A Study on Character-Position-Free On-line Handwritten Japanese and Chinese Text Recognition

文字位置自由オンライン手書き日本語・中国語文字列認識に関する研究

Jianjuan Liang

Supervisor: Prof. Masaki Nakagawa

Department of Computer and Information Sciences

Tokyo University of Agriculture and Technology

2016

Abstract

Recently, with the development and growing popularity of pen-based or touch-based input devices, such as smart phones and tablet PCs, on-line handwritten text recognition has been receiving large attention, especially for unconstrained text recognition. This trend is spreading into automobiles where drivers may input destinations to navigation systems by speech or written characters. In this situation, however, they often have to write characters without supported by wrist or elbow and without visual feedback. Moreover, they may stop writing while writing destinations to keep safe driving, namely, a text may be written in one or more steps, so that positions of characters or even strokes (a trajectory written from pen/finger down to up) become unstable. In this thesis, to develop an character-position-free on-line handwritten Japanese and Chinese text recognizers, we firstly collect handwritten text patterns written without supported by wrist or elbow and without visual feedback, and make models to produce such text patterns from normally handwritten text patterns extracted from the TUAT Kondate Japanese database and CASIA-OLHWDB2.1Chinese database, as well as the model for text patterns with characters completely overlaid. Then, we consider recognition methods which can recognize handwritten Japanese and Chinese text patterns produced from all the models.

To our best knowledge, it is common to use integrated segmentation and recognition method for normally handwritten Japanese/Chinese text recognition, to solve the character segmentation problem. It is more challenge for character-position-free handwritten text due to the fact that spaces between characters are very unstable. We considered two segmentation methods to solve it. One classifies each off-stroke between real strokes into a non-segmentation point, a segmentation point, and an undecided point according to the output of SVM model, we call it "candidate segmentation method". The other sets each off-stroke as an undecided point, we call it "undecided segmentation method". Both two segmentation methods evaluate the segmentation probability by SVM model. Then, the optimal segmentation-recognition path can be effectively found by Viterbi search in the candidate lattice, combining the scores of on-line and off-line character recognition, geometric context, linguistic context, as well as the segmentation scores by SVM. We test these two methods on generated character-position-free Japanese and Chinese sample patterns, as well as on collected handwritten Japanese text patterns. The results of experiments confirm that the undecided segmentation method yields better recognition rate than the candidate segmentation method, approaching the performance of the latest recognizer on normally handwritten horizontal Japanese and Chinese text patterns with no serious speed

restriction in practical applications.

In chapter 1, we briefly describe the background and the objective of this study. Then, we introduce the organization of this thesis.

In chapter 2, we mainly give a survey on the state-of-the-art methods for on-line handwritten text recognition, and the overlaid handwriting text recognition used on mobile phones with small surface. Then, we introduce current text input situation in the driving context, and character-position-free handwritten text recognition.

In chapter 3, we firstly describe collected handwritten text patterns without supported by wrist or elbow and without visual feedback. Then, we describe the normally handwritten text patterns in Kondate database and CASIA-OLHWDB database. Finally, we introduce 4 models to generate character-position-free Japanese and Chinese text patterns using normally handwritten horizontal Japanese and Chinese text patterns, respectively.

In chapter 4, we briefly describe the character recognition system combining on-line and off-line character recognizers, for each candidate character pattern in the candidate lattice.

In chapter 5, we describe the recognition methods for character-position-free on-line handwritten Japanese text recognition.

In chapter 6, we describe the linguistic context and geometric context for the path evaluation criterion to improve the text recognition accuracy.

In chapter 7, we describe experiments on generated character-position-free on-line handwritten Japanese and Chinese text datasets, and collected handwritten text patterns, as well as normally handwritten Japanese text patterns. Moreover, we compare the results of the proposed segmentation methods with the original recognizer, and give some analyses of recognition performance.

In chapter 8, we conclude this research and give several directions for the future work.

論文要旨

近年，スマートフォンやタブレット PC などのタッチ入力やペン入力端末の爆発的普及に伴い，オンライン手書き文字列認識に再び注目が集まっている．特に，制限無し，自由に手書き文字列認識に関心が高まっている．さらに，車載情報環境として，大型のタッチパネルが開発され販売されるようになっている．そこでは，行先を指書きし，認識させるなどが考えられる．この場合，通常のタブレット環境のように筆跡を確認しながら筆記することは期待できない，つまり，肘をつけずに，かつ，筆跡を確認しない．その結果,文字位置は非常に不安定になることを想定しなければならない.本研究では,こうした条件での利用を想定し,文字位置制限を緩和あるいは解除して手書き文字列を認識する方式を述べる．しかし，現状では，運転しながら車載タッチパネルに筆記することができないので，まず，研究室の学生らから，車載環境のように手や肘を付けず,目視による筆跡の確認なしにタブレット PC で筆記してもらった．そして，収集したデータに模して，従来のオンライン手書き文字列パターンデータベース（日本語の TUAT Kondate と中国語の CASIA-OLHWDB2.1）から文字位置自由オンライン手書き日本語・中国語文字列パターンを生成した.生成したパターンには,収集したものに近いパターン,より厳しく重なるパターン，そして，小さい筆記面に文字を完全に重ねて書かれるパターンも含めることにした.

オンライン手書き日本語/中国語文字列認識においては，文字ごとへの切出し問題を克服するために，切出しと認識の統合手法がよく利用されている．しかし，文字位置自由オンライン手書き文字列に対して，文字間隔が非常に不安定であるので，既存の仮切出し手法は利用できない．本研究では，2 つの手法を検討する．第一の手法は，各ストロークの切れ目に対する SVM の出力値を 2 つの閾値と比べて,それぞれを非切出しポイントと不確定ポイント，そして切出しポイントに分類する．非切出しポイントと切出しポイントは確定し，不確定ポイントでは，従来と同様に両方の可能性を検討する．この手法を切出し候補手法と呼ぶ．第二の手法では，各ストロークの切れ目をすべて不確定ポイントとする．これを不確定手法と呼ぶ．そして，両者に共通して，各ストロークに対する SVM の出力値から切出しの確からしさを求め，文字列候補パスの評価に利用する．候補ラティスから，Viterbi 手法で最適な文字列認識を探索する際には，文字認識の確からしさや幾何学的文脈および言語的文脈の確からしさに加えて，SVM モデルによる各文字の切出し確からしさも評価する.生成した文字位置自由オンライン手書き日本語・中国語文字列データベースと収集したデータで評価実験を行い，不確定手法は，切出し候補手法より高い認識率を達成し，かつ，通常の手書き文字列に近い精度で認識できることと認識速度も実用上問題がないことを確認した.

第 1 章「緒論」では，本研究の研究背景と研究目的について述べる．そして，本論

文の構成について述べる．

第２章「最新動向」では，本研究に関連する最新動向について述べる．つまり，従来のオンライン手書き文字列認識技術，オンライン重ね書き文字列認識技術，車載情報環境での手書き文字列入力の現状，及び文字位置自由手書き文字列認識の各研究分野について先行研究を紹介し，本研究の位置づけを明らかにする．

第３章「文字位置自由オンライン手書き文字列データベース」では，手や肘を付けず，目視による筆跡の確認なしにタブレット PC に筆記してもらった手書き文字列データパターンについて述べる．次に，既存の Kondate と CASIA-OLHWDB2.1 データベースを紹介する．そして既存の手書き文字列パターンから，文字位置自由手書き日本語・中国語文字列を生成する方法について述べる．

第４章「文字認識システム」では，デジタルペンなどから入力された時間情報を有するオンライン手書き文字パターンを処理する手書き文字認識システムについて述べる．ここでは，オンライン文字認識手法とオフライン文字認識手法を統合している．まず，多字種文字認識の高速化ために大分類についての概要を述べる．次に，オンライン手書き文字認識とオフライン手書き文字認識についてぞれぞれの概要，及びそれを構築する各構成要素を述べる．そして，統合手法を述べる．

第５章「文字位置自由オンライン手書き文字列認識」では，2 つの切出し手法に基づく文字位置自由オンライン手書き文字列認識手法について述べる．ここでは，文字列パターンに対するサポートベクトルマシン（SVM）を用いたストロークの分類方法，候補ラティスの生成方法，及び確率モデルによる文字列候補パスの評価基準とパラメーターの最適化について述べる．

第６章「言語の文脈処理と幾何学的な文脈処理」では，文字列認識精度を向上するために重要な評価要素技術である言語の文脈と幾何学的な文脈処理について述べる．

第７章「実験」では，評価実験の設定と実験結果，実験結果の分析について述べる．つまり，生成した文字位置自由オンライン手書き日本語・中国語文字列データベースと収集したパターン，普通の手書き日本語文字列で，日本語の場合は，提案した 2 つの切出し手法と従来のオンライン手書き日本語文字列認識手法との認識性能を比較した．中国語の場合は，提案した 2 つの切出し手法の認識性能を比較した．

第８章「結言」では，本論文の成果をまとめた上で，今後の課題について述べる．

Acknowledgements

First, I would like to express my gratitude to **Prof. Masaki Nakagawa** for giving me an opportunity to study in his laboratory. Although he is extremely busy, he always found time to guide me in study even in weekend. His responsible attitude is deeply appreciated and will guide my life in future.

Second, I would like to deeply thank **Dr. Bilan Zhu** for guiding me both in my study and in my life. She always gives me good advice when I have problem. She helped me to know well online recognizer for handwritten Japanese text.

Further, I would like to thank all of my teachers who assisted me with my study, and express my deepest gratitude to the EPSON international scholarship foundation for supporting me in the last year of my study.

I would also like to thank all my friends and laboratory-mates, especially Ms. Yuechan Hao, for their help and all the supports they provided.

Finally, I am indebted to my parents for their love, patience and understanding. A special thanks to my husband, for his love, encouragement and support. His support is priceless to the completion of my study. I would also like to thank my son for giving me so many happiness.

Content

1. Introduction

In this chapter, we describe the background and objective of this study. We also introduce the organization of this thesis.

1.1 Background

The research on on-line handwriting recognition began in the 1960s, and has been receiving intensive interest in the 1980s [1]. Most recognition systems, however, have writing constraint that characters are written in boxes to provide character segmentation prior to recognition. It is not natural and uncomfortable for people, due to people must pause every time they write one character.

In recent years, with the development and proliferation of pen-based and touch-based input devices, such as smart phones, tablet PCs, Anoto-pen [3], and electronic whiteboards and so on. These devices improve the precision of capturing the trajectories of pen tip movements and provide a comfortable writing interface, moreover, their writing area are getting larger than before. People tend to write text continuously without writing box constraint for recording information or communication. Therefore, on-line handwritten text recognition has been receiving large attention, especially for unconstrained text recognition.

