlines: Vocalized outlines and Shortforms. Vocalized outlines consist of a consonant outline with one or more surrounding vowel & diphthong symbol. The consonant outline is a connected sequence of features like circles, loops, hooks, straight and curved strokes. The vowel & diphthong symbols are small dots, dashes and ticks added in three standard position relative to the associated consonant outline. Shortform outlines are simple, single pen-strokes. Frequently used words and phrases are defined as shortforms for the convenience of usage. Figure 1.3 shows a sentence written in Pitman shorthand.

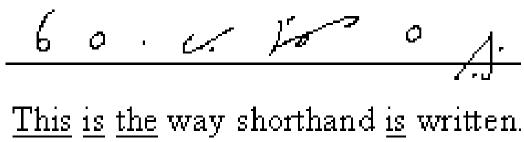
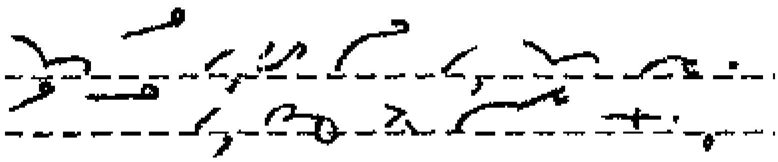


Figure 1.3: Pitman shorthand sample

(shortforms are underlined.)

Renqun shorthand is used to record Chinese Pinyin Like Pitman shorthand, Renqun shorthand includes two categories of shorthand: vocalized outlines and shortform outlines, and both of them are composed of rhymes and consonants. The consonant part is a sequence of straight and curved strokes and rhymes are either short primitives attached at the beginning/end of the pen-stroke or circles, loops and hooks with different sizes and orientations. Shortform outlines follow the abbreviated writing rules of Renqun notation which makes its outline simpler than the equivalent vocalized outline. Figure 1.4 shows a sentence recorded using Renqun shorthand.



知識豐富了，眼界擴大了，認識生活的能力提高了，寫作才有可能得心應手。

(With more knowledge, broader view point and deeper understanding ability, writing skill improves greatly.)

Figure 1.4: Renqun shorthand sample

There are substantial similarities between Pitman shorthand and Renqun shorthand. First, both of them are phonetic. Using very simple strokes of the pencil, the sounds of the words are recorded but not the letters themselves. Second, both of them are composed of two types of outlines: vocalized outlines and shortform outlines. Frequently used words and phrase are defined as shortforms for the convenience and speed of usage. Thirdly, both of them have similar shape features. They are composed of a sequence of geometric feature like straight lines, curved lines, circles, loops and hooks. It is because of the similarities between these two shorthand systems that algorithms firstly investigated for the recognition of Pitman shorthand can also be applied directly to the recognition of Renqun shorthand.

1.3 Objectives

Pen-computing has been an unrealized dream for many decades (Subrahmonia & Zimmerman, 2000). With the recent rapid advances in hand-held devices, entering textual rich information into these devices remains a serious bottleneck. With an achievable high input speed of over 100 wpm, shorthand is viewed as one of the most promising means of fast textual entry for the next generation of handheld computers and personal organizers.

As English and Chinese are the two most widely used languages, it is meaningful to investigate the potential of fast English and Chinese text entry on hand-held devices using Pitman shorthand and Renqun shorthand respectively. This research seeks to develop novel approaches to recognize the various components of Pitman shorthand (consonant outline, vowel & diphthong symbols and shortforms) and Renqun shorthand (rhymes & consonants). This research also seeks to propose a shorthand machine-compatible script or machinography by applying rules to improve machine readability of the Pitman and Renqun notations and overcome the deficiencies introduced by smooth junctions and boundaries with low angularity, which significantly deteriorate the machine readability of these shorthand systems. In addition, a demonstration online recognition system for Pitman shorthand will be implemented through collaboration with another research group at Nottingham University who are focusing on linguistic post-processing of Pitman shorthand. The demonstration system is to simulate the real interactive processes between human and computer using shorthand, demonstrate the feasibility of the approach and provide a good foundation for future work to commercialize the method.

1.4 Contributions

The key contributions of this research span academic analysis and recognition algorithms, a recognition demonstration system and discussion about the commercialization of shorthand recognition techniques. Specifically, the contributions are as follows:

(1) This research introduces techniques for shape recognition into the field of shorthand recognition in order to overcome the deficiency of previous shorthand recognition algorithms. Three algorithms are proposed to the recognition of different components in Pitman shorthand. They are: ①A template-based matching approach using Hausdorff Distance is developed to recognize Pitman shortforms. ②Vowel & diphthong symbols are classified according to their special geometric characteristics. ③The consonant outline, which is the most difficult part in shorthand recognition, is recognized using a two-stage approach (segmentation and classification). In addition, the two-stage approach

is also tailored to recognizing both vocalized outlines and shortforms of Renqun shorthand because of the similar geometric features between Pitman shorthand and Renqun shorthand.

(2) In order to ensure the "readable" quality of shorthand input, this research proposes new writing rules to improve the 'machinography' (Note: A machinography is a machine compatible handwriting system.) of the shorthand systems. For the smooth boundaries between consonant strokes which are impossible to accurately detect by mathematical approaches in Pitman shorthand, the new rule requires that a circle should be written near the smooth junction in the associated consonant outline.

(3) Another contribution is the demonstration recognition system for Pitman shorthand developed through collaboration with a research group from Nottingham University who are investigating the linguistic post-processing of Pitman shorthand. Implemented on a Fujitsu tablet PC under Window XP for Tablet PC, the demonstration system includes all procedures of a commercial shorthand recognition system - data collection, shape recognition, shape to phoneme transcription, phoneme to word transcription and graphical result output. Simulating the real-time interactive procedure between the stenographers and the recognition system, the demonstration system illustrates the feasibility of Pitman shorthand as a means of fast text entry and provided a good experimental environment for further research work.

1.5 Thesis Structure

This thesis contains 7 chapters. In chapter 2, the state-of-the-art in online handwriting recognition is surveyed. Related areas such as Korean character recognition, detection of dominant points in shape recognition which may be instructive to this project are also reviewed. A proposed system overview for shorthand recognition is described in chapter 3. Geometric features of both shorthand systems are also presented in detail. Chapters 4 and 5 describe the automatic recognition system for Pitman shorthand. Chapter 4 focuses on the most complicated part of Pitman shorthand – vocalized outline, whose recognition includes the segmentation and recognition of consonant outline components and the classification of vowel & diphthong symbols. A template-based approach to shortform classification is presented in Chapter 5. In Chapter 6, an integrated system for Renqun shorthand is introduced. Chapter 7 presents a Pitman shorthand simulation system and summarizes important technological factors in the commercialization process for a shorthand input system. In Chapter 8, the conclusion and contributions of the thesis are summarized and future work in this area is proposed.

CHAPTER

2

Review of related handwriting recognition techniques

2.1 Introduction

Handwriting recognition can be categorized into different methods based on how the handwritten data is presented to the recognition system. Offline handwriting recognition uses the static data from a scanned image as the input data, and online handwriting recognition works with dynamic data collected from a digitizing tablet. The raw data for an offline image of a few handwritten words sampled at 300 dots per inch requires several hundred kilo-bytes of storage space, which is much larger than a few hundred bytes required to store the series of x, y coordinates in the online case. (Plamondon & Srihari, 2000) Even though there is less data, the recognition rates reported are much higher for the online case in comparison to the offline case. (Nishimura & Timikawa, 2003)

In this thesis, only online recognition techniques are considered because the shorthand outline is an online pseudo cursive script and we are seeking a real-time, on-line method for fast text entry. The literature reviewed spans preprocessing methods, character recognition algorithms, shorthand recognition techniques and Korean character recognition.

2.2 On-line Handwriting Recognition

Since the 1980's, there has been a renewed interest in on-line handwriting recognition. In online handwriting recognition, the temporal information of writing is captured as a person writes and the recognition speed should be fast enough to keep up with the writing speed. Devices like a digitizer tablet with a pentip position sensitive surface are used to capture dynamic information about the pentip. Details of the position, pressure, velocity, acceleration and tilt angle of the pen are recorded as a function of time. Compared with off-line handwriting which uses an optical scanner to convert an image of writing to a bit pattern, on-line systems achieve superior performance. With the availability of more accurate electronic digitizers, more powerful CPUs, computationally expensive algorithms can be executed in real time to achieve better recognition performance. There are many commercial applications using the recent emergent powerful PDAs, such applications include on-line signature verification systems, handwritten notes, appointment entry, and instructional tools which interactively teach children to learn to write. The ultimate goal of on-line handwriting is to realize the dream of pen computing where recognition is accurately achieved without constraints on writing style, language used, and with the ability to respond to intuitive gesture-based editing commands.

2.2.1 Preprocessing

Preprocessing refers to the processing carried out prior to the recognition algorithms. It usually involves noise reduction, and normalization of the various aspects of the trace.

1. Noise reduction

The noise originates from the following sources: the quantization noise of the digitizer as well as the digitizing process itself, erratic hand or finger movements and the inaccuracies of the pen-up/pen-down indicator.

Smoothing is a process to average a point representing the path with respect to its neighbours. Yi et al. (2004) proposed an encoding method which combined smoothing and segmentation into a single step. The smoothing was only used to obtain the place of segmentation and the tangential direction. The position of the segmentation points and the extent of the resulting segments were derived from the original data.

Filtering can reduce the number of points by eliminating 'unimportant' points. For example, Hsieh et al. (2004) proposed a "minimum within-group variance" dynamic thresholding method to binarize aircraft images. If the size of a component after binarization was less than a threshold, it was considered as 'unimportant' and filtered out.

Smoothing and filtering could be performed in one operation. Manke et. al (1995) used a moving average window to remove sampling noise and smooth the trajectory simultaneously.

Wild point correction removes spurious points, usually caused by hardware problems. A penlift (Nair & Leedham,1991) is considered to be accidental if the distance between a pen-down and pen-up occurrence was less than 0.5 mm and the time interval between these two occurrences was less than 100ms.

Dehooking is to remove noises at the beginning or the end of a penstroke (Tappert et al., 1988). Such kind of noise usually looks like a hook which results from either erratic hand motion or inaccuracies in pen-down detection.

Stroke connection can remove extraneous pen lifts. Two separate strokes are connected manually if the distance between a pen-down and pen-up occurrence is small relative to the character size. (Brown & Ganapathy, 1983)

2. Normaliztion

Normalization refers to the reduction of geometric variations introduced by the writer and typically includes deskewing (slant correction), orientation normalization (baseline correction), size normalization (core height correction) and length normalization.

Deskewing corrects character slant. Characters slanting to the left or to the right are deskewed to put them all upright or to at least make them slant at the same angle (Kao & Don, 2005).

Using *baseline drift correction* a character or word can be oriented relative to a baseline or horizontal (Su & Wang, 2004).

Size normalization adjusts the character size to a standard by re-locating the origin to the lower left-corner or center of the character. Rosenthal et al. (1997) proposed an approach based on the Hough transform for the size and orientation normalization of online handwritten characters.

Stroke length normalization forces the originally sampled coordinates which are equidistant in time to be equidistant in space (Manke et al., 1995). It compensates for different sampling rates and varying writing speeds.

2.2.2 External segmentation

The goal of segmentation is to partition a word into regions, each containing an isolated and complete character. There are three categories of segmentation according to the way segmentation and classification interaction in the overall process (Casey and Lecolinet, 1996): classical approach, recognition-based approach and holistic approach. Holistic approach is segmentation free in which character is recognized as a whole. However, it only works on predefined lexicon or vocabulary. In the classical approach, segmentation is identified based on “character-like” properties. It cuts up the whole character into meaningful components through “dissection” techniques. In the recognition-based approach, segmentation is a by-product of recognition, which is driven by contextual analysis. Syntactic or semantic correctness of the overall result is emphasized in evaluating performance.

Structural features such as white space and pitch, projection, curvature extrema etc are used frequently to decompose the character into a sequence of sub-characters or strokes. Anquetil & Lorette (1997) extracted a set of “perceptual anchorage points”(points of discontinuity, angular points, cusps, points of inflection, extrema and multiple-points of k order) for the construction of a priori pertinent strokes. The segmentation approach proposed by Stefano et al. (2004) was achieved by exploiting curvature information extracted from the electronic ink at different level of resolution. Such information was then combined into a saliency map, through which the segmentation points were eventually found. Information from the saliency map was also used to select the optimal resolution to be used for describing the curvature of each stroke.

However, the above methods based on geometrical features are not adequate for difficult cases. For example, pitch-based methods cannot be applied if the width of the character is variable. Projection does not work well on inter-character connections, slanting characters or bad-quality images. Other approaches have been introduced to improve the segmentation results. Dynamic programming (Gao et al., 2005) was used to achieve segmentation. Using shape and position information of handwritten strokes, the cost function of possible segmentation paths was computed by dynamic programming. The one with minimal cost was selected as the optimal segmentation path. Artieres et al. (2000) explored the potential of segment models (SMs) for online handwriting recognition. From the authors' point of view, SMs offered an interesting alternative to classical Hidden Markov Models (HMMs) and allowed to address several shortcomings of other models. Unlike generalize HMMs in which observation modeling is at the frame level, SMs modeled a segment which is composed of several observations as the basic entity. In online character modeling, a segment corresponded to a stroke in a character. Such a segment will be modeled as a whole and the trajectory of the corresponding signal will be considered as an explicit function of time.

It is difficult to segment all handwritten words into perfectly separated characters without the feedback from recognition. Usually, contextual analysis is introduced to optimize the segmentation hypotheses. (Carbonnel & Anquetil, 2003; Han & Xu, 2004) The preliminary sharps obtained after segmentation are called “graphemes” or “pseudo-characters”, are classified into identifiable classes. A contextual mapping function from graphemes classes to symbols is constructed according to the recognition confidence, which determines whether to combine or to split grapheme classes. As summarized by Nohl et al. (1992), there are three types of approaches to finding the correct segmentation path. They were: (1) Finding the best-scoring segmentation for each lexicon entry; (2) Combining the lexicon and segmentation alternatives to form a single optimal path problem and establishing a one to one correspondence between paths through the graph and legal segmentation of lexicon entries; (3) Finding the K best segmentation/interpretations independent of lexicon and performing post-processes to eliminate interpretations not found in the lexicon.

2.2.3 Character Recognition

Many methods have been proposed and investigated in the area of on-line handwriting recognition. In this section, only those methods and techniques useful to this research such as shape recognition methods, curve matching, stochastic modeling and neural networks are reviewed.

2.2.3.1 Shape recognition methods

In shape recognition methods, static or dynamic geometric features are extracted for the character classification. These features can be either binary or nonbinary. Frequently used binary properties are the presence of descender or ascender features, dot or no dot on an i or j, circle or no circle etc. Hanaki & Yamazaki (1980) used a decision tree to find the type of unknown character. For example, the existence of a dot reduces the choice to either i or j. If the unknown character also has a descender, it must be j. However, alternative possibilities which are indispensable for post-processing or transcription are eliminated in the processing of the decision tree. Dynamic properties such as time sequence of zones, directions and extremes are frequently used as classification features. An approach based on sequential handwriting signals to on-line handwritten alphanumeric character was proposed by Li & Yeung (1997). An on-line handwritten character is represented by the combination of a sequence of dominant points and a sequence of writing directions between consecutive dominant points. The directional information of the dominant points is used for pre-classification and the positional information is used for fine classification. Okamoto & Yamamoto (1999) proposed an online character recognition approach using directional features and direction-change features. The directional features represent the location and direction of the character's coordinates. The direction-change features represent where and in which direction the character's written and unwritten imaginary stroke coordinates change, and the location of the circular parts of the character.

2.2.3.2 Curve Matching

Curve matching includes a set of templates and a similarity measurement to calculate the dissimilarity between an unknown character and the templates. The template which best matches the unknown character is assigned as the final result. The difference of x,y coordinates and directional change of feature points is calculated to measure the dissimilarity between the unknown character and the templates (Connell & Jain, 2001). A Dynamic Time Warping (DTW) (Sankoff & Kruskal, 1983) was proposed to measure the dissimilarity based on stroke measurement. The strokes and data points were matched in the same order as when they were produced and the first and last data points of the two characters were strictly matched against each other.

However, curve matching is computationally intensive. Much effort is devoted to pre-classification to reduce the number of templates and improve the efficiency of comparison. Li & Yeung (1997) proposed a pre-classification method using the distance between dominant points (pen-down/pen-up points, curvature extreme points) to roughly evaluate the similarity between an unknown character and a set of templates. Connell & Jain (2001) introduced two parameters *Dimension Tolerance* (T_d) and *Local Extensibility* (N_e) for pre-classification. Only templates whose number of feature points were smaller than $n-T_d$ and larger than $n+T_d$ (where n is the number of feature points of the unknown character) were eligible for detailed comparison. In the point-to-point comparison process, the ith feature point in the unknown pattern was compared with the feature points whose sequences ranged from $i-N_e$ to $i+N_e$.

2.2.3.3 Neural Networks

Numerous neural network approaches have been investigated intensively in character recognition. A multi-layered neural network with improved learning algorithm to evade a standstill in learning (Yamada et al., 1989) was proposed to classify handwritten numerals. Time delay neural network (TDNN) (Schenkel et al., 1995; Manke & Bodenhausen,1994; Matic et al., 1993) was applied to recognize online cursive handwritten characters. A set of back-propagation networks emulating Bayesian likelihood for each candidate category was used to recognise on-line Japanese Kanji (Toyokawa et al., 1993). Handwritten digitals were classified by a radial basis function (RBF) network (Lemarie, 1993). A Self-organization map (Wang et al., 1996), combined with principal component analysis (PCA), was used to recognize unconstrained handwritten numerals. A recurrent network (Albesano et al., 1992) realized a state transition model for word recognition.

Pittman (1992) pointed out that the type of ANN has little effect on the final recognition performance if the size of the training samples is large enough. How to choose features encompassing essential characteristics of the object has also been investigated intensively. Petersen et al. (2002) summarized eight categories of features to capture local geometric properties in pattern recognition and image processing. They were: points with a high curvature on the detected object contours; filter banks including wavelets; dedicated features such as stellate features (Karssemeijer & Brake, 1996) and OCR features; projection of the (sub)image onto the x- and y-axis; principal components obtained from the image; distances to feature space trajectories, which describe objects in all rotations, translations or scales; the Fourier descriptors derived from the image; the Zernike moments and the moments of Hu. Although neural networks have many advantages such as they are applicable to a wide variety of problems in pattern recognition and they are relatively easy to use and have outstanding input-output mapping abilities, many fundamental issues remain unresolved in this area. The first issue is how to ensure the transferability of a neural network. When trained to classify patterns obtained from one setting with a specific class distribution, a neural network will have a poorer and possibly unacceptably low performance when transferred to a novel setting with another class distribution. Another question (Petersen et al., 1999) is how to incorporate costs of misclassifications or the computational costs of features. There is always the danger of being captured by a local optimum matching solution rather than a global one.

2.2.3.4 Stochastic Models

Like elastic matching, the stochastic model is trained to represent the population of data for each class. The Hidden Markov Model (HMM) (Rabiner, 1989) is the most commonly used approach. HMMs have been used widely to represent state-to-state transitions within a character. These transitions provide a sequence of observations on the characters. Features are typically measured in the left-to-right direction. In such a system, segmentation is obtained as a by-product of recognition and no extra computation work for segmentation is needed. An optimal recognition is based on the maximization of a posteriori probability P(W|X), or, equivalently by the Bayes rule

$$P(\hat{W}, \quad X) = \max_{W} \quad P(X \mid W)P(W)$$

where P(X|W) is the probability that a specific pattern X is observed from the word W of prior probability P(W). Features extracted from the individual sample points were used to create the models. A "slide window" (Kornai,1997; Dehghan et al., 2000) which slides along the sample point sequence, was used to produce a feature sequence. Although HMMs model temporal relations very well, they need a large amount of training data and are computationally intensive. $V*N^2*T$ times of computation is needed to recognize a single character. Where V is the number of the patterns to be recognized, N is the number of the states in the HMM model, T is the number of the observation for the unknown pattern. "sub-stroke" or "nebulous" HMM (Tokuno et al., 2002; Hu et al., 1996) was proposed to solve this issue. By enabling shared patterns which are better trained with the same amount of training data, each character is denoted as the concatenation of different "substroke" HMM.

2.3 Recognition of shorthand

Over one decade ago, intensive research was carried out to demonstrate the potential of Pitman shorthand as a means of rapid text entry. The first investigation into the feasibility of using handwritten Pitman shorthand as a means of verbatim transcription as an aid for the deaf was performed by Brooks & Newell (1985). They developed a prototype transcription system to segment the outlines into pattern primitives and classify these primitives adopting a structural pattern recognition strategy. Leedham (1984) categorized the Pitman shorthand into two classes: shortforms and vocalized outlines. Dynamic programming and a syntactic method which interacts with a knowledge source derived from analysis of a large number of shorthand outlines (Leedham & Downton, 1987) were proposed to classify shortforms and vocalized outlines respectively. Nair & Leedham (1992) compared four basic dynamic programming algorithms and their variations in terms of their recognition performance and computational cost. Qiao & Leedham (1993) proposed an interactive heuristic search (IHS) method for the overall segmentation and interpretation of consonant outlines. The whole algorithm was composed of three parts: a bottom-up analyzer to generate a plan for each outline and its phonetic interpretation; a top-down analyzer to examine the symbolic data structure in the plan and a heuristic search scheme to control the interaction between the bottom-up process and the top-down analyzer. The communication between the bottom-up and top-down processes leads the direction of search towards the best or nearly the best interpretation of the consonant outline.

However, due to the limitations of portable computers at that time and the lack of suitable devices for a pen-based computer, this novel research work did not develop beyond laboratory prototypes. Firstly, the ergonomics of the data acquisition devices (writing tablet and instrumented pen) of that period required a writing technique which was disliked by many writers and consequently affected the quality of the shorthand outline. Secondly, with the undeveloped hardware techniques at that period, it was hard to commercialize the research work into an acceptable product. Recently, with the improvement of PDAs and other human-computer interaction technologies, it is timely to revitalize work in this area. However, previous work was not sufficiently accurate for a commercial automatic transcription system. The IHS approach could not guarantee to find the best interpretation for each consonant outline using the heuristic search. Dynamic programming techniques suffer from the problem of redundancy. That is to say, if the users define personalized shortforms with very similar shape features, the performance of the algorithm will degrade rapidly. The issue of vowel and diphthong symbol recognition was not fully investigated.

Recently, with the development of PDA and related IT technologies, the previous work in shorthand has been revitalized. Zhu et al (2002) proposed a novel approach using two back-propagation neural networks for the segmentation and classification of the consonant outlines. One neural network is applied to segment consonant outline and detect over-segmentation and another neural network is used to classify segmented primitives.

Limited progress has been achieved in the recognition of Renqun shorthand. Chen & Yu (1998) proposed a structural approach to segment and classify Renqun shortforms (shown in Figure 2.1). The shortform is firstly segmented into a string of primitives by detecting the location of boundary points between segment and circle, segment and hook, segment and dash. Afterwards, each primitive is classified by IF-THEN rules and dynamic programming. Finally the recognized shape features are sent to a transcription system for the Chinese phrase output.

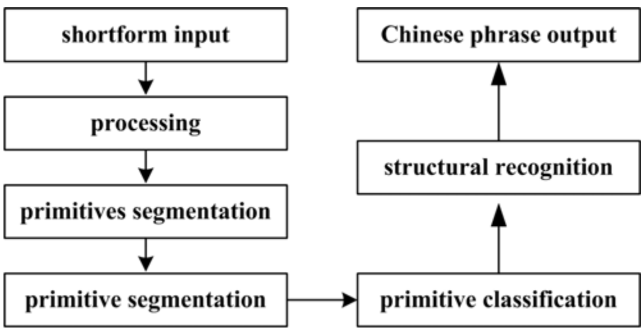


Figure 2.1: Structure of Renqun shorthand recognition

(adapted from Chen & Yu,1998)

Five novel rules were proposed by Chen et al. (2000) to find potential segmentation points (points of maximum curvature, intersection points and stroke junctions) in Renqun shorthand. Chen & Qiao (1997) used IHS to recognize Renqun shorthand. Input outlines were first segmented and coarsely classified using a knowledge-based heuristic search algorithm. Techniques like artificial intelligence, dynamic programming were introduced to classify the segmented primitives. According to our investigation, most of the systems take advantage of the same approaches in Pitman shorthand such as dynamic programming for the classification of basic consonants, interactive heuristic search (IHS) approach to the segmentation and classification of consonants. To some extent, research regarding Renqun shorthand could be viewed as the extension of the research in Pitman shorthand. Therefore, it suffers the same problem as the recognition approach to Pitman shorthand such as redundancy in dynamic programming and low accuracy of IHS.

2.4 Related methods

2.4.1 Recognition of Korean Characters

There are some loose similarities in the structure of Korean script and vocalized outlines in Pitman Shorthand. First, both of them are phonetic. Second, both of them are composed of a sequence of vowels and consonants using similar shape features like straight lines, curved lines and circles. Therefore it is worth reviewing the literature of Korean Script recognition to broaden the research scope.

Jung & Kim (2000) classified Korean character recognition methods into four categories according to the basic recognition unit: character (holistic recognition), stroke, stroke segment and grapheme. In the following sections, we will only discuss the last three segmentation-based approaches.

Lee et al. (1999) proposed an improved two-stage classification approach to recognize printed Korean characters. The first stage determined the positional relationship of the input character, and the second stage recognized subparts of the grapheme. All classifiers were implemented using a multi-layer back-propagation neural network. A validation process was also introduced to correct fatal errors in the first stage, which are unrecoverable in the second stage. The two-stage (two-level) idea has been used widely in the recognition of Korean scripts. The major differences of these approaches lie in:

(a) The realization approach: Neural networks such as back-propagation (BP) and Approximate Predictive Control (APC) (Paek et al., 1993), Adaptive Resonance Theory (ART) (Kim et al., 1998), dynamic programming (Shin, 2001), elastic matching (Kang & Kim, 2000) and HMM (hidden Markov model) (Kwon et al., 1997; Kim et al., 1998) have been selected to fulfill the goal of different stages;

(b) The recognition units in each stage: In the first stage, the recognition unit is the basic component of a character such as stroke code and relative stroke position (Paek et al, 1993), primitive stroke (Kim et al, 1998), stroke segment (Kwon & Kwon, 1993) and grapheme (Shin, 2001). Generally the recognition unit in the second stage is the whole character.

In the approach proposed by Jung & Kim (2000) to the recognition of on-line cursive Korean characters, a TDNN (time-delay neural network) was used in grapheme recognition to recognize the current grapheme and predicts the type of the grapheme that will appear in the next position. The outputs of TDNN were used as symbol probabilities (B matrix of HMM). The whole character recognition was finalized by finding the best-scoring path using the Viterbi algorithm.

Kim et al. (1996) used an ART-1 neural network to classify the cursive stroke of Korean characters. Fuzzy membership functions were introduced to discriminate a stroke from a group of similar strokes.

Jun et al. (1995) proposed a curvature based approach to recognize Korean characters. Firstly, the strokes were split at any one of the following three conditions: (a) change in rotating direction; (b) sudden change in direction; (c) excessive direction (stroke too long). Afterwards, some splitting points were merged to produce comparable parts with the pre-defined curvature models.

2.4.2 Polygonal Approximation

If a shorthand script is viewed as a curve rather than a character, it can also be represented by a sequence of dominant points which denote its major sharp features. It is possible to introduce approaches to shape recognition, such as polygonal approximation for the shorthand segmentation.

There are many polygonal approximation approaches such as the split-and-merge approach (Xiao et al., 2001; Ray & Ray, 1995), the scale-based approach (Rattarangsi & Chin,1992; Pikaz & Averbuch, 1996), the genetic algorithm based approach (Yin, 1998; Huang & Sun, 1999) and the dominant points or angle detection approach (Teh & Chin, 1989; Wu & Wang, 1993; Ray & Ray, 1992, 1994; Inesta et al., 1998). In this review, we focus on the algorithm that is able to be highly respectful to the shape via the detection of the least number of points rather than the algorithm favoring the selection of a set of perceptually correct and informatively rich vertices which minimize the approximation error.

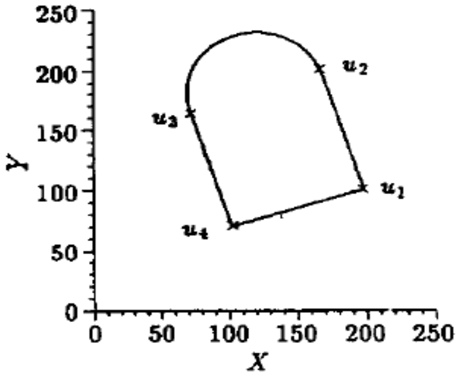
Teh & Chin (1989) proposed a famous approach to detect dominant points. First the support region of the point P_i was determined by the ratio of perpendicular distance between P_i and the chord $P_{i-n}P_{i+n}$ over the length of the chord $P_{i-n}P_{i+n}$. The above ratio was then used to compute the curvature of the point P_i. However, this method is sensitive to noise and the support region of all points in the curve is symmetrical. Ray & Ray (1992) pointed out that there was no reason why the support region should be symmetrical and proposed an approach to compute the left region and the right region respectively. The right support region was extended from P_i until some value of j where $\theta_{i,j-1} < \theta_{i,j} \leq \theta_{i,j+1}$.

The rule for left support region was vice versa. Both methods mentioned above were parameter free which avoided the issue of choosing suitable parameter values. Inesta et al. (1998) proposed a method to suppress the collinear points that contain little information about the shape. If the angle formed by three continuous dominant points was smaller than a predefined threshold, the middle point was deleted from the set of dominant points. Asada & Brady (1986) classified dominant points into two types: corner and smooth join. Their mathematical properties were described as follows:

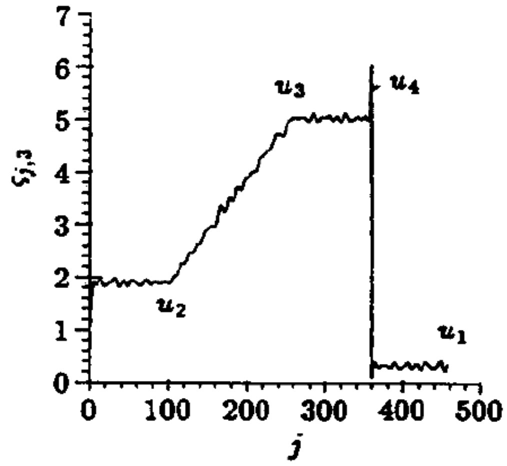
Corner: is an isolated curvature change, for which the tangent to the contour is discontinuous (points u_1 and u_4 in Figure 2.2(a).); and

Smooth join: is an isolated curvature change, for which the tangent is continuous, but the curvature is discontinuous (points $u_2$and u_3 in Figure 2.2(a).).

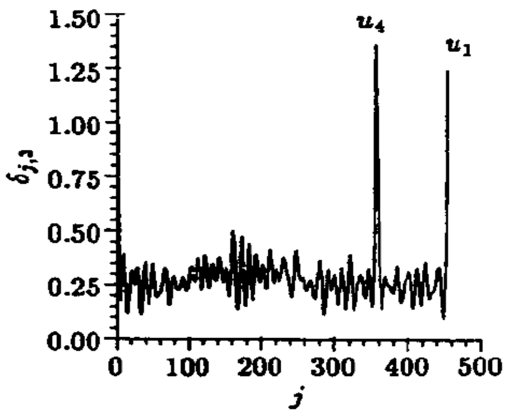
Chen et al. (1996) defined two parameters: k chain-code $\xi_{j,k} = \tan^{-1}\left[\frac{y_j - y_{j-k}}{x_j - x_{j-k}}\right]$ and k differential chain-code $\sigma_{j,k} = \tan^{-1}\left[\frac{y_{j+k} - y_j}{x_{j+k} - x_j}\right] - \tan^{-1}\left[\frac{y_j - y_{j-k}}{x_j - x_{j-k}}\right]$ to distinguish a sharp corner from a smooth join. As shown in Figure 2.2, a corner is the point whose k chain-code is discontinuous and differential k chain-code is a spike. A smooth join is the corner point of k chain-code curve, which can be found by ordinary piecewise linear approximation. Koplowitz & Plante (1995) proposed an algorithm to detect corners. The maximal straight-line distance was used as an indicator of the curvature for each point. The point whose the chain code was different from its preceding one was listed as a potential candidate for future selection.



(a) x,y coordinates of the input outline



(b) k chain-code $\xi_{j,k}$ (k=3, m=458)



(c) k differential chain-code $\sigma_{j,k}$ (k=3, m=458)

Figure 2.2: A mouse like shape **(Chen et al, 1996)**

Polygonal approximation usually uses line segments to fit curves. In order to fit an arc with minimal error, many broken points will be produced. So it is necessary to introduce high order curves to reduce the number of broken points and improve approximation performance. Pei & Horng (1996) proposed an approach for polygonal approximation using both straight-line and curves, in which a line segment was viewed as a special type of arc whose curvature is zero. Unlike commonly used error measures such as integral absolute error, integral square error, "area" was chosen as the quantity of the error measure and dynamic programming was used to find the optimal combination of dominant points.

In other work by Horng & Li (2001), line segments and circular arcs were used to fit digital planar curves. It was based on the following two criteria:

I. The straight-line segments are preferred over the circular arcs. A large coefficient is given to line segments for the priority. Circular arcs are adopted only if they considerably simplify the representation;

II. Longer analytic curve segments (line segments or circular arcs) are preferred. In other words, less segmentation is preferred to produce simple and clear representations.

2.5 Summary

In this chapter, on-line handwriting recognition techniques, previous work in shorthand recognition, Korean character recognition and shape recognition are reviewed. It is concluded that techniques in online handwritten character recognition do not work well on the recognition of shorthand because they rely heavily on the special geometric features of characters. In order to break through the current research dilemma, techniques in shape recognition are introduced for the shorthand recognition if a shorthand outline is treated as a curve rather than a handwritten character. In the recognition of consonant outlines which is the most complicated part of the recognition engine, "the region of support" is introduced to detect potential stroke junctions of consonant outlines and polygonal approximation is used to adjust the position of segmentation points. In the classification of Pitman shortforms, curve matching techniques are introduced to measure the dissimilarities between the input shortform and the corresponding templates. Moreover, processes in the recognition of Korean script (preprocessing, corner point detection, feature extraction, grapheme recognition) are tailored for the segmentation & classification (preprocessing, dominant point detection, primitive classification) of Pitman consonants and Renqun shorthand.

CHAPTER

3

System overview

This chapter analyses the geometric features of Pitman shorthand and Renqun shorthand. Based on the conclusion drawn in Chapter 2, techniques in shape recognition techniques such as "region of support", polygonal approximation and Hausdorff Distance are introduced for the recognition of shorthand outline. Besides, the preprocessing module and some prerequisites of the recognition system are discussed in detail.

3.1 Geometric Analysis of Pitman Shorthand

Invented by Sir Isaac Pitman in 1837, Pitman Shorthand has been widely taught and used in 75 countries throughout the world. Based on examination results obtained from the Pitman Examination Institute, it is estimated that there are over 500,000 practising Pitman shorthand writers and more than 80 Pitman training centers all over the UK, Ireland, India and the Middle East, training 50,000 people every year. The major reason why Pitman shorthand is chosen in this research is due to its excellent 'machinography', a term proposed by Wellisch (1978) to evaluate the suitability of a script for machine recognition. Brooks (1985) proposed the following guidelines for a machinography:

- The script should represent an efficient, error free, and reasonably natural method of interaction between men and machines;

- For reasons of compatibility, the script should possess the same linguistic structure as the local dominant script;
- The script should consist of a small number of simple pattern primitives which are both easily written and recognized. Further, the ease of writing and recognition of each symbol should be proportional to the frequency of use of that symbol;
- Ideally, the symbols of the alphabet should correspond directly to the individual pattern primitives constituting the alphabet of the structural recognition grammar;
- The script should possess structure on numerous levels, and there should be good correspondence between the linguistic and physical significance of primitives, so that a small physical change represents only a minor linguistic change;
- The script must also possess a simple well-defined physical and linguistic syntax, which may be described in terms of a straightforward structural recognition grammar;
- Finally, for practical reasons, the physical boundaries between adjacent pattern primitives should be marked in some obvious way.

According to Leedham & Qiao (1992), the Pitman shorthand notation exhibits many of these qualities. For example:

- The fundamental components of the handwriting system are simple geometric shapes (hooks, loops, circles, straight and curved lines etc);
- Each outline represents a single word or phrase;
- The script is fully defined in terms of stroke sequences and pen-stroke directions;
- Consonants and vowels are easily separated;
- Most boundaries between phonetic features are clearly indicated by closure, angularity or high curvature.

The above qualities make Pitman shorthand an ideal means of fast data entry for the next generation of PDA products.

There are two forms of Pitman shorthand: Pitman New Era and Pitman 2000. Although New Era style is able to record at higher speed, it is more complicated than Pitman 2000 and more difficult to automatically recognize and transcribe. Pitman 2000 (Pitman,1980) is easier to learn and more widely used. In Pitman notation, the English language is represented as having 40 phonemes. These are further classified into 24 consonants, 12 vowels and 4 diphthongs and each of them is assigned a simple pen-stroke. Leedham & Downton (1987) categorized Pitman shorthand into two types: vocalized outline and shortform. Vocalized outline consists of a single consonant outline with surrounding vowel and diphthong symbols. The consonant outline is a connected sequence of simple features like circles, loops and hooks, straight and curved strokes. As shown in Figure 3.1, there are 24 types of basic consonant symbols. Three lengths (normal, doubled and halved) and different thickness of strokes are used to convey additional information.

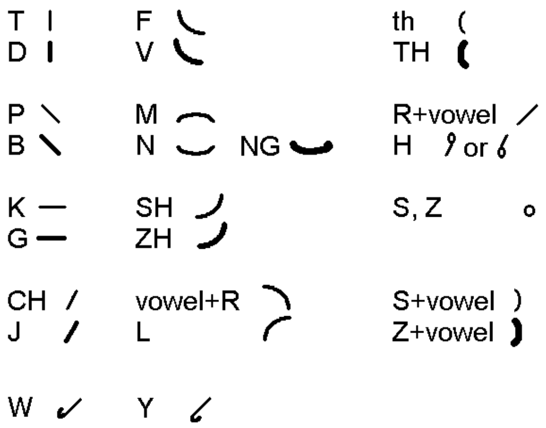


Figure 3.1: Consonant strokes in Pitman shorthand

The vowel and diphthong symbols are dots, dashes and small ticks attaching the associated consonant outline. The vowel sounds are represented by dots or dashes written in one of three positions with respect to the consonant stroke. The thickness of these dots and dashes determines if the vowel is a long vowel or a short vowel. Each of the vowel primitives may be written either before or after the consonant stroke. Diphthongs are special symbols written in a similar manner to vowels but may be attached to the consonant stroke in an outline. An example of a vocalized outline and its constitute of phonetic features is shown in Figure 3.2.

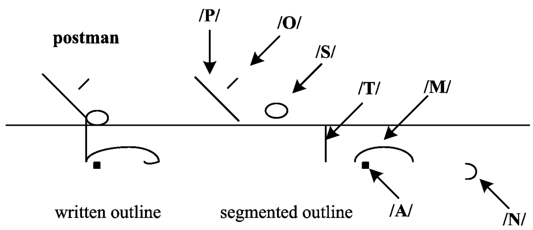


Figure 3.2. Vocalized outline “postman”

Approximately 90 frequently occurring words are defined as shortforms which are single, simple pen-strokes for the convenience of usage. The use of shortforms increases the recording speed significantly. Previous studies have noted that a typical shorthand passage consists of between 30% and 50% shortforms (Leedham & Downton, 1987). Writers also frequently use a small number of personalized shortforms in their working environment. To distinguish a shortform from a vocalized outline is relatively simple if the writers are constrained to include at least one vowel or diphthong in all outlines rather than shortforms.

Based on the geometric features of Pitman shorthand, the recognition of Pitman shorthand should be classified into shortform recognition, consonant recognition, and vowel & diphthong recognition. Different algorithms should be employed for these three components.

3.2 Geometric Analysis of Renqun shorthand

The history of Chinese shorthand can be traced back to 1896 when four kinds of Pinyin-based shorthand drafts were proposed for the high speed recording of Chinese language. In 1921, the publication of a book named “Basic Rules of Chinese Shorthand” represented a milestone of the establishment of Chinese shorthand. In the next two decades, Chinese shorthand experienced a fast development period. More than 10 kinds of Chinese shorthand systems were invented and used in different situations. However, only two of them -- Renqun shorthand and Yawei shorthand remain in usage today. Renqun shorthand is popular in the southern part of China especially in provinces like Guangdong, Fujian etc, and Yawei shorthand is widely used in the northern part of China. Although Yawei shorthand is able to record words at a higher speed than Renqun, it is more complicated and requires more time for users to learn. In this research, Renqun shorthand is selected as the second part because it has similar geometric features to those in Pitman shorthand hence successful recognition techniques for Pitman shorthand are also applicable to Renqun, at least at the theoretical level.

Invented by Liao Renqun at the beginning of 1980's, Renqun shorthand (Liao, 1985) has been used by a wide audience of writers, office secretaries and newspaper, court and governmental reporters due to its simplicity and precision. As a means of recording Mandarin phonetically, Renqun shorthand could achieve up to 100 wpm recording speed. Figure 3.3 is a typical sentence written in Renqun shorthand. Renqun shorthand. It is estimated that there are 150,000 Renqun shorthand writers and 24 training centers training 20,000 people every year in China.

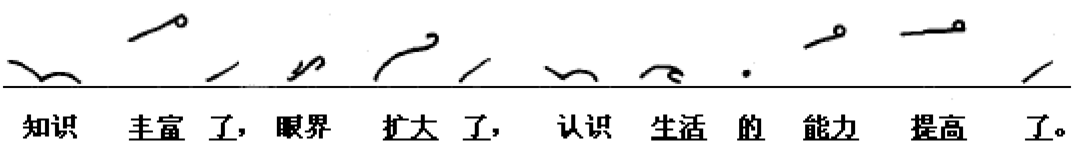


Figure 3.3: A sentence written in Renqun shorthand (Shortforms are underlined.)

Like Pitman Shorthand, Renqun shorthand adopts simple straight lines, curved lines, and geometric symbols to represent the initial consonant and rhyme parts of Chinese syllables. Like Pitman shorthand, those frequently used words or phrases are also represented by abbreviated symbols called shortforms in Renqun shorthand. These shortforms are composed of either the first syllable notation or the second one, and sometimes they also make use of the combination of initial consonant symbol of the first syllable and rhyme notation of the second one. As for a typical text recorded using Renqun shorthand, shortforms usually account for a high proportion, at least up to 50% of the outlines. Furthermore these shortforms can simplify the automatic recognition by omitting the usual translation from syllable to Chinese character since these shortforms accord directly with Chinese phrases. However, the template-based approach used for the recognition of Pitman shortform is not applicable to Renqun shortforms because of differing writing rules and geometric features.

According to the rules of the Chinese pinyin system, 296 pinyin (sound) notations are represented by the combination of 21 consonants and 35 rhymes. However, only 17 consonant symbols and 28 rhymes are introduced in Renqun shorthand since consonants and rhymes of similar pronunciations are merged. The consonant symbols are either straight lines or shallow curves with two different lengths (length ratio is 2:3) and three slanting directions as shown in Figure 3.4.

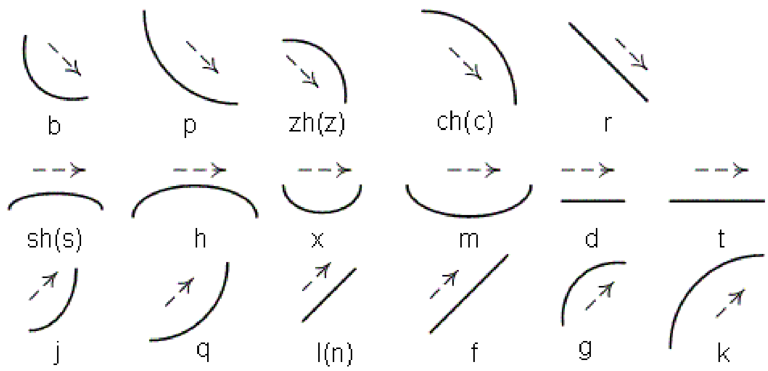
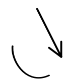
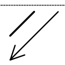
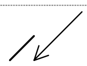
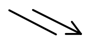
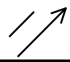
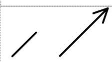
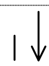
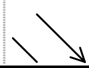


Figure 3.4: Consonant symbols in Renqun shorthand

Two types of rhymes are defined in Renqun. The first type is circles, loops and hooks with different sizes (small or big), written in different position (up or down referring to the stroke to which it attaches) and direction (same or opposite referring to the stroke to which it attaches) referred to the associated consonant outlines. The second type is straight lines or curves attached at the beginning/end of the consonants. Detailed features of these two types of rhymes are summarized in Table 3.1.

Table 3.1: Rhymes in Pitman shorthand

Rhyme	**I**	*ü*	**Uang**	**ue (** *üe* **)**	**U**	**Un (** *ün* **)**	**ao**
Symbol	Small circle	Small circle+tail	Small circle	Small circle(+tail)	Small circle	Small circle(+tail)	Big loop
Position	Up	Up	Up	Up	Down	Down	Up
Direction	Same	Same	Same	Same	Same	Same	Same
Rhyme	**uo/o**	**A/ia**	**Ie**	**ua**	**en/eng**	**Ai**	**uai**
Symbol	Big loop	Small hook	Small hook	Small hook	Small hook	Large hook	Large hook
Position	Down	Up	Down	Down	Down	Up	Down
Direction	Same	Same	Same	Same	Opposite	Same	Same
Rhyme	**An**	**Uan**	**Ang**	**iang**	**ong/iong**	**uen**	**uei/ei**
Symbol	Small hook/circle+ horizontal line	Large circle + horizontal line	Small circle+tail	Small circle+tail	Small circle+tail		
Position			Up	Up	Down		
Direction	Same/opposite	Same/opposite	Same	Same	Same		
Rhyme	**Ian**	*üan*	**in/ing**	**iao**	**E/er**	**Ou**	**iou**
Symbol							

In Renqun shorthand, 367 Independent shortforms (ordinary shortforms) and 265 Raised shortforms are used to represent frequently used words or phrases. Two reference lines are introduced to distinguish them. As shown in Figure 3.5, independent shortforms should be written above the second reference line and below the first reference line. The position of a raised shortform is a little bit higher and should be written above the first reference line. A previous study (Chen & Yu, 1998) demonstrated that almost every raised shortform accords with an independent shortform and there are, in total, 436 different shapes of shortforms.

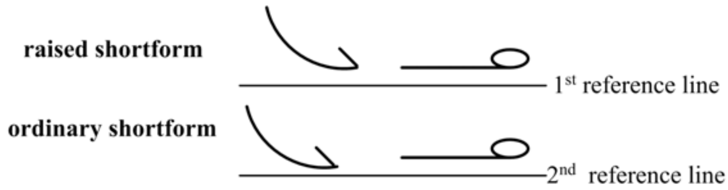


Figure 3.5: Shortforms and corresponding Chinese characters

3.3 Comparison of Renqun and Pitman

According to the above analysis, a comparison summary is given in Table 3.2. Both Pitman shorthand and Renqun shorthand record the phonetics of the speech. That is, their words are represented by their sounds rather than their spelling. These two shorthand systems are composed of similar shape features like straight lines, curved lines, circles, loops and hooks. Like Pitman shorthand, Renqun shorthand is also composed of vocalized outline and shortforms. But there are still differences between them. Pitman vocalized outline consists of a single consonant outline surrounding by vowel & diphthong symbols. A Renqun vocalized outline consists of rhymes and consonants. In Pitman shorthand, stroke thickness conveys additional information. Pitman shortforms are single, simple penstrokes, and Renqun shortform consists of rhymes and consonants. Renqun shortforms represent meaningful adjectives, nouns or notional verbs, and Pitman shortforms denote frequently used prepositions or auxiliary verbs which convey little meaning outside a sentence. Approximately 94 frequently used English words and phrases defined as Pitman shortforms. There are 436 Renqun shortforms which consist of more than 50% of words in a typical Renqun shorthand passage. Therefore, the recognition accuracy of shortforms plays an important role for the performance of the recognition system.

Table 3.2: Comparison of two shorthand systems

<table>
<tr><th colspan="2">Comparison</th><th>Renqun shorthand</th><th>Pitman shorthand</th></tr>
<tr><td colspan="2" rowspan="3">Similarity</td><td colspan="2">Record words phonetically</td></tr>
<tr><td colspan="2">Similar geometric features (straight, curved lines, hooks, circles and loops)</td></tr>
<tr><td colspan="2">Similar basic consonant symbols</td></tr>
<tr><td rowspan="8">Dissimilarity</td><td>Language</td><td>Chinese</td><td>English</td></tr>
<tr><td>Stroke length</td><td>For consonant, two kinds of stroke lengths (2:3)
For rhymes, short primitives</td><td>Normal, doubled, halved</td></tr>
<tr><td>Stroke thickness</td><td>Not concerned</td><td>Same strokes with different thickness represent different phonetic features</td></tr>
<tr><td>Number of primitives per outline</td><td>1-2</td><td>2.6 on average</td></tr>
<tr><td>Vowels and dipthongs</td><td>N/A</td><td>Yes</td></tr>
<tr><td rowspan="3">Shortforms</td><td>Approx. 436</td><td>Approx. 94</td></tr>
<tr><td>complex, following the simplified writing rules</td><td>simple, single pen-stroke</td></tr>
<tr><td>Representing meaningful words</td><td>Representing meaningless words such as preposition, auxiliary verb</td></tr>
</table>

3.4 System structure

According to the above detailed discussion about the geometric features of Pitman shorthand and Renqun shorthand, both Pitman Shorthand Recognition System (PSRS) and Renqun Shorthand Recognition System (RSRS) are proposed in this research.

3.4.1 Preprocessing

The recognition processes of PSRS and RSRS are quite similar and both of them have preprocessing modules in order to remove noise caused by instability in the writing and digitization process.

The preprocessing includes eight steps as follows:

(1) Size normalization: size normalization, in which the width of the bounding box is scaled to a standard width and the height of the bounding box is adjusted to maintain the original aspect ratio, enables the subsequent recognition algorithms to deal with any size of input using pre-defined thresholds.

(2) Removal of accidental pen-lifts: Pen lifts with distance between a pen-up and pen-down occurrence was less than 0.5mm and the time interval was less than 100 ms are removed as 'jitter' points. The above two parameters should be small enough to filter out accidental pen lifts which usually occur when the direction of the pen is changed.

(3) Minimum distance filtering: The point whose distance from its previous one is less than a predefined minimal threshold is removed as a repetitive and redundant one. In the system, the threshold is 0.2mm which means the pen is stationary or moving very slowly.

(4) Straight line average smoothing: The middle point of three consecutive points moves to 'best fit' the line formed by the first and the third points.

(5) Minimum distance filtering: Applying minimum distance filtering again to reduce redundant points produced by step 3.

(6) Straight line average smoothing with angle constraint: This step is similar to step 3 except it excludes those sudden angular change points, which include significant features of the input outline. The position of the sharp corner is determined by the equation: $\left|\Phi_1 + \Phi_2 - \Phi_3 - \Phi_4\right|$>threshold. The reason to use ± 2 samples for the calculation is to distinguish genuine corners from small noise glitches which still need smoothing. Experiments showed that the value of the threshold is set to 60^o to be most effective. If the threshold is too small, the outline will remain jagged; if the threshold is too large, the algorithm will be turned into straight line average smoothing in step 4.

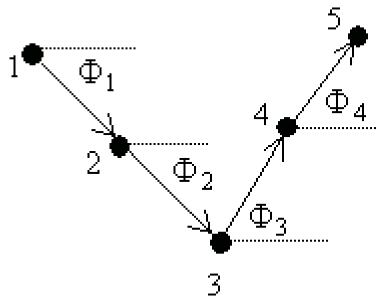


Figure 3.6: Angles measured to detect sharp corner

Steps 2 - 6 have been applied successfully by Nair & Leedham (1991) in the preprocessing procedure on a Pencept Penpad 310 digitiser with a resolution of 120 samples/second.

(7) Time normalization: The original x,y coordinates are divided into equi-distance sequences.

(8) Gaussian filter (Connell & Jain, 2001): The aim of the filter is to reduce noise in the input outline. The original coordinates are denoted as X^{org}, Y^{org} . The Gaussian-filtered coordinates are denoted as X,Y, which are expressed as follows

$$X = \sum_{i=-n\sigma}^{n\sigma} \xi_i X_{t+i}^{org} \qquad Y = \sum_{i=-n\sigma}^{n\sigma} \xi_i Y_{t+i}^{org} \text{ , where } \xi_i = \frac{e^{-\frac{i^2}{n\sigma^2}}}{\sum_{j=-n\sigma}^{n\sigma} e^{\frac{-j^2}{n\sigma^2}}}$$

where σ is the standard deviation of the Gaussian distribution, which accounts for the degree of noise reduction. The choice of σ and n is a tradeoff between the reduction of noise and the retention of major features of the input outline. The larger the values of σ and n, the smoother the curve, the more likely corners will lose their angularity and vice versa.

3.4.2 Prerequisite

The readability tests conducted by Leedham & Downton (1986) concluded that the input shorthand must be of as high a quality as possible and must conform to the shorthand writing conventions if an automatic recognition system is expected to produce readable outputs. In the samples collection processes of this project, the writers were required to write shorthand outlines which should be readable by any person familiar with the shorthand notations. Personalization of shorthand styles, which happens frequently among professional writers, will deteriorate the performance of the transcription system and even make the shorthand virtually unreadable to other persons. We believe that if a stenographer regularly used an automatic recognition system he will be constrained to produce outlines with consistent high quality provided by the feedback of the transcription system.

Besides the requirement for the quality of samples collected in the experimentations, there are three prerequisites which provide theoretical foundation for the recognition system dealing with pen pressure (stroke thickness), length of stroke and reference line for writing.

In an online recognition system, the temporal or dynamic information of the writing are captured in real-time. This information includes the order of the strokes, the direction of the writing for each stroke, the tilt of the pen and the writing speed within each stroke. In this research, temporal information used in the recognition algorithms includes x, y coordinates, pen up/down indicators and time stamps. Other information such as pen pressure and pen tilt is regarded as unreliable. Therefore, the recognition system is not able to differentiate between the light and heavy strokes which are feature's of Pitman shorthand. The unique determination of each stroke depends on the context of the sentence in the linguistic analysis.

In a typical shorthand passage, the consonant strokes form a large percentage of the shorthand outlines. In both Pitman shorthand and Renqun shorthand, consonant strokes with different lengths convey different syllables. For example, Pitman uses halve and thickened N or M stroke or just halve the length of N stroke to indicate the succeeding sound of D or the suffix /MENT/ respectively. And all curved strokes are doubled in length to present the addition of the syllables –TER, -DER, -THER and –TURE. According to the experiment the variation in stroke length (Leedham & Downton, 1986), 15% of normal length Pitman outlines overlap with the distribution of half and double length strokes. That is to say, the classification rate of normal length can only be around 85% without the use of contextual information. To avoid such ambiguity, stroke length is not considered in the recognition system. Two consonant strokes are classified to be 'similar' if they have the same shapes even if their lengths are different. For each classified stroke, its type and length will be transferred for the post-processing. If the length lies in the 15% overlap region, both possibilities (normal & half or normal & double) would be enumerated for the word-level or sentence-level linguistic analysis.

In both Pitman and Renqun shorthand systems, the writing line is used as a reference for writing. Outlines with the same shapes and sizes but different positions to the reference line represent different syllables. In Pitman notation, there are three groups of Pitman outlines depending on their position relative to the single reference line. They are:

(a) Complete outline above the reference line;

(b) Outline on the reference line;

(c) Part of the outline cutting through the reference line.

Renqun notation uses two writing lines to fix the position of each outline. There are three groups of Renqun outlines according to their position relative to the two reference lines:

(a) Complete outline on the first reference line;

(b) Outline between two reference lines;

(c) Outline below the first reference line and cutting through the second reference line.

In the system, the outline's position to the references line is not considered. Two outlines are classified to be 'similar' if they are composed of a sequence of 'similar' strokes even if their position relative to the reference line is different.

After information such as the category of classified strokes, the length of each stroke, the outline's position to the reference line has been classified by the recognition system, it is transferred to the transcription system for the correctly text output.

3.4.3 Recognition of Pitman shorthand

As we have already stated, there are three types of Pitman shorthand patterns: shortforms, consonant outlines, and vowel & diphthong symbols. Therefore Pitman shorthand recognition system includes three core algorithms for shortform classification, consonant outline recognition and vowel & diphthong classification, as shown in Figure 3.7.

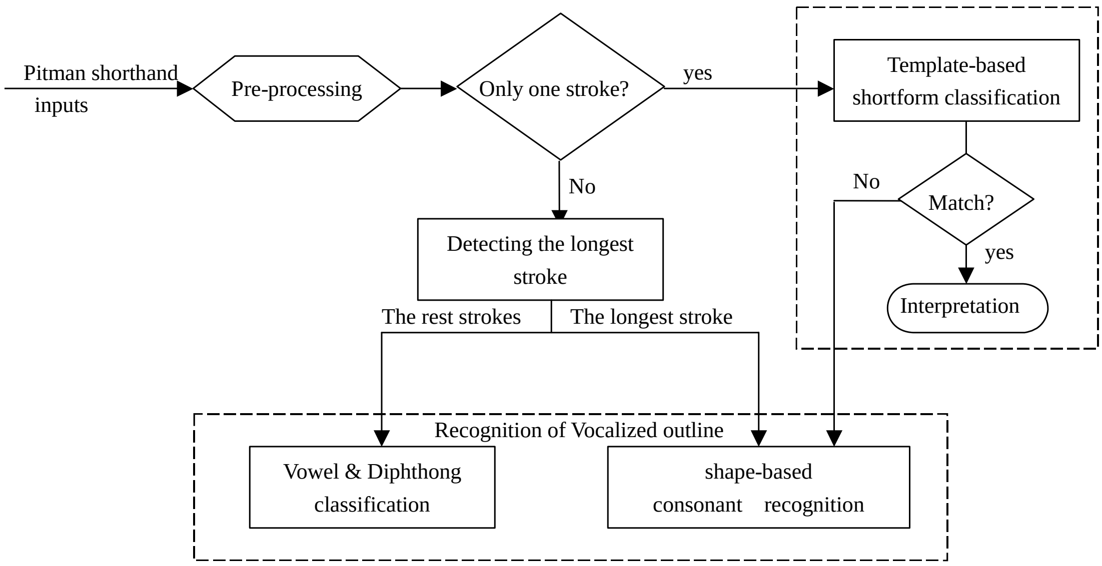


Figure 3.7: Diagram of PSRS

It can be seen from Figure 3.7 that shortforms are distinguished from vocalized outlines by the number of pen-strokes. If the outline consists of a single pen-stroke, it is assumed as a shortform, which can be classified directly by a template-matching approach. If the input outline consists of two or more pen-strokes, it is assumed to be a vocalized outline, for which recognition involves two parts: consonant outline and surrounding vowel and diphthong symbols. In the recognition of consonant outlines, a ranked list of interpretation of each stroke is obtained after segmentation and classification. It is worth emphasizing that all possible interpretations of geometric features are preserved and passed to the transcription system for phonetic interpretation and conversion to correctly spelt English text.

3.4.4 Recognition of Renqun shorthand

According to the comparison between Pitman shorthand and Renqun shorthand, it can be concluded that it is not necessary for RSRS to deal with Renqun handwriting in terms of shortforms and vocalized outlines and the algorithms for recognizing Renqun shorthand should be designed for rhymes and consonants since both vocalized outline and shortform are composed of rhymes and consonants.

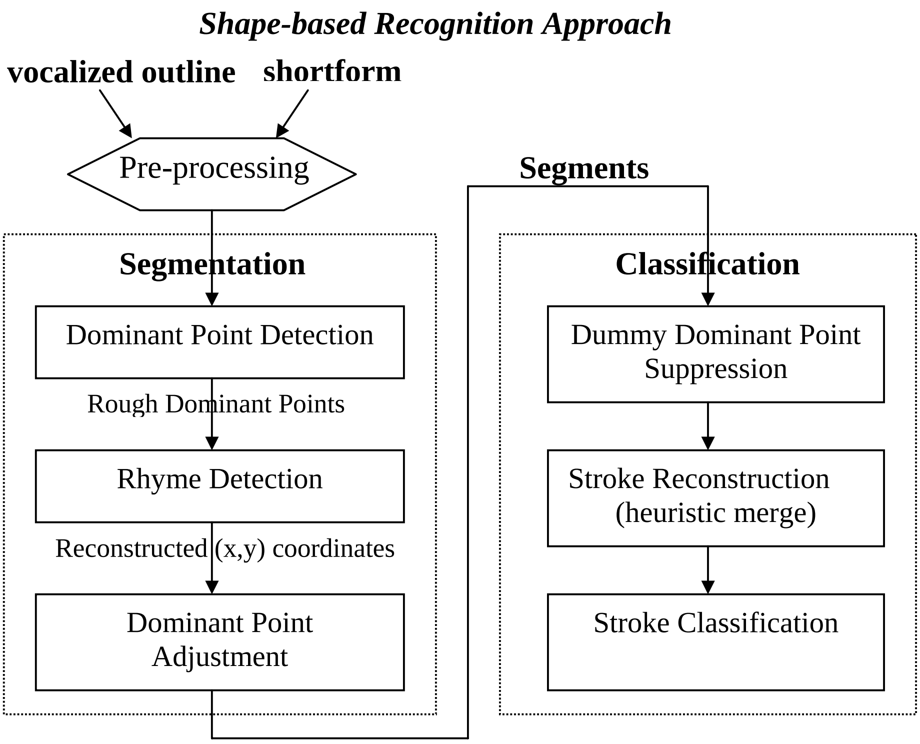


Figure 3.8: Diagram of RSRS

The diagram of the recognition system for Renqun shorthand is shown in Figure 3.8 where the recognition approach to Renqun shorthand includes two-stages called segmentation and classification. This recognition approach used in RSRS is called a shape-based two-stage approach since dominant point detection and adjustment in the segmentation stage mainly rely on techniques in shape recognition. In detail, a non-parametric approach based on “the region of support” is used to detect rough dominant points, which indicate potential junctions of stroke features. Afterwards, the positions of dominant points are adjusted by polygonal approximation after rhymes like short primitives, circles, loops and hooks have been detected. Over-segmentation is preferred in the segmentation and more than enough dominant points are obtained. In the classification stage, dummy dominant points are suppressed to form a valid path for the segmented strokes and a standard back-propagation neural network is used for the stroke classification.

CHAPTER 4

Recognition of the vocalized outline in Pitman shorthand

In Pitman shorthand, vocalized outlines are composed of a single consonant outline surrounded by vowel & diphthong symbols. In the following sections, recognition of the consonant outline and vowel & diphthong symbols are introduced respectively. For each outline, its sequences of x,y coordinates and chain-code values are calculated firstly for the convenience of usage in the following recognition algorithms.

4.1 Recognition of consonant outline

The consonant outline, the most difficult part of a Pitman shorthand outline to recognize, is segmented and classified by the shape-based two-stage approach shown in Figure 4.1. In the segmentation stage, a non-parametric approach is firstly used to select rough dominant points, which indicate potential junctions of stroke features. Afterwards, the position of dominant points should be adjusted by polygonal approximation after special geometric features like circles, loops and hooks have been detected. In order to avoid losing information rich dominant points, over-segmentation is preferred in the segmentation stage. In the classification stage, dummy dominant points are firstly suppressed and merged. Afterwards, a combination of segments (paths) are searched using a heuristic merge algorithm.

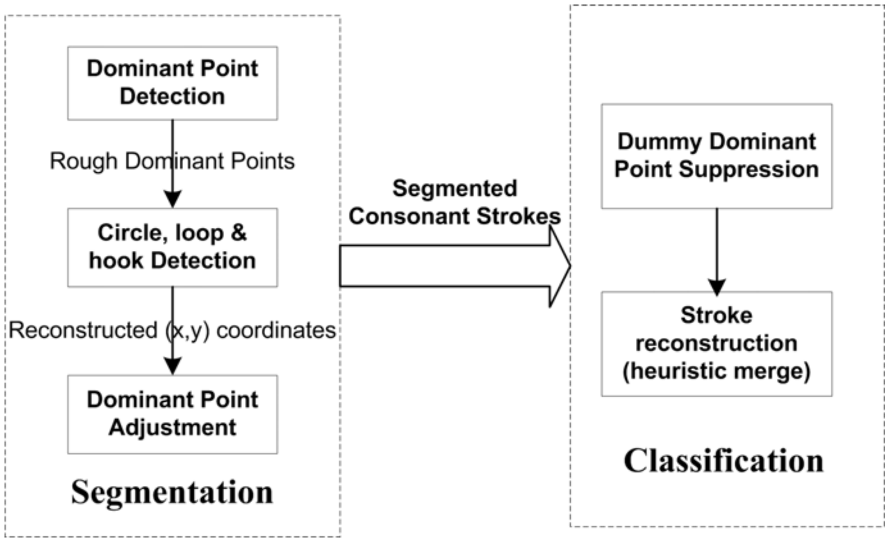


Figure 4.1: Diagram of the shape-based two-stage approach

4.1.1 Segmentation

The purpose of the segmentation stage is to produce a sequence of segmented consonant strokes which are easily classified in the next stage. Below, the three steps of this stage are described:

4.1.1.1 A non-parametric dominant point detection approach

Unlike conventional curvature-based approaches which are sensitive to noise, the approach proposed here is robust and independent of parameters. Specifically the approach is based on the dominant point detection method proposed by Teh & Chin (1989).

Teh & Chin pointed out that determining the region of support is the main issue in dominant point detection. Incorrectly chosen regions of support may cause the measures of significance to be computed over inappropriate neighborhoods which may subsequently cause dominant points to be discarded. The region of support is symmetrical which is determined as follows:

For any point p_i, l_{ik} is the length of the chord joining the points p_{i-k} and p_{i+k} as $l_{ik} = \left|\overline{p_{i-k}\,p_{i+k}}\right|$, d_{ik} is the perpendicular distance of the point p_i to the chord $\overline{p_{i-k}p_{i+k}}$. d_{ik} is signed and calculated by the following formula $d_{ik} = \frac{[x(i+k)-x(i-k)]\times[y(i-k)-y(i)]-[x(i-k)-x(i)]\times[y(i+k)-y(i-k)]}{\sqrt{[x(i+k)-x(i-k)]^2+[y(i+k)-y(i-k)]^2}}$, where $x(i)$ and $y(i)$ are the x, y coordinates of point p_i.

Start with k=1. Compute l_{ik} and d_{ik} until

Rule 1: $l_{i,k} \geq l_{i,k+1}$ or

Rule 2: $d_{ik}/l_{ik} \geq d_{i,k+1}/l_{i,k+1}$ for $d_{ik} > 0$

$d_{ik}/l_{ik} \leq d_{i,k+1}/l_{i,k+1}$ for $d_{ik} < 0$

Then the region of support of point p_i is the set of points which satisfy either Rule 1 or Rule 2 and are denoted as $D(p_i) = \{p_{i-k,}, ..., p_{i-1}, p_i, p_{i+1,} ..., p_{i+k}\}$

Our algorithm consists of three steps: determination of region of support, computation of significance and suppression of dummy dominant points.

(1) Determination of region of support

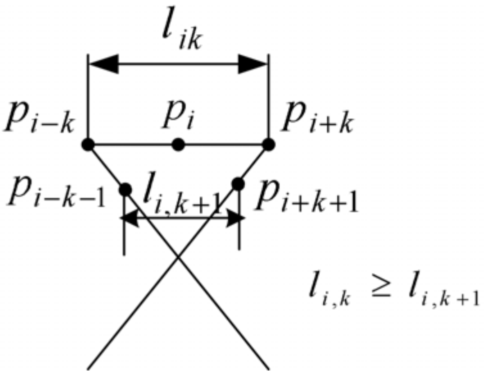
Besides Rules 1 and 2, two extra rules are added to determine the region of support. They are

Rule 3: $l'_{i-k,i+k} \geq l'_{i-k,i+k+n}$ or $l'_{i-k,i+k} \geq l'_{i-k-n,i+k}$

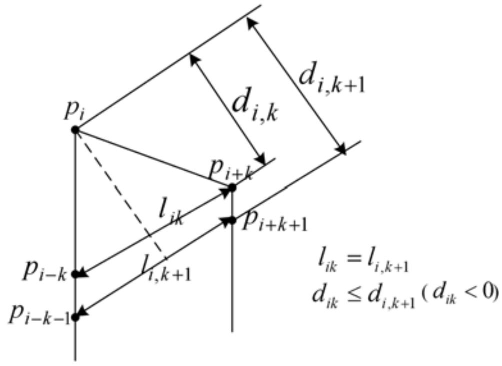
Rule 4: $d'_{i,i-k,i+k} \geq d'_{i,i-k-n,i+k}$ or $d'_{i,i-k,i+k} \geq d'_{i,i-k,i+k+n}$

Where $l'_{x,y}$ is the length of the straight line from point x to point y; $d'_{x,y,z}$ denotes the perpendicular distance from point x to the line between point y and point z.