This trend is spreading into automobiles with large surface for vehicle-mounted touch panels, where drivers may input destinations to navigation systems by speech or written characters. In this situation, however, they often have to write characters without supported by wrist or elbow and without visual feedback. Moreover, they may stop writing while writing destinations to keep safe driving, namely, a text may be written in one or more steps, so that positions of characters or even strokes (a trajectory written from pen/finger down to up) become unstable. Therefore, we need to develop a handwriting recognizer to recognize handwritten text patterns written under this environment. Except the special automobile environment, this recognition system also can be applied to the need of recognizing text written without visual feedback, such as writing memos for students in the class while listening and watching the blackboard, and recoding some information in the meeting.

Since we do not have actual on-line handwritten text patterns collected from vehicle-mounted touch panels while driving, we have collected a small amount of text patterns from students with the condition that they write specified short phrases on a tablet PC without supported by wrist or elbow and without confirming previously written strokes, namely, without visual feedback for the safety of driving, which meets the requirement for display system operation while a vehicle is in motion [4]. This is a simulated environment where a driver would write destinations under such the condition. Then, we make models to produce such text patterns as much as possible from normally handwritten text patterns, as well as the model for text patterns with characters completely overlaid, which provides a convenient input choice especially writing Kanji characters by finger. We call them character-position-free handwritten text patterns.

The character-position-free handwritten text recognition becomes more challenging than normal handwritten text due to the unstable position between characters, especially the loss of horizontal character shift in overlaid handwritten text. This thesis mainly focuses on resolving the character segmentation problem, and constructing a robust character-position-free on-line handwritten Japanese/Chinese text recognition system.

1.2 Objective

The research objective is to develop a handwriting recognizer based on the existing system [5], which can recognize character-position-free on-line handwritten Japanese/Chinese text patterns, and apply it into automobile environment.

The Japanese is a large character set language, which includes thousands of ideographic characters of Kanji, two sets of phonetic characters (Hiragana and Katakana), alphanumeric, and symbols. Most Kanji character patterns are composed of multiple subpatterns called radicals, which are shared among many Kanji character patterns. The Chinese also includes tens of thousands of traditional characters and simplified ones. In Chinese, About 5000 characters are frequently used in daily life [7].

Due to the large character set, the unstable character positions and the divergence of writing styles, it is a challenging research work to segment correctly characters before recognition as human for machine. To solve this problem, we follow the over-segmentation-based recognition framework as shown in Figure 1-1. In the over-segmentation stage, we consider two segmentation methods.

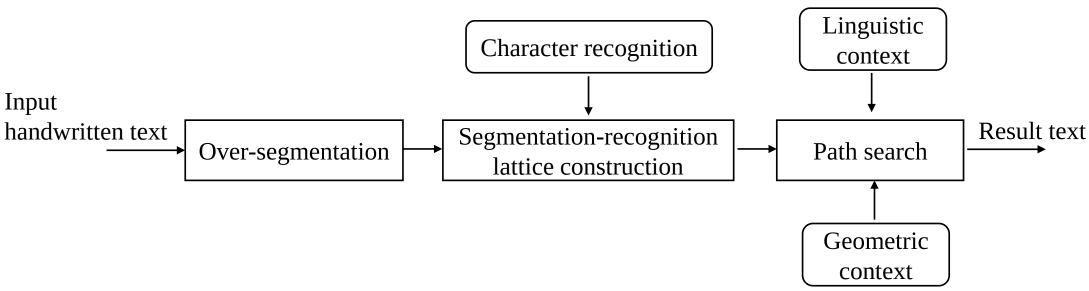


Figure 1-1 Flow chart of over-segmentation-based recognition.

1.3 Structure of this thesis

In chapter 1, we briefly describe the background and the objective of this study. Then, we introduce the organization of this thesis.

In chapter 2, we mainly give a survey on the state-of-the-art methods for on-line handwritten text recognition, and the overlaid handwriting text recognition used on mobile phones with small surface. Then, we introduce current text input situation in the driving context, and character-position-free handwritten text recognition.

In chapter 3, we firstly describe collected handwritten text patterns without supported by wrist or elbow and without visual feedback. Then, we describe the normally handwritten text patterns in Kondate database and CASIA-OLHWDB database. Finally, we introduce 4 models to generate character-position-free Japanese and Chinese text patterns using normally handwritten horizontal Japanese and Chinese text patterns, respectively.

In chapter 4, we briefly describe the character recognition system combining on-line and off-line character recognizers, for each candidate character pattern in the candidate lattice.

In chapter 5, we describe the recognition methods for character-position-free on-line handwritten Japanese text recognition.

In chapter 6, we describe the linguistic context and geometric context for the path evaluation criterion to improve the text recognition accuracy.

In chapter 7, we describe experiments on generated character-position-free on-line handwritten Japanese and Chinese text datasets, and collected handwritten text patterns, as well as normally handwritten Japanese text patterns. Moreover, we compare the results of the proposed segmentation methods with the original recognizer, and give some analyses of recognition performance.

In chapter 8, we conclude this research and give several directions for the future work.

2. State of the art

In this chapter, we mainly review the state-of-the-art recognition methods for on-line handwritten text recognition, and on-line handwriting overlaid text recognition which has been applied in small-surface devices, such as smart-phones. Then, we introduce current text input situation in the driving context. Finally, we introduce the character-position-free handwritten text recognition, we may consider on-line handwriting overlaid text as an extremely special case of character-position-free handwritten text patterns.

2.1 On-line handwritten text recognition

With the development of pen-based or touch-based devices, such as tablet PCs, digital pens and electric whiteboards and so on, the writing area of these devices becomes larger than before. People tend to write text continuously with little constraints. The demand for improving the handwriting text recognition is still increasing to meet potential many applications. On-line handwritten text recognition has been receiving larger attention, especially for unconstrained text recognition.

In general, handwritten text pattern recognition methods divided into on-line recognition and off-line recognition [6]. On-line recognition recognizes text patterns captured from a pen-based or touch-based input device where a series of trajectories of pen-tip or finger-tip movements are recorded, while off-line recognition recognizes text patterns captured from a scanner or a camera device as two dimensional images. Due to the on-line handwritten text pattern includes both temporal information of pen-tip or finger-tip movements and spatial shape information, the on-line handwriting recognition can yield higher recognition accuracy than off-line recognition. Moreover, on-line handwriting recognition provides friendly interaction and adaptation capability for users, such as the recognition result is showed and updated at the same time while writing, user can respond to the recognition result to correct misrecognition.

The research on on-line handwriting recognition started in the 1960s and has been receiving intensive interest from the 1980s. Tappert et al. [1] made a comprehensive survey before the 1990s. Nakagawa gave a survey focused on on-line handwritten Japanese characters recognition [2]. Since the 1990s, the research efforts have been aiming at the relaxation of constraints to ensure successful recognition, such as writing in boxes and the compliance with standard shapes. In recent survey papers, Plamondon et al. [6] mainly reviewed the advances of western handwriting recognition. Liu et al. [7] reviewed the advances in on-line Chinese and Japanese handwriting recognition from the 1990s. Recently, Zhu et al. [9] reviewed the on-line handwriting Japanese character recognition and its practical applications.

The handwritten Japanese/Chinese text recognition is more challenging than western language due to the large character set. Japanese character set consists of various characters: symbols, numerals, hiragana and katakana (called Kana), and Kanji characters of Chinese origin. Hiragana and katakana are phonetic characters. Kanji characters are ideographic characters, which have divided into two classes: JIS (Japanese Industrial Standard) first level set and JIS second level. The JIS first level set contains 2,965 common use characters, which are necessary for reading the newspaper, and the JIS second level set contains 3,390 characters less common and special characters for naming.

Chinese characters sets consist of traditional Chinese characters mainly used in Taiwan, and simplified Chinese characters used in the mainland of China. The simplified Chinese characters includes two character sets, one contains 3,755 characters and the other contains 6,763 characters, where the first set is a subset of the second one, were announced as the National Standard GB2312-80. The traditional Chinese set includes 5,401 characters. In both simplified and traditional Chinese, about 5,000 characters are frequently used [7].

Moreover, most Kanji/Chinese character patterns are composed of multiple subpatterns, called radicals, which are shared among many Kanji character patterns. In Kanji character patterns, some are simple consisting of a single radical, while others are complex with multiple radicals.

In addition, the various writing styles also obstruct handwritten text recognition. The handwritten scripts are generally classified into three typical styles: regular style, fluent style and cursive style. The regular style is also referred to as block style or hand-printed style, which is written carefully with keeping fairly strict proper stroke number and order. The fluent style is often called "cursive" style, which is close to peoples' practical writing and is written faster with fewer strokes, and some characters are connected together. The current recognition systems can recognize regular script with high accuracy, whereas the recognition of fluent or cursive style still remains unsolved and requires more intensive research efforts. The fluent or cursive script is the target of most recognition systems, which features greater variability of stroke-order and stroke-number within character and occurs frequently in practical writing.

Therefore, it is impossible to segment characters unambiguously in handwritten text recognition. Many works have focused on resolving the segmentation problem. These proposed methods can be roughly classified into the following categories: segmentation-based method, integrated segmentation and recognition method, and segmentation-free method.

2.1.1 Segmentation-based method

The segmentation-based method attempts to segment characters before character recognition solely according to geometric layout features, such as character size, position, and interrelationship.

Tseng et al. [13] proposed a segmentation method based on merging strokes and dynamic programming for the off-line handwritten Chinese characters recognition. It firstly extracts the strokes of the off-line characters to build the stroke bounding boxes. Then, the stroke bounding boxes are heuristically merged as a candidate character or a part of candidate character pattern using knowledge-based merging operations. Finally, the best segmentation boundaries is found by dynamic programming method. This method, however, is feasible only for neatly handwritten text. The segmentation performance mainly relies on the extracting stroke algorithm from characters.

Lu et al. [14] proposed a method to segment handwritten Chinese destination addresses of mail pieces. It merges subassemblies of Chinese characters based on the structural features of Chinese characters and the topological relations of subassemblies, namely, left-right, upper-lower and inside-outside relations. This pure structure-based segmentation method, however, is only suitable for handwritten text patterns without connected characters.

Zhao et al.[15] presented a two-stage approach to segment unconstrained off-line handwritten Chinese characters. In the first segmentation stage, according to the vertical projection and background skeleton, a horizontal handwritten Chinese character text is coarsely segmented into several blocks, and the blocks of connected characters are identified. The candidate segmentation points are found. In the second stage, connected characters are separated using geometric features of strokes, then the fine segmentation paths are extracted using fuzzy decision rules, which classify the candidate segmentation points. This segmentation method can resolve parts of connected characters. The segmentation accuracy of characters, however, is 81.6% on 1,000 unconstrained handwritten Chinese character texts. Wei et al. [16] proposed a new approach for connected Chinese characters, where the best segmentation path can be found by genetic algorithm.

Liang et al. [17] proposed a metasynthetic method to segment off-line handwritten Chinese character texts. For non-touching characters, it firstly applies the Viterbi algorithm to obtain the candidate segmentation paths, then a dynamic programming algorithm is applied to merge components. For touching characters, it firstly extracts candidate segmentation paths according to background and foreground information, and extracts peripheral features for each candidate segmentation path. Then the best segmentation path is found by the mixture probabilistic density function whose parameters are obtained by the EM algorithm.

Furukawa et al. [18] proposed a segmentation method for online unconstrained handwritten Japanese texts using off-stroke (between strokes) features. In this method, the handwritten text is pre-segmented into basic segments, and a segmentation graph is constructed, where a node stands for a candidate segmentation point, an edge stands for a candidate character pattern, which is created by merging one or more basic segments. Then, it extracts the features of each candidate character pattern, which include temporal and geometric features, and proposed off-stroke features within candidate character patterns and between candidate character patterns. Based on the assumption that each feature distribution fits a normal distribution, the candidate segmentation pattern likelihood can be calculated from these extracted features using a probabilistic model. Finally, an optimal segmentation path on the segmentation graph is found by dynamic programming (DP). The character segmentation rates, however, is 75.6% of all characters.

2.1.2 Integrated segmentation and recognition method

Handwritten Japanese/Chinese text recognition is challenging problem due to the fact that spaces between characters are not obvious, and many Kanji characters comprise radicals with internal gaps, as well as character touching. Without character recognition cues and linguistic context, characters in handwritten text patterns cannot be segmented unambiguously. A feasible solution to overcome the ambiguity of character segmentation is called the integrated segmentation and recognition method. Liu et al. [8] evaluated several common pattern classifiers based on this integrated segmentation and recognition framework, which includes neural classifiers, discriminative density models, and support vector classifiers, on handwritten numeral texts recognition. They demonstrate that superior text recognition performance can be achieved with appropriately designed classifiers even with simple pre-segmentation and without using geometric context in post-processing.

The integrated segmentation and recognition method is classified into segmentation-free and over-segmentation-based methods [10], [11]. The two methods are also called implicit segmentation and explicit segmentation methods, respectively. Segmentation-free methods will be introduced in the next section.

Over-segmentation-based methods [5], [12], [19], [20], [21], [22], [23], [24], attempt to split character patterns at their true boundaries and classify the split character patterns. Character patterns may also be split within them, but they are merged later. This is called over-segmentation. The over-segmentation-based method is mainly accomplished in two steps: over-segmentation and path search. The handwritten text pattern is firstly over-segmented into primitive segments, and each segment composes a single character or part of a character. The primitive segments are combined to generate candidate character patterns, and then a segmentation lattice is constructed as shown in Figure 2-1, where a node stands for a candidate segmentation point and an edge stands for a candidate character pattern. Each candidate character pattern can obtain several similar character classes with the corresponding class scores by character recognition, and then segmentation-recognition candidate lattice is constructed, where each path in the lattice corresponds to a segmentation-recognition paths (hypothesis), which is evaluated by combining the character recognition, linguistic context and geometric context. Finally, the optimal recognition result text is found by searching for the optimal segmentation-recognition path with maximum score or minimum cost.

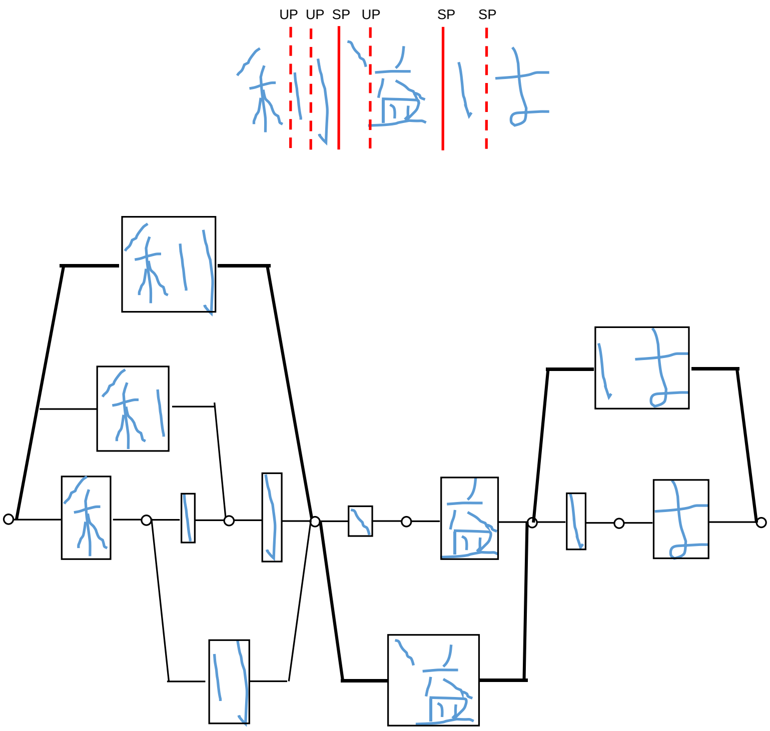


Figure 2-1 Segmentation lattice. (SP is segmentation point and UP is undecided point.) The thickly marked path is the correct segmentation path.

(1) Path evaluation

The key issue in over-segmentation-based text recognition is how to evaluate of candidate segmentation-recognition paths (segmentation hypotheses) in the candidate lattice. A desirable criterion should make the path of correct segmentation have the maximum score. Probabilistic model based on the maximum a posteriori (MAP) criterion [25] is one of the frequently used methods for segmentation hypothesis evaluation [5], [26], [27].

An early text class probability model can be found in [28]. Assume that a handwritten text pattern X is segmented into a sequence of segments $S = s_1 s_2, \cdots, s_n$ (note that there are many segmentation candidates even with the same text length), where s_i stands for a candidate character pattern, and is assigned to a text class $C = c_1 c_2, \cdots, c_n$, where character c_i is assigned to s_i by a character recognition. The a posteriori probability of the text class is defined as:

$$P(C|X) = \sum_{S} P(C, S|X) \tag{2-1}$$

The segmentation candidate is constrained to have the same length of C, that is $|S| = |C| = n$. The candidate character patterns are represented by the feature vectors $X = x_1 x_2, \cdots, x_n$. To avoid summing over multiple segmentation candidates in Eq. (2-1), the optimal text class can be decided by

$$C^* = arg \max_C \max_S P(C, S|X). \tag{2-2}$$

This is to find for the optimal segmentation candidate S for each text class. Using the Bayesian law, $P(C, S|X)$ is decomposed into

$$P(C, S|X) = \frac{P(X|C, S)P(C, S)}{P(X)} = \frac{P(X|C, S)P(S|C)P(C)}{P(X)} \tag{2-3}$$

Assuming context independence of character shapes, it can by approximated as:

$$\begin{aligned} P(C, S|X) &\approx P(C) \prod_{i=1}^{n} \frac{P(x_i|c_i, s_i)P(s_i|c_i)}{P(x_i)} \\ &= P(C) \prod_{i=1}^{n} \frac{P(c_i, s_i|x_i)}{P(c_i)} \\ &= P(C) \prod_{i=1}^{n} \frac{P(c_i|s_i, x_i)P(s_i|x_i)}{P(c_i)} \end{aligned} \tag{2-4}$$

where $P(s_i|x_i)$ stands for the probability of geometric context, the priori probability of text class $P(C)$ stands for the linguistic context. It is often approximated by a bigram language model for an open vocabulary:

$$P(C) = P(c_1) \prod_{i=2}^{n} P(c_i|c_{i-1}) \tag{2-5}$$

Assume that the character recognition is not related into the geometric context, $P(c_i|s_i, x_i)$ can be replaced by $P(c_i|x_i)$. Ignoring the geometric context, $P(s_i|x_i)$ can be viewed as a constant, and the text class probability in Eq. (2-4) is approximately

$$P(C, S|X) \approx P(C) \prod_{i=1}^{n} \frac{P(c_i|x_i)}{P(c_i)} \tag{2-6}$$

where $P(c_i|x_i)$ stands for the posterior probability of the candidate character pattern x_i being recognized as c_i. In literature [28], $P(c_i|x_i)$ is approximated by the output of a multi-player perceptron (MLP) classifier.

In handwritten Japanese text recognition, Nakagawa et al. [23] proposed a text class probability model incorporating the geometry of inter-character gap. The candidate pattern sequence is denoted by $S = s_1 g_1 s_2 g_2, \cdots, s_n g_n$, where s_i represents the geometric features of the *i*-th character pattern, which includes the width and height of bounding box, and g_i represents the geometric features between adjacent two character patterns. In the Eq. (2-3), $P(X)$ is omitted because it is independent of text class. $P(C)$ is estimated by a bigram model. Hence, $P(C, S|X)$ is approximated by

$$
\begin{aligned}
P(C, S|X) = P(X|C, S)P(S|C)P(C) \\
&\approx \prod_{i=1}^{n} P(x_i|c_i) \\
&\times \prod_{i=1}^{n} P(s_i|c_i)P(g_i|c_i c_{i+1}) \times \prod_{i=1}^{n} P(c_i|c_{i-1})
\end{aligned}
\tag{2-7}
$$

where $P(x_i|c_i)$ is the likelihood of pattern x_i with respect to class c_i, which is estimated by a character classifier. $P(s_i|c_i)$ and $P(g_i|c_i c_{i+1})$ can be seen as character likeliness and between-character compatibility, respectively. Finally, by taking log of the both sides in Eq. (2-7), the all score of a path is the summation of product of probabilistic likelihood in the right-hand side. The literature [20], [29], [30], [31] have used the similar evaluation criterion.

If the character classifier is trained to be resistant to non-characters, namely, all defined classes are assigned low confidence values on non-character patterns. Without geometric context score, it can still give high text recognition accuracy [7]. The text pattern is classified to

$$C^* = arg \max_C P(C)P(X|C) = \arg\max_C P(C) \prod_{i=1}^{n} P(x_i|c_i) \tag{2-8}$$

By assuming $P(C)$ is equal, the classification criterion is further simplified to

$$C^* = arg \max_C P(X|C) = \arg\max_C \prod_{i=1}^{n} P(x_i|c_i) \tag{2-9}$$

A text pattern can be segmented into variable lengths of character pattern sequences. However, since the likelihood measure is usually smaller than one, the summation criterion is often biased to paths with fewer characters, namely short path. This will raise the segmentation error of merging multiple characters into one candidate pattern. To overcome this bias, Tulyakov et al.

[32] proposed a normalized text probability score as follows:

$$C^* = \arg\max_{C} \left(\prod_{i=1}^{n} P(x_i|c_i) \right)^{1/n} \tag{2-10}$$

The normalized criterion, obtained by dividing the summation criterion by the number of segmented characters (segmentation length), tends to over-split characters.