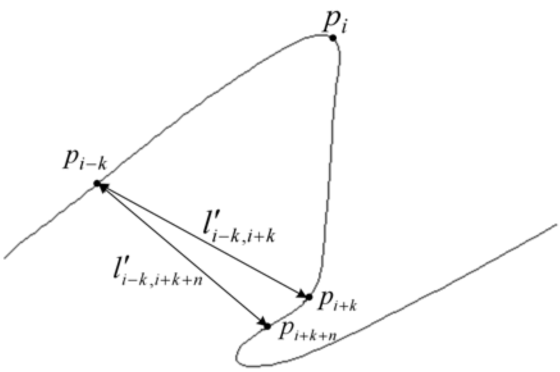
Figure 4.2 illustrated the above four rules graphically.



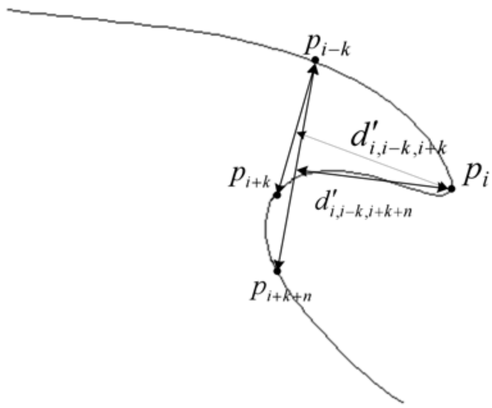
(a) Rule 1



(b) Rule 2



(c) Rule 3



(d) Rule 4

Figure 4.2: Illustration of rules for determination of region of support

The aim of introducing the Rule 3 and Rule 4 is to prevent hook points (as shown in Figure 4.2 (c) (d), point p_{i+k} is hook point which indicates the existence of a hook) from being falsely eliminated. Using the rules by Teh & Chin, point p_{i+k} cannot be detected. The region of support of point p_i will extend to point p_{i+k+n}, which will erase point p_{i+k} in the subsequent deletion procedure. As shown in Figure 4.2(a), Rule 3 prevents the region of support of point p_i from extension at point p_{i+k+n} since $l'_{i-k,i+k} \geq l'_{i-k,i+k+n}$. Figure 4.2(b) illustrates Rule 4 in the same way where $d'_{i,i-k,i+k} \geq d'_{i,i-k,i+k+n}$.

As for those non-dominant points whose chain-code change is zero, their region of support does not need to be calculated.

(2) Computation of significance

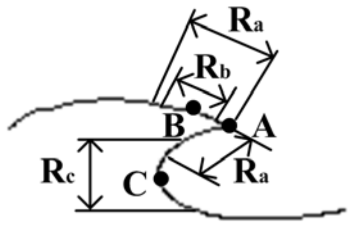
Curvature is used to represent the significance of each point. Since the input outline was time normalized in the preprocessing stage, the curvature of point i is computed as:

$$K_i = \left| \tan^{-1}\{(y_{i+k} - y_i)/(x_{i+k} - x_i)\} - \tan^{-1}\{(y_i - y_{i-k})/(x_i - x_{i-k})\} \right|$$

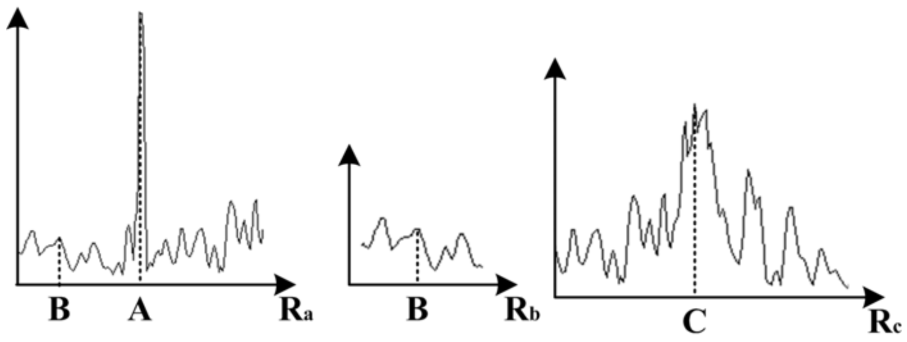
As a parameter to reduce noise, k is related to the resolution of the digitizer. If the resolution is low, k should have a large value and vice versa. In the system, k =3 is used.

(3) Suppression of dummy points

For each point and its region of support, only one point with the maximal curvature value remains after the suppression procedure.



(a) The region of support of points A, B and C



(b) Curvature of Ra, Rb and Rc

Figure 4.3: Dominant point detection processes

Figure 4.3 illustrated the above three processes. As shown in Figure 4.3(a), the region of support of three points A, B and C on the input outline is Ra, Rb and Rc respectively. The curvature curve of each region of support is showed in Figure 4.3(b). As point B and point A lie in the same region of support of point A, one of them will be deleted in the dummy dominant point suppression stage. And point B is merged because its curvature value is smaller than that of point A.

The above approach can be used to produce dominant points of the consonant outlines. Afterwards, a series of adjustments are needed to achieve high performance even under adverse conditions.

4.1.1.2 Detection of circles, loops and hooks

As shown in Figure 4.4, for different circles and loops in shorthand, both the number and the position of the dominant points detected by the approach described in section 4.1.1.1 are variable. In order to guarantee the overall performance, those special geometric features like loops, circles and hooks should be found first.

Figure 4.4: Different dominant points for circles and loops

<u>Circles and loops detection</u>

Abuhaiba et al. (1994) replaced all loops in off-line handwritten Arabic characters with a new prominent point before extracting their dynamic information using the following two rules:

Rule 1: “Given any vertex in the loop, there is a path from that vertex to itself which visits all the other vertices in the loop only once, and which traverses all the links in the loop only once.”

Rule 2: “No two vertices in the loop are connected by a path which crosses the interior of the loop.”

It is relatively easy to detect the loop in Arabic scripts since each character contains one or at most two adjacent loops occurring either at the beginning or the end of the pen-stroke. But the above approach is not suitable to shorthand systems in which both the number and the position of loops are uncertain.

As shown in Figure 4.5, there are five kinds of circles and loops defined in Pitman and Renqun shorthand whose characteristics are illustrated in Table 4.1.

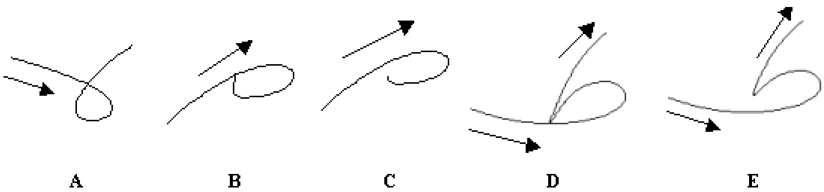


Figure 4.5: Different kinds of loops in Pitman shorthand

Table 4.1: Characteristics of different types of loops in Pitman shorthand

Loop Type	Characteristics
Type A	Fully closed with crossover
Type B	Fully closed, no crossover
Type C	Incomplete closure
Type D	Fully closed with direction reversal
Type E	Incomplete closure with direction reversal

(a) Approach to detecting type A loop

The algorithm consists of four steps as follows: Firstly, the input outline is converted into a 16-direction Freeman chain code sequence l (l=0,1,2….15). Secondly, the input outline is separated into several regions at points whose preceding and succeeding chain code values are different. Each region is of a constant chain code value and is viewed as a straight line. Thirdly, each region is compared with all its previous regions to determine whether there is any overlapping, which can be detected by examining the position, the height and the width of the region. Fourthly, coordinates of the intersection point of two overlapping regions are calculated. If the intersection point lies in both regions, it is assumed that a loop has been detected.

Compared with the method proposed by Abuhaiba et al. (1994), our approach evaluates the probability of overlapping between two regions at first and filters out unpromising combinations like region 2 and 3 in Figure 4.6, which improves the computational efficiency greatly.

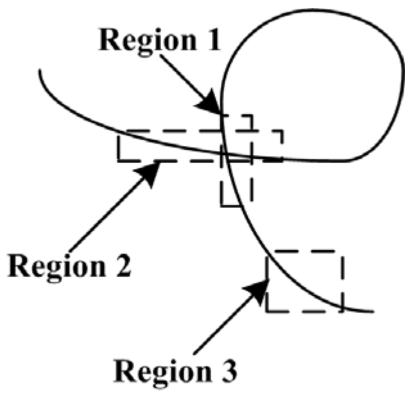


Figure 4.6: Regions to detect intersection points

(b) Approach to detecting type B and type C loops

In order to apply the above approach to both Type B and Type C loops, the two original patterns should be changed into Type A with the extension technique at suitable place. Clearly, Type B and Type C loops lie at the beginning, the end of an input outline, therefore a fixed length is extended at the beginning and the end point of a pen stroke respectively (Figure 4.7). If the extension is made at the start point, the extension direction should be opposite to the direction of pen stroke. If the extension is made at the end point, the extension direction should be in the same as the direction of the pen stroke.

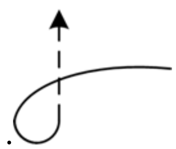


Figure 4.7: Extension made for detecting type B and type C loops

Afterwards, these extended Type B and Type C loops can be handled with the algorithm for Type A loop.

(c) Approach to detecting type D and type E loops

Type D and Type E loops always occur in the middle of an input outline, therefore a extension should be made at the corner point of a pen stroke (Figure 4.8).

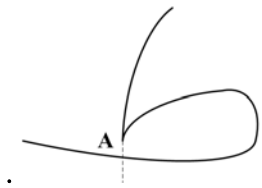


Figure 4.8: Extension made for detecting type D and type E loops

A corner point can be selected by its angular change value. From Figure 4.9, it is clear to see that the angular change of corners (Ψ_A and Ψ_D) is larger than that of smooth joins (Ψ_B and Ψ_C). In our system, Ψ_i =60° obtained by experimentation, is used to distinguish the sharp corner from the smooth join.

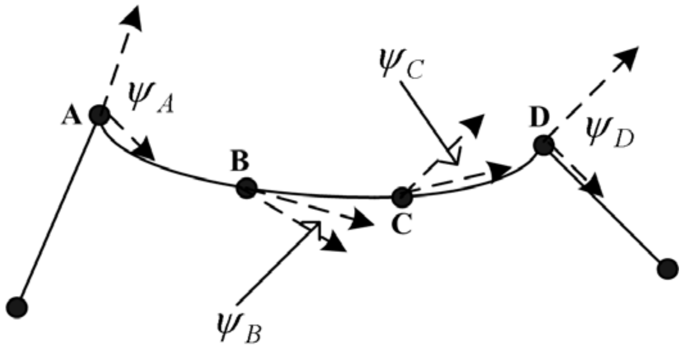


Figure 4.9: Angular change of dominant points

(A and B are corners, B and C are smooth joins)

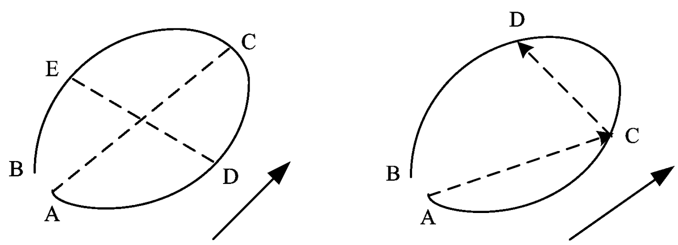
Afterwards, these extended Type D and Type E loops can be handled with the algorithm for type A loop.

(d) Properties of circles and loops

Besides the location of circles and loops, other relevant information such as its type (circle or loop), closed or unclosed (detected using the extension technique or not), orientation (clockwise or anti-clockwise), size (small or large) should also be identified.

As shown in Figure 4.10(a), the type of circle/loop can be determined using the following steps. After the start point A and the end point B of a loop/circle have been detected, point C, which has maximal distance to A, is found. Then a pair of points, D and E, lying in the middle of A and C, or C and B respectively, are identified. In our algorithm, the larger one between $|AC|/|DE|$ and $|DE|/|AC|$ is used to determine whether it is a circle or a loop. If it is smaller than a predefined threshold, it is assumed to be a circle; otherwise, it is assumed to be a loop. In our system, the threshold is 1.8. The larger the threshold, the greater the possibility of classifying as a circle; the smaller the threshold, the greater the possibility of classifying as a loop.

As show in Figure 4.10(b), the orientation can be detected by the 16-direction Freeman chain code of AC and DC, which represents their relative position. For example, if the chain code value of AC and DC are 1 and 7 respectively, the orientation of the circle/loop is anti-clockwise.



(a) Classification of circle and loop (b) Orientation of loop and circle

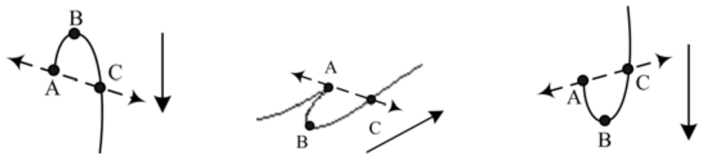
Figure 4.10: Properties of circles and loops

Since the input outline has been normalized in the preprocessing, the size of loops and circles is proportional to the number of pixels.

Hook Identification

A hook could lie at the beginning, the end or in the middle of an outline. If the distance between two consecutive dominant points is smaller than a predefined threshold (in our system, 5 mm which denotes the minimal distance of two hook points, is used), the hook identification algorithm is triggered.

Extension is made at corner point A (dashed lines in Figure 4.11) to find any intersection possibilities with the input outline. The direction of the extension is orthogonal to line AB. If an intersection point C is located, it is assumed that a hook is found.



(a) Hook at the beginning (b) Hook in the middle (c) Hook at the end

Figure 4.11: Hooks encountered in shorthand outlines

Using the same techniques described in circles & loops detection, the orientation and size of hooks can be determined.

A rough dominant point set, which includes the following three types of dominant points, is obtained after the above procedures.

(1) Special points: Points which represent the existence of hooks, circles or loops. After circles, loops and hooks have been detected, new dominant points are added at the positions of intersections or hook points to replace other nearby dominant points. The newly added dominant points cannot be moved or merged any more, whilst pixels in the circles, loops and hooks are ignored in the following processes.

(2) Smooth joins: Points lie at the smooth region of the input outline. These points are more likely to be suppressed in the following procedures.

(3) Corners: Points lie at the angular region of the input outline. Compared with smooth joins, they are less likely to be suppressed. The rule to distinguish corners from smooth joins has been illustrated in section 4.1.1.2.

4.1.1.3 Split-and-merge by polygonal approximation

In order to adjust the position of dominant points, a split-and-merge algorithm based on polygonal approximation is employed. Split is emphasized, which tends to produce over-segmentation. Merge is carried out cautiously to avoid the loss of important segmentation points. The detail of the algorithm is as follows:

(1) For any two consecutive dominant points, two polygonal approximation parameters are calculated. The first refers to the ratio of the chord length over the straight length of the two dominant points; the second refers to the area between the chord and the straight line.

(2) If any of the two parameters exceeds predefined thresholds, a split is triggered. In the system, the first parameter should be larger than 1.2 and the second parameter should be larger than 150. The two parameters are a relatively small value to favour over-segmentation. If the value of the parameters are two low, the split algorithm will be useless; if the values of the parameters are too high, too many segmentation points will be produced although most of them contains little information and have to be merged by the stroke reconstruction process in the classification stage. The point with the longest perpendicular distance to the straight line between the two dominant points is chosen as a new dominant point. If the parameters are less than a predefined threshold, the second dominant point is merged. In the system only when the first parameter is smaller than 1.05, which means the two dominant points are collinear, merging is carried out. If the value of the parameter is too high, many information rich dominant points will be merged; if the value of the parameter is too low, no points even collinear points will be merge. For example, if this parameter is less than or equal to 1, merge algorithm will never be triggered. The beauty of the split algorithm is that it is helpful to detect smooth junctions causing large polygonal approximation parameters. Figure 4.12 illustrates the split procedure for the outline AB. Smooth junction C can be detected by splitting the outline from A to B if any polygonal approximation parameters between AB is larger than the predefined threshold. The above procedure is implemented recursively until polygonal

approximation parameters between any two consecutive dominant points satisfy the above criteria.

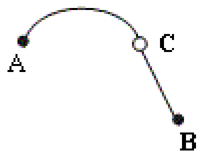


Figure 4.12: Illustration of the split algorithm

After the spilt-and-merge procedure, the following criteria is applied to adjust the position of dominant points: For any three consecutive dominant points, the middle one should be the point with the maximal perpendicular distance to the straight line joining the other two points. This guarantees each dominant point lies at the most suitable place.

4.1.2 Classification

In the segmentation stage, over-segmentation is preferred which produces more than enough dominant points. Therefore, suppression of dummy dominant points is the major objective in the classification. First, obvious dummy dominant points are merged. Afterwards, a heuristic search algorithm is applied to find the optimal path of the consonant outline. In the above processes, a stroke/non-stroke classifier is constructed to classify whether each segment is a stroke or not. And another neural network classifier is used to classify individual strokes. Finally a feature-to-phoneme constructor is introduced to convert the pattern primitives obtained in the shape recognition system into phonetic representations.

4.1.2.1 Neural network classifiers

There are 24 types of basic consonant symbols defined in Pitman shorthand. Since circles, loops and hooks have been classified in the segmentation stage and stroke thickness is not considered, only 14 types of basic consonant symbols remains (see Figure 4.13).

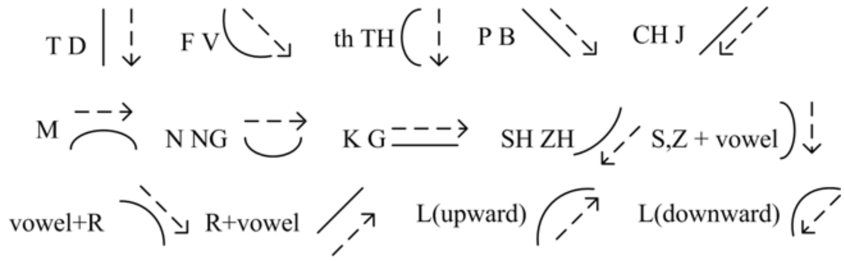


Figure 4.13: 14 types of basic consonant symbols

Two back-propagation neural networks are constructed as stroke/non-stroke classifier and individual stroke classifier respectively. Both of them use the same input features extracted as follows:

(1) Feature extraction

The input vector of the neural network represents the major features of the basic consonant strokes. Three aspects should be taken into consideration to extract representative features. They are:

First, based on the direction of the straight line from the start point to the end point, the basic consonant symbol can be classified into five major categories as shown in Figure 4.14.

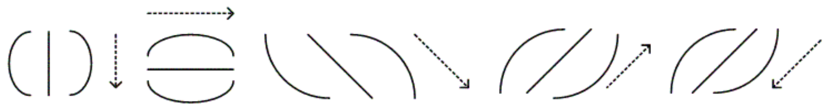


Figure 4.14: Five kinds of basic consonant strokes

(The arrow represents the direction of the line from the start point to the end point)

Second, for each category, the maximal and average perpendicular distance of all points to the straight line between the start point and the end point are effective to distinguish whether the consonant stroke is a straight line or a curve. Distance of the point {x(i),y(i)} is calculated using the following formula

$$d = \frac{[x(end)-x(start)]\times[y(start)-y(i)]-[x(start)-x(i)]\times[y(end)-y(start)]}{\sqrt{[x(end)-x(start)]^2+[y(end)-y(start)]^2}}$$

Where $x(start)$, $y(start)$ and $x(end)$, $y(end)$ represent x, y coordinates of the start point and the end point respectively.

Thirdly, if the input outline is a curve written in the clockwise direction, the distance *d* of points in the input outline is negative; if the input outline is a curve written in the anti-clockwise direction, the distance *d* is positive; if the input outline is a straight line, the distance *d* could distribute in the two sides of zero. So the percentage of points with positive and negative distances could be used as input vectors.

Based on the analysis above, nine inputs were extracted as follows:

(1) The ratio of the straight length over the length of the chord from the starting point to the end point;

(2) The ratio of the width over the perimeter of the bounding box of the input outline;

(3) The ratio of the height over the perimeter of the bounding box of the input outline;

(4) The sine values of the direction of the straight line from the starting point to the end point;

(5) The cosine values of the direction of the straight line from the starting point to the end point;

(6) The maximal, average distance of all the points in the input outline to the straight line from the start point to the end point;

(7) The average distance of all the points in the input outline to the straight line from the start point to the end point;

(8) The percentage of negative distance points;

(9) The percentage of positive distance points.

The above 9 features were separately normalized to the range of 0 to 1.

(2) Training processes

a. Stroke/non-stroke classifier

A 9-25-2 back-propagation neural network was used to classify whether each segment is a stroke or not. Two categories of patterns: positive pattern and negative pattern are used as training samples. The positive patterns are 14 types of basic consonant strokes. The negative patterns are non-stroke patterns resulting from the possible combinations of valid basic consonant strokes. For example, the intended two strokes  may be erroneously classified as the single stroke .

However, there are so many possible combinations for non-stroke patterns. It is a challenge for a small size of neural network to learn all kinds of possibilities. A stroke/non-stroke discriminator (see Figure 4.15) is introduced to filter out certain apparently non-stroke patterns. Only when the input outline is classified as a stroke by the discriminator, will the neural network process it. The discriminator and the stroke/non-stroke neural network are called as stroke/non-stroke classifier.

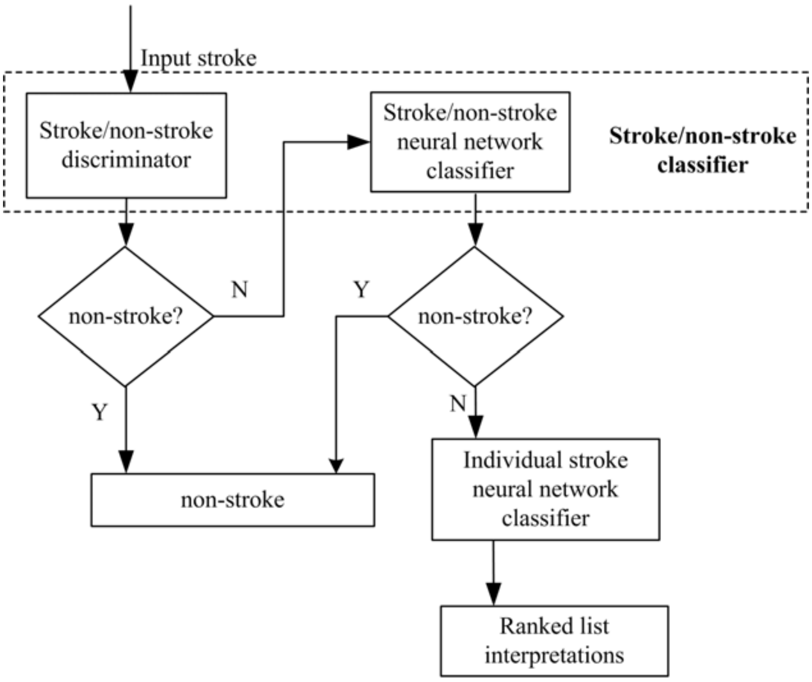


Figure 4.15: Diagram of the non-stroke discriminator

Two criteria used in the stroke/non-stroke discriminator are described as follows:

Criteria 1: The outline containing a sharp corner is a non-consonant pattern (eg, outline ABC in Fig. 4.16).

Criteria 2: If there are other dominant points located between the start point and the end point (that means the outline is not a straight line) and the percentage of neither positive distance points nor negative distance points is predominant, such an input outline is also a non-consonant pattern (eg, outline CDEF in Fig. 4.16).

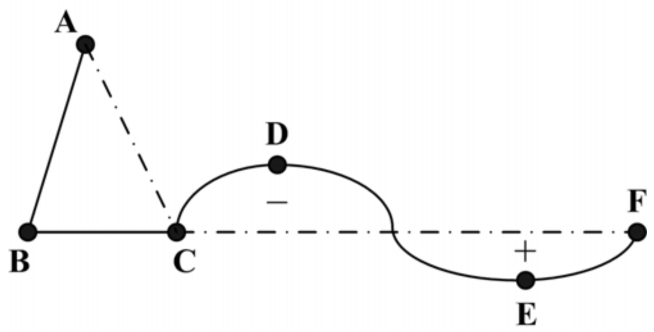


Figure 4.16: Two situations of non-consonant patterns

In order to achieve good performance, a wide range of examples that exhibit all different characteristics were compiled to train the neural network. In total, 420 positive samples and 410 negative samples were used. Positive and negative samples were selected manually from the Pitman shorthand outline written by three writers specially to train the neural network. For example, in the outline of the word ‘clergyman’ shown in Figure 4.17, strokes ***bc, cd, de*** and ***ef*** are marked as positive samples and strokes ***abc, cde*** and ***efg*** are marked as negative samples. For each sample, nine features in section 4.1.2.1 were extracted and fed into the neural network as the input vector. For negative samples, the desired output is 0. For positive samples, the desired output is 1. The accuracy of the neural network using the 5-fold cross-validation strategy (The sample data was divided into 5 subsets of equal size. The neural network was trained 5 times, each time leaving out one of the subsets from training for testing.) is shown in Table 4.2.

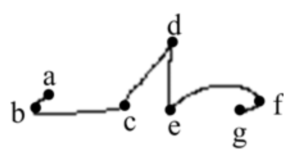


Figure 4.17: Positive and negative training samples

Table 4.2: Accuracy of stroke/non-stroke neural network classifier

Average accuracy	**Training**	**Testing**
Stroke classification	99.11%	97.62%
Non-stroke classification	98.78%	96.34%
Average	98.95%	96.98%

b. Individual stroke neural network classifier

A 9-30-14 back-propagation neural network was used to classify the class that the stroke belongs to. 14 output dimensions denote 14 types of strokes. Three writers were asked to write each type of stroke 20 times. As shown in Figure 4.18, all samples were required to be written in a bounding box and strictly followed the writing rules of Pitman notation. In total, 840 samples with 60 samples for each stroke type were collected. The average classification rate obtained using 5-fold cross-validation strategy is shown in Table 4.3.

The outputs of the neural network classifier denote the posterior possibility of each type of stroke. The classification result is a ranked list interpretation including the following two types of strokes: The first type is the stroke whose corresponding neural network output is larger than the pre-set threshold (to retain all possibilities, a small value of 20% was used), and the second type is the stroke with similar patterns to the segment. For two symbols, the similarity stands for: The difference of their Freeman chain codes from the start point to the end point is less than or equal to 1. All these possibilities are ordered by their possibilities and transferred to the transcription system for semantic analysis.

Table 4.3: Accuracy of stroke neural network classifier

Stroke type	Training accuracy	Testing accuracy
T 1:	87.5%	75%
T 2:	100%	100%
T 3:	93.75%	83.33%
T 4:	100%	96.43%
T 5:	100%	91.66%
T 6:	97.92%	100%
T 7:	100%	100%
T 8:	93.75%	75%
T 9:	100%	100%
T 10:	100%	100%
T 11:	100%	100%
T 12:	95.83%	66.67%
T 13:	100%	75%
T 14:	100%	66.67%
Average	97.76%	87.84%

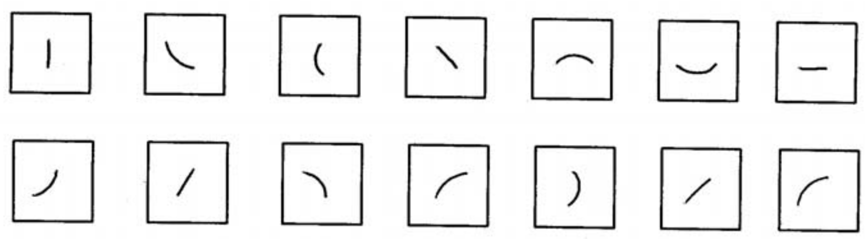


Figure 4.18: Training samples

4.1.2.2 Deletion of dummy dominant points

In this step, obvious unnecessary dominant points are merged. Two notions: "minimal stroke length" (MinSL) and "maximal stroke length" (MaxSL) are introduced as suppression criteria. In PSRS, MinSL is equal to the length of a halved length stroke and MaxSL is equal to the length of a doubled length stroke.

Suppose there are four consecutive dominant points p_{i-1}, p_i, p_{i+1}, p_{i+2} obtained in the segmentation stage and points p_{i-1}, p_{i-2} … are all reliable dominant points (the algorithm starts from the start point p_0, which is definitely a reliable dominant point). Starting from p_{i-1}, if the length from dominant point p_i to p_{i+1} ($dis(p_i, p_{i+1})$) is smaller than MinSL and both $dis(p_i, p_{i+2})$ and $dis(p_{i-1}, p_{i+2})$ are smaller than MaxSL, the algorithm based on recognition confidence to detect whether point p_i or p_{i+1} should be merged, is triggered. The confidence of strokes from dominant point p_{i-1} to p_i is denoted as $cf(p_{i-1,i})$, which denotes the reliable degree of this stroke. It is equal to the maximal output of the stroke neural network classifier. Then point p_i and point p_{i+1} are identified as follows:

(1) Start point, end point, intersection points and hook points are defined as special points.

If point p_i is a special point and p_{i+1} is not, point p_{i+1} is deleted;

If point p_{i+1} is a special point and p_i is not, point p_i is deleted;

If both point p_i and point p_{i+1} are special points, both points are preserved.

(2) Calculate $cf(p_{i-1,i})$ and $cf(p_{i-1,i+1})$.

If $cf(p_{i-1,i}) \geq 2 \times cf(p_{i-1,i+1})$ and $cf(p_{i-1,i}) \geq 50\%$, point p_{i+1} is deleted;

If $cf(p_{i-1,i+1}) \geq 2 \times cf(p_{i-1,i})$ and $cf(p_{i-1,i+1}) \geq 50\%$, point p_i is deleted;

Else go to step 3;

(3) Calculate $cf(p_{i,i+2})$ and $cf(p_{i+1,i+2})$.

If $cf(p_{i,i+2}) \geq 2 \times cf(p_{i+1,i+2})$ and $cf(p_{i,i+2}) \geq 50\%$, point p_{i+1} is deleted;

If $cf(p_{i+1,i+2}) \geq 2 \times cf(p_{i,i+2})$ and $cf(p_{i+1,i+2}) \geq 50\%$, point p_i is deleted;

Else go to step 4;

(4) If $cf(p_{i-1,i}) + cf(p_{i,i+2}) > cf(p_{i-1,i+1}) + cf(p_{i+1,i+2})$, point p_{i+1} is deleted;

If $cf(p_{i-1,i+1}) + cf(p_{i+1,i+2}) > cf(p_{i-1,i}) + cf(p_{i,i+2})$, point p_i is deleted.

The above procedures were implemented recursively until the distance between any two consecutive dominant points are larger than MinSL and smaller than MaxSL.

4.1.2.3 Stroke reconstruction by heuristic merge

If n segments are left, at most 2^{n-1} possible combinations are available. The search path will grow exponentially if nearby points are merged one by one to produce all possibilities. In this system, a heuristic merge algorithm is proposed to reduce the computational complexity of stroke reconstruction. For each dominant point p_i, a parameter named "importance" of a point ($\mathrm{Im}(p_i)$), which is the length from point p_i to the nearest point having a different chain code value from point p_i, is defined to implement heurist merge. If point p_i lies in a smooth region, $\mathrm{Im}(p_i)$ is large and p_i should be merged with high priority. Otherwise, if p_i lies in a sharp region, $\mathrm{Im}(p_i)$ is small and the possibility for point p_i as a correct segmentation point is high.

The following is the detailed search procedures:

First, the "importance" values of all dominant points are calculated. The maximum and minimum values are denoted as max(Im) and min(Im) respectively. All these values are referred to as a set of merge thresholds, which is denoted as $T = \{\mathrm{Im}_1 ..., \mathrm{Im}_i ,.... \mathrm{Im}_{n+1}\}$, where $\min(\mathrm{Im}) \leq \mathrm{Im}_i \leq \mathrm{Im}_{i+1} \leq \max(\mathrm{Im})$. For any merge threshold Im_i, points satisfied the following three criteria should be merged one by one:

(a) Points whose "importance" value is larger than Im_i have the possibility to be merged.

(b) For each point, its merge priority is directly proportional to its "importance" value.

(c) If a point is merged, the distance between its preceding point and its succeeding point should not exceed MaxSL.

Afterwards, the combination of segments generated from the merge are classified by a stroke/non-stroke classifier. If all segments are classified as stroke, it is assumed that a valid path is found. Otherwise, the combination is regarded as invalid. For each available path, its "confidence" is equal to the summation of maximum interpretation of each segment divided by the number of segments. The path with the largest "confidence" value is selected as the finalized one. For two paths with the same "confidence", the one obtained at a smaller Im_i is regarded as more reliable. In this system, all available paths and the ranked list interpretation of each segment are ordered by their "confidence" and transferred to the transcription system.

4.1.2.4 Feature to phoneme conversion

In some cases, a phonetic representation is composed of more than one feature primitives. For example, outlines of "W" and "Y" are classified separately as a small hook (or) and a stroke written upwards . Therefore, an additional step is needed to correctly interpret clustered primitives into related phonemes. This process is referred as "Feature to Phoneme Conversion".

Five production rules (Htwe et al, 2004) and related phonemes are proposed as follow:

(a) Direct translation rule: all consonants except Y, W and H

(b) Primitive combination rule: W, Y, H

(c) Primitive combination and reverse ordering rule: PL, BR, etc, PR, BR, etc., FR, VR, etc., and FL, VL etc

(d) Feature Detection rule: SES, ZES circles, ST, STER loop, N, F, V, SHUN hook, suffix –SHIP hook, suffix –ING/INGS dot

(e) Length Detection rule: MD, ND, suffix –MENT, half length strokes, double length strokes

4.1.3 Issue of smooth junctions

In Pitman shorthand, there are three types of boundaries between adjacent primitives: low angularity ($<90^o$), high angularity ($\geq 90^o$) and smooth junction. Figure 4.19(a) shows some examples of smooth junctions in which the dots represent the place for the right segmentation. Clearly, it is difficult to locate these types of dominant points in the smooth regions using mathematical tools.

The notion of a ‘machinography’ was proposed by Brooks (1985) as a script specially designed or tailored for the purpose of machine communication. Later investigations (Leedham & Qiao, 1992) pointed out that compared with other shorthand and speedwriting systems, Pitman shorthand exhibits many similarities to a machinography which makes it a potentially successful means of fast input of text to computer. However, smooth junctions deteriorate the role of Pitman shorthand as a machinography since it makes physical boundaries unclear for machine detection. It has been shown that (Ma et al., 2004) smooth junction degraded the recognition accuracy by up to 15%. Therefore, it is possible to resolve the smooth junction problem from human behavior perspective. We define a machine compatible shorthand based on constraints proposed by Choi & Bang (2000) to redefine handwritten Korean character to improve its machine readability. The following three criteria should be followed strictly in the machine compatible Pitman shorthand:

(1) The basic units in Pitman shorthand should not be modified. Basic consonant strokes, vowel and diphthong symbols denote the basic property of Pitman shorthand.

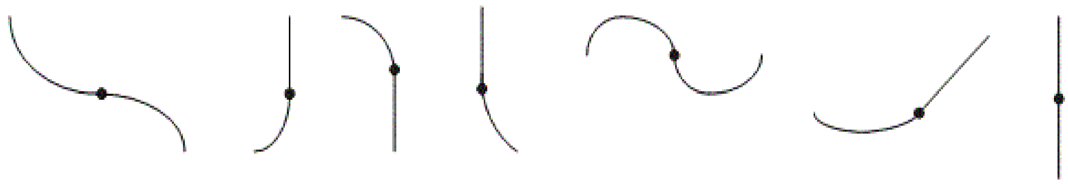
Changing any of them would be equal to inventing another shorthand system, which is counter to the original objective.

(2) The new rules should be as simple as possible so that the writer is able to learn and apply these rules easily and quickly. At the same time, the burden that the new rules put on the human side should be as light as possible.

(3) The new rules should not cause any new ambiguity. A new rule that disambiguates smooth junctions but introduces other confusions is not a good one.

In order to improve the machine compatibility of smooth junction, two strategies had been investigated in this research. The first strategy is to reshape a shorthand at smooth junction with special features, for example, a small tick (see Figure 4.19(b)) to produce a maximal curvature point intentionally, our experiments show this strategy doesn't work well because the newly introduced symbol is likely to be recognized as a hook due to their similar geometric features. The second strategy is to introduce a kind of special symbol around smooth junctions for recognition. In this case, once this special symbol appears at a place around a shorthand stroke, the point near the special symbol will be recognized as a smooth junction. Given the second strategy, either a dot or a tick should not be treated as a "smooth junction indicator" since they have been defined as vowel & diphthong symbols in Pitman notation. A circle was finally chosen due to its simplicity rather than angular triangle, rectangle or other shapes.

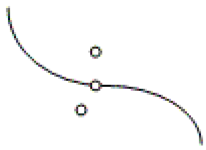
In the machine compatible shorthand, a circle is required to be written near the smooth junction in order to help the computer system detect smooth junctions. There is no specific requirement of the position and the size of the circle. It can be seen from Figure 4.19(c) that the circle can be written anywhere near the smooth junction. A point in the associated consonant outline with minimal Euclidean distance to the circle is chosen as a new dominant point for segmentation. Both the efficiency of the proposed rule and the burden it places on the human is evaluated later.



(a) Examples of smooth junction in Pitman shorthand

(The dot indicates the boundary point)

(b) Tick added at the place of smooth junction



(c) Three acceptable places for the circle to indicate the smooth join position

Figure 4.19: Smooth junctions in Pitman shorthand

4.2 Vowel and diphthong recognition

In Pitman shorthand, two vowel symbols (a dot and a dash) are used to represent six simple long vowel sounds and six short vowel sounds. They are placed alongside a consonant stroke to represent a vowel-consonant or consonant-vowel pairs as shown in Figure 4.20.

Figure 4.20: Relation of vowel and consonant stroke

The representations of four common diphthong symbols in Pitman shorthand are shown in Figure 4.21.

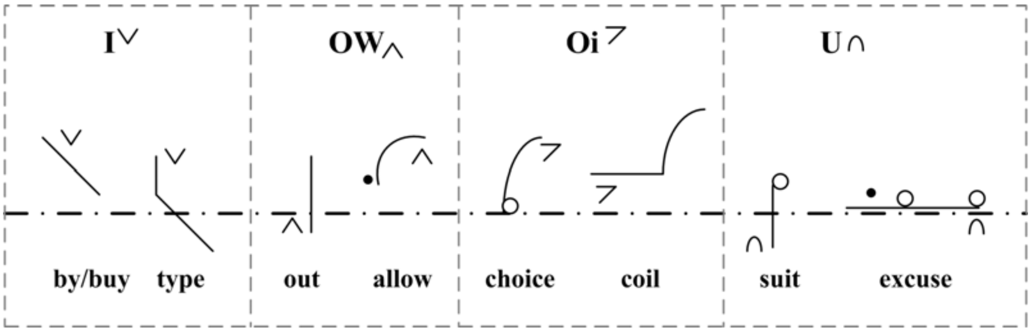
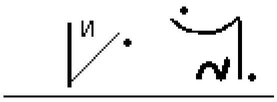


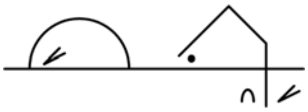
Figure 4.21: Diphthong symbols

Diphones and triphones are introduced in Pitman shorthand to represent two and three consecutive vowels respectively. As a special diphthong symbol with a small tick attaching at the end, the triphone (see Figure 4.22) is used to represent any vowel immediately following the diphthong (representing three vowels in one sign). The diphone (See Figure 4.23) is used to represent two vowels in words such as ‘layer’, ‘creation’ and ‘heaviest’ which occur consecutively and are pronounced separately.



diary annuity

Figure 4.22: Triphones



layer repudiate

Figure 4.23: Diphone

In summary, there are four different types of vowel and diphthong symbols in Pitman shorthand as shown in Figure 4.24.

(a) Vowels (b) Diphthongs (c) Diphone (d) Triphones

Figure 4.24: Vowel and diphthong symbols in Pitman shorthand

4.2.1 Classification of symbol type

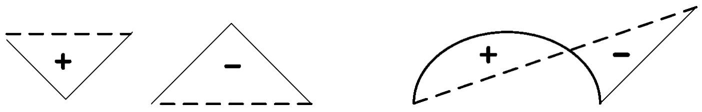
There are four steps involved in the classification procedure:

(1) Circles required by the proposed new rule in section 4.1.2 should be detected first. If the length from the start point to the end point is less than 1.5 mm, it is assumed that a circle has been found and a point in the consonant outline with nearest Euclidean distance to the circle is added as a new dominant point representing a smooth junction.

(2) If the whole outline could be contained in a 1.5mm square (a value representing the largest square which could contain the dot. If the value is too high, other small symbols like dash will be wrongly classified as dot. The lower the value, the more attention writers have to pay to write dots within the maximal range set by the value) and the time duration exceeds 100ms (a parameter to distinguish dot from a 'jitter' point in section 3.4.1), the outline is a dot.

(3) If the maximal perpendicular distance of all points to the line between the start point and the end point is smaller than a predefined threshold, the outline is a dash. In the system, the threshold is 0.5mm, which is equal to the allowed maximal perpendicular distance of a dash, is used in the system. The lower the value, the more attention writers have to pay in case that perpendicular distance of the dash is out of the maximal range set by the value. If the value is too high, other curves will be wrongly classified as a dash.

(4) The percentage of points with positive and negative distance to the line from the start point to the end point is used to distinguish triphones from other symbols. As shown in Figure 4.25, vertical distance of all points in diphthongs and the diphone have the same sign. But in triphones, they have different signs. For a triphone, the attached tick should be removed by the following algorithm before going to step 5.

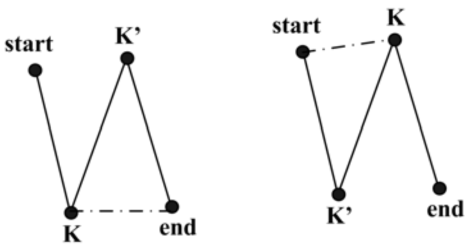


(a) Diphthong (b) Triphone

Figure 4.25: Sign of distance

(vertical line from dashed line to written line)

The following three steps were used to remove the attached tick from the triphone: Firstly, point K with maximal perpendicular distance to the line from the start point to the end point was found. Figure 4.26 shows two possible positions where point K might appear. Secondly, a dashed line from point K to either the start or the end point which has the longer distance to K was added to find point K'. Point K' is the point with the maximal perpendicular distance to that line. Thirdly, the start point, point K and point K' were combined to construct a new diphthong symbol without the attached small tick.



(a) (b)

Figure 4.26: Recognition of triphones

(5) Two parameters: the direction of two lines as shown in Figure 4.27 (line 1 is formed by the start point and point *P*; line 2 is formed by point *P* and the end point. *P* is the point with maximal perpendicular distance to the line from the start point to the end point), and the area between the two lines and corresponding chords (grey regions in Figure 4.27), were used to classify the four diphthongs and diphone.

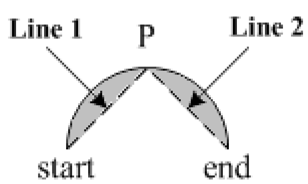


Figure 4.27: Classification of diphthongs

4.2.2 Place-sequence of vowels and diphone

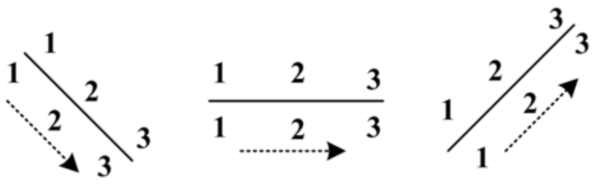
Vowels and diphone can be placed in three places relative to a consonant stroke, namely, at the beginning, the middle, and the end, which are accordingly called first-place, second-place, and third-place respectively as shown in Figure 4.28(a).

Vowels and diphones at different places represent different vowel-sounds in English language according to the rules in Table 4.4.

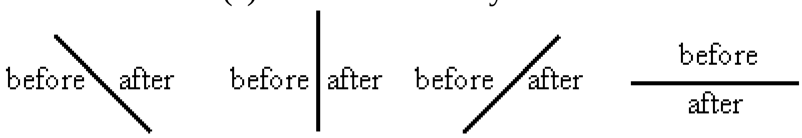
Table 4.4: Representation of vowels and diphone at different places

Representation		**1st place**	**2nd place**	**3rd place**
Vowel	**Dot**	*ah* (fade) or $\breve{a}$ (fed)	$\bar{a}$ (raid) or $\breve{e}$ (red)	$\bar{e}$ (eel) or $\breve{i}$ (ill)
	Dash	*aw* (pawed) or $\breve{o}$ (pod)	$\bar{o}$ (rote) or $\breve{u}$ (rut)	$\overline{oo}$ (pool) or $\breve{oo}$ (pull)
Diphone		*ah* or $\breve{a}$ and any vowel immediately following (sahib)	$\bar{a}$ or $\breve{e}$ and any vowel immediately following (layer, laity, surveyor)	$\bar{e}$ or $\breve{i}$ and any vowel immediately following (real, amiable)
Diphone		*aw* and any vowel immediately following (flawy, drawer)	$\bar{o}$ and any vowel immediately following (showy, poet, heroic)	$\overline{oo}$ and any vowel immediately following (shoeing, bruin)

The sequence information (see Figure 4.28(b)) represents the order of reading vowels with their strokes: with downstrokes and upstrokes, from left to right and for horizontal strokes, from top to bottom.



(a) Places of vowel symbols



(b) Sequence of vowel symbols

Figure 4.28: Place-sequence of vowel symbols

It should be noted that the place-sequence of vowels and diphone obtained from the recognition system is unreliable because stenographers write vowels and diphones inaccurately and often omit them for the sake of speed. So place-sequence can be used as references in the transcription system.

(a) Place information

Firstly, dominant point ***K*** in the consonant outline (see Figure. 4.29) with minimal Euclidean distance to the symbol is found. Secondly, point ***H*** lying between dominant point ***K-1*** and dominant point ***K+1*** and with minimal perpendicular distance to the symbol is found. Finally, the positional relationship between point ***K*** and point ***H*** is used to determine the place. If point ***H*** lies between ***K*** and ***K+1***, the symbol may locate at either place 1 or place 2. If the ratio of the length of chord ***KH*** over the length of chord ***K,K+1*** is larger than 35% (A parameter determining the final place of the symbol. The larger the parameter, the possibility for place 2 is smaller and the possibility for place 1 or place 3 is larger. This parameter can be chosen freely to adapt different writing styles. Generally, it should not exceed 50%), the place is 2; otherwise, it is 1. If point ***H*** lies between ***K-1*** and ***K***, the symbol may locate at either place 3 or place 2. If the ratio of the length of chord ***HK*** over the length of chord ***K-1,K*** is larger than 35%, the place is 2; otherwise, it is 3.

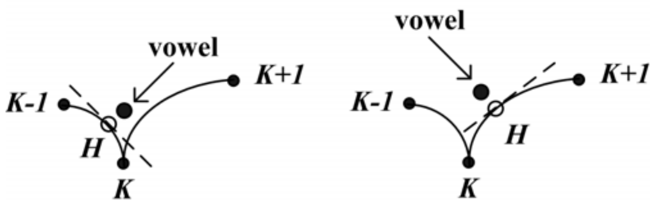


Figure 4.29: Place-sequence detection

(b) Sequence information

After point ***H*** is found in section 4.2.2(a), a reference line (dashed line in Figure 4.29) which is the tangent to the chord that point ***H*** lies in, is added at ***H***. Two parameters are used to determine the sequence. The first refers to the sign of the perpendicular distance from the vowel symbol to the line. The second refers to the eight-direction chain code value of the line. Detailed rules are listed in Table 4.5.

Table 4.5: Sequence table

Sign of distance / Chain-code of the line	**Positive(+)**	**Negative(-)**
0~3	Before	After
4~7	After	Before

For a vowel lying between two pen-strokes, both possibilities are listed. As shown in Figure 4.30, with reference to stroke 1, the vowel lies in the 3rd place and read after stroke 1; with reference to stroke 2, it lies at the 1st place and read before stroke 2.

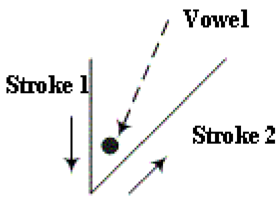


Figure 4.30: Ambiguity of vowel position

4.3 Evaluation

4.3.1 PSIS System

Shorthand outlines were written on a 12″ x 12″ WACOM ARTZII digitizing tablet, which is a piece of pressure sensitive hardware used to capture the handwriting. The tablet comes with an Ultrapen that is used for writing on the tablet to produce the image on the screen. A software system named ***Pen Stroke Information System*** (PSIS, Lim, 2000) was used to capture input sequences as well as display the written outlines. The tablet has an active area of 12 inches x 12 inches, which gives a coordinate range of 30480 x 30480. So one unit of coordinate on the tablet is 0.001 cm, which is named as ***Tunit.*** The sampling rate is 200 points per second.

The PSIS system captures dynamic information of the pentip such as its position, velocity, acceleration and pen pressure as a function of time. In the system, only the sequence of sampled (x,y) coordinates of the pen-tip movement was used. Another important part in data acquisition is to make an appropriate rule which enables the system to identify different pen-strokes according to both Euclidean distance and time difference. In the system, the rule of pen-stroke extraction is:

Rule 1: If two consecutive sample points have a Euclidean distance greater than 0.5mm and a time difference greater than 100 ms, they are considered to belong to different pen-strokes. This rule is used to detect pen-down and pen-up actions. The value of two parameters is equal to the value of the parameters used in the second step of preprocessing to remove accidental pen-lift (section 3.4.1).

Rule 2: For two continuous pen-strokes, if the minimal Euclidean distance between pen-stroke *n* and its preceding pen-stroke *n-1* is smaller than 1cm, they are considered to belong to the same shorthand outline, vice versa.

To acquire the shorthand outlines for the evaluation, a data entry form was required. The design of the form took into consideration the two following aspects: the first aspect is the characteristic of the shorthand. As shorthand outlines can be written on, above or through the writing line, there should be enough space for the writer to write the outline. Therefore, a 3x3 cm^2 cell with a reference line in the middle was designed for the entry of each single outline. The second consideration is the separation of outlines by software. Two consecutive cells are required to have a spacing of 1cm. That is also the reason that the value of the parameter in Rule 2 is equal to 1cm. The layout of the data entry form used in the data collection is given in Figure 4.31.

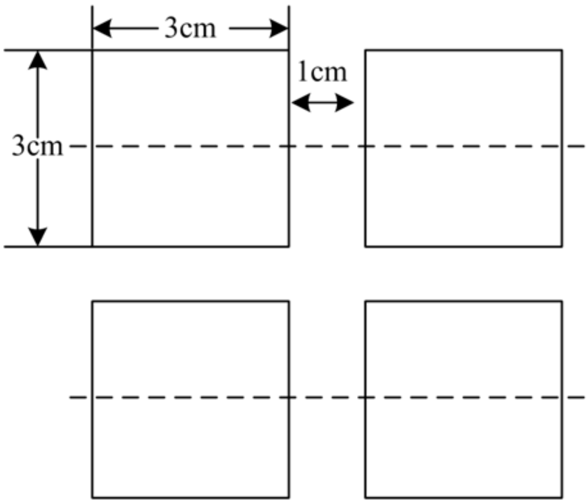


Figure 4.31: Design of data entry form

4.3.2 Data Collection

Two experiments were carried out to evaluate the approach to the recognition of vocalized outlines and the efficiency of the rules in defining machine compatible shorthand for the issue of smooth junction. Three shorthand writers were involved in the experiments.

There are two types of thresholds used in the algorithm. The first type is used to recognize consonant outlines. It includes: the angular change value Ψ to distinguish corner from smooth join, threshold for the loop/circle detection, the threshold to trigger hook detection algorithm, "minimal stroke length" (MinSL) and "maximal stroke length" (MaxSL) for the stroke reconstruction. Among the four thresholds, the last two are relatively crucial to the recognition accuracy. But they are not writer specific because each outline has been time & size normalized in the preprocessing. If the writers were able to adjust the thresholds according to their writing habits, the recognition accuracy could be improved. And the second type of thresholds is used to determine the place-sequence of vowel & diphthong symbols. It has little impact on the system performance as place-sequence is regarded as unreliable recognition result and is seldom used in the post-processing process.

In the first experiments, 50 vocalized outlines (see Appendix A) which include all features and possible combinations of Pitman shorthand primitives except for the smooth junctions were selected for the evaluation. Three shorthand writers were asked to write each sample eight times. 1127 outlines were collected in total. Table 4.6 shows the statistics of the samples. The average number of primitives per outline in the samples is 4.2, while it is 2.6 in commonly used Pitman shorthand (Leedham,1984), which means the outlines used here are more difficult cases than would typically be encountered. The recognition performance is expected to improve in the case of standard Pitman shorthand. As shown in Figure 4.32, the low angularity and high angularity boundaries are easily distinguished by human vision. The percentage of boundaries with low angularity is around $1229/3087 \approx 40\%$

Table 4.6: The statistics of the samples

	Number of primitives						Total
	2	**3**	**4**	**5**	**6**	**7**	
Number of outlines	58	397	514	116	23	19	1127
Number of circles/loops	25	301	323	70	44	0	763
Number of hooks	25	214	160	94	22	17	532
Number of low angularity	33	429	628	117	22	0	1229
Total number of boundaries	58	794	1542	464	115	114	3087

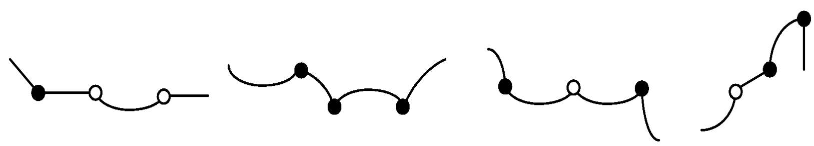


Figure 4.32: Different type of boundaries

(High angularity boundaries are indicated by dark circles; low angularity boundaries are indicated by white circles.)

The second experiment was to evaluate the efficiency of the rules proposed in section 4.1.3 to solve the problem of smooth junctions. 20 vocalized outlines (See Appendix A) each containing one smooth junction were selected as samples. Three writers were asked to write each outline eight times without and with the introduction of new rules respectively. Table 4.7 shows the statistics of the outlines.

Table 4.7: Statistics of the outlines

	Number of primitives					Total
	2	3	4	5	6	
Number of outlines without the new rule	65	155	179	25	40	**464**
Percentage	14%	33.41%	38.58%	5.39%	8.62%	-
Number of outlines with the new rule	63	157	170	24	47	**461**
Percentage	13.66%	34.06%	36.87%	5.21%	10.20%	-

4.3.3 Experimental Results

In the first experiment, the accuracy of the recognition of consonant outline and vowel & diphthong symbols were evaluated respectively. Table 4.8 shows the accuracy rate for the consonant outlines. Overall, a correct recognition rate of 75.33% was achieved. Obviously, the classification accuracy (99.18%) is much higher than that of segmentation (75.95%).

Table 4.8: The experimental results of the consonant outline recognition

	Number of primitives						Total	Average Rate
	2	3	4	5	6	7		
Number of outlines	58	397	514	116	23	19	1127	-
Number of outlines Without Segmentation errors	46	282	419	73	18	18	856	-
Rate of segmentation	79.31%	71.03%	81.52%	62.93%	78.26%	94.74%	-	75.95%
Number of outlines correctly classified *	46	281	415	71	18	18	849	-
Rate of classification	100%	99.65%	99.04%	97.26%	100%	100%	-	99.18%
Total accuracy	79.31%	70.78%	80.74%	61.21%	78.26%	94.74%	-	75.33%

* The outline is considered classified correctly if the correct path is found and the correct class is included in the ranked list interpretations. The order of the ranked list is not considered.

Table 4.9: The experimental results for segmentation

	Number of primitives						**Total**	**Average Rate**
	2	**3**	**4**	**5**	**6**	**7**		
Number of segments	58	794	1542	464	105	108	3071	-
Number of correctly segments	47	675	1432	433	105	108	2800	-
Rate of segments	81.03%	85.01%	92.87%	93.32%	100%	100%	-	91.18%
Number of correctly detected circles/loops	23	299	321	66	43	0	752	-
Rate of circles/loops	92%	99.34%	99.38%	94.29%	97.73%	--	-	98.56%
Number of correctly detected hooks	25	196	144	84	20	16	485	-
Rate of hooks	100%	91.59%	90%	89.36%	90.91%	94.12%	-	91.17%

Table 4.9 lists the recognition accuracy of different processes in the segmentation stage. The accuracy rate of dominant point detection, circle/loop detection and hook detection are 91.18%, 98.56% and 91.17% respectively. In the dominant point detection process, most errors were due to the missing of boundaries with low angularity. As shown in Figure 4.33(a), the rate of segments is inversely proportional to the percentage of low angularities. In Figure 4.33(b), the right segmentation point indicated by the circle is failure to be detected since it lies in the smooth region of the input outline. Figure 4.33(c) shows two typical cases where the basic consonant stroke ◡ is not correctly classified as primitive /N/ or /NG/, but as a hook.

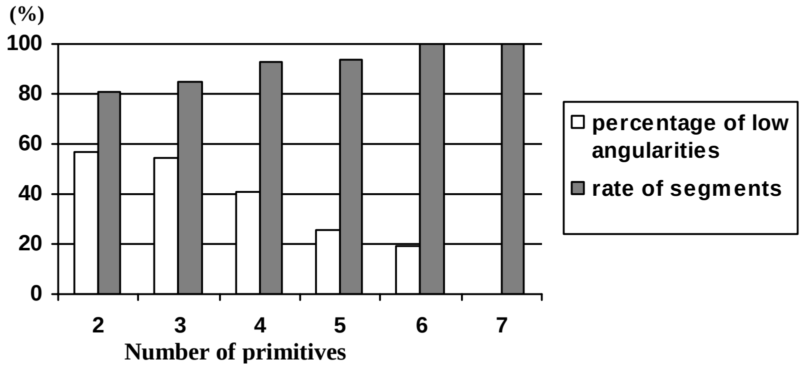
Table 4.10: The experimental result of stroke reconstruction

	Number of primitives						**Total**	**Average Rate**
	2	**3**	**4**	**5**	**6**	**7**		
Number of outlines without Segmentation errors	46	286	423	78	18	18	869	-
Number of outlines correctly reconstructed	46	282	419	73	18	18	856	-
Accuracy	100%	98.60%	99.05%	100%	100%	100%	-	98.50%

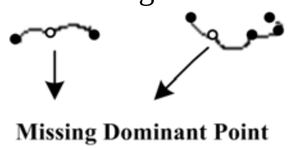
Table 4.11: Accuracy of stroke neural network classifier

	Number of primitives						**Total**	**Accuracy Rate**
	2	**3**	**4**	**5**	**6**	**7**		
Number of primitives for classification	92	846	1676	365	108	126	3213	-
Number of primitives in the first place of rank list	86	775	1516	316	95	117	2905	-
Number of primitives in the top three places of rank list	92	841	1662	361	108	125	3189	-
Percentage of primitives in the first place of rank list	93.48%	91.61%	90.45%	86.58%	87.96%	92.86%	-	90.41%
Percentage of primitive in the top three of rank list	100%	99.41%	99.16%	98.90%	100%	99.21%	-	99.25%

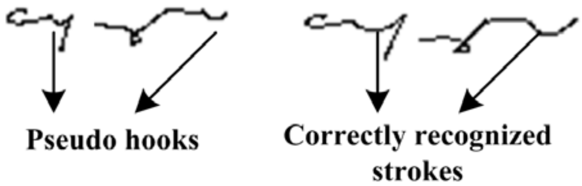
Table 4.10 and Table 4.11 show the experimental results in the classification stage. The accuracy of stroke reconstruction was 98.50%. According to our analysis, errors made in the stroke reconstruction were mainly due to a non-stroke pattern wrongly classified as a stroke pattern (see Figure 4.34). The accuracy of neural network classification was 90.41% if only primitives in the first place of rank list are considered and it increased to 99.25% if primitives in the top three places of rank list were considered. Figure 4.35 lists five groups of primitives which are easily confused with each other.



(a) Relationship of the low angularities and segmentation rate



(b) Typical error in segments



(c) Hook classification error

Figure 4.33: Typical errors in the segmentation stage

In the two-stage approach to the recognition of a consonant outline, segmentation and classification are cascaded and errors made in the preceding stage (segmentation) cannot be recovered in the succeeding stage (classification). A parallel system in which different stages work independently or incorporating high-level linguistic knowledge to correct mistakes in the segmentation is expected to improve the performance significantly.

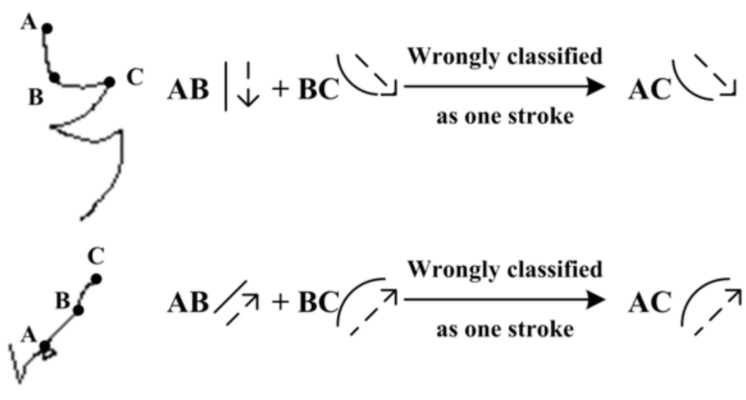


Figure 4.34: Examples of typical mistakes made in stroke reconstruction

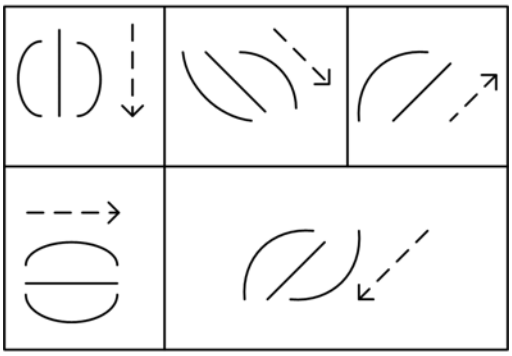


Figure 4.35: Five groups of primitives with similar patterns

It is a common practice for stenographers to omit vowel and diphthong symbols for the sake of speed. Table 4.12 compared the number of vowel & diphthong symbols required by the Pitman notation and the number of vowel & diphthong symbols actually written. The classification rate is 96.86%. Most of the classification errors come from the ambiguity between the two diphthong symbols ∧ and ∩.

Table 4.12: The experimental results for the vowel and diphthong symbols

	Number of vowels and diphthongs			Correctly recognized	Recognition rate
	Number1*	**Number2****	**Percentage**		
Writer1	913	632	69.22%	620	98.10%
Writer2	1088	532	48.90%	521	97.93%
Writer3	1155	941	81.47%	898	95.43%
Total	3156	2105	66.70%	2039	96.86%

* The number of vowel and diphthong symbols included in the samples

** The number of symbols actually written

In the experiment, many vowels were not written at the places required by the Pitman notation. In order to evaluate their place-sequence, it was necessary to manually check the place-sequence for each vowel symbol in the collected samples. The classification was considered to be correct if the output of the recognition was the same as the checked result even if the place-sequence of the vowel symbol does not follow the writing rules of the Pitman notation. Detailed evaluation results are showed in Table 4.13. The overall place-sequence rate is 81.05%. The place rate is 90.36% and it is higher than the sequence rate of 84.04%. Mistakes occurred when vowels were written in the middle of two consecutive places (1^{st} place and 2^{nd} place, 2^{nd} place and 3^{rd} place).

Table 4.13: The experimental results of the place-sequence of the vowels

	Number of vowels	**Number of vowels with correctly classified place**	**Place rate**	**Number of vowels with correctly classified sequence**	**Sequence rate**	**Number of vowels with correctly classified place-sequence**	**Place-sequence rate**
Writer1	632	568	89.87%	518	81.96%	500	79.11%
Writer2	532	468	88.00%	452	84.96%	431	81.01%
Writer3	941	866	92.03%	799	84.91%	775	82.36%
Average	2105	1902	90.36%	1769	84.04%	1706	81.05%

The results of the second set of experiments to evaluate the efficiency of the rule to solve the issue of smooth junctions, are shown in Tables 4.14, 4.15 and 4.16. With the proposed new rule, the overall recognition rate was improved by 55.88% (from 37.53% to 93.41%) at the cost of 15.29% increase in time. (The average writing time the writer used to write each vocalized outline.) We can see in Figure 4.36 that the increased time is inversely proportional to the number of primitives the outline contains. In other words, the new rule affected the writing time of simple outlines more than that of complicate outlines. Figure 4.37 compared the relationship between the increased recognition rate and writing time. It is obvious that the proposed rule improves the recognition rate significantly while placing relatively little burden on the writer.

Table 4.14: Comparison of recognition rate with and without the proposed rule

		Number of primitives										Average	
		2		3		4		5		6			
		WT	W	WT	W	WT	W	WT	W	WT	W	WT	W
Recognition Rate	Writer1	53.33%	100%	28.30%	88.46%	32.14%	85.71%	25%	87.50%	53.33%	93.75%	34.69%	89.12%
	Writer2	66.67%	100%	37.50%	95.83%	28.07%	90.74%	12.50%	85.71%	56.25%	100%	39.22%	94.56%
	Writer3	45.83%	100%	33.93%	96.36%	36.84%	94.83%	25%	85.71%	56.25%	100%	38.51%	96.27%
	Total	55.56%	100%	33.12%	93.55%	32.35%	90.47%	20.83%	86.36%	55.32%	97.92%	37.53%	93.41%

Table 4.15: Comparison of writing time with and without the proposed rule

		Number of primitives										Average	
		2		3		4		5		6			
		WT	W	WT	W	WT	W	WT	W	WT	W	WT	W
Writing time(ms)	Writer1	492	606	512	593	558	642	578	657	621	664	552	632
	Writer2	513	629	520	624	549	614	571	639	613	683	553	638
	Writer3	450	534	458	547	513	601	556	608	600	674	516	593
	Average	485	590	497	588	375	619	568	635	611	674	540	621

WT The experiment without the proposed rule; **W** The experiment with the proposed rule

Table 4.16: Percentage increase in time when adding the proposed segmentation indicator

		Number of primitives					Total
		2	3	4	5	6	
Percentage of increased time	Writer1	23.17%	15.82%	15.05%	13.67%	6.92%	14.93%
	Writer2	22.61%	20%	11.84%	11.91%	11.42%	15.56%
	Writer3	18.67%	19.43%	17.15%	9.35%	12.33%	15.39%
	Average	21.48%	18.42%	14.68%	11.64%	10.22%	15.29%

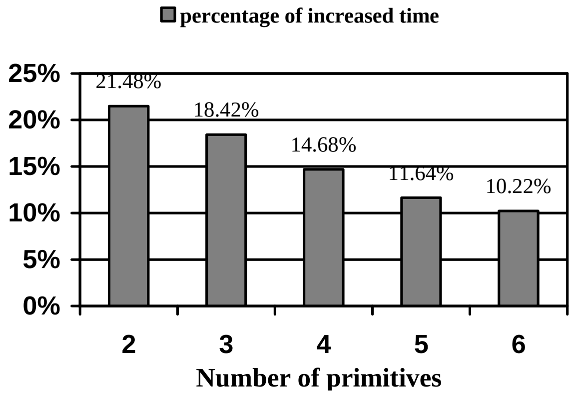


Figure 4.36: Number of primitives vs. percentage of increased time

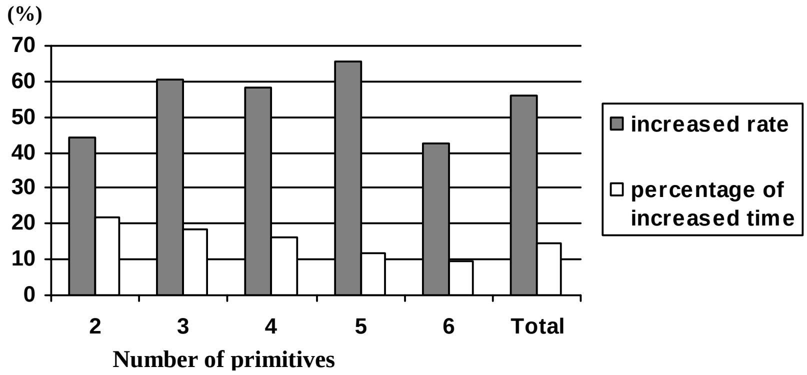


Figure 4.37: Comparison of increased recognition rate and writing time

4.4 Conclusion

In this chapter, the recognition of Pitman vocalized outlines is fully discussed. A new solution to the segmentation and classification of consonant outlines, which is the most complicated part in Pitman shorthand, is proposed. We view Pitman shorthand as a curve rather than an online handwritten character and introduce approaches in shape recognition such as dominant point detection by region of support, and polygonal approximation parameters for its recognition. Approaches to the classification of vowel & diphthong symbols and place-sequence detection of vowel symbols are also discussed. In order to solve the problem of smooth junctions, a new rule defining machine compatible shorthand is proposed to improve the 'machinography' of Pitman notation. Evaluation on 1127 consonant outlines written by three shorthand writers on a Wacom digitizer showed that the proposed two-stage recognition approach achieved a correct recognition rate of 75.33%. Another experiment on 461 consonant outlines with one smooth junction each showed that, with the new proposed rule, recognition rate was improved by 55.88% (from 37.53% to 93.41%) at the cost of 15.29% increase in writing time.