To solve the problems, Zhu et al. [5] proposed a robust context integration model for on-line handwritten Japanese text recognition. By labeling primitive segments, the proposed path evaluation criterion can not only integrate the character shape information into recognition by introducing some adjustable parameters, but also is insensitive to the number of segmented character patterns because the summation is over the primitive segments. The path evaluation criterion is expressed as follows:

$$f(X,C) = \sum_{i=1}^{n} \left\{ \sum_{h=1}^{6} [\lambda_{h1} + \lambda_{h2}(k_i - 1)] logP_h + \lambda_{71} logP\left(g_{j_i}\middle|SP\right) \right.$$
$$\left. + \lambda_{72} \sum_{j=j_i+1}^{j_i+k_i-1} log\, P\left(g_j\middle|NSP\right) \right\} + m\lambda \tag{2-11}$$

where $P_1, P_2, P_3, P_4, P_5,$ and P_6 stands for the probabilities of trigram ($P(c_i|c_{i-2}c_{i-1})$), character pattern sizes ($P(b_i|c_i)$), inner gaps ($P(q_i|c_i)$), single-character positions ($P(p_i^u|c_i)$), pair-character positions ($P\left(p_i^b\middle|c_{i-1}c_i\right)$) and character recognition ($P(x_i|c_i)$), respectively. k_i is the number of primitive segments contained in the candidate character pattern x_i. $\lambda_{h1}, \lambda_{h2}$ ($h = 1\sim7$) and λ are the weighting parameters. g_i is the between-segment gap feature vector. If the adjacent two segments is within a true character, the label is NSP (non-segmentation point), otherwise is SP (segmentation point). Due to the character recognition is estimated by the combination score of on-line and off-line isolated character recognizers, Zhu et al. [33] divided the character recognition into two parts $P(x_i^{on}|c_i)$ and $P\left(x_i^{off}\middle|c_i\right)$, where x_i^{on} denotes the on-line features of x_i, x_i^{off} denotes the off-line features of x_i. $P(x_i^{on}|c_i)$ and $P\left(x_i^{off}\middle|c_i\right)$ are estimated by the score of the on-line recognizer and off-line recognizer, respectively. Then the path evaluation criterion in Eq. (2-11) is changed as follows:

$$f(X,C) = \sum_{i=1}^{n} \left\{ \sum_{h=1}^{7} [\lambda_{h1} + \lambda_{h2}(k_i - 1)] logP_h + \lambda_{81} logP\left(g_{j_i}\middle|SP\right) \right.$$
$$\left. + \lambda_{82} \sum_{j=j_i+1}^{j_i+k_i-1} log\, P\left(g_j\middle|NSP\right) \right\} + m\lambda \tag{2-12}$$

Under this same path evaluation criterion, Gao et al. [34], [35] reduced the text recognizer size for hand-held devices by compressing each component in this text recognition system. It compresses MQDF2 based off-line character recognizer by linear discriminant analysis (LDA), vector quantization and data type transformation, and selects an elastic matching based on-line recognizer. This recognition method has been successfully applied in smart phones and tablets.

In handwritten Chinese text recognition, to overcome the problem of sensitivity of the path length, Wang et al. [36] used the similar path evaluation criterion for real-time recognition of on-line handwritten sentences. The path evaluation is the combination of multiple contexts as follows:

$$f(X,C) = \sum_{i=1}^{n}\{k_i logP(c_i|x_i)+\lambda_1 logP(c_i|c_{i-1})+\lambda_2 logP(c_i|g_i^{uc})+\lambda_3 logP\left(z_i^p = 1\middle|g_i^{ui}\right) + \lambda_4 logP\left(c_{i-1},c_i\middle|g_i^{bc}\right)+\lambda_5 logP\left(z_i^g = 1\middle|g_i^{bi}\right)\} \tag{2-13}$$

where $P(c_i|x_i)$ is given by the character classifier, $P(c_i|c_{i-1})$ is a bigram language model, $P(c_i|g_i^{uc})$ and $P\left(z_i^p = 1\middle|g_i^{ui}\right)$ stand for the unary class-dependent (*uc*) and unary class-independent (*ui*) geometric score, respectively. $P(c_{i-1},c_i|g_i^{bc})$ and $P\left(z_i^g = 1\middle|g_i^{bi}\right)$ stand for binary class-dependent (*bc*) and binary class-independent geometric score (*bi*), respectively. Compared to literature [33], they added unary and binary class-independent geometric information to evaluate the path.

Wang et al. [21] also used the similar path evaluation criterion [36] for off-line unconstrained handwritten Chinese text recognition. Paths are evaluated from the Bayesian decision view by combing character recognition scores, class-dependent and class-independent geometric contexts, and linguistic context. The recognition performance on the HIW-MW test set [37] achieved the character-level accurate rate of 91.86% and correct rate of 92.72% using word class bigram.

Li et al. [24] proposed a new probabilistic model for off-line unconstrained handwritten text recognition to evaluate possible segmentation hypotheses. The path evaluation criterion as shown in Eq. (2-14) can be implemented in a simply way that follows Bayesian rules using just two classifiers, one is MQDF based isolated character recognizer, which has been trained by a linear discriminant analysis (LDA) –based negative training strategy using non-character patterns, the other is a the character verifier to check whether a candidate character pattern is true character or not, which can be transformed to posterior probability of a five-class MQDF classifier, including Chinese class, digit class, punctuation class and two classes of non-characters. The proposed method achieved the character-level recognition rates of 80.15% with a bigram language model on HIT-MW test set.

Zhou et al. [38] proposed a new method for on-line handwritten Chinese/Japanese text recognition by defining the high-order semi-Markov conditional random fields (CRF) on the candidate lattice to directly estimate the posterior probability of segmentation-recognition paths. In this semi-CRF model, it fuses the scores of character recognition, geometric and linguistic contexts in a principled MAP framework. This method has yielded superior text recognition performance compared to the state-of-the-art methods on the test sets of CASIA-OLHWDB (Chinese) [39], TUAT Kondate (Japanese) and ICDAR 2011 Chinese handwriting recognition competition.

The weighting parameters in the path evaluation criterion were sometimes determined by trial and error to yield higher text recognition performance. In recent years, some works have applied the supervised text-level learning approach to estimate the weighting parameters by minimizing the text recognition error. Zhu et al. [5] optimized the weighting parameters for on-line handwritten Japanese text recognition using genetic algorithm (GA). They also compared with the minimum classification error (MCE) criterion [40] optimized by stochastic gradient decent [41], and showed that GA-based optimization method yields better text recognition performance than MCE. Wang et al. [36] optimized the combing weights by MCE learning for on-line handwritten Chinese text recognition. The parameters in MCE learning are learned by stochastic gradient decent. Zhou et al. [26] proposed learning the weights by minimizing the negative log-likelihood (NLL) loss under the framework of CRF, and compared its performance with MCE criterion. Zhou et al. [38] modified NLL loss by adding a margin term to improve the generalization performance of parameter learning in semi-CRF.

(2) Path search

The search of optimal path for handwritten Japanese/Chinese text recognition is not trivial due to the large number of candidate segmentation-recognition paths in the candidate lattice. Moreover, the search is complicated when using word-level language models because the word segmentation is again a combinatorial problem [21]. The exhaustive search strategy that computes the scores of all segmentation-recognition paths and then selects the optimal one is computationally expensive.

Heuristic search algorithms that evaluate only a portion of segmentation-recognition paths have been commonly used in handwritten text recognition. The speech recognition field has contributed many efficient search algorithms based on dynamic programming (DP) and beam search [42].

If the segmentation-recognition path is scored by the accumulated cost form, the optimal path can be easily found by dynamic programming algorithm [5], [23], [29], [30], [31]. Under the normalized criterion, however, DP algorithm does not guarantee finding the optimal path. Beam search strategy has been employed. Among the partial paths ending at an intermediate node in the candidate lattice, beam search retains multiple partial paths with high scores for extension, the retained partial paths are also called beam width. All the retained partial paths of the parent nodes are extended to each child, where several high-score partial paths are again retained. At the terminal node, the path of highest score in the retained paths is as the optimal path. Liu et al. [8] used the beam search to find the optimal result for handwritten numeral text recognition.

On the other hand, according to the order of node generation in the heuristic research, the search algorithms can be divided into character-synchronous and frame-synchronous search [10]. The frame-synchronous is also called time-synchronous search. Liu et al. [43] proposed lexicon-driven text recognition approach for Japanese mail address reading using character-synchronous beam search strategy. The all address phrases are stored in a trie structure lexicon. Due to the beam search is used to expand all the nodes of same depth in the search space synchronously and proceeds by depth until there is no open node to expand, the character-synchronous beam search is appropriate for lexicon-driven text recognition. Zhu et al. [44] proposed lexicon-driven approach for on-line handwritten Japanese disease names recognition using frame-synchronous beam search. It restricts the character categories of recognizing each candidate character pattern from the trie lexicon of disease names and preceding paths during path search, as well as the length of disease names. The beam search is used to expand all the nodes of same segment in the search space.

2.1.3 Segmentation-free method

The over-segmentation-based method tries to over-segment handwritten text at the all possible character boundaries. For cursive writing with character overlapping and touching, however, handwritten texts are not easily segmented, over-segmentation-based method may result in misrecognition. In this case, a segmentation-free method is appropriate.

The segmentation-free methods, which mostly combined with hidden Markov model (HMM)-based recognition [11], simply slice the word or text pattern into frames (primitive segments) with moving a sliding window along word or text pattern, and label the sliced frames, which are concatenated into characters during recognition. Figure 2-2 shows an example of segmentation-free method for a handwritten Japanese text, the red rectangle is a sliding window with W width.

Su et al. [45] [46] proposed a segmentation-free strategy based on HMM for off-line realistic Chinese handwritten text recognition. The handwritten text is first converted to observation sequence by sliding windows with extracting features for each frame. In the training stage, embedded Baum-Welch algorithm is adopted to train character HMMs. In the testing stage, the optimal text result maximizing the a posteriori is found by Viterbi algorithm.

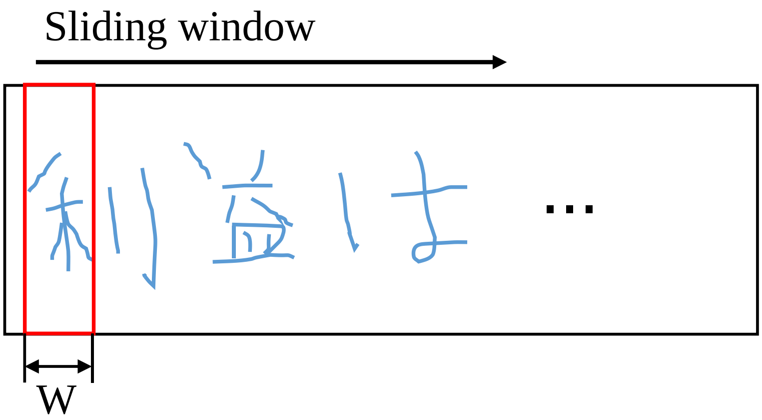


Figure 2-2　Segmentation-free method. W is the width of sliding window.

Recently, Messina et al. [47] proposed a segmentation-free method for handwritten Chinese text recognition based on multidimensional long-short term memory recurrent neural networks (MDLSTM-RNN). In this network model, the input image is divided into regularly spaced "frames" (vertical slices of the input image) by the scanning procedure, then an output vector of probabilities is produced for each frame, where each element of vector is associated to one of the characters. The free parameters in the network are trained to minimize the negative log-likelihood of the sequence of characters predicted by the network and the target transcription using the connectionist temporal classification (CTC).

Segmentation-free methods avoid character segmentation prior to recognition, and reduce expenditure in preparation of training data without labeling each character in texts. In the segmentation-free strategy, it only inputs the handwritten text images and their underlying character string, the system aligns automatically each character to its position in the text image and then estimates the model of that character. However, such methods do not sufficiently incorporate character shape information.

In on-line handwritten Japanese/Chinese text recognition, since over-segmentation-based methods can better utilize character shapes, the over-segmentation-based method is effective and more efficient compared with segmentation-free method [7]. Moreover, over-segmentation-based methods produce less primitive segments since they attempt to find the true boundaries of character patterns.

2.2　On-line overlaid handwriting text recognition

On-line handwritten continuous Japanese/Chinese text recognition has been receiving large attention, and the character-level accuracy is larger than 90% [5], [36], [38]. The touch screen size of handheld devices (e.g. smart phones), however, restricts the usage of continuous handwriting input method, especially in the case that users want to write text by their finger.

To solve the problem, there is one way to allow the user to write characters continuously on such a small touch screen with their finger on top of each other, namely, characters written overlaid without pauses as shown in Figure 2-3. Then the handwriting recognition system can automatically segment and recognize overlaid characters. At the same time, the overlaid handwriting text recognition causes more challenges than normal handwritten text recognition.

Since there is no spatial interval between characters, it is more difficult to segment characters. On the other hand, this writing way may confuse users due to that the handwritten characters are all displayed at the same area. Users cannot see clearly what he/she is currently writing. Moreover, users may slow down their writing speed or even stop writing and wait for the screen to be cleared.

Figure 2-3 An example of overlaid handwriting text.

There have been already some works on the overlaid handwritten text segmentation and recognition. In overlaid Japanese handwriting recognition, Shimodaira et al. [48] introduced substroke based Hidden Markov Models (HMMs) with a bigram language model for overlaid handwritten Japanese text. Using a bigram model consisting of 1,016 Japanese educational Kanji and 71 Hiragana characters, the character recognition rates are 74.9% for free stroke order patterns collected from people, and 91.1% for fixed stroke order patterns, artificially created from the isolated character database. Tonouchi et al. [49] proposed an on-line overlaid handwriting recognition system based on stroke-level discrete Markov Models They restricted the character set to recognize to non-Kanji characters, which include 81 hiragana characters and 5 symbols only, and applied Kana to Kanji conversion (KKC) in order to reduce the writing time of Kanji, especially by finger.

In overlaid English handwriting recognition, Bharath A. et al. [50] proposed a HMM-based recognition system for overwritten lowercase words. The promising result that word recognition rate is 89% with a 20K word lexicon, has been reported. Recently, Kienzle et al. [51] proposed a new input method on a small touch screen by writing words with drawing characters on top of each other. In this method, it never recognizes characters but detects character boundaries to untangle and render the handwritten message. The ink strokes with a word are automatically segmented into characters using a character segmentation algorithm, which builds four end-of-character models trained discriminatively with AdaBoost. The word boundary, however, is marked by space gesture.

In overlaid Chinese handwriting recognition, Zou et al. [52] proposed a quick segmentation method based on an artificial neural network to detect the first stroke of each character, and then let previous characters fade out for the clear viewing of the current character. The quick segmentation method is also used in the recognition system to speed up the whole recognition process. Overlaid handwritten text is recognized by combining scores of character recognition and bigram scores. Wan et al. [53] proposed a method of combining stroke level evaluation by SVM model and character level evaluation based on character recognition scores, bigram scores and geometric scores and then proposed a strategy to filter out correct segmentations. They report that the strategy performs better than the DP algorithm. Lv et al. [54] proposed a real-time overlaid handwriting recognition method under the integrated segmentation and recognition framework. It firstly over-segments a stroke sequence based on SVM into primitive segments, which may be concatenated into candidate characters during the next path search. Then, it searches for a best path by integrating class-independent unary and binary geometric scores, character classification score and bigram linguistic score.

On the other hand, the overlaid handwriting recognition may have another application for input text by writing in the air. Using various sensors such as Kinect [55], and Leap Motion Controller [56], we can detect the movement of an index finger so that text written by finger movement can be recognized.

Zhang et al. [57] proposed a finger-writing Chinese character recognition system for the character written in the air using a Kinect sensor. It firstly segments the hand from the cluttered background by a depth-skin-background mixture model. The fingertip then is detected from various hand poses by a dual-model switching algorithm. Finally, the finger-writing trajectory is generated by linking all detected fingertip positions and then reconstructed as an inkless character, which is recognized by a MQDF-based character classifier.

Chen et al. [58] proposed recognition methods for air-writing isolated characters and overlaid words with constraints on stroke orders and uppercase letters. The air-writing data is recorded by a six-degree-of-freedom motion tracking system, which includes a push-to-write model to signal the beginning and ending of writing by holding a button. The word is written with each letter overlaid in the same “virtual box” in space. They used Hidden Markov models for the air-writing characters, and statistical models for words by concatenating clustered ligature models and HMM-based letter models. In literature [59], they proposed an air-writing system for the overlaid words written in the same “virtual box” using Leap, which automatically detects and recognizes words. In detection stage, a window-based approach was proposed to automatically detect the air-writing event, and the consecutive writing events are converted into a writing segment. In recognition stage, the writing segments are recognized by a similar HMM-based recognizer used in [58].

2.3 Writing in automobile environment

Nowadays, it is common that computing-based systems and communications technologies are within modern cars providing various functions to drivers, such as electronic maps and navigation systems, information and entertainment systems, and satellite radio. Such systems may be already integrated by the car manufacturer or later added freely by the user. It is evident that using these functions while driving increase driver distraction and workload, and so, there are very clear rules about when they may and may not be used in Japan [4]. This is reflected in many navigation systems, which present a warning that the driver should not interact with the system while the car is in motion. The results of a survey, however, show that over one in ten (11%) of all drivers claim to input the destination on their satellite navigation while driving [60]. This survey was conducted by Privilege Insurance in England. This suggests that users have a desire for text input whiling driving.

Currently, there have been already some means for text input to interact with a navigation system in the automobile context. They are mainly classified into three methods, the first one is typing on a touch screen keyboard, the second one is speech recognition, and the third one is handwriting recognition.

Under the special driving situations, speech recognition is a more natural way of inputting text. Several researches show the benefit of speech input [61], [62]. Tsimhoni et al. [61] compared destination-entry methods between speech recognition and typing on a touching screen keyboard on examination of driving performance, glance behavior, task partitioning and subjective evaluation. The results showed that the use of speech recognition was better than touch-screen keyboard while driving, due to destination entry using a touch-screen keyboard took a longer task completion time, and significantly increased driving workload.Maciej et al. [62] compared speech-based interfaces with manual controls for different in-vehicle-information-systems (IVIS) while driving. These systems include audio, telephone with name selection, navigation system with address entry and point-of-interest selection. The results showed that speech interfaces improved driving performance, gaze behavior and subjective distraction for all systems except the navigation system with point-of-interest entry which requires multiple visual confirmations. These improvements, however, are not strong enough to reach the performance level of driving without IVIS.

Although speech recognition shows promise as a largely non-visual/manual input method, it is not free of certain limitations including usability and problems related to the underlying speech recognition technology, due to the speech recognition system can fail to recognize spoken input correctly or worse due to the similar sounding words and various noise interferences in the car environment, such as driving noises, wind noise (e. g. open windows), climate control fans, music and conversations [63].

Technological progress and price reductions have pushed a growing use of touch-based interfaces to control various functions in automobile environment. In the recent years, some researches have shown the potential for handwriting recognition for inputting text in the driving context.

Kamp et al. [64] did a quantitative experiment to evaluate different input devices (touchpad, keyboard and voice) for an in-vehicle Internet navigator intended for different types of tasks on a driving simulator. The drive can write any alphanumeric pattern by finger on the touchpad device, which was located on the steering wheel. The handwritten character was recognized by the recognition system. Though it is not possible to confirm that the touchpad device can be globally regarded as an easier and safer way to interact during driving than the keyboard, for the "name" instruction, the touchpad is clearly much more advantageous than keyboard with almost half the operating time and error rate of the keyboard. The handwriting touchpad may be considered as an input choice for drivers to some extent.

Burnett et al. [65] compared handwriting recognition to a conventional on-screen keyboard for entering an address into a navigation system whilst driving. The study results showed that handwriting touchpads do offer particular advantages over a conventional on-screen keyboard system in cars, such as time taken to enter a destination while driving was reduced and driving performance (e. g. speed variability) was improved when participants utilized handwriting. The location of the input device, however, plays an important role with regard to the dominant hand (right hand or left hand).

Inspired by the research work [65], Kern et al. [66] investigated handwriting as a text input method and explored further locations for the input and output interfaces while driving by creating different prototypes. These prototypes allow text input on the steering wheel or in the central console, and provide visual feedback (output) on the input surface or the dashboard. Their study on a driving simulator showed that handwritten text input by finger on a touchscreen mounted on the steering wheel is well accepted by users and that the visual feedback should be presented in the dashboard area or on the steering wheel. Moreover, compared to text input in the central console, the number of corrective actions and the remaining errors were significantly decreased (25% less) on the steering wheel.

2.4 Character-position-free handwritten text recognition

With the development and popularity of touch-based or pen-based input devices outside of the automotive domain, such as PDAs, smartphones, and tablet PCs, these devices may apply to or be embedded into the in-vehicle systems with the constraint of the car cockpit and safety concerns. Moreover, the existing handwritten Japanese/Chinese text recognition technologies have obtained higher character-level accuracy (over 90%) [5], [36], [37]. So there is a need for text input. It offers potential benefits to drivers to integrate the handwriting recognition technologies in the in-vehicle systems, such as navigation systems and information systems. Handwriting any text (e.g. destination), however, is a secondary task and disturbs the driver's visual attention and makes him/her distracted. There are several proposed interaction methods mentioned in section 2.3, attempt to balance between the text input by writing for alphanumeric data and driving task.

Due to Japanese/Chinese language has thousands of character categories with different shape, it is more challenge to input Japanese/Chinese text by writing in automobile environment. To solve this problem, in our previous work [67], we proposed a new input way that allows drivers input characters without visual feedback, namely, character-position-free writing as shown in Figure 2-4, to reduce driver's distraction as much as possible. We then considered methods for character-position-free on-line handwritten Japanese text recognition. Moreover, writing Kanji characters with many strokes by finger is not easy, especially without visual feedback, so we also considered the case of completely overlaid handwritten text recognition. By following the integrated segmentation and recognition framework, we considered two segmentation methods in the over-segmentation stage, one classifies each off-stroke between real strokes into non-segmentation point (NSP), segmentation point(SP) or undecided point (UP, maybe considered as NSP or SP) according to the output of SVM, the other sets each off-stroke as UP. From the results, our proposed methods have obtained a promising recognition rates.

The previous research works on the overlaid handwriting text recognition have introduced in the section 2.2. On the other hand, there are several research works on splitting of touching characters in off-line handwritten text recognition [68]. The touching characters mean multiple characters in a connected component, also called a touching pattern.