CHAPTER 5

Classification of shortforms in Pitman shorthand

The total context independent Pitman shortform dictionary comprises a set of 94 commonly occurring words and phrases (see Appendix B). The unique determination of each shortform depends on the position of the outline with respect to the writing line, and the pressure or thickness of the outline. If the thickness of the pen-stroke and the position to the reference line are not considered, only 55 different shortform shapes remain. Shortforms accord with words or phrases directly which simplifies the usual translation from syllable to character in the transcription system. The output of the classification is the exact transcription if the shortforms can be uniquely distinguished by considering its outline shape. If the shortforms are dependent on their position relative to the reference line or writing pressure, its possible transcription is given. Before the classification of shortforms, noise in the outline has been reduced by applying the preprocessing steps 2-8 described in section 3.4.1.

5.1 Templates Construction

Shortforms are classified using a template-based approach. Users can personalize their own shortforms by adding or deleting templates in the database. In order to create a new template, a selection group of at least three samples is required. Among the three samples, the one with maximal Fisher Criteria should be selected as a template. The Fisher Criteria is calculated by $F = S_B / S_W$, where S_B is the between class distance, and S_w is the within class distance. S_B is set to 1 if the template database is empty or does not include different types of templates. S_w is the sum of distances between the input outline and the other outlines in the selection group. The maximal distance between the selected outline and the other outlines in the selection group is saved as radius R_r of the template, which is used to evaluate the reliability of the classification. Templates created by the above procedure are of both large between-class distances and small within-class distances. This means they can improve the accuracy of classification.

5.2 Outline Classification

5.2.1 Pre-classification

The pre-classification filtering to remove useless templates involves three steps:

(1) If the input outline could be contained within a 1.5mm square, it is a dot.

(2) If the length from the start point to the end point is less than 1.5mm, the outline is a circle.

The above two parameters are obtained by trial-and-error from experimentation. The lower the value, the more attention the writers have to pay in case that dot/circle extends outside the maximal range set by the threshold value. If the value of the parameter is too high, other symbols are likely to be classified as a dot/circle.

(3) The ratio of the number of points in the input outline over the number of points in the template should not be larger than 2 and not less than ½ (2 and ½ are used since each point is allowed to be compared twice at most).

5.2.2 Dissimilarity measure using polar coordinates

For any two points m_i and n_j in the input outline and template, their dissimilarity is calculated on the basis of a polar coordinate system as shown in Figure 5.1. The start point of the pen-stroke is defined as an origin. Each point in the outline is defined by its distance and its angle referred to the origin.

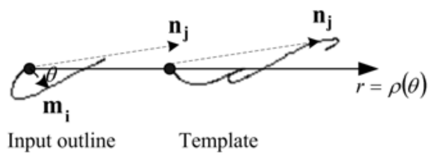


Figure 5.1: Polar coordinates

$d(m_i, n_j) = d_l(m_i, n_j) + d_\theta(m_i, n_j)$, where $d_l(m_i, n_j) = \left| r_{m_i} - r_{n_j} \right|$ (r_{m_i} and r_{n_j} are distances from the origin to point m_i and point n_j respectively) and $d_\theta(m_i, n_j)$ is the angular distance which is calculated by the approach proposed by Chen et al.(2002).

$$d_\theta(m_i, n_j) = \min(r_{m_i}, r_{n_j}) \times \sin(\theta(m_i, n_j))$$

$\theta(m_i, n_j)$ denotes the smallest positive intersecting angle between m_i and n_j as shown in Figure 5.2.

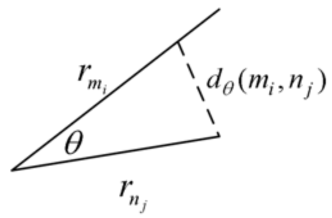


Figure 5.2: Illustration of angular distance d_θ

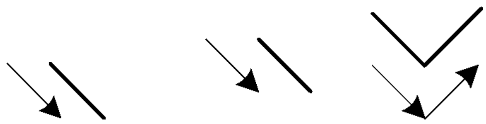
5.2.3 Hausdorff distance

In the area of shape matching, the Hausdorff Distance has been used frequently to evaluate the degree of mismatch between the input outline and templates (Gastaldo & Zunino, 2002; Huttenlocher et al., 1993). Here, a generalization of the Hausdorff Distance

(Dubuisson & Jain, 1994) which is insensitive to the single 'outlying' point of *M* or *N*, is used to measure the dissimilarity between the input outline and the template:

$$h(M,N) = \sum_{m_i \in M} \min_{n_j \in N} \left\| m_i - n_j \right\| / p$$

where *M* is the input outline, *N* is the template, p is the number of points in *M*.



(a) Input outline *M* (b) Templates N_1 and N_2

Figure 5.3: Shorforms

As shown in Figure 5.3, for the same outline *M*, its Hausdorff distance to template N_1 and N_2 are equal to zero. To avoid such ambiguity, the Hausdorff distance $h(N,M)$ is added to calculate the dissimilarity from the template to the input outline. The finalized Hausdorff distance includes two parts:

$$\begin{aligned} H &= h\left(M,N\right) + h\left(N,M\right) \\ &= \frac{1}{P_m} \sum_{m_i \in M} \min_{n_j \in N, \left|m_i - n_j\right| < \delta_1} \left| d\left(m_i, n_j\right) \right| + \frac{1}{P_n} \sum_{n_j \in N} \min_{m_i \in M, \left|n_j - m_i\right| < \delta_1} \left| d\left(m_i, n_j\right) \right| \end{aligned}$$

For point m_i in the input outline, point n_j in the template which satisfies $\left|m_i - n_j\right| < \delta_1$ could be selected for detailed comparison. Parameters δ_1 and δ_2 define the permissible tolerance for local warping. Large δ_1 and δ_2 may lead to possible redundancy; while small δ_1 and δ_2 may impose constraints on the writing. In the system, $\delta_1 = M/4$ and $\delta_2 = N/4$ were used. In calculating $h(M,N)$, each point m_i in the outline M can be selected twice maximally. And each point n_j in the outline N can be selected twice maximally in calculating $h(N,M)$.

The dissimilarity between the input outline and the template is directly proportional to their Hausdorff distance. The output of the classification process is a ranked list of templates whose radius R_r is larger than the dissimilarity between the template and the input outline. Preserving the ranked list is helpful to overcome the problem of redundancy since it is difficult for the system to distinguish outlines with similar geometric features especially when writers do not write them discriminatively. If there are no templates whose radius is larger than the dissimilarity, the template with minimal distance to the input outline is chosen as the final result. However, such a result is considered unreliable.

5.3 Evaluation results

Each writer was asked to provide five sets of data files, namely A_1 - A_5, B_1 - B_5, C_1 - C_5 respectively. Each data file included three samples of each shortform. During the collection of these shortforms, the writers were instructed to ensure that shortforms with similar shapes should be distinguishable by human vision. Writer A and writer B also wrote each outline three times for the construction of two sets of templates A_T and B_T respectively. During the collection of templates, the writers were required to pay more attention to the writing rules, which made the collected templates similar to standard shortforms. Appendix C shows templates A_T & B_T and samples written by different writers.

A series of evaluations were carried out to compare the recognition accuracy and processing time of the proposed approach and dynamic programming (Note: Nair & Leedham (1992) reported accuracy rate from 93% to 100% in the classification of Pitman shortform. However, shortforms used in this experimentation are different from samples used in the previous work. Pitman 2000 shortforms have little redundancy which degrades the performance). The first experiment was conducted to evaluate the dependency of each algorithm on the reference template. Templates A_T and B_T were used as reference templates separately with A_1 - A_5 used as input data files. As shown in Table 5.1, the classification rate of the proposed approach is 91.76%, which is 91.76%-

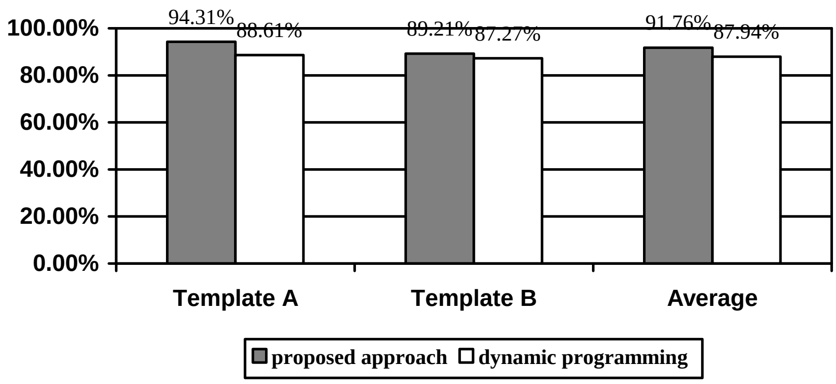
87.94%=3.82% higher than that of dynamic programming. For both templates A_T and B_T, the proposed approach achieved higher accuracy rate than dynamic programming (see Figure 5.4(a)). However, the proposed approach relies more on the choice of reference template from the result shown in Figure 6.4(b) & (c). Its classification rate increased 5.1% (from 89.21% to 94.31%) while dynamic programming only increased 1.34% (from 87.27% to 88.61%) if the template was selected from the outlines written by the same writer. That means the proposed approach is a writer-dependent method.

Table 5.1: Results of the first experimentation

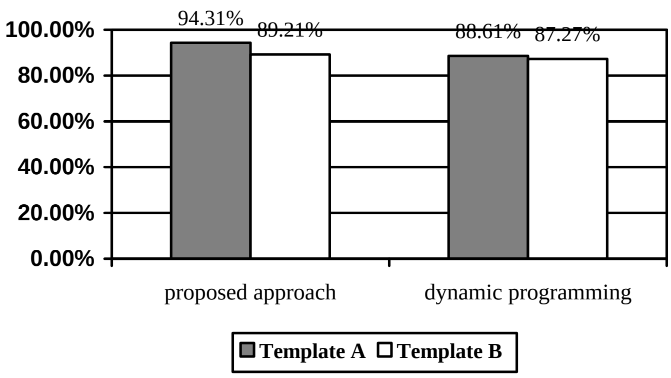
		A_1	A_2	A_3	A_4	A_5	Average
Proposed approach	**Template A_T**	95.76%	93.94%	94.55%	94.55%	92.73%	94.31%
	Template B_T	90.30%	89.70%	88.48%	87.87%	89.70%	89.21%
	Average	93.03%	91.82%	91.52%	91.21%	91.22%	91.76%
Dynamic programming	**Template A_T**	89.70%	90.30%	87.27%	89.09%	86.67%	88.61%
	Template B_T	88.48%	87.87%	86.67%	87.27%	86.06%	87.27%
	Average	89.09%	89.09%	86.97%	88.18%	86.37%	87.94%

The second experiment was conducted on all data files with reference template A_T and B_T respectively. From the result shown in Table 5.2, the proposed approach achieved higher recognition accuracy (91.62% vs 86.97%) with more computational effort (24.40 seconds vs 15.76 seconds). (The processing time is the total time of matching all input outlines with templates divided by the number of the input outlines. In order to calculate the processing time, a function named "clock" is applied to tell how much processor time used in the recognition process. And the processing time in seconds is approximated by dividing the clock return value by the value of a constant **CLOCKS_PER_SEC** defined in Microsoft C.) The computational complexity of the proposed approach is $O[(m^2+n^2)/2]$ and the computational complexity of the dynamic programming is $O(mn)$. where m is the number of points in input pattern M and n is the number of points in the template N. It is obvious that $O[(m^2+n^2)/2] \geq O(mn)$ so the proposed approach needs longer computing time.

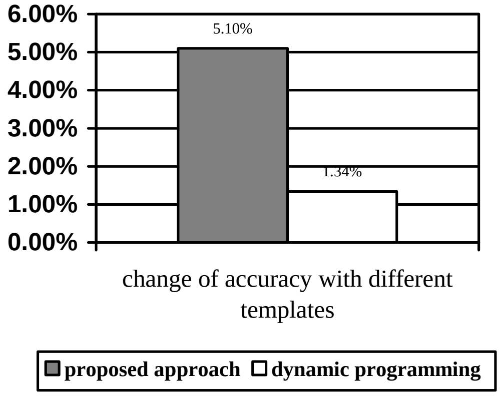
The comparison of recognition rate and processing time of both approaches using different templates is shown in Figure 5.5.



(a) Different classification approaches, same template



(b) Different templates, same classification approaches

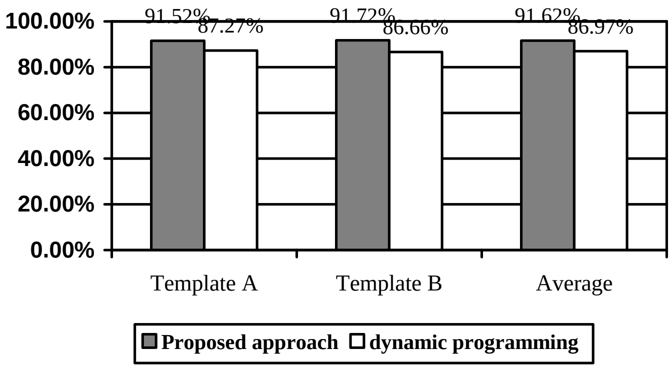


(c) Change of recognition rate using different templates

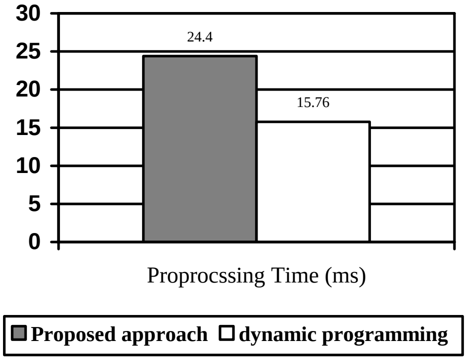
Figure 5.4: Comparison of recognition rate

Table 5.2: Results of the second experiment

		Accuracy Rate			
		Template A	**Template B**	**Average**	**Average processing time (secs.)**
Proposed approach	**Writer A**	94.55%	89.09%	91.82%	23.34
	Writer B	90.30%	93.94%	92.12%	24.87
	Writer C	89.70%	92.12%	90.91%	24.98
	Average	**91.52%**	**91.72%**	**91.62%**	**24.40**
Dynamic programmin g	**Writer A**	88.48%	87.87%	88.18%	15.34
	Writer B	89.09%	86.67%	87.88%	16.07
	Writer C	84.24%	85.45%	84.85%	15.87
	Average	**87.27%**	**86.66%**	**86.97%**	**15.76**



(a) Comparison of recognition rate



(b) Comparison of processing time

Figure 5.5: Result comparison of different approaches

Dynamic programming attains the maximum coincidence of two patterns by warping or stretching one pattern onto the other. Its performance degraded significantly in the situation of high redundancy especially when two outlines of different sizes have the same geometric features such as and , and . As a point-to-point comparison approach, the proposed new approach can distinguish redundancy patterns more effectively. Errors usually occur when two outlines have both similar sizes and similar shape features such as and , and , and . Table 5.3 lists redundancies in Pitman shortform.

Table 5.3: Redundancy in Pitman shortforms

No.	Word	Shortform	Word	Shortform
1	Dear		Trade/toward	
2	Accord/according(to)		With	
3	Can not		Would	
4	Have		For	
5	Therefore		their/there	
6	are/which, hour/our		and/ought, should/who	
7	put/to be, be		all/of, too/to	
8	From/very		Particular	
9	Influence		Information	
10	Influence		Any/in, thing	
11	Information		Any/in, thing	
12	Information		Any/in, thing	
13	Thank/think		that/without	
14	Wonderful/ly		Anyone	
15	Shall		Will	
16	How		You	
17	We		gentleman/gentlemen	
18	Therefore		from/very	
19	Their/there		Particular	

In the third experiment, data files A_1 - A_3 were used to generate a new reference template T with data files A_4 - A_5, B_1 - B_5, C_1 - C_5 used as input. The purpose of this experiment was to evaluate the impact of different categories of templates on the system performance. Unlike standard templates A_T and B_T, template T was obtained from normal handwriting without any constraints. Compared with the classification rate obtained in the second experiment using template A_T and B_T, the accuracy of the proposed approach degraded 86.97%-85.76%=1.21% compared to dynamic programming 91.62%-89.60%=2.02% using non-standard template T. The generalization of the proposed approach is not as good as dynamic programming as its accuracy degraded a little bit more than that of dynamic programming when non-standard templates were used although the degraded accuracy is not statistically significant compared with the overall accuracy around 80% -90%.

Table 5.4: Results of the third experiment

Accuracy	Proposed approach		Dynamic programming	
	Template T	Processing time (secs.)	Template T	Processing time (secs.)
Writer A	91.22%	24.01	86.97%	16.04
Writer B	89.09%	24.15	85.46%	17.10
Writer C	88.48%	23.98	84.85%	15.97
Average	**89.60%**	**24.05**	**85.76%**	**16.37**

5.4 Conclusion

In this chapter, a template-based approach to classifying Pitman shortform is proposed. Hausdorff Distance is introduced to measure the dissimilarity between the input outline and the reference templates. Three experiments were carried out to evaluate the proposed approach from both recognition accuracy and computational cost. Standard and non-standard templates were used to evaluate the performance of the classification system respectively. The experiments show that the proposed approach achieved higher classification accuracy for both types of templates. But it needs more computational time. The accuracies of both approaches become lower when non-standard templates were used. The generalization of the proposed approach needs to be improved. Its performance relies more on the choice of the template. The generalization ability of the proposed approach is hopefully to be improved by using multiple templates for a same shortform.

CHAPTER

6

Recognition of Renqun shorthand

As a means of recording Mandarin phonetically, Renqun shorthand adopts simple straight lines, curved lines or geometric symbols to represent the initial consonant and rhyme parts of Chinese syllables. In Renqun shorthand, 45 basic symbols including 17 initial consonants and 28 rhymes produce all 297 pinyin (sound) notations. Like Pitman shorthand, those frequently used words or phrases are also represented by abbreviated symbols called shortforms in Renqun notation. These shortforms are composed of either the first syllable notation or the second one, and sometimes they also make use of the combination of initial consonant symbol of the first syllable and rhyme notation of the second one. In a typical text of Renqun shorthand, shortforms usually count for a high proportion, of up to 50%. These shortforms can simplify the automatic recognition by omitting the usual translation from syllable to Chinese character since these shortforms accord directly with a Chinese phrase. Therefore, it is vital in the research of Renqun shorthand recognition to define accurate shortform recognition techniques.

Before the recognition, handwritten input to the Renqun shorthand recognition system (RSRS) are also required to be processed by the pre-processing module of PSRS (section 3.4.1) in order to remove noise and normalize input data. Composed of a sequence of rhymes and consonants, Renqun handwritten shorthand is also recognized in two stages as shown in Figure 6.1. In the segmentation stage, a non-parametric approach is firstly used to select rough dominant points, which indicate potential junctions of stroke features. Afterwards, the positions of dominant points are adjusted by polygonal approximation after rhymes like short primitives, circles, loops and hooks have been detected. In the classification stage, dummy dominant points are initially suppressed and merged. A heuristic merge algorithm is used to find all possible combinations of segments (path) for the stroke reconstruction. Finally, a neural classifier is used to classify segmented consonant strokes.

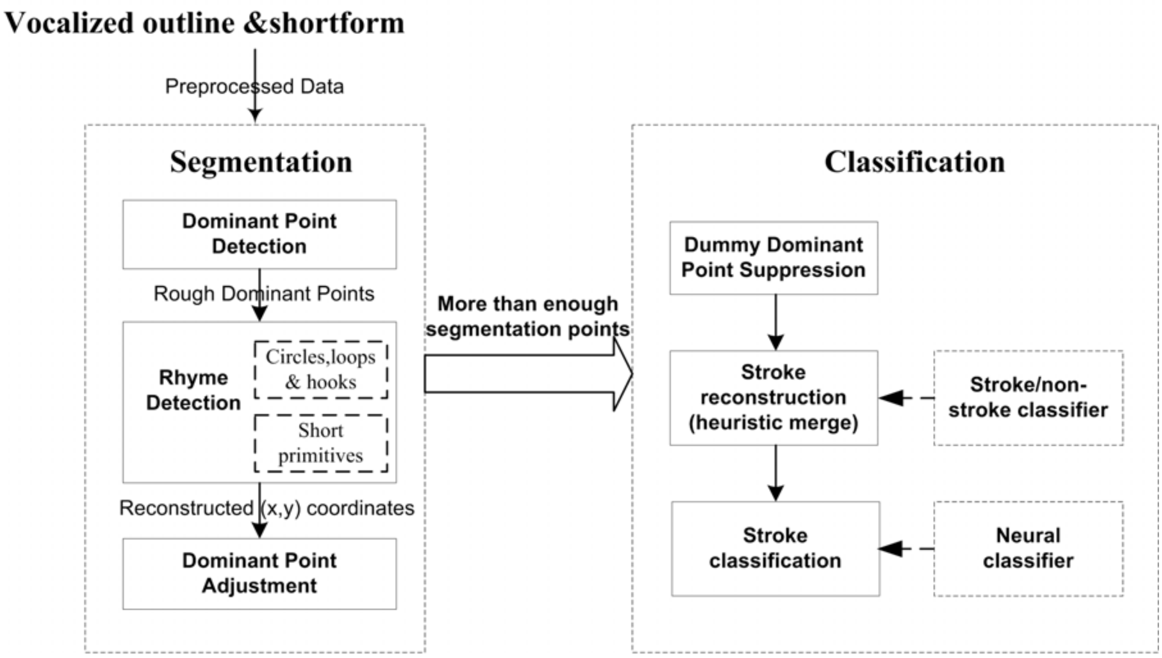


Figure 6.1: Diagram of the Renqun shorthand recognition system

The recognition procedures of Renqun shorthand are similar to the approach to the segmentation and classification of Pitman consonant outlines described in Chapter 4. In the recognition of Renqun shorthand, rhymes and consonants, which are fundamental components of both vocalized outline and shortform, are processed by the shape-based two-stage approach. However, the recognition procedures are different due to the special shape characters of Renqun shorthand. Table 6.1 compares the differences of the shape-based two-stage approach when it is applied to Pitman and Renqun respectively. The two-stage approach can be used to recognize vocalized outlines and shortforms in Renqun shorthand, but it is only applicable to Pitman consonant outlines. Most of processes in segmentation and classification are similar except a short primitive detection process is added to classify short primitives in Renqun shorthand and different parameters (MinSL and MaxSL) are used for the stroke reconstruction in the classification stage.

Table 6.1: The shape-based two-stage recognition approach in difference scenarios

<table>
<tr><th colspan="2">Two-stage recognition approach</th><th>Pitman</th><th>Renqun</th></tr>
<tr><td colspan="2">Applicability</td><td>Consonant outline</td><td>Vocalized outline,
Shortform</td></tr>
<tr><td colspan="2">Recognition elements</td><td>Consonants</td><td>Rhymes and consonants</td></tr>
<tr><td rowspan="4">Segmentation</td><td>Dominant point detection</td><td colspan="2">Same</td></tr>
<tr><td>Circles, loops & hooks detection</td><td colspan="2">Same</td></tr>
<tr><td>Short primitive detection</td><td>N/A</td><td>Yes</td></tr>
<tr><td>Adjustment by polygonal approximation</td><td colspan="2">Same</td></tr>
<tr><td rowspan="2">Classification</td><td>Parameters used in dummy dominant points suppression</td><td>MinSL=short stroke length
MaxSL=long stroke length</td><td>MinSL=halved stroke length
MaxSL=doubled stroke length</td></tr>
<tr><td>Neural network</td><td colspan="2">Neural network with the same configuration,
but trained with different training samples</td></tr>
</table>

Given the similar recognition process for Renqun shorthand and the consonant outline recognition for Prtman shorthand, only the key processes of RSRS are emphasized here in terms of rhyme detection and consonant classification.

6.1 Rhyme detection

As shown in Figure 6.2, there are two types of rhymes defined in Renqun shorthand: circles, loops and hooks with two different sizes (small or large) and orientations (clockwise or anti-clockwise) and short primitives attached at the beginning or the end of the consonants.

(a) Circles, loops & hooks

(b) Short primitives

Figure 6.2: Rhymes in Renqun shorthand

Circles, loops and hooks can be detected using the same approach described in section 4.1.1.2. In Renqun shorthand, only Type A, B and C circles/loops exist, which simplifies the procedures of circle/loop detection.

Recognition of short primitives consists of two steps: Firstly short primitives in an input outline are detected. Then dynamic programming is applied to calculate the distance between the detected short primitive and the templates. Afterwards approximate matching is introduced to rank all primitives having similar structures.

(1) Short primitive detection

According to the writing rules of Renqun shorthand, short primitives are either small tails or independent short strokes attached at the start or end of the outline. If the length between the start/end point and the nearest corner point Q is less than pre-defined threshold, it is assumed that a short primitive exists. If the dominant point is an intersection point, the short primitive is the tail of a circle or loop. Otherwise, it should be classified according to the templates shown in Figure 6.2(b).

(2) Dynamic programming

Due to its simplicity and good performance in matching patterns with different geometric features as stated in Chapter 5, dynamic programming (Leedham & Downton, 1987) was chosen to classify short primitive according to the templates shown in Figure

6.2(b). First, the input pattern $A = \{a_1, a_{2,} a_3 ... a_{i,} ...\}$ and template pattern $B = \{b_1, b_{2,} b_3 ... b_{j,} ...\}$ are represented by a string of I and J vectors with a_i and b_j as individual vectors in the input and reference patterns respectively. The optimum template match TM between two patterns is calculated by the following formula $TM = s(I, J)/(I + J - 1)$, where $s(I, J)$ is given by the recurrence relation

$$s(i, j) = f(i, j) + \min \begin{cases} s(i-1, j) \\ s(i-1, j-1) \\ s(i, j-1) \end{cases}$$

Where $f(i, j) = |a(i) - b(j)|$

The recurrence relation is calculated with the initial condition as follows:

$$s(1,1) = f(1,1) = |a(1) - b(1)|$$

$$s(1, j) = f(1, j) + s(1, j-1)$$

$$s(i,1) = f(i,1) + s(i-1,1)$$

$s(I, J)$ denotes the optimal path through the $f(i, j)$ matrix from $f(1,1)$ to $f(I, J)$. Its value measures the degree of dissimilarity between the input and the template. The smaller the value of $s(I, J)$ (and TM) the more similar are the two shapes for comparison. If $s(I, J) = 0$, the input pattern and the template are identical in both shape and size. Distances between the short primitive and each template are calculated in dynamic programming and used as a similarity measure to find the nearest neighbors of the short primitive.

(3) Approximate matching

This is true that handwritten outlines differ between writers, even for the same writer over a long period of time. One problem in the recognition system is that short primitives are frequently written in a peculiar and unclear manner. Therefore, approximate pattern matching is used to cope with uncertain short primitives. Templates with similar angular structures are defined as its nearest neighbor and included in the final result set ranked by the distances calculated in dynamic programming. As an example, the template has the minimum distance to a short primitive, similar patterns and are included in its nearest neighbors. Three neighborhoods as shown in Figure 6.3 are defined in our system.

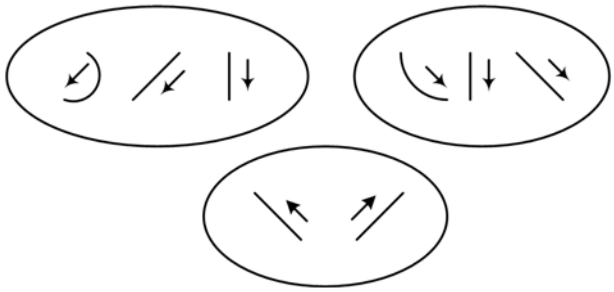


Figure 6.3: Three neighborhoods used in approximate matching

6.2 Consonant classification

The angle between the primitive outline and the horizontal line in Renqun shorthand is between 15^o and 25^o (Liao, 1985), different from the angle of 30^o to 60^o in Pitman shorthand (Pitman, 1989). Therefore, Renqun shorthand is more likely to have boundaries of low angularity, which significantly deteriorates the overall system performance. In our system, it is required that the primitives of Renqun shorthand should be written with a slant angle of around 45^o. From the comparison in Figure 6.4, the outline written with a slant angle of 45^o produced fewer boundaries with low angularity than it is written in a slant angle of 15^o or 25^o. As a matter of fact, many Renqun shorthand writers have already done so to improve the 'readability' of the handwritten shorthands. The newly defined writing rule improves the 'machinography' of Renqun shorthand as well as the compatibility of the two shorthand systems. It also enables the basic consonant strokes in the two shorthand systems to be classified by the neural network with similar configuration.

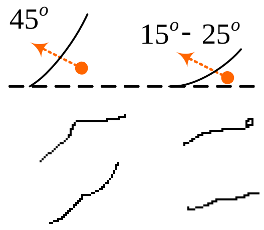


Figure 6.4: Comparison of Renqun outlines written with different slanting angles

Classification of Renqun consonants is similar to the classification of Pitman consonants described in section 4.1.2. It involves three processes: dummy dominant point suppression, stroke reconstruction and consonant classification. Two neural networks are constructed to perform stroke/non-stroke classification and individual stroke classification respectively. The nine input features of the neural network are extracted as described in section 4.1.2.1(1). The configuration and training process of the neural networks are as follows:

(1) Stroke/non-stroke neural network classifier

459 positive samples and 410 negative samples were collected to train a 9-25-2 back-propagation network. Positive and negative samples were selected manually from the Renqun shorthand outline written specially to train the neural network. For example, in the outline of the phrase ‘zhanshi (戰士)’ shown in Figure 6.5, strokes ***ab, bc*** and ***cd*** are marked as positive samples and strokes ***abc*** and ***bcd*** are marked as negative samples.

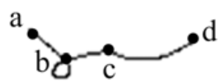


Figure 6.5: Positive and negative training samples

5-fold cross-validation was used to train the neural network. The correct classification rate is shown in Table 6.2.

Table 6.2: Accuracy of stroke/non-stroke neural network classifier

Average accuracy	Training	Testing
Stroke classification	99.18%	96.74%
Non-stroke classification	98.48%	96.34%
Average	98.83%	96.54%

(2) Individual stroke neural network classifier

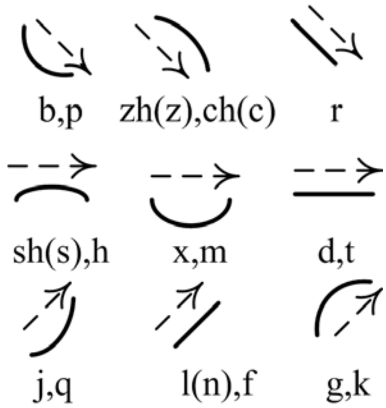


Figure 6.6: Nine types of consonant symbols in Renqun shorthand

If the length of the stroke is not considered, there are 9 types of basic consonant symbols (see Figure 6.6) in Renqun shorthand. A standard 9-14-9 back-propagation neural network was used to classify the class that the stroke belongs to. Nine output dimensions denote 9 types of strokes. Four writers were asked to write each type of stroke 20 times. In total, 720 samples with 80 samples for each stroke type were collected. As shown in Figure 6.7, all samples were required to be written in a bounding box and strictly followed the writing rules of Renqun notation. The average classification rate obtained using 5-fold cross-validation strategy is shown in Table 6.3. The sample data was divided into 5 subsets of equal size. The neural network was trained 5 times, each time leaving out one of the subsets from training for testing.

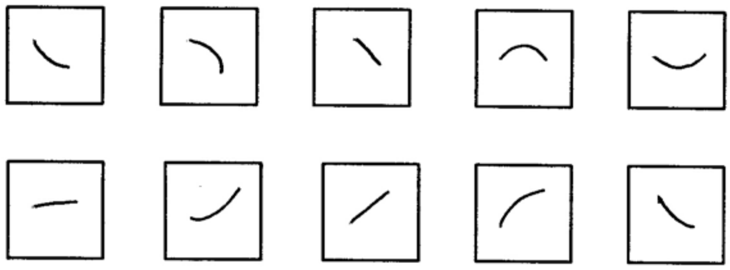


Figure 6.7: Samples used for the training of stroke neural network classifier

Table 6.3: Accuracy of stroke neural network classifier

Stroke type	Training accuracy	Testing accuracy
T 1:	**100%**	**100%**
T 2:	**92.19%**	**81.25%**
T 3:	**95.31%**	**87.5%**
T 4:	**100%**	**100%**
T 5:	**100%**	**87.5%**
T 6:	**100%**	**100%**
T 7:	**96.87%**	**93.75%**
T 8:	**100%**	**100%**
T 9:	**93.75%**	**75%**
Average	**97.57%**	**91.67%**

The trained neural networks are used for the deletion of dummy dominant points and stroke reconstruction using the same algorithms in 4.1.2.2 and 4.1.2.3 except for the two parameters MinSL and MaxSL chosing from the basic consonants of two types of lengths (2:3) respectively.

6.3 Evaluation

Three experiments were carried out to evaluate the overall performance of the two-stage (segmentation & classification) approach. Three writers were involved in the experiments. Before the formal data collection, each writer was given a sample sheet and did half an hour's practice. Using the Wacom ART II digitizer, the sample data were collected using the same PSIS system as described in section 4.3.1.

In the first experiment, fifty vocalized outlines (see Appendix D) including all features and their combinations in Renqun shorthand were selected for the evaluation. The samples chosen in the evaluation were complicated cases and the average number of primitives per outline was 2.1.

All outlines were required to be written with a slant angle of about 45^o. In total, 1677 outlines were collected from three shorthand writers (551 from writer A, 568 from writer B, 558 from writer C). Table 6.4 shows the statistics of the samples collected in the first experiment. The primitives of each outline are required to be written with a slant angle of about 45^o. There are 391 boundaries with low angularity, which occupies about 391/1713=23% of the total number of boundaries. According to our investigation, if primitives of each outline are written with a slant angle between 15^o and 25^o, there will be 683 boundaries with low angularity, which occupies nearly 772/1713=45% of the total number of boundaries. It will significantly deteriorate the machine readability of Renqun notation.