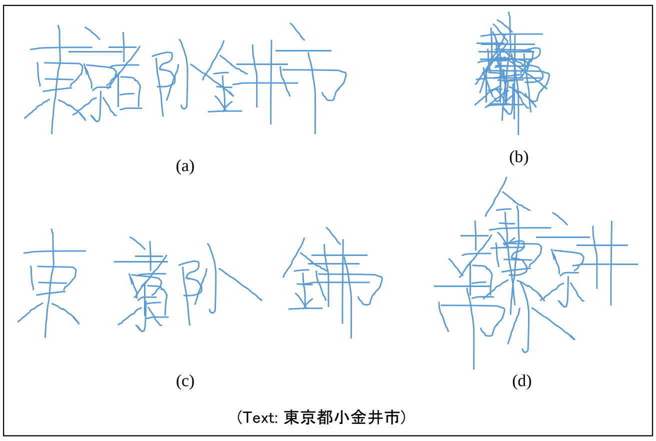


Figure 2-4　**Character-position-free handwriting for the same text.**

3. Character-position-free on-line handwritten text database

In this chapter, we describe character-position-free on-line handwritten text patterns, which include two parts. One is a small amount of collected text patterns, the other is generated datasets from normally handwritten Japanese and Chinese text patterns.

3.1 Collected character-position-free text patterns

Due to we do not have actual on-line handwritten text patterns collected from vehicle-mounted touch panels while driving, we need to investigate handwritten text patterns in such driving context.

From the guidelines for in-vehicle display systems, published by JAMA (Japan Automobile Manufacturers Association, Inc.) [4], where the following requirements are related to this study:

- The operation of a display system shall not result in a marked obstruction of forward field visibility.
- Information to be presented by a display system shall not cause the driver to gaze at the screen continuously.
- Preferably, a display system is so designed that its display of information can be discontinued by the driver.
- Information, such as the reporting of system state and operation that is displayed in response to the data inputted by the driver shall be quickly and easily comprehensible.

According to the requirements, a driver should write text without visual feedback, moreover, in one or more steps to keep safe driving.

We collected a small amount of text patterns from 10 participants in our laboratory with physical conditions simulated driving environment as much as possible. The collecting process is as follows:

First, we randomly prepared 50 phrases as shown in Table 3-1, most of them are short and the Japanese address string with many Kanji categories, and printed them with 10 phrases per one page.

Table 3-1　Phrases for collecting character-position-free text patterns.

No.	Text	No.	Text
1	北海道札幌市中央区北一条西２－１	26	駒場東大前駅東口
2	青森県三戸郡名川町	27	別府タワー
3	大字下名久井字青柳４番地１	28	三井アウトレットパーク多摩南大沢
4	神奈川県横浜市西区みなとみらい	29	西調布駅
5	忠生市民センター	30	東京都葛西臨海水族園
6	港区麻布地区総合支所	31	浦和競馬場
7	八王子駅南口	32	羽村市動物公園
8	墨田区東駒形コミュニティ会館	33	市民球場
9	アウトレットパーク多摩南大沢	34	SICILIA 料理カンナ
10	大和スポーツランド	35	巣鴨新田駅
11	板橋区下赤塚区民事務所	36	調布市立図書館染地分館
12	高島そば	37	オリンピック公園
13	高幡不動駅	38	民俗資料館
14	温泉やすらぎの湯	39	船の科学館
15	秋津駅南口	40	新井薬師前駅北口
16	東京ドーム	41	西府駅北口
17	青山公園	42	中道海浜公園
18	丸の内オアゾ	43	新宿駅西口
19	江北駅	44	SHIBUYA109
20	目黒区役所	45	武者小路実篤記念館
21	秋川渓谷瀬音の湯	46	有楽町駅国際フォーラム口
22	京成小岩駅北口	47	品川区立荏原図書館
23	吉祥寺駅北口	48	大特価セール
24	首都大学東京陸上競技場	49	テスト用の文章
25	日比谷公園	50	筆圧が高いので

Second, the participant freely wrote each phrase on a tablet PC without supported by wrist or elbow, and without confirming previously written strokes, namely without visual feedback. Figure 3-1 shows the collecting scene.

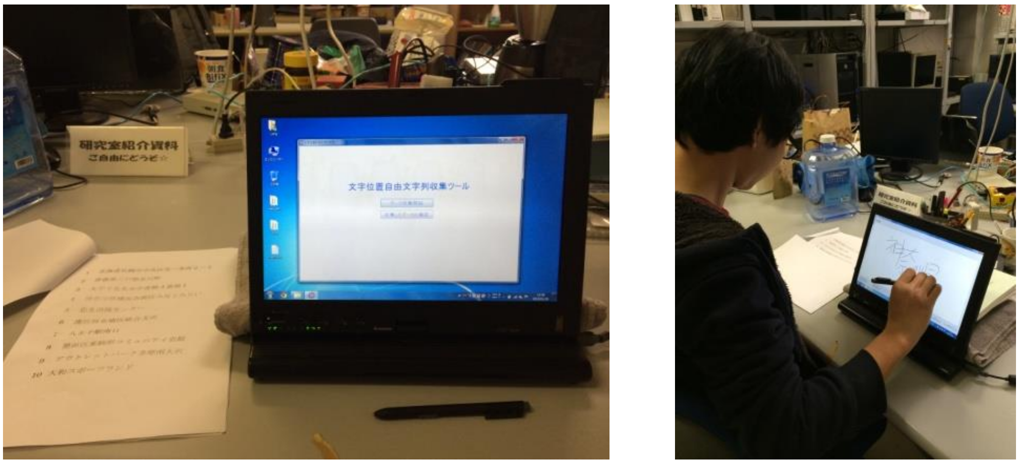


Figure 3-1 Scene of collecting character-position-free text patterns.

Then, we collected character-position-free on-line handwritten Japanese text patterns from 10 participants in our laboratory. Table 3-2 shows the participants' information.

Table 3-2 Participants' Information.

No.	Gender	Native Language	Dominant Hand	Writing Hand	Job
1	Male	Japanese	Right	Right	Student
2	Male	Japanese	Right	Right	Student
3	Male	Japanese	Right	Right	Student
4	Male	Japanese	Right	Right	Student
5	Male	Japanese	Right	Right	Student
6	Male	Vietnamese	Right	Right	Foreign Student
7	Male	Vietnamese	Right	Right	Foreign Student
8	Male	Vietnamese	Right	Right	Foreign Student
9	Female	Chinese	Right	Right	Foreign Student
10	Female	Chinese	Right	Right	Foreign Student

Finally, we annotated each stroke and checked these collected text patterns whether some characters missed in a text or not by an ink annotation tool as shown in Figure 3-2. If some characters have missed or written incorrectly in a collected text pattern, the participant should write the text again. Figure 3-3 shows examples of collected handwritten text patterns.

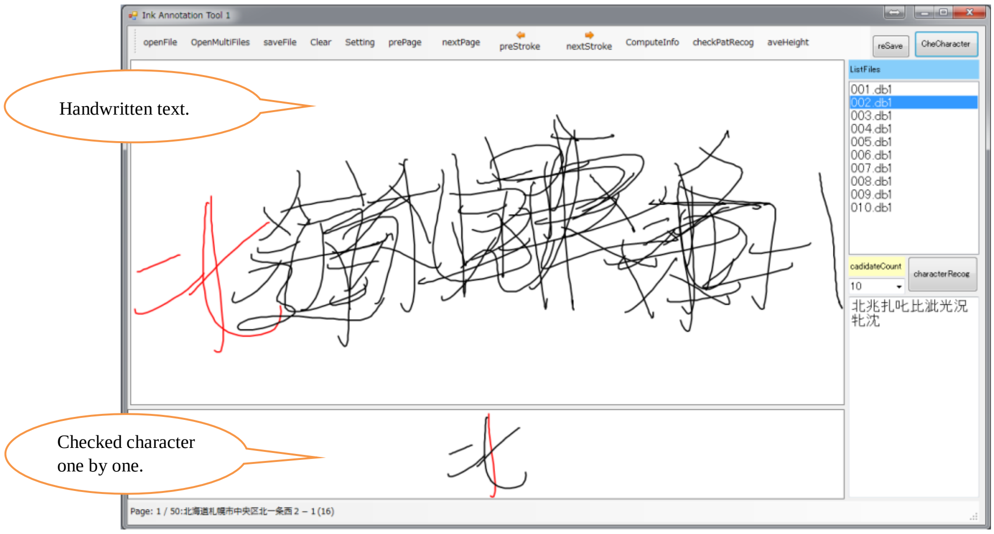


Figure 3-2　Ink annotation tool for checking collected handwritten text patterns.

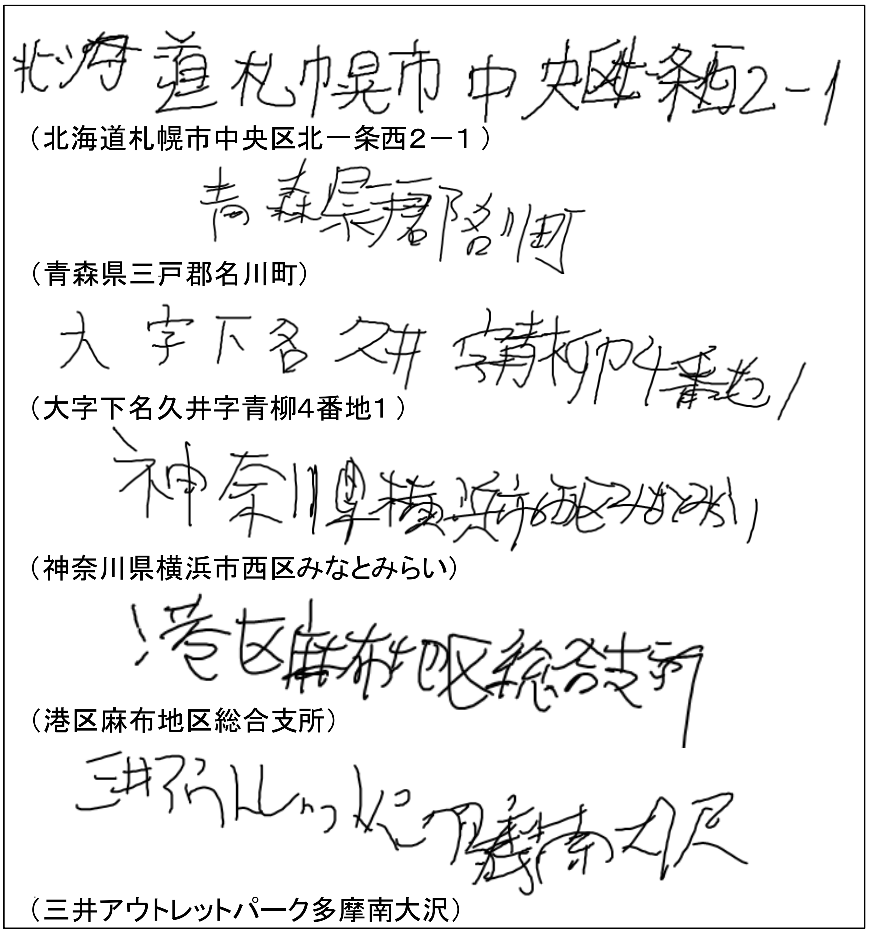


Figure 3-3　Examples of collected handwritten text patterns. The ground truth in bracket of each text pattern is placed under it.

The collected handwritten text patterns are so small to simulate exactly handwritten patterns under driving context. However, it showed some useful information to some extent. Based on the analysis of collected handwritten text patterns, characters are often partially overlapped on each other or separated randomly while stroke positions are relatively stable within a character pattern although displaced strokes make character patterns sometimes difficult to read.

Next, in order to develop a handwriting recognizer which can recognize on-line Japanese/Chinese text patterns written without physical support and visual feedback, we make models to produce such text patterns from normally handwritten text patterns in Kondate database [69] and CASIA-OLHWDB2.1 [39], as well as the model for text patterns with characters completely overlaid.

3.2 Kondate database

Kondate [69] is a database of on-line handwritten patterns mixed of texts, figures, tables, maps, diagrams and so on. In this research, we only use the part of on-line handwritten texts, which initially has been collected in Japanese from 100 people at Tokyo University of Agriculture and Technology (TUAT) in Japan.

As for on-line handwritten Japanese texts in Kondate, the most text patterns were collected by writing natural sentences taken from a Japanese newspaper on display integrated tablets. The writing style was not constrained so that most of the characters were written fluently although some people write in regular style due to their writing habit. Moreover, the writers write freely without any writing grids and even without guidelines.

Therefore, Kondate database covers any direction text patterns, such as horizontal, vertical, diagonal, horizontal and vertical mixed text and so on. Figure 3-4 shows some examples of on-line handwritten Japanese texts in Kondate.

In this research, we only use horizontal handwritten Japanese text patterns extracted from Kondate, and call it “Kondate_h”. Its statistic information is shown in Table 3-3.

Table 3-3 Statistic information of Handwritten Japanese text Database.

Database	#writers	#text lines	#character patterns	#character categories
Kondate_h	100	13,685	139,779	1,161
Dataset 1 to 4	100	15,389	129,076	1,123

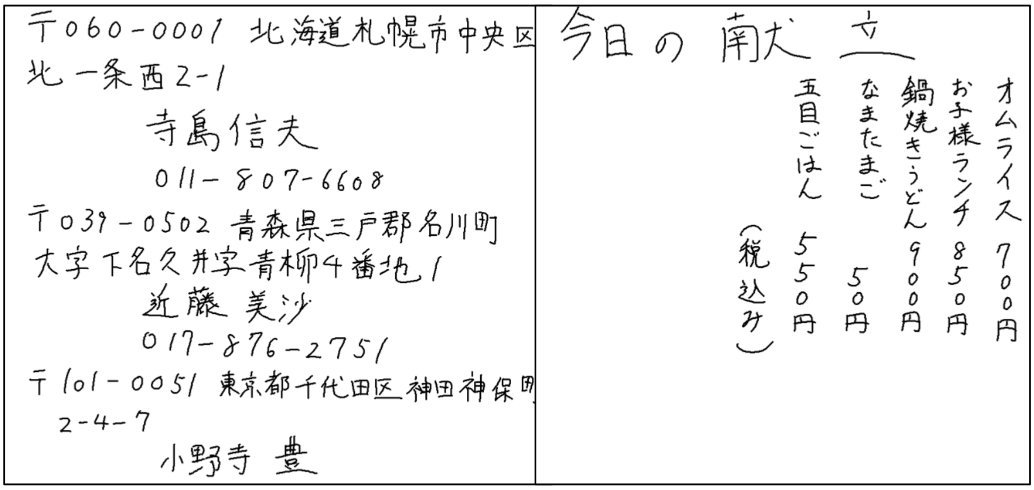


Figure 3-4　Examples of handwritten Japanese texts in Kondate.

3.3 CASIA-OLHWDB database

The CASIA-OLHWDB Chinese handwriting databases [39] built by the Institute of Automation of Chinese Academy of Sciences (CASIA), contain both on-line isolated characters and unconstrained handwritten texts. The on-line handwritten samples are collected by 1,020 writers using Anoto pen, and divided into six datasets, three for isolated characters (DB1.0-1.2) and three for handwritten texts (DB2.0-2.2).

In the on-line handwritten text datasets, DB2.1 involves more character categories than DB2.0 and DB2.2. In the thesis, therefore, we use the handwritten text dataset DB2.1, namely, CASIA-OLHWDB2.1, to generate the character-position-free Chinese handwritten text patterns. The dataset DB2.1 including 429,083 characters of 2,256 character categories is partitioned into standard training subset of 240 writers and test subset of 60 writers. The training text set contains 13,758 text lines from 1200 text pages, while the test text set contains 3,524 text lines from 300 text pages. The detail information of DB2.1 is shown in Table 3-4. Figure 3-5 shows some handwritten text pages of DB2.1.

Table 3-4　Statistic information of Handwritten Chinese text Database.

Database	Subset	#writers	#text lines	#character patterns	#character categories
CASIA-OLHWDB2.1	Training set	240	13,758	343,333	2,256
	Test set	60	3,524	85,750	2,251
CASIA-OLHWDB2.1		300	17,282	429,083	2,256

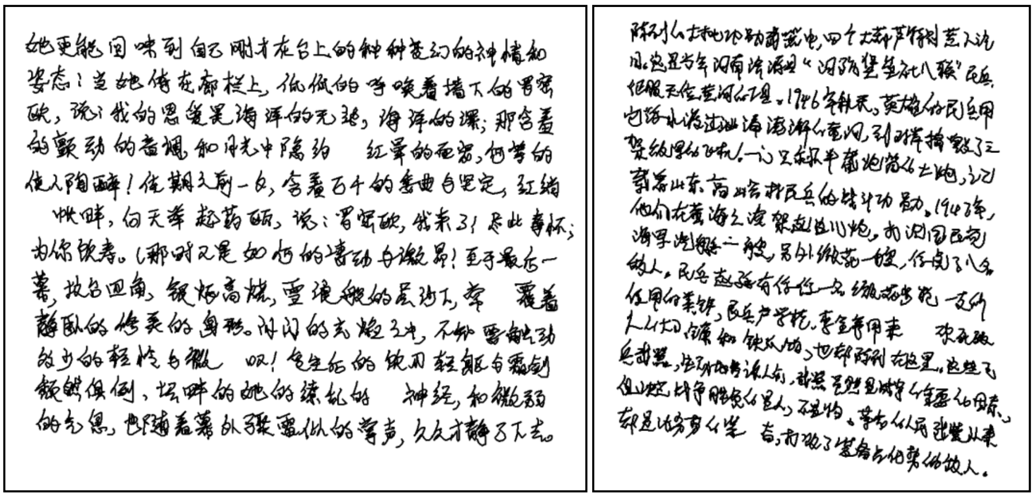


Figure 3-5 Examples of handwritten Chinese text patterns in CASIA-OLHWDB2.1.

3.4 Generated character-position-free text patterns

Due to the long text would not be written in the driving context, we delete text lines composed of more than 30 characters from Kondate_h/CASIA-OLHWDB2.1. Moreover, we divide a text line at the punctuation mark, and delete all punctuation marks following the previous research [54] since their recognition is difficult as independent symbols but they can be easily input by soft keys or gestures.

Therefore, based on these new normally handwritten patterns, we make 4 models and generate 4 datasets by changing the parameters to adjust character positions. We call them character-position-free handwritten Japanese text models and character-position-free handwritten Japanese text datasets, respectively.

For the random number generation, we compared the uniform random and the normal random. The uniform random generator generates text patterns more similar to the collected patterns. Therefore, we choose it to generate the random distance for all models.

Model 1 simulates handwritten text patterns where a character is placed randomly from a half character-size to a full-size advanced from left to right with 10% variations vertically according to Eq. (3-1) where d_x and d_y stand for the horizontal distance and the vertical distance from the previous character to the next character, respectively, $\bar{x}$ and $\bar{y}$ are the average width and the average height of characters in handwritten text, respectively. Generated handwritten text patterns are stored in Dataset 1.

$$
\begin{aligned}
d_x &\sim U([0.5 * \bar{x}, 1.0 * \bar{x}]) \\
d_y &\sim U([-0.1 * \bar{y}, 0.1 * \bar{y}])
\end{aligned} \tag{3-1}
$$

Model 2 simulates handwritten text patterns where a character is placed randomly from 0.4 times to 1.5 times the character-size advanced from left to right with 10% variations vertically according to Eq. (3-2). It has a wider variation horizontally compared with Model 1. Generated handwritten text patterns are stored in Dataset 2.

$$
\begin{aligned}
d_x &\sim U([0.4 * \bar{x}, 1.5 * \bar{x}]) \\
d_y &\sim U([-0.1 * \bar{y}, 0.1 * \bar{y}])
\end{aligned} \tag{3-2}
$$

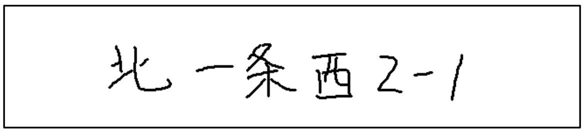
Model 3 simulates overlaid handwritten text patterns where characters are overlaid on previous characters according to Eq. (3-3). Generated overlaid handwritten text patterns are stored in Dataset 3.

$$
\begin{aligned}
d_x &\sim U([-0.1 * \bar{x}, 0.1 * \bar{x}]) \\
d_y &\sim U([-0.1 * \bar{y}, 0.1 * \bar{y}])
\end{aligned} \tag{3-3}
$$

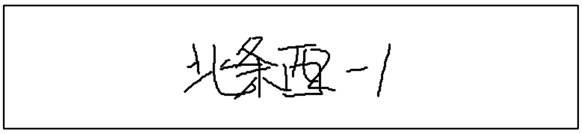
Model 4 simulates handwritten text patterns where a character is placed randomly in any direction (left, right, top, bottom etc.) of the immediately preceding character according to Eq. (3-4). Generated handwritten text patterns are stored in Dataset 4.

$$
\begin{aligned}
d_x &\sim U([-1.0 * \bar{x}, 1.0 * \bar{x}]) \\
d_y &\sim U([-1.0 * \bar{y}, 1.0 * \bar{y}])
\end{aligned} \tag{3-4}
$$

Figure 3-6 shows an original handwritten Japanese text pattern and generated patterns by Model 1 to Model 4, while Figure 3-7 shows an original handwritten Chinese text pattern and generated patterns by Model 1 to Model 4.

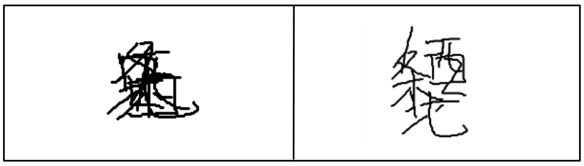


(a) An original handwritten text pattern.