Table 6.4: The statistics of the samples

	Number of primitives			
	Writer A	**Writer B**	**Writer C**	**Total**
Number of outlines	551	568	558	1677
Number of boundaries	561	582	570	1713
Number of low angularity	128	136	127	391
Number of circles/loops	391	405	387	1183
Number of hooks	242	253	245	740
Number of short Primitives	220	228	234	682

Table 6.5: The experimental results of the segmentation

	Writer A	**Writer B**	**Writer C**	**Total**	**Average Rate**
Number of segments	561	582	570	1713	-
Number of correctly segments	516	533	530	1579	-
Rate of segments	91.98%	91.58%	92.98%	-	92.18%
Number of circles/loops	128	136	127	391	-
Number of correctly detected circles/loops	387	396	376	1162	-
Rate of circles/loops	98.98%	97.78%	97.16%	-	98.22%
Number of hooks	242	253	245	740	-
Number of correctly Detected hooks	222	228	219	669	-
Rate of hooks	91.74%	90.12%	89.39%	-	90.41%
Number of short primitives	220	228	234	682	-
Number of correctly detected short primitives	193	202	207	602	-
Rate of short primitives	87.73%	88.60%	88.46%	-	88.27%

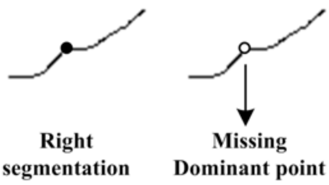
Table 6.5 lists the accuracies of different procedures in segmentation stage. The recognition rate of segments, circles/loops, hook and short primitives were 92.18%, 98.22%, 90.41% and 88.27% respectively. Among all procedures in the segmentation stage, short primitive detection had the lowest accuracy. From the machine printed Renqun outline shown in Figure 6.8(a), we can see that the length of short primitive is so short that it has little impact on the appearance of the whole outline. Therefore, it is relatively difficult for the recognition system to detect them accurately. In addition, writers tended to write short primitives very cursorily as shown in Figure 6.8(b) and (c) (The place of short primitives are indicated by the dashed circles.), which increased the

difficulty of short primitive detection. Fortunately, such errors do not impact greatly on the performance of the system. Chord BAC will be classified as the same category of the chord AC by the neural network if short primitive AB was not recognized. And a missing short primitive could be recovered in the transcription system with contextual information as each consonant always has one corresponding rhyme according to the pronunciation rules of Chinese Pinyin.

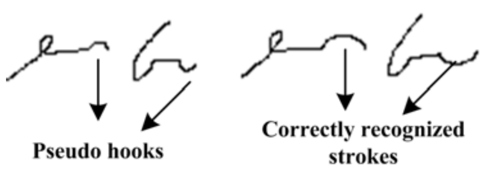
(a) (b) (c)

Figure 6.8: Illustration of short primitive errors

Figure 6.9 shows typical errors in other segmentation stages. Errors in segmentation mainly came from failure of detecting dominant points at smooth boundaries (see Figure 6.9(a)). Figure 6.9(b) shows two typical cases where the basic consonant stroke which represent primitive /N/ or /NG/ is wrongly classified as a hook. The above scenario happened particularly when the input outline was small so that the size of hook was relatively large causing some confusion between hook and stroke.



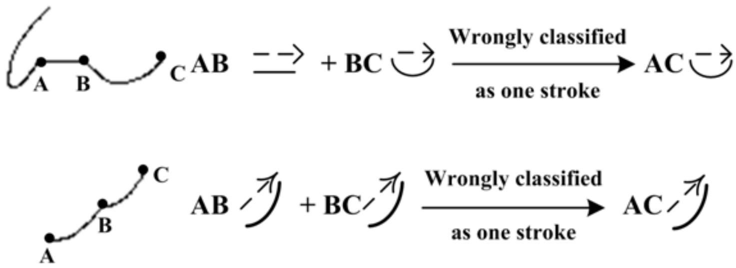
(a) Segmentation error



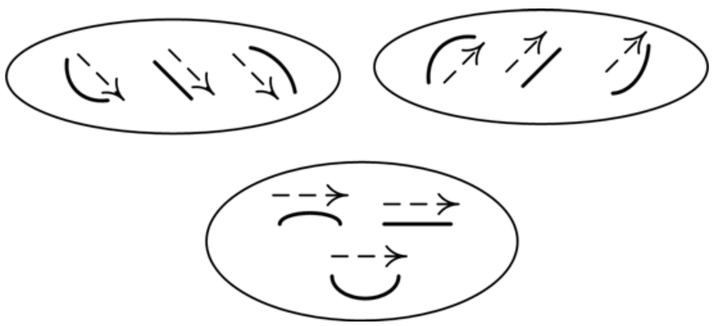
(b) Hook classification error

Figure 6.9: Typical errors in the recognition of Renqun shorthand

Table 6.6 and Table 6.7 show experimental results in the classification stage. Due to the geometric simplicity of Renqun outlines, the rate of stroke reconstruction and primitive classification achieved recognition rates of 99% and 99.15% (if primitives in the top three places of the ranked list were included). If the ranked list were not considered, only one primitive with the highest classification rate would be included in the final optimal result. Figure 6.10 (a) shows some typical errors in the classification stage in which a non-stroke pattern was wrongly classified as a stroke pattern. This type of error contributed the majority of mistakes in the stroke construction. Figure 6.10(b) lists three groups of primitives that are easily confused with each other during the stroke classification.



(a) Examples of typical mistakes made in stroke reconstruction



(b) Three groups of primitives
Figure 6.10: Typical errors in the classification

Table 6.6: The experimental results of stroke reconstruction

	Writer A	Writer B	Writer C	Total	Average Accuracy
Number of outlines Without any segmentation errors	464	466	482	1412	-
Number of outlines Correctly reconstructed	460	462	476	1398	-
Accuracy	99.14%	99.14%	98.76%	-	99%

Table 6.7: The experimental results of the neural network classifier

	Writer A	Writer B	Writer C	Total	Average percentage
Number of primitives needed to be classified	1135	1119	1168	3422	-
Number of primitives in the first place of rank list	1053	1036	1055	3144	-
Percentage of primitives in the first place of rank list	92.77%	92.58%	90.32%	-	91.88%
Number of primitives in the top three places of rank list	67	80	102	249	-
Percentage of primitives in the top three places of rank list	98.68%	99.73%	99.06%	-	99.15%

The performance to the recognition of vocalized outline is summarized in Table 6.8. The accuracy of classification (98.58%) was much higher than that of segmentation (84.20%). Compared with Pitman outlines, Renqun outlines contain fewer primitives and fewer low angularity boundaries. That is the major reason that a correct overall recognition rate of 83% was achieved, which is significantly higher than the accuracy of 75.33% observed in the recognition of Pitman consonant outline.

Table 6.8: The experimental results for vocalized outlines

	Writer A	**Writer B**	**Writer C**	**Total**	**Average Rate**
Number of outlines Without any segmentation errors	464	466	482	1412	-
Rate of segmentation	84.21%	82.04%	86.38%	-	84.20%
Number of outlines correctly classified *	458	461	473	1392	-
Rate of classification	98.70%	98.93%	98.13%	-	98.58%
Total accuracy	83.12%	81.16%	84.76%	-	83.00%

* The outline is regarded as being classified correctly if the correct path is found and the correct result of each segment is included in the final ranked list. The order of the ranked list is not considered.

The second experiment was carried out to evaluate the impact of different writing angles on the accuracy of the recognition system. Using the same 50 vocalized outline in the first experimentation, 1521 samples (489 from writer A, 529 from writer B, 503 from writer C) with a slanting angle between 15^o and 25^o were collected. As the comparison shown in the Figure 6.11, for the same Renqun outline, primitives written with a slanting angle between 15^o and 25^o are likely to produce more low angularity boundaries. Table 6.9 shows the statistics of the samples. Boundaries with low angularity occupied 759/1558=48.72% of the total number of boundaries, which is 48.72%-23%=25.72% higher than samples with a slanting angle of 45^o in the first experiment. Detailed recognition results are shown in Table 6.10 and Table 6.11. The segmentation accuracy and the overall accuracy were 72.91% and 71.57% respectively. Among all processes in the segmentation stage, the recognition rate in the second experiment is lower than that in the first experiment. From the comparison shown in Figure 6.12, the newly introduced rule produced less low angularity boundaries, and improved the recognition performance consequently.

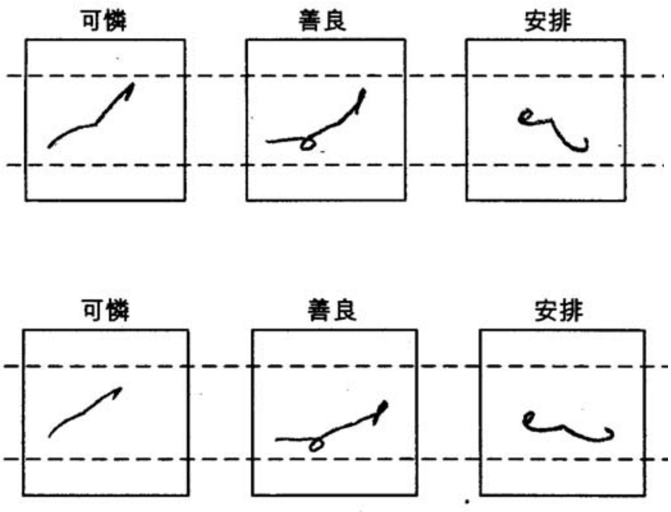


Figure 6.11: Outlines with different slanting angle

Table 6.9: The statistics of the samples

	Number of primitives			
	Writer A	**Writer B**	**Writer C**	**Total**
Number of outlines	489	529	503	1521
Number of boundaries	497	539	522	1558
Number of low angularity	255	287	217	759
Number of circles/loops	342	387	361	1090
Number of hooks	205	222	213	640
Number of short Primitives	195	212	205	612

Table 6.10: The experimental results of the segmentation stage

	Writer A	**Writer B**	**Writer C**	**Total**	**Average Rate**
Number of segments	497	539	522	1558	-
Number of correctly segments	403	442	422	1267	-
Rate of segments	81.09%	82.00%	80.84%	-	81.32%
Number of correctly detected circles/loops	309	372	345	1026	-
Rate of circles/loops	90.35%	96.12%	95.57%	-	94.13%
Number of correctly Detected hooks	186	197	190	573	-
Rate of hooks	90.73%	88.73%	89.20%	-	89.53%
Number of correctly detected short primitives	169	182	179	530	-
Rate of short primitives	86.67%	85.85%	87.32%	-	86.60%

Table 6.11: The experimental results of vocalized outline

	Writer A	Writer B	Writer C	Total	Average Rate
Number of outlines Without any segmentation errors	358	400	378	1136	-
Rate of segmentation	72.03%	74.21%	72.41%	-	72.91%
Number of outlines correctly classified *	351	390	374	1115	-
Rate of classification	98.05%	97.50%	98.94%	-	98.15%
Total accuracy	70.62%	72.36%	71.65%	-	71.57%

* The outline is regarded as being classified correctly if the correct path is found and the correct result of each segment is included in the final ranked list. The order of the ranked list is not considered.

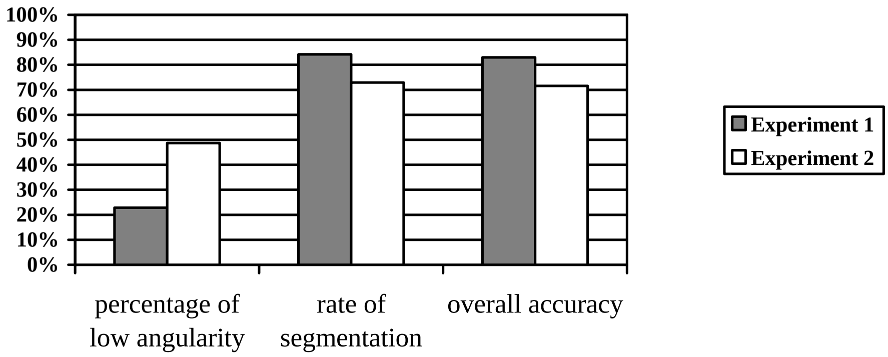


Figure 6.12: Results comparison of two experiments

In the third experiment, 436 Renqun shortforms were written on a sample sheet and each writer was asked to write each shortform with a slant angle of about 45^o five times. Figure 6.13 shows some examples of Renqun shortforms. As mentioned previously, Renqun shortforms follow the simplified writing principles of Renqun shorthand. The average number of primitives per outline was 1.2. Most of the outline contains only a single penstroke which does not need segmentation. The detailed evaluation result is shown in Table 6.12. The overall accuracy is 84.07%. Recognition rates of each procedure, which is the ratio of the correctly recognized samples in this procedure over the total number of samples, are also listed. It is not surprising that the recognition performance is a little bit higher than that of vocalized outline since it involves relatively simple geometric features. Errors made in the recognition of shortform were the same as those in the recognition of vocalized outline such as missing dominant points in the low angularity boundaries, and failure to detect short primitives.

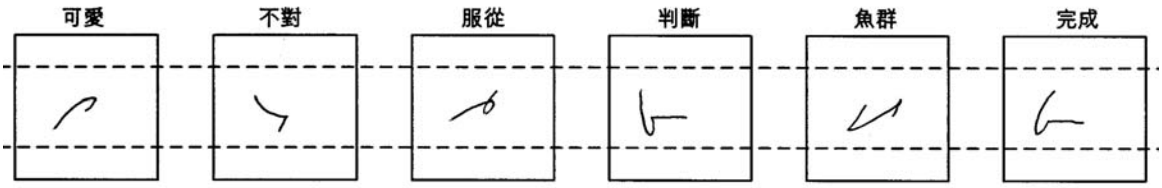


Figure 6.13: Renqun shortform

Table 6.12: Evaluation results of Renqun shortform

Recognition stages		Writer 1	Writer 2	Writer 3	Average
Segmentation	Dominant points	92.11%	91.56%	92.91%	92.19%
	Circles & loops	100%	97.86%	98.39%	98.75%
	Hooks	92.35%	91.83%	94.58%	92.92%
	Short Primitives	91.20%	90.78%	87.75%	89.91%
Classification		95.68%	98.05%	98.32%	97.35%
Average accuracy		83.79%	80.77%	87.65%	84.07%

6.4 Conclusion

In this chapter, an overall solution to the recognition of Renqun shorthand is described. The research could lead to a solution to high speed Chinese text entry on hand-held devices using Renqun shorthand. Composed of a sequence of rhymes and consonants, Renqun handwriting shorthand is also recognized by the shape-based two-stage approach. A new writing rule is also proposed to improve the 'machinography' of Renqun shorthand. Experiments following the new writing rule showed that the proposed approach recognized vocalized outline and Renqun shortform at rates of 83% and 84.07% respectively. For vocalized outlines without the new rule, boundaries with low angularity occupy nearly 50% of the total number of boundaries and the recognition accuracy degraded to 71.57%.

Future work should be focused in the following two directions: The first is to improve the performance of the recognition system in aspects of effectively detecting hooks, dominant points at smooth boundaries and locating short primitives accurately. The second is the post processing of the recognition results to readable Chinese characters. The transcription procedure includes recognized phoneme-to-Pinyin conversion as well as Pinyin-to-Chinese character conversion. So far, little work has been reported in this area.

CHAPTER

7

Discussion, conclusions and recommendations

7.1 A Pitman shorthand online system

A prototype on-line system was implemented on a Fujitsu Lifebook T3010 Tablet PC with 12.1 inch TFT-XGA/ 1024 X 768 pixel transmissive LCD. The T3010 Tablet PC contains a 1.4 GHz processor running Windows XP Tablet PC Edition. All of the recognition and transcription software was implemented as an application-level program written in C and JAVA running on top of Windows. By architecting the system in this way using a high level language, standard operating system, and a general purpose tablet PC, the effort required to port to other handheld devices running on Windows Mobile is significantly eased.

The prototype Pitman Shorthand Online Recognition System layered architecture, as shown in Figure 7.1, has four components. The lowest layer is the operating system that is the interface to the hardware. The operating system chosen for the system was Windows XP; however, any standard operating system may be used. Two of the remaining components are the recognition system and transcription system that contain the core recognition and transcript algorithms for operations such as segmentation, classification, as well as translation. These modules are plugged together to deal with handwritten outlines.

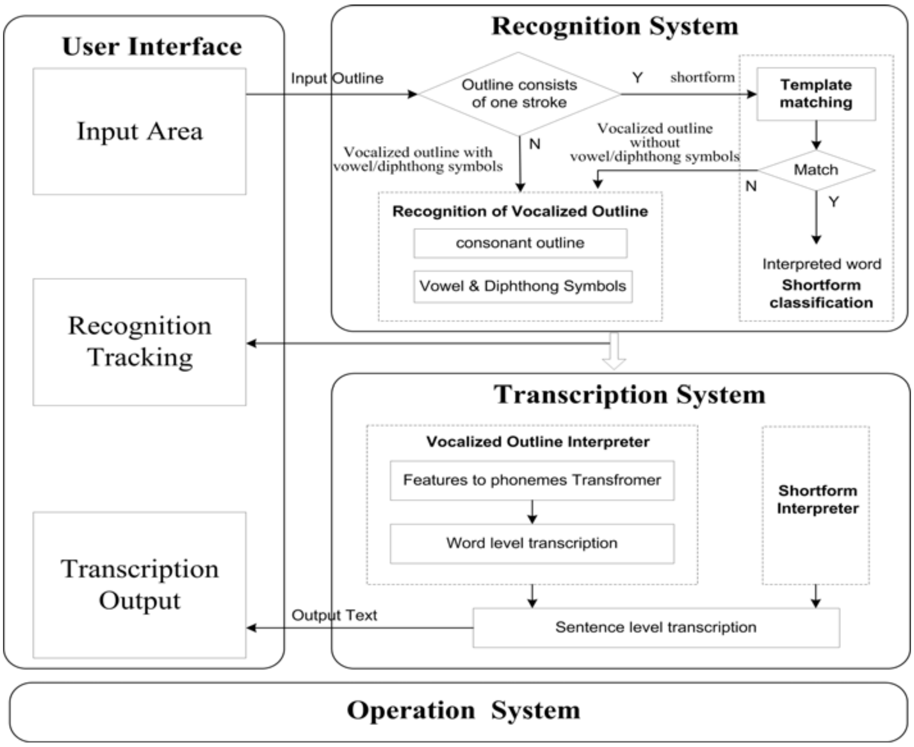


Figure 7.1: System diagram of the prototype Tablet PC PSRS

Figure 7.2 shows the user interface of the demonstration system. The user interface consists of an input area and two output areas (recognition output and transcription output). The reason why the recognition output is included here is to enable users to track the recognition performance, since this is a simulation system for use in the laboratory. The possible interpretations of the input outline are displayed in the transcription area and listed according to their possibilities. In order to track and analyze the complete process of recognition and interpretation of Pitman Shorthand, all of the same basic consonant outline and its associative vowel & diphthong symbols can be tracked in the recognition output area where the feature sets are ordered by their ranked list in the same row. Different basic consonants are shown in different rows.

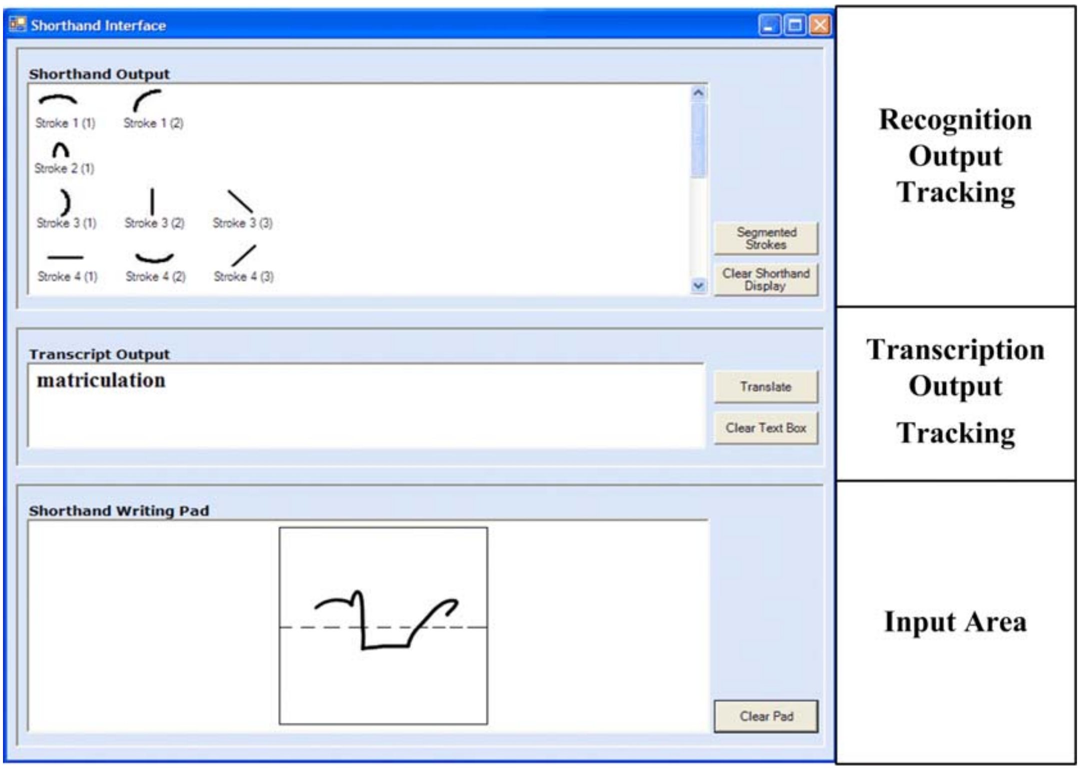


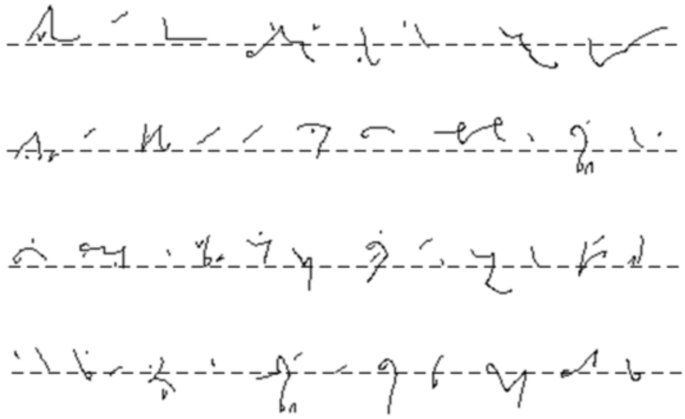
Figure 7.2: System interface

7.2 Evaluation

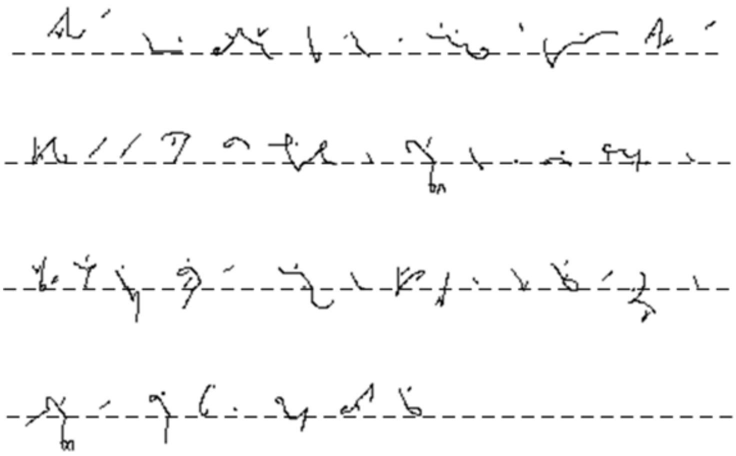
Most of the existing algorithms were developed on a general desktop computer with a external Wacom digitizer. They are portable to the Tablet PC platform which has three distinct advantages over the previous arrangement: it has an inherent active digitizer, it has a large hard drive for permanent storage, and it is able to provide an identical development and debugging environment as the PC. These three factors allowed us to use the same developmental and debugging tools that were used on a PC. The only effort was optimizing the pre-processing algorithms since the input interface was changed from a pen-writing tablet to the transmissive screen of the Table PC. In order to migrate into the Tablet PC environment, the preprocessing stage of the recognition system has to be modified to accommodate the feature of the digitizer. First, the x,y coordinates of each sample points should be adjusted because the origin of the Euclidean coordinate system lies in different positions of the interface. Second, the Tablet PC's digitizer is more noisy (i.e., the cursor position "jitters" between pixels when you hold the pen stationary). In order to reduce noise effectively, a Gaussian filter in step 8 of the preprocessing is enhanced by using a large value of σ and n ($\sigma = 4, n = 5$).The choice of σ and n is a tradeoff between the reduction of noise and the reservation of major features of the input outline. The larger the values of σ and n, the smoother the curve, the more likely corner will lose its angularity and vice versa.

A sample sentence (see Appendix E) composed of words from the most frequently used 5000 English words provided at “edit” (http://www.edict.com.hk/textanalyser/) was selected for the evaluation. The sentence consists of 20 shortforms and 29 vocalized outlines which include 58 consonant strokes, 11 hooks, 10 circles and 40 vowel and diphthong symbols. In order to focus on the performance of the recognition accuracy, vocalized outlines used in the evaluation contain no smooth junctions. The average number of primitives per vocalized outline was 2.3.

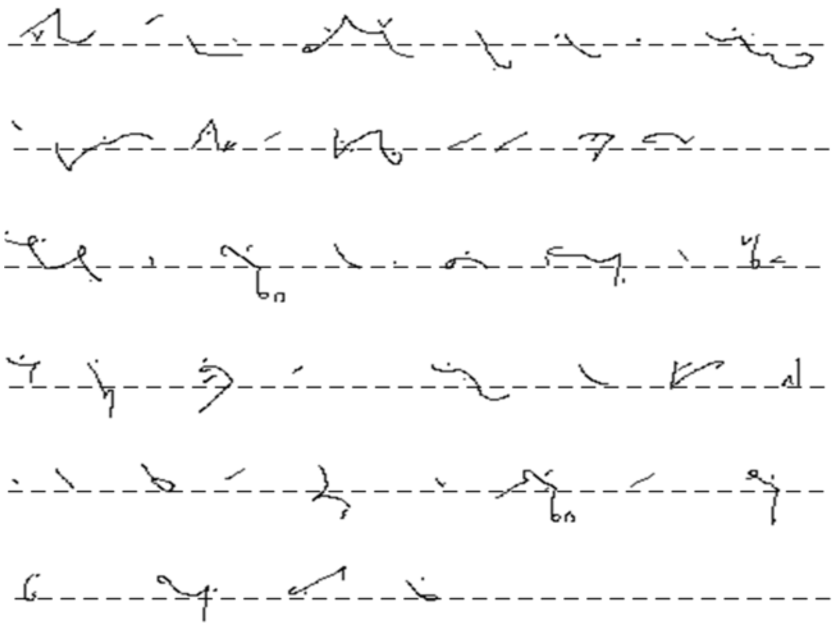
Three groups of three writers took part in the evaluation. The first group were proficient shorthand writers. The second group consisted of non-frequent users of Pitman shorthand. The third group were Pitman shorthand beginners. Before the experimentation, they were given some time to practice until they considered they had become proficient and familiar with writing on the transmissive screen of the Tablet PC using the special pen after half an hour’s practice. Due to the sliding of the pen tip on the transmissive screen, writing on a Tablet PC is slower (around 80-90 wpm) than the natural writing in a pen and paper environment (around 100 wpm). A sample sheet with all standard outlines of the sentence was provided to each writer. Each writer was asked to write the sentence three times. Examples of the outlines collected are shown in Figure 7.3. Outlines were written on the transmissive screen of the Table PC word by word and their information was saved automatically in separate data files with pen-down and pen-up indicators for each stroke. After finishing one outline, the writer needs to click the ‘Clear pad’ button to clear the writing area and the recognition results of the previous outline would show on the two output areas. In addition, the writers were encouraged to compare the recognition result with their writing by pressing the 'Translate' button.



(a) Samples written by proficient shorthand writer



(b) Samples written by non-frequent users of Pitman shorthand



(c) Samples written by shorthand beginner

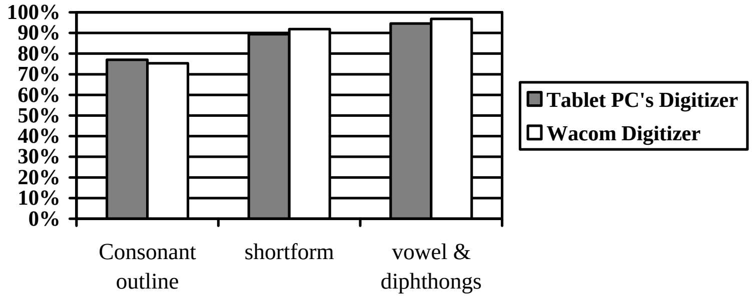
Figure 7.3: Examples of handwritten shorthand as written on the Tablet PC

7.2.1 Recognition accuracy

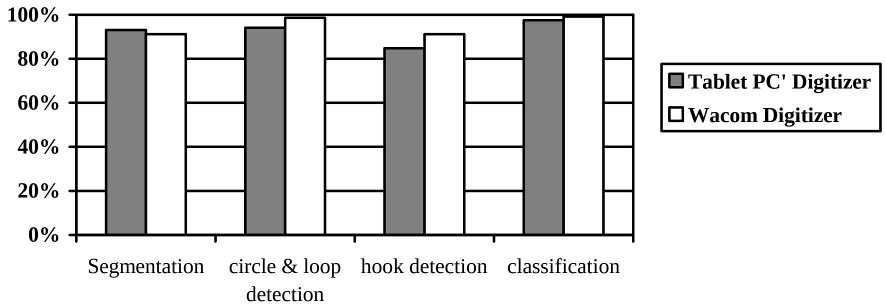
Table 7.1 shows the recognition rate of the shape recognition system. It can be seen that the algorithm is accurate at identifying dominant points (segmentation) as well as circle and loop detection in the consonant outline. The average accuracies are 93.1% and 94.1% respectively. However, the accuracy of hook detection was only about 84.8%. The correct classification rate of vowels and diphthongs was 97.5%. The overall recognition rate for vocalized outlines is 77%. The position-sequence of vowels was not evaluated as it is common practice for stenographers to write them inaccurately and sometimes even omit vowels and diphthongs. The accuracy of shortform recognition was 89.4%. We also observed an interesting phenomenon that the accuracy of beginners is higher than that of the other groups. The reason could be attributed to the beginners might have few “bad” writing habits which disturb the recognition system and they paid more attention to the feedback of the recognition system. Figure 7.3 shows the examples produced by different group of writers. Samples written by shorthand beginners are closer to “text-book” quality. Scripts written by writers in the first and the second groups are relatively small (encouraged by Pitmans), which caused confusion between hooks and strokes due to larger relative size of hooks. In the experiments, shorthand beginners were willing to change their writing styles (eg. adjusting the size of hook in cases that were classified as a stroke /N/ or /NG/, writing boundaries with low angularity more clearly) from the feedback of the demonstration system to achieve higher recognition accuracy. Proficient writers were less likely to change the style of their writing and there were consistent errors (eg. failure in hook detection due to its small size, failure in detecting Type D circles) in their writing. Given the accuracies for different stages and different signals, the average overall accuracy was 82.1%.

Table 7.1: Evaluation results

Stage		Group 1	Group 2	Group 3	Average
Consonant outline	Segmentation	91.4%	93.1%	94.8%	93.1%
	Circle, Loop	90.0%	97.8%	94.4%	94.1%
	Hook	82.8%	83.8%	87.9%	84.8%
	Classification	97.7%	98.0%	96.9%	97.5%
	Overall	75.6%	75.3%	80.2%	77.0%
Vowel & diphthong		92.2%	94.4%	97.2%	94.6%
Shortform		90.0%	88.9%	89.4%	89.4%
Overall accuracy		**81.5%**	**80.9%**	**84.0%**	**82.1%**



(a) Evaluation results of different components in Pitman shorthand



(b) Evaluation results of different stages for the recognition of consonant outline

Figure 7.4: Comparison of the experimental results on different digitizers

As shown in Figure 7.4, the evaluation results obtained from the table PC's digitizer are similar to the results on the Wacom digitizer (Chapter 4 and 5), which proved the robustness of the algorithms. For vocalized outlines, it is not surprising that the recognition rate is a little higher than the 75.33% observed in section 4.3 because the samples used in this evaluation are simpler and contain less low angularity boundaries.

7.2.2 Transcription accuracy

For word-level transcription, the transcription system (Note: transcription is related to the work at Nottingham University) no longer needs to be concerned with shortforms beyond the necessity to distinguish at the semantic level between shortforms having two or more associated words (see Appendix B). As for vocalized outline, the transcription process (Htwe et al., 2004) includes three steps as follows:

a. Approximate pattern matching

Approximate pattern matching is, in fact, a heuristic approach in which uncertain penstrokes are coped with and wrong vowel locations are estimated. It uses the nearest neighbor query and the heuristic based on the similarity between the two patterns. Here, similarity stands for, “having similar angular structure” for consonant kernels, “having similar shape” for circular components and “having similar location and shape” for vowel primitives.

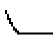
b. Features to phonemes conversion

In some cases, a phonetic representation is composed of more than one feature primitives obtained in the recognition system. For example, outlines of “W” and “Y” are classified separately as a small hook (or) and a stroke written upwards . Therefore, an additional step is needed to correctly interpret clustered primitives into related phonemes. This process is referred as “Feature to Phoneme Conversion”.

Five production rules were adopted in the system. They are: “Direct Translation”, “Primitive Combination”, “Primitive Combination and Reverse Ordering”, “Feature Detection” and “Length Detection”.

c. Phoneme ordering

In the recognition system, vowels are then recognized from left to right order and appended at the end of the consonant primitives. Therefore, the phoneme ordering requires correct insertion of vowels among the consonant kernels. The system uses “Dominant Point based Insertion (DPI) algorithm” for phoneme ordering in which vowels are assigned to related consonants using dominant point information and inserted at the right order using the sequence information.

The experimental results were evaluated from the following three main aspects: accuracy of final text output; consequence of inconsistent writing styles from writers to writers; and the impact of approximate pattern matching algorithm in the recovery of classification errors. Experimental results show that 78.6% of the written outlines can be interpreted correctly, but the remaining 21.4% failed to produce related words. A very interesting phenomenon observed in this experiment is that 8.2% of perfect transcription occurs in the presence of recognition errors, which proves the beauty of the approximate pattern-matching algorithm. However, approximate matching sometimes produces too many possibilities for a single outline. Another important factor observed in this experiment is the omission of vowels in an outline causes a major failure in transcription. And the system hits a complete failure when the input outlines are legible to human readers, but are not exactly consistent with the writing rules of Pitman shorthand. For example, word ‘books’ () is written as ; word ‘storage’ () is written as  . Although the natural feeling of writing is a primary concern of our handwritting recognition research, disagreement with writing rules is not allowed in our system.

7.3 Commercialization of the shorthand system

For analysis of the commercialization process of a shorthand recognition system, a Pitman shorthand demonstration system, which provides valuable direction and hints for future work, was developed. Critical technological issues in commercializing shorthand recognition system were discussed. Investigation into the commercialization of a shorthand system is meaningful to enable Pitman shorthand recognition techniques to smoothly pass “the valley of death” in the lifecycle.

7.3.1 Commercialization Process of Shorthand recognition system

Like any other technology transfer from the laboratory to reality, the commercialization of a shorthand recognition system also involves many factors spanning technological issues, organizational issues, and industrial issues to social issues. According to the theories about technology innovation and commercialization (Zmud, 1984; Betz, 1987), the commercialization process of new technologies can be divided into several phases: Conceiving Phase, Investigating Phase, Developing Phase, Introducing Phase, and Expanding Phase. In different phases, there are different driving forces that have different impacts on the commercializing process. As shown in the Figure 7.5, the four key driving forces that drive the solution into the next stages are research interests, technology, market, and business models. In this project, a Pitman online demonstration system was developed to provide a prototype from the laboratory work to the real application.

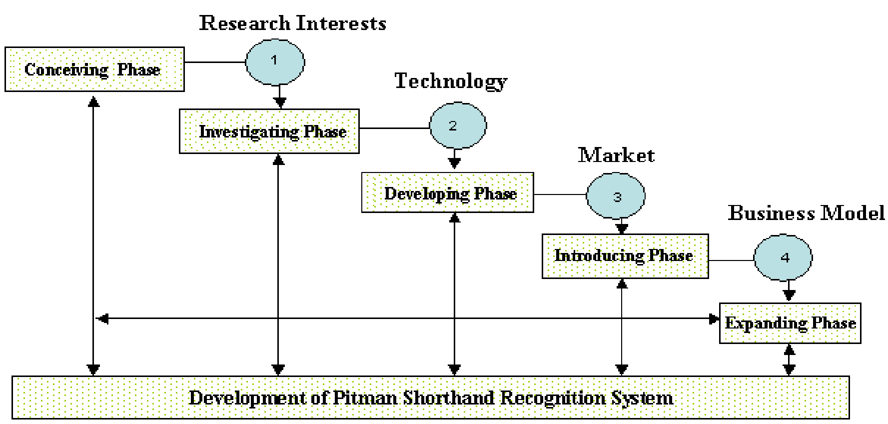


Figure 7.5: Commercialization process

There are two elements working in concert that drive the commercialization process of a Pitman Shorthand recognition system: "requirement pull" and "technology push". "Requirement pull" is a need looking for an automation system for Pitman Shorthand recognition from stenographers. "Technology Push" is a situation when the technology professionals are looking for opportunities to benefit stenographers from a developed recognition technology. Although Pitman Shorthand has been widely used around the world, the total number of users is limited despite its high recording speed. The reasons are many. The first and foremost is that shorthand writing is a complex skill that needs good training and plenty of practice. From another perspective, the rapid development of computing technology and the widespread availability of portable hand-held devices enable Pitman Shorthand writing to meet increasingly high-speed recording requirement. Therefore, the driving force of "technology push" is playing a more important role in the commercialization process of Pitman Shorthand recognition system than "requirement pull" does, especially in the current lifecycle phase of the Pitman Shorthand Recognition Solution.

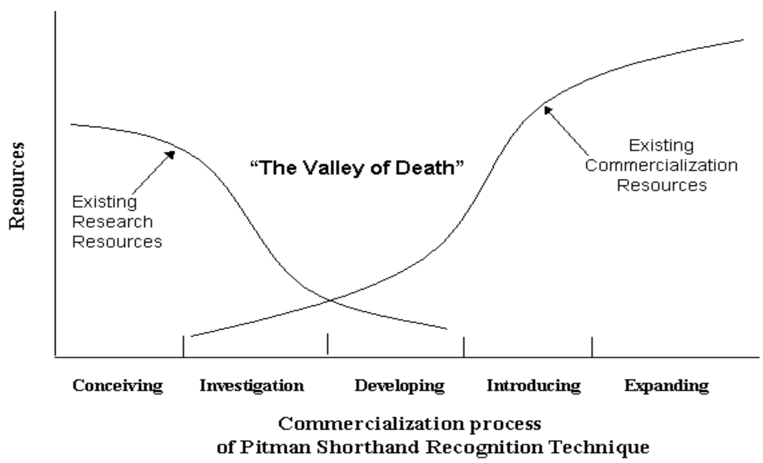


Figure 7.6: The valley of death of Pitman shorthand system

According to the current development level and resource availability, the Pitman Shorthand recognition system is now located in "Death valley" of its lifecycle. Studies show that researcher should pay more attention to technological issues when a technology is located in its “Death valley”. The attention to the key issues will benefit facilitating the movement of the Pitman Shorthand recognition system from an academic demonstration system to a successful commercial application. Therefore, whether involved in the commercialization or not, the researcher is well advised to pay more attention to the technological issues and carefully develop their first demonstration recognition system in order to pass successfully through “Death valley”. This logic also explains why critical issues regarding the commercialization of Pitman Shorthand recognition system become the central topic of this section.

7.3.2 Critical issues of commercialization

In 2005, there were 14.2 million PDA units shipped worldwide. (a 23 percent year-over-year increase). IDC forecasts that the worldwide handheld market will hit 150 million units by the end of 2006. The market potential of shorthand system is determined by the number of shorthand writers among PDA users. Given that one among one thousand users were interested in the shorthand entry system, there would be 1.5 million units equipped with shorthand recognition & transcription system.

Besides the potential market analyzed above, there are two core technological dimensions that are critical to develop a successful Pitman Shorthand Recognition System: algorithm performance and software principles. Algorithm performance is mainly determined by the outline recognition accuracy and transcription accuracy, although the transcription speed is also a related factor. Software principles taken here are vital to develop a successful demonstration and commercial application of a Pitman Shorthand recognition system. They have also great impact on the commercialization process of a handwritten Pitman Shorthand recognition system.

(1) Algorithm performance

As a matter of fact, some algorithms concerning recognition and interpretation of handwritten Pitman Shorthand had been put forward as early as more than two decades ago. But these algorithms could not at that time be integrated into any Pitman Shorthand recognition solutions. On one hand, this can be ascribed to their inherent defects that cannot achieve high recognition accuracy. On the other hand, due to the limited computing technology at that time, computationally intensive algorithms could not work effectively and efficiently. In recent years, the rapid development of computing technology and mobile intelligent devices make it possible to achieve better recognition and interpretation performance of handwritten Pitman Shorthand by introducing computationally intensive algorithms.

Recent research experiences have shown the great potential of these computationally intensive algorithms. In the latest research, a recognition approach which is aimed at recognizing all components of Pitman Shorthand has been proposed. This integrated approach takes advantage of neural network and techniques in shape recognition (such as polygonal approximation, detecting of segmentation points by region of support, Hausdorff Distance, etc.) and achieved much better performance than previous research. At the same time, researchers at Nottingham University, focus on the computer transcription of Pitman shorthand, particularly from the point of view of linguistic post-processing. Through collaboration of the two research groups, an overall solution to Pitman shorthand has been developed and implemented successfully.

Besides algorithm performance, recognition speed is also a key factor in the commercialization process. One of critical success factors of commercializing a Pitman Shorthand recognition system is that it should be able to run effectively on various device platforms including PDAs, smartphones, laptops, and Tablet PCs. The recognition aspect can be ported to DSP processor or other FPGA based hardware. Therefore we should pay more attention to those algorithms that possess low computing requirement as well as good performance in the commercialization process. Currently, most algorithms regarding the recognition and interpretation of Pitman Shorthand are based on PCs and Tablet PCs, few algorithms has been evaluated on PDAs. Therefore, increasing and evaluating the applicability of recognition algorithms among various platforms would become another key issue for academic researchers when they are engaged in improving algorithm accuracy.

(2) Software principle

The commercialization of a Pitman Shorthand recognition system is different from its academic research. Software principles including modularity, hardware independence, and software portability have important influences on the commercialization process of software application, especially for those running mobile platforms.

A typical Pitman Shorthand online recognition system normally consists of a recognition system and transcription system that are often from different research groups or different development groups. Therefore modularity is extremely useful because it allows people to focus their internal work rather than the overall issues. Also, with modularity, the preexisting recognition system and transcription system from different groups may be reused for a new entire system. Since these modules have already been written and tested, development and debugging time is reduced. These standard interfaces also make it easy to change the algorithm used in the Pitman Shorthand recognition system.

With hard independence and portability, a Pitman Shorthand recognition system can be first written for an architecture that has a better development and testing environment (eg. Windows running on a Table PC). Once the system is ready for launch, it can then be ported to the target platforms, for example, PDAs with Palm OS, or smartphones with Symbian OS.

7.4 Conclusion

This chapter has discussed the critical technological issues of commercializing a Pitman Shorthand recognition system with a demonstration system that illustrates the current state of Pitman Shorthand recognition techniques. These critical issues in the technological field come from two aspects: algorithm performance and software principles. Given the current social and technological environment that Pitman Shorthand is facing, it is essential to review the critical issues in order to pass the "the valley of death" in the lifecycle of Pitman Shorthand recognition technique.

In order to facilitate the commercialization of Pitman shorthand, research on fields below is vital. The first and most important issue is to develop a sentence-level shorthand recognition system by enabling multiple shorthand outlines to be entered simultaneously. Secondly, a powerful middle variable tracker is demanded among researchers so that they can debug results in the internal processes such as dominant points obtained in the segmentation, features converted from phoneme, etc. With such debugging tools, researchers could effectively improve their algorithms. Thirdly, because different people have different writing habits, a flexible configuration module which end users are able to adjust recognition algorithm parameters, is important to the success of the commercialization of a shorthand recognition system.

CHAPTER

8

Summary of this research

8.1 Achievements

This research concentrates on investigating the potential of Pitman shorthand and Renqun shorthand as a means of fast text entry for both English and Chinese. The research work includes three aspects: recognition of two shorthand systems, shorthand machine-compatible rules, and a demonstration system.

With an appreciation of the deficiency of previous recognition algorithms, some techniques in shape recognition are introduced into this research. Along with the introduction of shape recognition techniques, three algorithms are proposed to the recognition of different components in Pitman shorthand based on their special geometric features. As for Pitman shortforms, a template-based approach is developed to recognize those single, simple penstrokes. For any two points in the input pattern and reference templates, polar coordinate system is used to measure their distance and Hausdorff Distance is introduced to measure the dissimilarity between the input outline and templates. As for the recognition of the consonant outline, the most difficult part in shorthand recognition, a two-stage approach including the segmentation and the classification of each consonant outline is proposed and investigated. In the segmentation stage, dominant points are detected by using “the region of support” to measure their significance. Afterwards, special geometric features such as circles, loops and hooks are detected. Finally, polygonal approximation is introduced for the implementation of a split-and-merge algorithm. In the classification stage, dummy dominant points are suppressed and a heuristic merge algorithm is used for the stroke reconstruction.

Compared with Pitman shorthand, both vocalized outlines and shortforms of Renqun shorthand are composed of rhymes and consonants and have similar shape features. Due to the similar geometric features of the two shorthand systems, the two-stage approach used for the recognition of Pitman consonants is revised for recognizing both the vocalized outlines and shortforms of Renqun shorthand. In the segmentation stage, after dominant points and rhymes like circles, loops and hooks have been detected. A short primitive detection module is tailored to detect short rhymes in Renqun outlines. In the classification, a stroke/non-stroke classifier is constructed for the stroke reconstruction and a standard neural network is trained for the stroke classification.

To overcome the inherent properties which deteriorate the ‘machinography’ of the two shorthand systems, new rules are proposed to improve their machine readabilities. For Pitman shorthand, there are three types of boundaries among adjacent primitives: low angularity, high angularity and smooth junction. Accurate detection of boundaries among adjacent primitives has great impact on the final recognition accuracy of shorthand. Unlike the boundaries of low angularity and high angularity, smooth junctions are unlikely to be accurately detected by mathematical approaches. Therefore, a new writing rule which requires a circle to be written near the smooth junction and a point in the associated consonant outline with minimal Euclidean distance to the circle is chosen as a new dominant point for segmentation. There is no specific requirement of the position and the size of the circle. Evaluation results showed that the proposed new machine compatible shorthand puts little burden on human writing and the readability of shorthand systems from the new writing rules improved significantly while placing relatively little burden on the writer. For Renqun shorthand, the primitives are required to be written with a slant angle of about 45^o rather than the 15^o to 25^o defined in the shorthand system. Evaluation results showed that the new writing requirement reduces the percentage of boundaries with low angularity among the total number of boundaries and improves the classification rate significantly.

Based the research achievements, critical technological commercialization issues of shorthand recognition system are fully discussed from two aspects: algorithm performance and software principles. Moreover, in order to facilitate the research process, a demonstration system which includes all procedures of a commercial shorthand recognition system like data collection, shape recognition, transcription and graphical result output is implemented on a Tablet PC through collaboration with researchers from Nottingham University. Simulating the real-time interactive procedure between the stenographers and recognition system, the demonstration system demonstrated the feasibility of Pitman shorthand as a means of fast text entry and provided a good experimental environment for further research work.

8.2 Future work

As for the shape recognition of shorthand, the further work should focus on recognition techniques of consonant outlines. The smooth boundaries between strokes can be effectively detected by introducing new writing rules in our system. However the low angularity boundary between strokes and the boundary between a hook and a stroke remain a major bottleneck to the accurate recognition of shorthand shapes. In the future the feedback information of classification must be used in relevant recognition algorithms so that initial segmentation errors can be modified when poor or unlikely primitive classification occurs. Besides, the future work will also concentrate on the incorporation of higher-level linguistic knowledge in the interpretation of the shorthand outlines.

As for the interpretation of shorthand, the future work will search for more powerful interpretation algorithms to build up sentence-level transcription system since current research work has mainly focused on word-level semantic analysis.

8.3 Publications

A number of publications have resulted as a consequence of this research. These are listed bellow:

1. Ma Yang, Graham Leedham, Colin Higgins, Swe Myo Htwe, Segementation and recognition of phonetic features in handwritten Pitman shorthand, Pattern Recognition, accepted for publication in December, 2004 (in press)

2. Ma Yang, Graham Leedham, Colin Higgins, Swe Myo Htwe, Segmentation and recognition of vocalized outline in Pitman shorthand, V1, pp.441-444, Proceedings of the 17th International Conference on Pattern Recognition, Cambridge, United Kingdom, 23-26 August, 2004

3. Ma Yang, Graham Leedham, Colin Higgins, Swe Myo Htwe, On-line recognition of handwritten Pitman shorthand for fast mobile text entry, Proceedings of the IEEE 3rd International Conference on Information Technology and Applications, Vol. I, pp.686-691, July 1-4 2005, Sydney, Australia

4. Ma Yang & Graham Leedham, On-line recognition of Renqun shorthand for fast mobile Chinese text entry, Proceedings of the 12th Conference of the International Graphonomics Society, pp. 64-68, Salerno, Italy, 26-29 June 2005.

5. Ma Yang & Graham Leedham, An on-line automatic recognition system for Pitman's shorthand, Proceedings of TENCON'05, IEEE Region 10 Technical Conference, Melbourne, Australia, 21-24 November 2005.

6. Ma Yang & Graham Leedham, Critical technological issues of commercializing a Pitman shorthand recognition system, Proceedings of the Fifth International Conference on Information, Communications and Signal Processing (ICICS 2005), Bangkok, Thailand, 6-9 December 2005.

7. Swe Myo Htwe, Colin Higgins, Graham Leedham and Ma Yang, Post-processing of handwritten phonetic Pitman's shorthand using a Bayesian network built on geometric attributes, In Pattern Recognition and Image Analysis, Lecture Notes in Computer Science 3687, Springer, Sameer Singh, Maneesha Singh, Chid Apte, Petra Perner (Eds.), pp. 569-579, 2005.

8. Swe Myo Htwe, Colin Higgins, Graham Leedham and Ma Yang, Transliteration of online handwritten phonetic Pitman's shorthand with the use of a Bayesian network, Proc. of the 8th International Conference on Document Analysis and Recognition, Vol. 2, pp. 1090-1094, Seoul, Korea, 29 Aug – 1 Sep 2005.

9. Swe Myo Htwe, Colin Higgins, Graham Leedham & Ma Yang, Evaluation of Feature Sets in the Post Processing of Handwritten Pitman's Shorthand, Proceedings of the 9th International Workshop on Frontiers in Handwriting Recognition, ISBN 0-7695-2187-8, pp. 359-364, Kokubunji, Tokyo, Japan, 26-29 October 2004.

10. Swe Myo Htwe, Colin Higgins, Graham Leedham & Ma Yang, Post processing of handwriting Pitman's shorthand using unigram and heuristic approaches, Published in Lecture Notes in Computer Science: Document Analysis Systems VI, 3163, Springer-Verlag, pp. 332-336, Proceedings of the IAPR workshop on document analysis systems, University of Florence, Italy, 8-10 September 2004.

11. Swe Myo Htwe, Colin Higgins, Graham Leedham & Ma Yang, Knowledge based transcription of Pitman's handwritten shorthand using word frequency and context, Proceedings of the 7th IEEE International Conference on Development and Application Systems, pp. 508-512, Suceava, Romania, 27- 29 May 2004

12. Swe Myo Htwe, Colin Higgins, Graham Leedham & Ma Yang, Linguistic processing of phonetic primitives in on-line handwritten Pitman's shorthand, submitted to Pattern Recognition Letters (under review)

13. Ma Yang, Graham Leedham, On-line recognition of handwritten Renqun shorthand for fast mobile Chinese text entry, submitted to Pattern Recognition Letters (under review)

14. Ma Yang, Graham Leedham, Colin Higgins, Swe Myo Htwe, Investigation of techniques for the recognition of online Pitman outlines, Submitted to IEEE Transactions on Consumer Electronics (under review)

Bibliography

I.S.I.Abuhaiba, M.J.J.Holt and S.Datta (1994) **Processing of Off-Line Handwritten Text: Polygonal Approximation and Enforcement of Temporal Information**, CVGIP: Graphical Models And Image Processing, vol.56, No.4, pp.324-335

D. Albesano, R. Gemello, F. Mana, (1992) **Word Recognition With Recurrent Network Automata,** International Joint Conference on Neural Networks, 7-11 June 1992, Baltimore, USA, vol.2, pp: 308-313

E. Anquetil, G. Lorette, (1997) **Perceptual model of handwriting drawing application to the handwriting segmentation problem**, Document Analysis and Recognition, 1997., Proceedings of the Fourth International Conference on, vol. 1, 18-20 Aug. 1997, pp:112 – 117

T. Artieres, J.M. Marchand, P. Gallinari, B. Dorizzi, (2000) **Multi-modal segmental models for on-line handwriting recognition**, Pattern Recognition, 2000. Proceedings. 15th International Conference on, vol. 2, 3-7 Sept 2000, pp: 247 – 250

H.Asada, M.Brady (1986) **The Curvature Primal Sketch**, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol.8, pp.2-14

F.Betz., (1987) **Managing Technology-Competing Through New Ventures – Innovation, and Corporate Research**, Englewood Cliffs, NJ: Prentice Hall.

C.P.Brooks, (1985) **Computer Transcription of Pitman's Handwritten shorthand for the deaf**, PhD Thesis, Department of Electronics, University of Southampton

C.P. Brooks, A.F. Newell, (1985) **Computer Transcription of Handwritten Shorthand as An Aid for the Deaf—A Feasibility Study**, International Journal of Man Machine Studies, pp. 45-60

M. K. Brown, S. Ganapathy, (1983) **Preprocessing Techniques for Cursive Script Word Recognition**, Pattern Recognition, vol.16(5), pp: 447-458

S. Carbonnel, E. Anquetil, (2003) **Lexical post-processing optimization for handwritten word recognition**, Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on, 3-6 Aug. 2003, vol.1, pp: 477 - 481

R.G. Casey, E. Lecolinet, (1996) **A survey of methods and strategies in character segmentation**, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, Issue 7, pp:690 – 706

J.M. Chen, J.A.Ventura, C.H.Wu (1996) **Segmentation of Planar Curve into Circular Arcs and Line Segments**, Image and Vision Computing, vol.14, pp.71-83

J.Chen, M.K.Leung, Y.Gao,(2002) **Noisy Logo Recognition Using Line Segment Hausdorff Distance**, Pattern Recognition, vol.36, pp. 943-955

X. P. Chen, Y. Z. Qiao, (1997) **On-line Recognition of Renqun's Handwritten Shorthand**, Proceedings of the Fourth International Conference on Document Analysis and Recognition, 18-20 Aug. 1997, Ulm, Germany, pp.941 –944.

X.P. Chen, Yu, T.C. (1998) **A Study On Recognition Method of Shortforms in Chinese Handwritten Shorthand**, Proceedings of the Fourth International Conference on Signal Processing, 12-16 Oct., 1998, Beijing, China, pp.1245 –1248.

X.P.Chen, T.C. Yu, R. W. Dai (2000) **Automatic Recognition of On-line Handwritten Chinese Shorthand**, Journal of Software, vol.11, pp.1361-1367.

B.Y. Choi, S.Y. Bang, (2000) **Constraints on Handwriting Korean Character to Improve the Machine Readability**, International Journal of Industrial Ergonomics, vol.26, pp. 549-559

S.D.Connell, A.K.Jain (2001) **Template-base Online Character Recognition**, Pattern Recognition, vol.34, No.1, pp.1-14

M. Dehghan, K. Faez, M. Ahmadi, M. Shridhar, (2000) **Holistic Handwritten Word Recognition Using Discrete HMM and Self-organizing Feature Map**, 2000 IEEE International Conference on Systems, Man, and Cybernetics, 8-11 Oct. 2000, Nashville, Tennessee, USA, vol.4, pp: 2735 –2739

M.P. Dubuisson, A.K. Jain,(1994) **A Modified Hausdorff Distance for Object Matching**, Proceeding of the 12th international conference on pattern recognition, 9-13 Oct.1994, Jerusalem, Israel, vol.1, pp. 566-568

Xue Gao; Lallican, P.M.; Viard-Gaudin, C.; (2005) **A Two-stage Online Handwritten Chinese Character Segmentation Algorithm Based on Dynamic Programming**, Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on, 29 Aug.-1 Sept. 2005, vol.2, pp:735 - 739

P. Gastaldo, R. Zunino, (2002) **Hausdorff Distance for Target Detection,** 2002 IEEE International Symposium on Circuits and Systems, Scottsdale AZ, vol.5, pp. 26-29

Y. Han; D. Xu; (2004) **SketchEditor: a freehand pen editor**, Signal Processing, 2004. Proceedings. ICSP '04. 2004 7th International Conference on, August 2004, vol.2, pp:1344 – 1347

S. Hanaki, T. Yamazaki, (1980) **On-line Recognition of Handprinted Kanji Characters**, Pattern recognition, vol.12, pp.421-4219

J.H.Horng, J.T.Li (2001) **A Dynamic Programming Approach for Fitting Digital Planar Curves with Line Segments and Circular Arcs**, Pattern Recognition Letters, vol.22, No.1, pp.183-197

J.W. Hsieh, J.M. Chen, C.H. Chuang, K.C. Fan, (2004) **Novel Aircraft Type Recognition with Learning Capabilities in Satellite Images**, International Conference on Image Processing, ICIP '04. 24-27 Oct. 2004, Singapore, vol. 3, pp:1715 – 1718

S.M.Htwe, C. Higgins, C.G. Leedham, Y.Ma, (2004) **Evaluation of Feature Sets in the Post Processing of Handwritten Pitman's Shorthand**, 9th International Workshop on Frontiers in Handwriting Recognition, Tokyo,Japan, 2004, pp.359-364

J.Y.Hu, M.K.Brown, W.Turin (1996) **HMM Based On-line Handwriting Recognition**, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol.8, No.10, pp.1039-1045

S.C.Huang,Y.N.Sun (1999), **Polygonal Approximation Using Genetic Algorithms**, Pattern Recognition, vol.32, pp.1409-1420

D.P. Huttenlocher, G.A. Klanderman, W.J. Rucklidge, (1993) **Comparing Images Using the Hausdorff Distance**, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol.15, pp. 850 –863

J.M.Inesta, M.Buendi, M.A.Sarti (1998) **Reliable Polygonal Approximation of Imaged Real Objects Through Dominant Point Detection**, Pattern Recognition, vol. 31, No.6, pp.685-697

B.H.Jun, M.Y.Kim, C.S.Kim, W.S.Kim, J.Kim (1995) **On-Line Cursive Korean Character Recognition by Using Curvature Models**, Proceedings of the 3rd International Conference on Document Analysis and Recognition, 14-16 August, Montreal, Que., Canada, pp.1051-1054

K. Jung, H.J.Kim (2000) **On-line Recognition of Cursive Korean Characters Using Graph Representation**, Pattern Recognition, vol. 33, No.3, pp.399-412

C.H. Kao, H.S, Don, (2005) **Skew Detection of Document Images Using Line Structural Information**, Third International Conference on Information Technology and Applications, 04-07 July 2005, Sydney, Australia, vol. 1, pp: 704 – 715

H.Kang, H.J.Kim (2000) **Design of An Interface on PDA For Korean**, IEEE Transaction On Consumer Electronics, vol. 46, No.3, pp.834-838

N. Karssemeijer, G.M.T. Brake, (1996) **Detection of Stellate Distortions in Mammograms**, IEEE Transaction on Medical Imaging vol.15, No.5, pp: 611–619.

S.K.Kim, J.W.Kim, H.J.Kim (1996) **On-line Recognition of Cursive Korean Characters Using Neural Networks**, Neurocomputing, Vol. 10, No.3, pp.291-305

S.K.Kim, S.M.Park, J.K.Lee, H.J.Kim (1998) **On-line Recognition of Korean Characters Using ART Neural Network and Hidden Markov Model**, Journal of Systems Architecture vol.44, No.12, pp.971-984

J.Koplowitz, S.Plante (1995) **Corner Detection For Chain Coded Curves,** Pattern Recognition, vol. 28, No.6, pp.843-852

Kornai,A (1997) **An Experimental HMM-based Postal OCR System**, 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing, 21-24 April 1997, Munich, Germany, vol.4, pp: 3177 -3180

O.S.Kwon, Y.B.Kwon (1993) **An On-Line Recognition of Hangul Handwriting Using Dynamic Generation of Line Segments**, Proceeding of The 20th KISS Spring Conference, Korea Information Science Society, vol.1, pp.151-154

J.O.Kwon, B.Sin, J.H.Kim (1997) **Recognition of On-line Cursive Korean Characters Combining Statistical And Structural Methods**, Pattern Recognition, vol.30, No.8, pp.1255-1263

J.S.Lee, O.J.Kwon, S.Y.Bang (1999) **Highly Accurate Recognition of Printed Korean Characters Through an Improved Two-stage Classification Method**, Pattern Recognition, vol. 32, No.12, pp.1935-1945

C.G.Leedham (1984), **Computer Acquisition and Recognition of Pitman's Handwritten Shorthand**, PhD Thesis, Department of Electronics, University of Southampton

C.G. Leedham and A.C. Downton, (1986) **On-line Recognition of Pitman's Shorthand: An Evaluation of Potential**, Int. J. Man-Machine Studies, Vol.24, pp. 375-393

C.G. Leedham, A.C. Downton, (1987) **Automatic Recognition and Transcription of Pitman's Handwritten Shorthand---An Approach to Shortforms**, Pattern Recognition, vol.20, pp.341-348.

C.G.Leedham, Y.Qiao, (1992) **High Speed Text Input to Computer Using Handwriting,** Instructional Science, vol.21, pp.209-221

B. Lemarie, (1993) **Practical Implementation of A Radial Basis Function Network for Handwritten Digit Recognition,** Proceedings of the Second International Conference on Document Analysis and Recognition, 20-22 Oct. 1993, Tsukuba City, Japan, vol.1, pp: 412 – 415

J. R. Lewis. (1999) **Input rates and user preference for three small-screen input methods: Standard keyboard, predictive keyboard, and handwriting.** Proceedings of the Human Factors and Ergonomics Society 43rd Annual Meeting - HFES '99, 425-428. Santa Monica, CA: Human Factors and Ergonomics Society.

X.L.Li, D.Y Yeung, (1997) **On-line Handwritten Alphanumeric Character Recognition Using Feature Sequence**, Pattern Recognition, vol.30, no.1, pp.31-44

R.Q. Liao, (1985) **Rules of Renqun shorthand**, Guangdong Publishing Ltd, China

K.H. Lim (2000) **Extraction of Dynamic Pen-Stroke Information**, Final Year Project, School of Applied Science, Nanyang Technological University

Y.Ma, G.Leedham, C.Higgins, S.M.Htwe, (2004) **Segmentation and Recognition of Vocalized Outline in Pitman Shorthand**, 17th International Conference on Pattern Recognition, 23-26 August, 2004, Cambridge, UK, vol.1, pp.441-444

S. Manke, U. Bodenhausen, (1994) **A Connectionist Recognizer for On-line Cursive Handwriting Recognition,** IEEE International Conference on Acoustics, Speech, and Signal Processing, April 19-22 1994, Adelaide, South Australia, vol.1, pp: 633 – 636

S. Manke, M. Finke, A. Waibel, (1995) **NPen: A Writer Independent, Large Vocabulary Online Cursive Handwriting Recognition System**, Proceedings of the Third International Conference on Document Analysis and Recognition, 14-16 Aug. 1995, Montreal, Canada, vol.1, pp.403 – 408

N. Matic, I. Guyon, J. Denker, V. Vapnik, (1993) **Writer-adaptation for On-line Handwritten Character Recognition,** Proceedings of the Second International Conference on Document Analysis and Recognition, 20-22 Oct. 1993, Tsukuba City, Japan, vol.1, pp: 187-191

Masayoshi Okamoto and Kazuhiko Yamamoto, (1999) **On-line Handwriting Character Recognition Using Direction-change Features that Consider Imaginary Strokes**, Pattern Recognition, vol. 32, No. 7, pp.1115-1128

A.Nair, C.G.Leedham, (1991) **Preprocessing of Line Codes for Online Recognition Purposes**, Electronics Letters, vol.27, No.1, pp.1-2

A. Nair, C.G. Leedham, (1992) **Evaluation of Dynamic Programming Algorithms for the Recognition of Shortforms in Pitman's Shorthand**, Pattern Recognition Letters, vol.13, pp.605-612

H. Nishimura, T. Timikawa, (2003) **Offline character recognition using online character writing information**, Proceedings of the Seventh International Conference on Document Analysis and Recognition, 3-6 Aug. 2003, Edinburgh, Scotland, vol.1, pp: 168 - 172

C.R.Nohl, C.J.C.Burges, J.I.Ben, (1992) **Character-based handwritten address word recognition with lexicon**, U.S. Postal Service 5th Adv. Technol. Conf., Washington, DC, November,1992, Vol.3, pp.167-182

S.H.Paek, Y.S.Hwang, S.Y.Bang (1993) **On-Line Korean Character Recognition by Using Two Types of Neural Networks**, Proceedings of 1993 International Joint Conference on Neural Networks, 25-29 Oct, Nagoya, Japan, vol.3, pp.2113-2116

S.C.Pei, J.H.Horng (1996) **Optimum Approximation of Digital Planar Curves Using Circular Arcs**, Pattern Recognition, vol. 29, No.3, pp.383-388

M.E.Petersen, W.R.M.Dassen, J.H.C. Reiber, (1999) **Sequential Selection of Discrete Features for Neural Networks –A Bayesian Approach to Building A Cascade**, Pattern Recognition Letter, vol.20, pp.1439-1448

M.E.Petersen, D.De Ridder, H.Handels, (2002) **Image Processing with Neural Networks - A Review**, Pattern Recognition, vol. 35, pp.2279-2301

A.Pikaz,A.Averbuch (1996) **On Automatic Threshold Selection for Polygonal Approximation of Digital Curves**, Pattern Recognition, vol.29,No.11, pp.1835-1845

J.A.Pittman (1992) **Recognizing Handwritten Text**, MCC Report, Austin

Pitman (1980), **Pitman 2000 shorthand—First Dictionary**, Pitman Publishing Ltd, ISBN 0 273 00885 4

Pitman(1989), **Shorthand for today**. Pitman Publishing Ltd., ISBN 0 7299 0106 8

R.Plamondon, S.N.Srihari, (2000) **On-line and Off-line Handwriting Recognition: A Comprehensive Survey**, IEEE Trans. On Pattern Analysis and Machine Intelligence, vol.22, No.1, pp. 63-84

Y. Qiao, C.G. Leedham, (1993) **Segmentation and Recognition of Handwritten Pitman Shorthand Outlines Using An Interactive Heuristic Search**, Pattern Recognition, vol.26, pp. 433-441

L.R.Rabiner (1989) **A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition**, Proc. of the IEEE, Vol.77, No.2, pp.257-286

A.Rattarangsi, R.T.Chin (1992) **Scale-Based Detection of Corners of Planar Curves**, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol.14, No.4, pp.430-449

B.Ray, K.S.Ray (1992) **An algorithm for detection of dominant points and polygonal approximation of digitized curves**, Pattern Recognition Letters, vol. 13, pp.849-856

B.K.Ray, K.S.Ray (1994) **A Non-Parametric Sequential Method For Polygonal Approximation of Digital Curves**, Pattern Recognition, vol.15, pp.161-167

B.K.Ray, K.S.Ray (1995) **A New Split-And-Merge Technique for Polygonal Approximation of Chain Coded Curves**, Pattern Recognition Letters, vol.16, No.2, pp.161-169

A. S. Rosenthal, J. Hu, M. K. Brown, (1997) **Size and orientation normalization of on-line handwriting using Hough transform**, IEEE International Conference on Acoustics, Speech, and Signal Processing, 21-24 April 1997, Munich, Germany, Vol. 4, pp: 3077 - 3080

D.Sankoff, J.B.Kruskal, (1983) **Time Warps, String Edits and Macromolecules: the Theory and Practice of Sequence Comparison**, Addison-Wesley

M. P. Schambach (2003) **Model Length Adaptation of an HMM based Cursive Word Recognition System,** Proceedings of Seventh International Conference on Document Analysis and Recognition, Aug.3-6,2003, Edinburgh, Scottland, UK, vol. 1, No. 1, pp. 109-113

M.Schenkel, I. Guyon, D. Henderson (1995) **On-line Cursive Script Recognition Using Time-delay Neural Networks and Hidden Markov Models**, Machine Vision and Applications, Vol. 8, No. 4, pp.215-223

J.P. Shin (2001) **Online Cursive Hangul character Recognition Based on Dynamic Programming**, Sixth International Conference on Document Analysis and Recognition, 10-13 Sep, Seattle, WA, USA, pp.810-814

De Stefano, M. Garruto, A. Marcelli, (2004) **A Multiresolution Approach to On-line Handwriting Segmentation and Feature Extraction**, Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, vol. 2, 23-26 Aug. 2004, pp:614 - 617

Y.M. Su, J.F. Wang, (2004) **Decomposing Chinese Characters into Stroke Segments Using SOGD Filters and Orientation Normalization**, ICPR 2004. Proceedings of the 17th International Conference on Pattern Recognition, 23-26 Aug. 2004, Cambridge, UK, vol.2, pp: 351 – 354

J. Subrahmonia, T. Zimmerman, **Pen Computing: Challenges and Applications**, Proceedings. 15th International Conference on Pattern Recognition (ICPR) Vol. 2, 3-7 Sept 2000, Barcelona, Spain, pp:60 - 66

C. C. Tappert, C. Y. Suen, T. Wakahara, (1988) **On-line Handwriting Recognition - A Survey**, 9th International Conference on Pattern Recognition, 14-17 Nov. 1988, Rome, Italy, vol. 2, pp. 1123-1131

C.H.Teh,R.T.Chin (1989) **On the Detection of Dominant Points on Digital Curves**, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol.11, no.8, pp.859-872

J.Tokuno, N.Inami,S.Matsuda (2002) **Context-Dependent Substroke Model for HMM-Based On-line Handwriting Recognition**, Proceeding of the Eighth International Workshop on Frontiers in Handwriting Recognition(IWFHR'02), August 6-8, 2002, Niagara-on-the-Lake, Ontario, Canada, vol.1, pp.78-83

K. Toyokawa, K. Kitamura, S. Katoh, H. Kaneko, N. Itoh, M. Fujita, **An On-line Character Recognition System for Effective Japanese Input,** Proceedings of the Second International Conference on Document Analysis and Recognition, 20-22 Oct. 1993, Tsukuba City, Japan, vol.1, pp: 208–213

S. Wang, C. Shu, S.W. Xia, (1996) **A Hybrid Approach to Unconstrained Handwritten Numerals Recognition,** 3rd International Conference on Signal Processing, 14-18 Oct. 1996, Beijing, China, vol.2, pp: 1334 – 1337

H.H.Wellisch (1978) **The Conversion of Scripts—Its Nature, History and Utilization**, John Wiley and Sons

W.Y.Wu, M.J.Wang (1993) **Detecting the Dominant Points by the Curvature-Based Polygonal Approximation**, Graphical Models and Image Processing (GVGIP), vol.55, No.2, pp.79-88

Y.Xiao, J.J.Zou, H.Yan (2001) **An Adaptive Split-And-Merge Method for Binary Image Contour Data Compression**, Pattern Recognition Letters, vol. 22, No. 3, pp.299-307

K. Yamada, H. Kami, J. Tsukumo, T. Temma, (1989) **Handwritten Numeral Recognition by Multilayered Neural Network with Improved Learning Algorithm,** International Joint Conference on Neural Networks (IJCNN), 18-22 June 1989, Washington, USA, vol.1, pp: 259-266

J. Yi, C. Lee, J. Kim, (2005) **Online Signature Verification Using Temporal Shift Estimated by the Phase of Gabor Filter**, IEEE Transactions on Signal Processing, vol. 53, Issue 2, Part 2, pp: 776 – 783

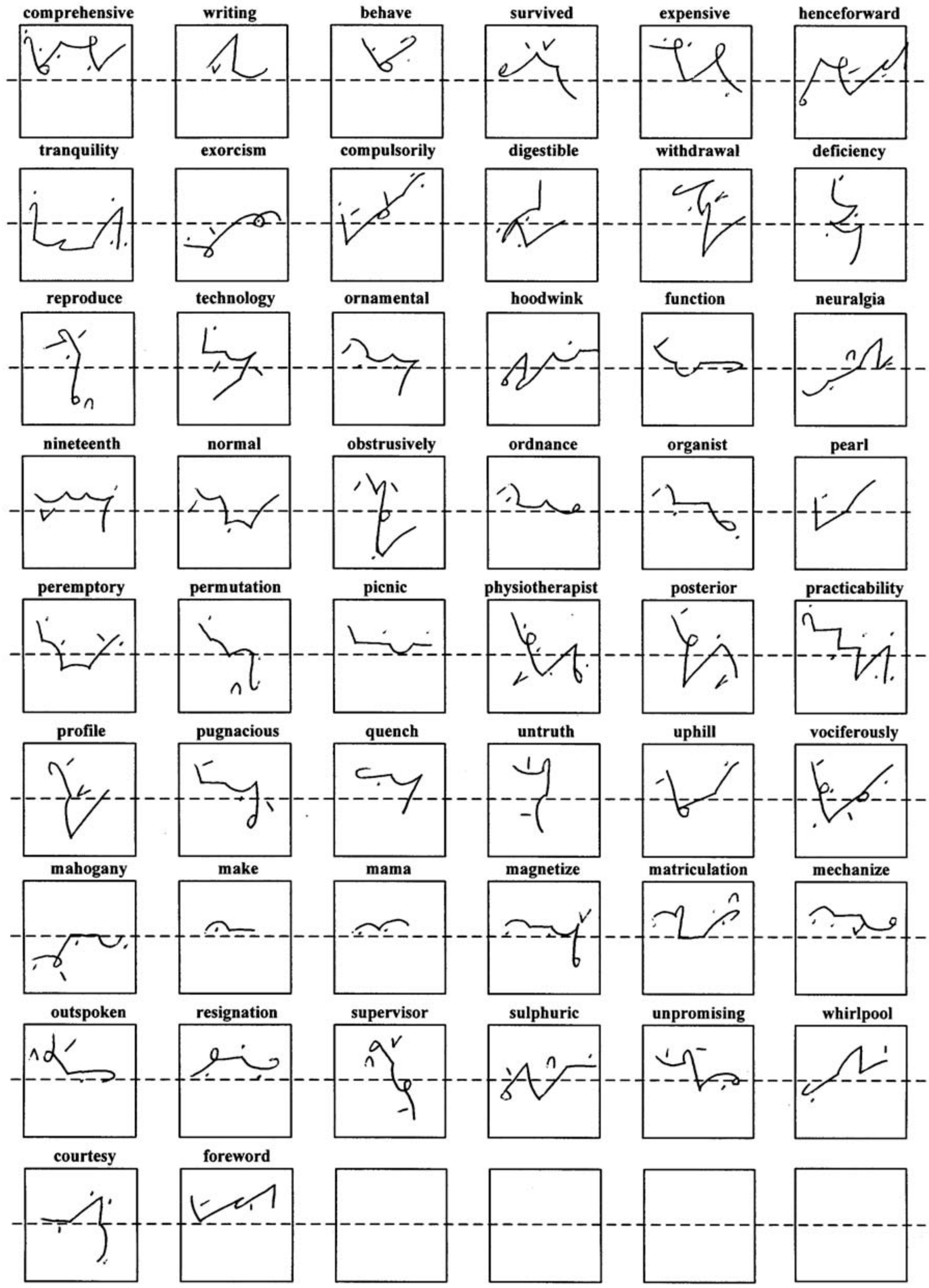
P.Y.Yin (1998) **A New Method For Polygonal Approximation Using Genetic Algorithm**, Pattern Recognition Letter, vol.19, pp.1017-1026

R. W. Zmud, (1984) **As Examintion of "pushing-pull" Theory Applied to Process Innovation in Knowledge Work**, Management Science, vol. 30, pp.727-738

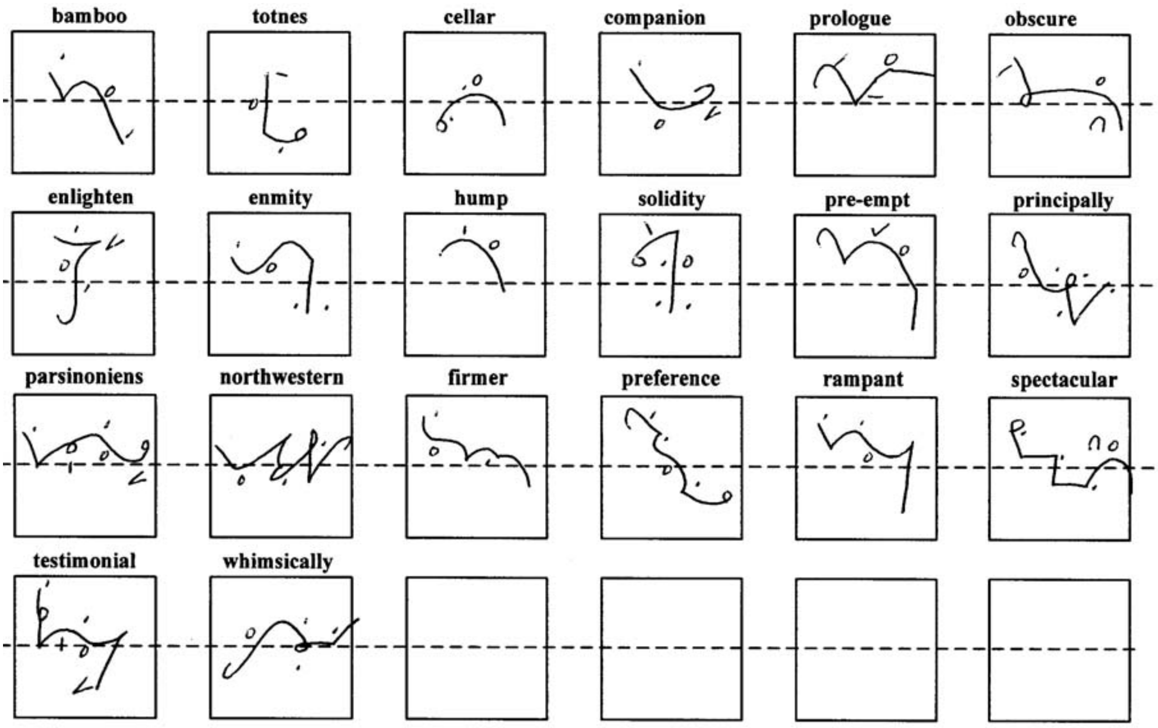
M. Zhu, Z. Chi, X..P. Wang, (2002) **Segmentation and Recognition of On-line Pitman Shorthand Outlines Using Neural Network**, Proc. International Conference on Neural Information Processing, Singapore, 18-22 Nov. 5, pp. 2454-2458

Appendix A

1. Vocalized outlines used in the first experimentation



2. Vocalized outline used in the second experimentation



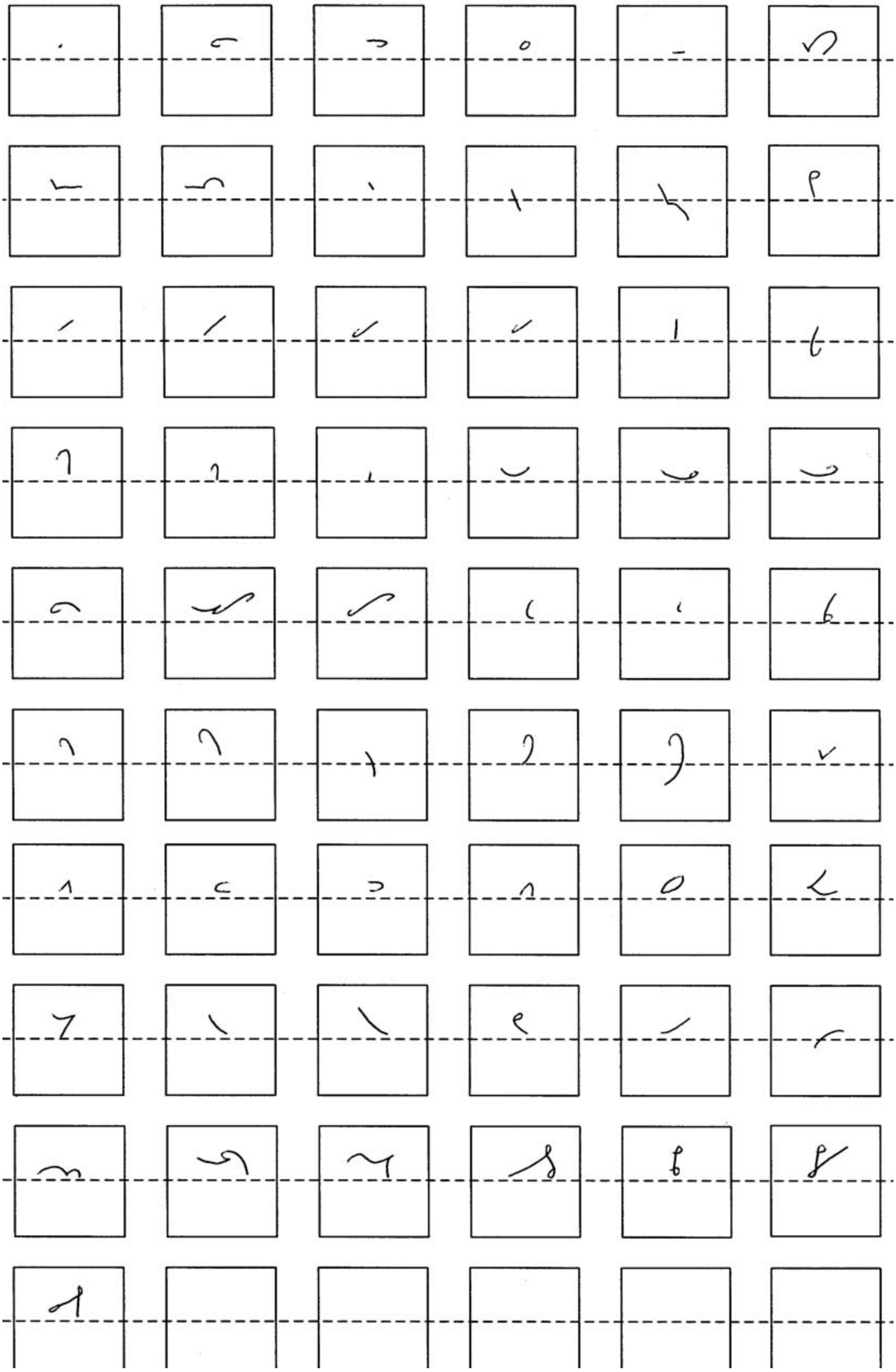
Appendix B

Shortforms in Pitman 2000

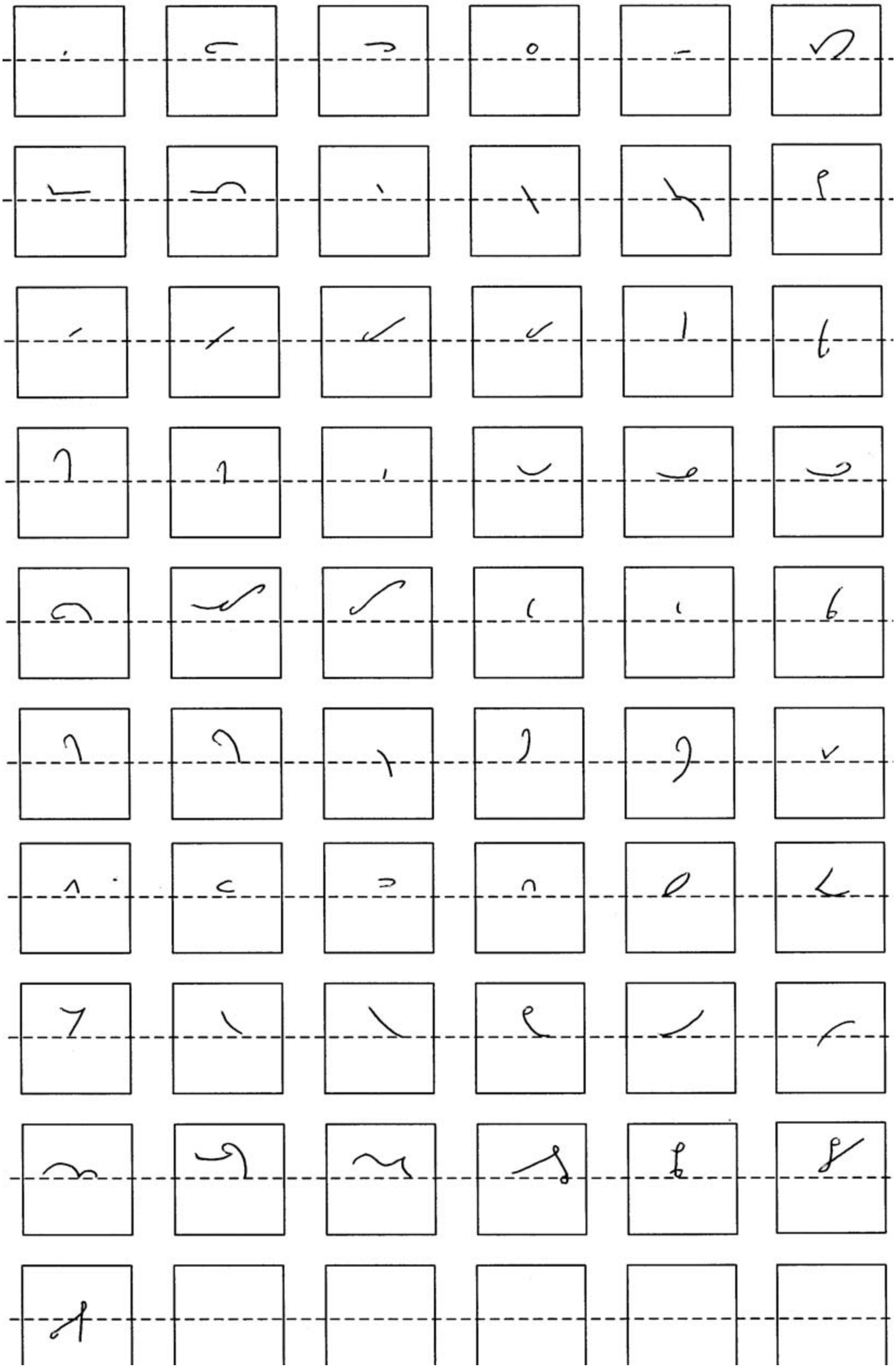
No.	Word	Shortform	No.	word	shortform
1	a/an, the		2	accord/according (to)	
3	can not		4	as/has, is/his	
5	Could		6	Also	
7	altogether, together		8	commercial/ly	
9	all/of, too/to		10	put/to be, be	
11	Before		12	subject	
13	and/ought, should/who		14	are/which, hour/our	
15	We		16	gentleman/gentlemen	
17	do/it, dollar/has		18	difficult	
19	Dear		20	trade/toward	
21	but/he, o/oh/owe/on		22	Any/in, thing	
23	Influence		24	information	
25	More		26	anyone	
27	Wonderful/ly		28	thank/think	
29	that/without		30	This	
31	Particular		32	from/very	
33	year, your		34	their/there	
35	Therefore		36	Eye/I	
37	How		38	With	
39	Would		40	You	
41	First		42	January	
43	Knowledge		44	For	
45	Have		46	several	
47	Shall		48	Will	
49	immediate		50	influential	
51	manufacture		52	responsible/ibility	
53	satifaction		54	satisfactory	
55	yesterday				

Appendix C
Shortforms Samples

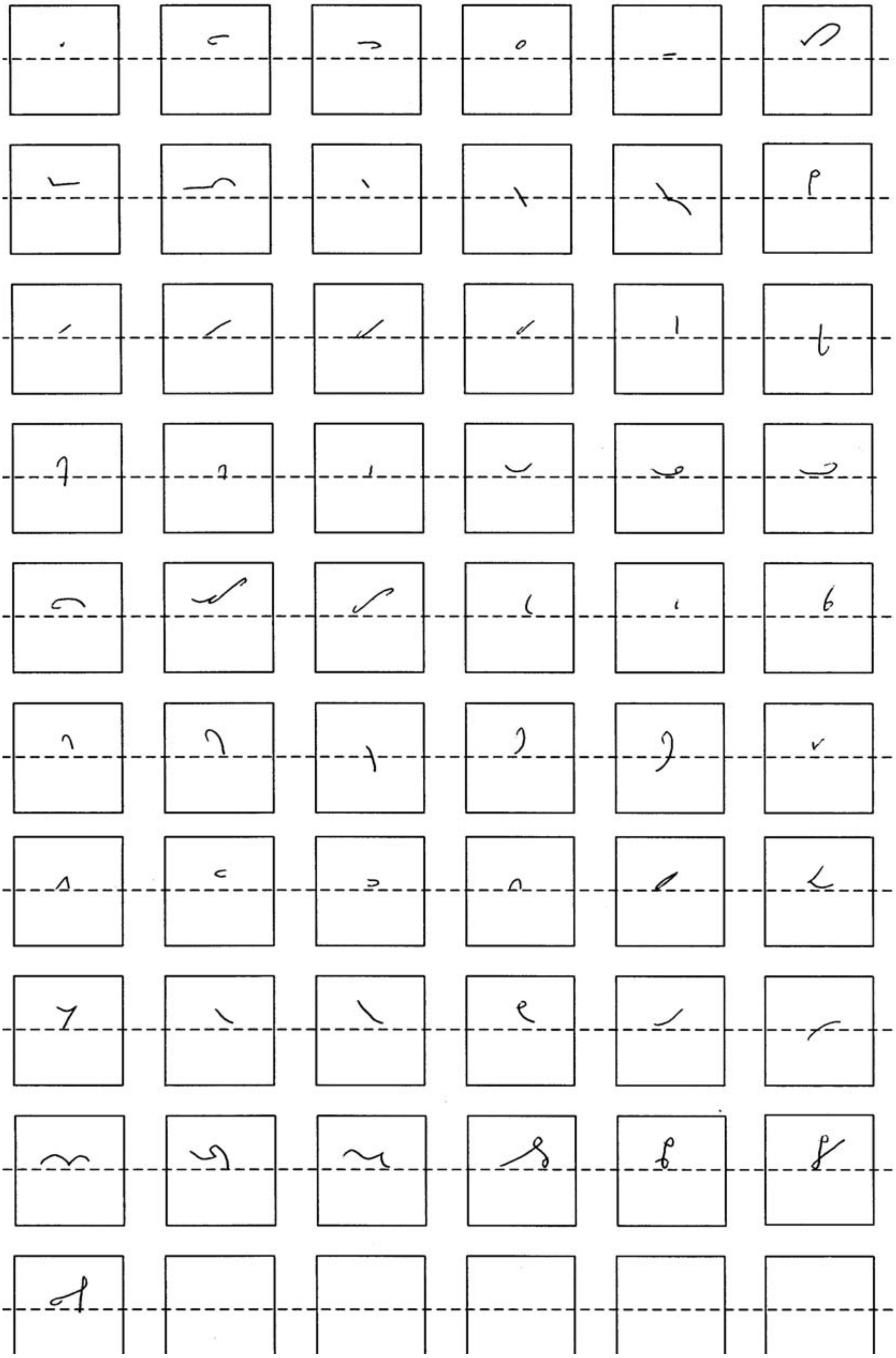
1. Standard template A_T

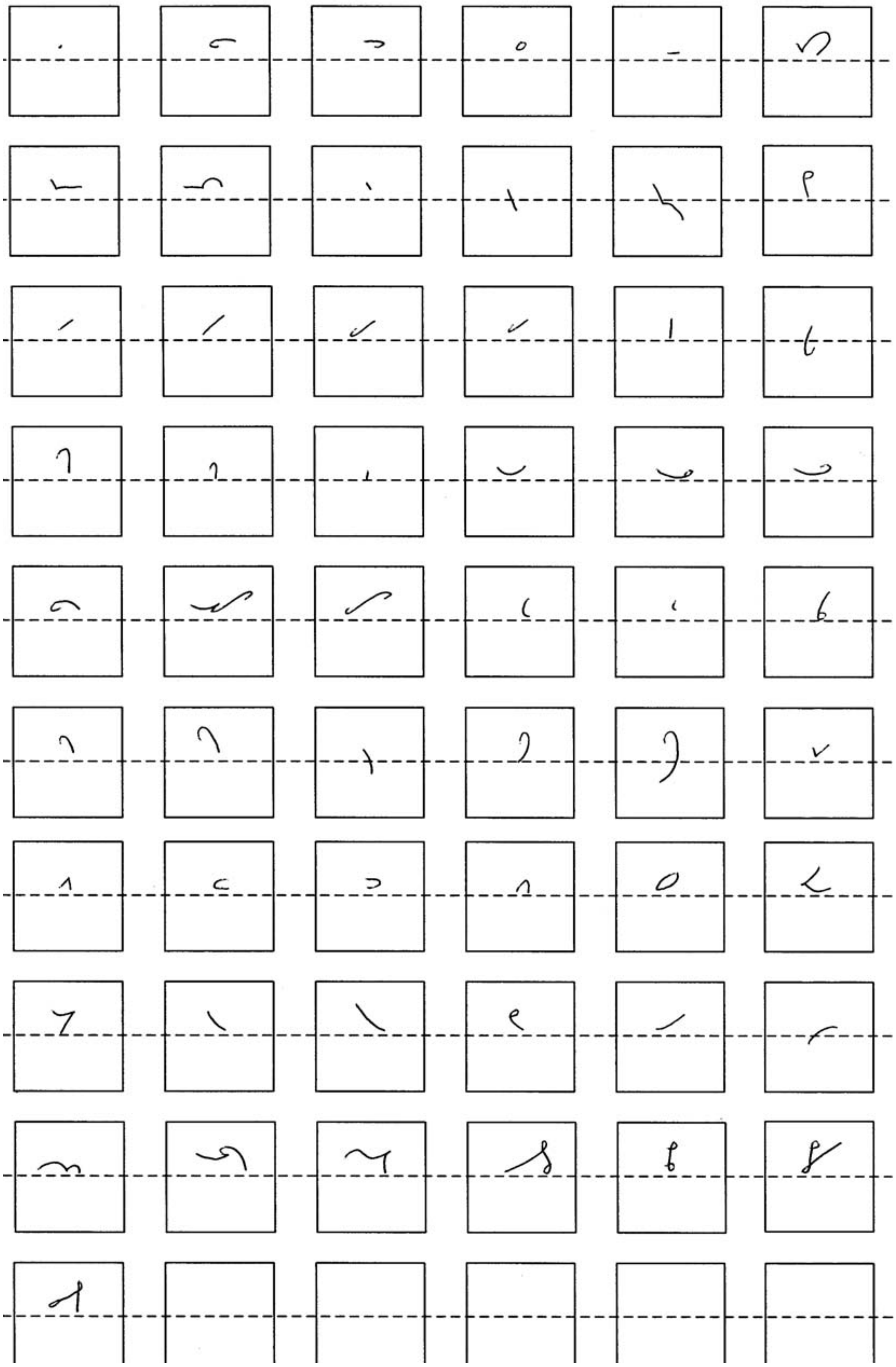


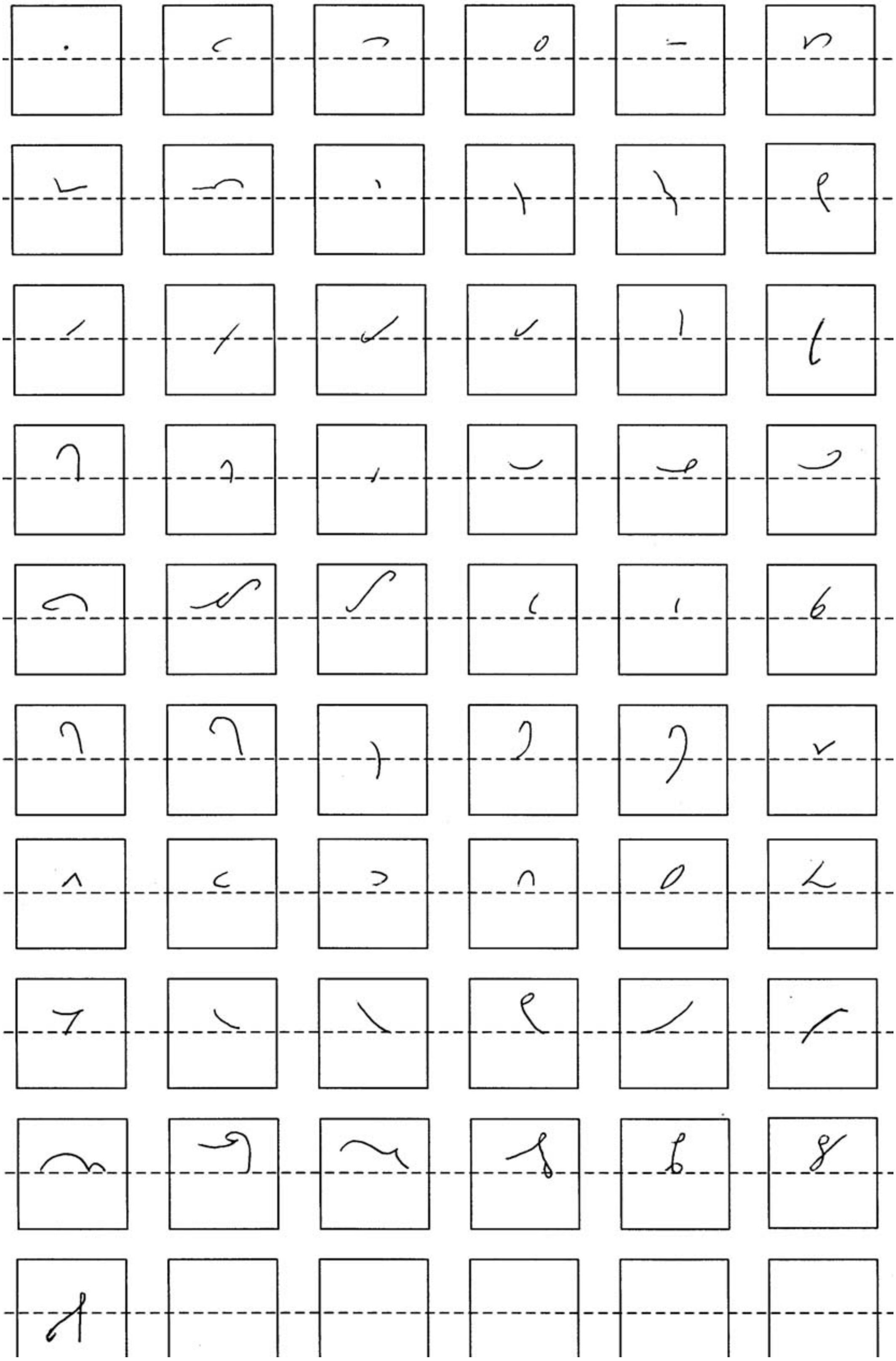
2. Standard template B_T



3. Samples written by different writer A, B and C

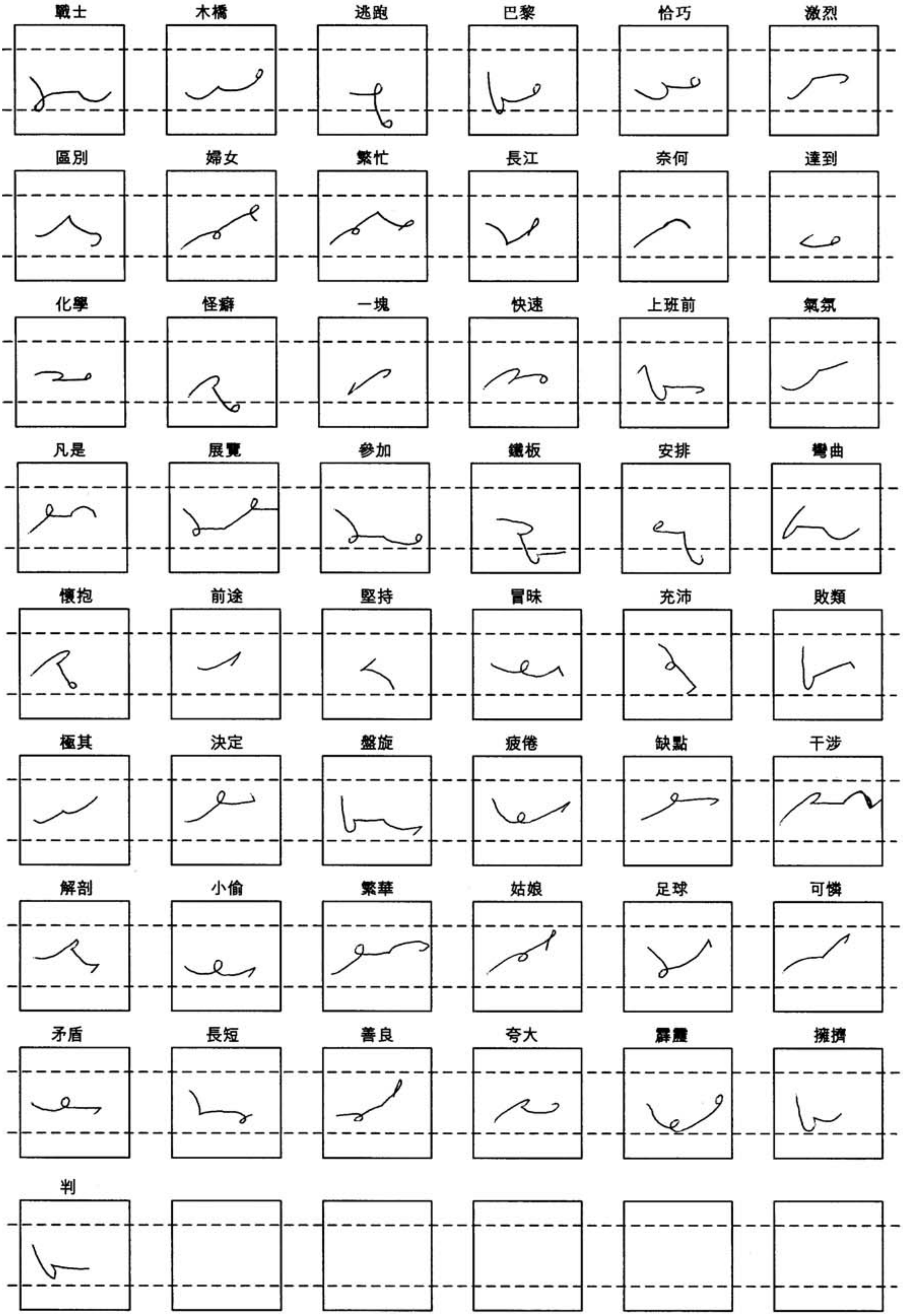






Appendix D

Renqun outlines used in the evaluation



Appendix E

Samples used in the evaluation on Table PC