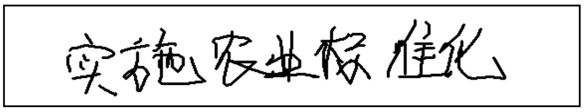
(b) Generated Dataset 1.

(c) Generated Dataset 2.



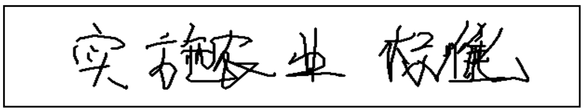
(d) Generated Dataset 3(Left) and Dataset 4 (Right).

Figure 3-6 Examples of generated character-position-free handwritten Japanese text patterns.

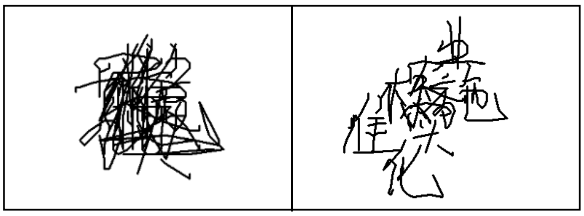


(a) An original handwritten text pattern.

(b) Generated Dataset 1.



(c) Generated Dataset 2.



(d) Generated Dataset 3 (Left) and Dataset 4 (Right).

Figure 3-7 Examples of generated character-position-free handwritten Chinese text patterns.

4. Character recognition system

In this chapter, we describe the isolated character recognition system combining the on-line character recognizer and the off-line character recognizer, for each candidate character pattern in the candidate lattice. The lattice is constructed for a character-position-free handwritten Japanese/Chinese text pattern.

Figure 4-1 shows the flow chart of the on-line character recognition system. The input to the system is an on-line character pattern, which includes a time sequence of coordinates of pen-tip or finger-tip movements. After feature extraction stage, a fast coarse classification is commonly used to first select a small subset of candidate classes which the input character pattern is expected belong to, for speeding up the recognition of the Japanese/Chinese large character set. Then, the input character pattern is classified into one of these candidate classes in the fine classification stage including on-line and off-line character recognizers. Finally, the system outputs the top N ($N \geq 1$) candidate classes.

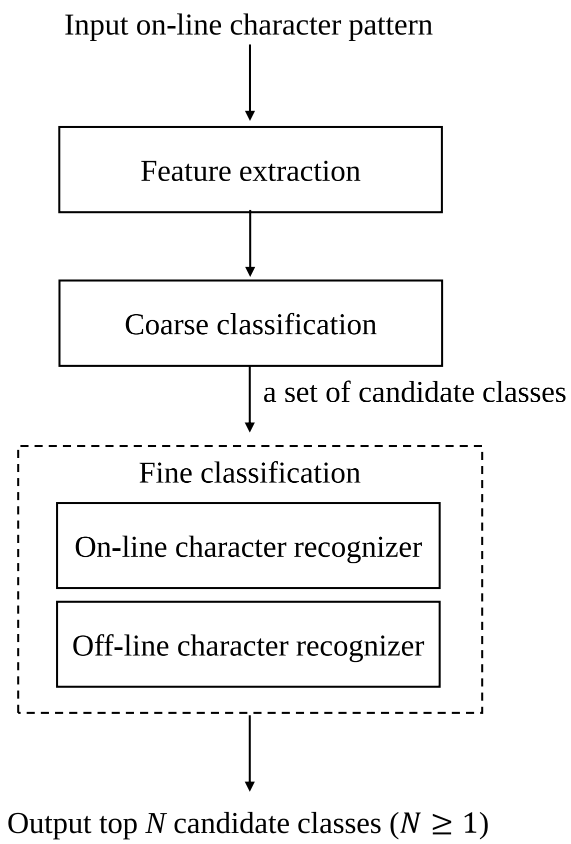


Figure 4-1 Flow chart of a character recognition system.

4.1 Coarse classification

Japanese and Chinese language are large character set. The Japanese character set includes thousands of ideographic characters of Chinese origin (Kanji), two sets of phonetic characters (Hiragana and Katakana), numerals and symbols, while the simplified Chinese language used mainly in the mainland of China has about 6,763 characters.

These large character categories affect not only the recognition accuracy but also the recognition speed. To improve the recognition speed, a common approach is to perform coarse classification before the fine classification [7]. In general, the coarse classification employs simpler classification algorithms or fewer features in order to select a small subset of candidates out of a very large character set quickly. Then, the fine classification will be used on these selected candidates to match an input character pattern so that the whole recognition time is reduced.

In the character recognition system, the coarse classification is based on off-line method without using the time sequence information from an input on-line character pattern, which includes the following steps: (1) nonlinear normalization, (2) directional features extraction, (3) feature reduction by Fisher linear discriminant analysis (FLDA), (4) character classification by a simple distance measure, i.e., Euclidian distance, to select candidates from thousands of character categories. The processing step 1 to step 3 is same as that of the off-line character recognizer introduced in the section 4.3.

Fine classification after coarse classification combines an on-line character recognizer and an off-line character recognizer to select the top character categories with the largest similarities from these candidates obtained by coarse classification as the output results.

The on-line character recognition method is based on stroke analysis and uses the structural features such as sampling points and line segments. Therefore, it is robust against character shape variations, while it is weak at collecting global character pattern information. In contrast, the off-line character recognition method is based on the image of character pattern. It uses un-structural features such as directional features, gradient histogram features and projection features, and so, it is robust against noises and stroke-order dependence but very weak against character shape variations. That is why combining the on-line and off-line character recognizers for fine classification to enhance robustness.

4.2 On-line character recognizer

On-line character recognizer recognizes a time sequence of coordinates of pen-tip or finger-tip movements, namely, an on-line character pattern as shown in Figure 4-2, which capture the temporal information on the pen movements, such as the number and order of pen strokes, the direction of the writing for each pen stroke and the speed of the writing within each pen stroke.

We introduce the Markov random field (MRF)-based on-line character recognizer, which consists of these components: linear normalization, feature points extraction, and matching the feature points with the states of each character class based on MRF model. Figure 4-3 shows its processing steps. In this recognizer, it firstly normalize the input on-line pattern by a linear method to keep the horizontal and vertical ratio. Second, it extracts feature points by a recursive method [70]. It then uses a MRF model to match the feature points with the states of each character class and obtain a similarity for each character class. Finally, it selects the character class with the largest similarity as the recognition result.

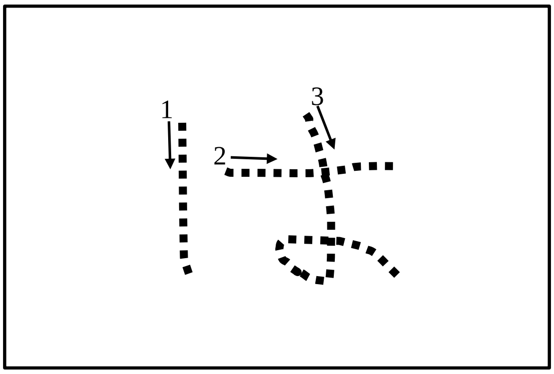


Figure 4-2　An example of on-line character pattern.

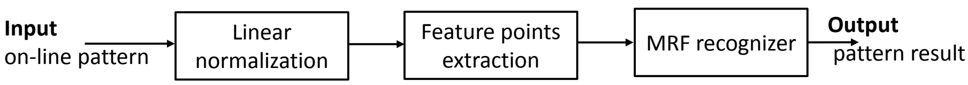


Figure 4-3　Flow chart of on-line character recognizer.

4.2.1　Linear normalization

Linear normalization is considered to be the most important pre-processing factor for on-line character recognition, which linearly mapped the character pattern onto a standard plane by interpolation or extrapolation. The size and position of character is controlled such that normalized plane in x and y dimension is filled. The implementation of interpolation/extrapolation is influential to the recognition performance [71], [72]. After linear mapping, the character pattern is not deformed except the change of aspect ratio.

It is better to fill both dimensions of normalized pattern (standard pattern), for alleviating feature extraction and classification, while the deformation is enlarged. In aspect ratio adaptive normalization (ARAN), however, the dimensions of the standard plane are not necessarily filled [73]. Depending on the aspect ratio, the normalized image is centered in the plane with one dimension filled. Assume the standard plane is square and the side length is denoted by L. Denote the width and height of the input on-line pattern as W_1 and H_1, and that of the corresponding normalized one as W_2 and H_2, the aspect ratio is defined by

$$R_2 = \begin{cases} W_2/H_2\,, & if\ W_2 < H_2 \\ H_2/W_2\,, & otherwise \end{cases} \tag{4-1}$$

The normalized pattern is filled one dimension by $max(W_2, H_2) = L$. That is, to keep the aspect ratio unchanged, the normalized image does not necessarily fill both dimensions. According mapping direction, the linear normalization can be divided into the forward mapping and backward mapping. The linear forward mapping is shown in Eq. (4-2), and the linear backward mapping is shown in Eq. (4-3), where α and β are parameters computed by Eq. (4-4).

$$x' = \alpha x, \qquad y' = \beta y \tag{4-2}$$

$$x = x'/\alpha, \qquad y = y'/\beta \tag{4-3}$$

$$\alpha = W_2/W_1\,, \qquad \beta = H_2/H_1 \tag{4-4}$$

4.2.2 Feature points extraction

For on-line character recognition, we use the feature points to express the on-line pattern rather than original coordinate sequence of the pattern, in order to reduce the computation complexity and discard repeated sampled coordinates. Before feature extraction, the input pattern is normalized to 128 x 128 pixels by linear normalization described in the previous section.

We extract feature points using the method by Ramner [70]. For each stroke, first, the start and end points are picked up as feature points. Then, the farthest point from the straight line between adjacent feature points is selected as a feature point if the distance is greater than a threshold value. This selection process continues recursively until no more feature points are selected. Figure 4-4 gives an example of the process of feature point extraction for a stroke.

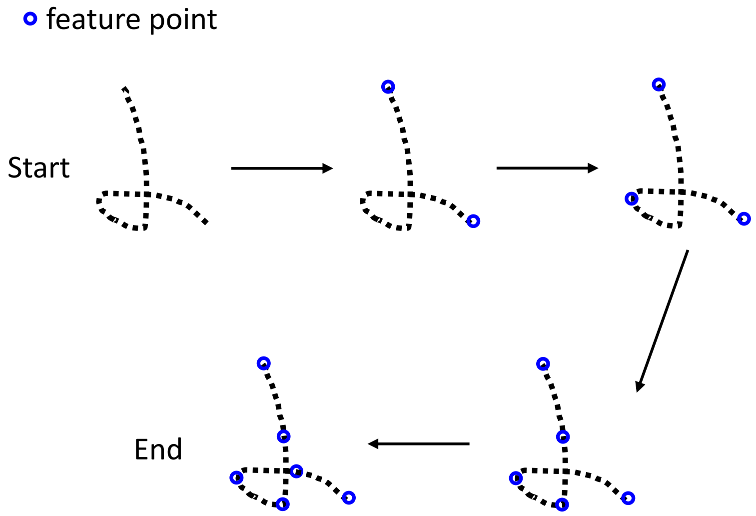


Figure 4-4 Feature point extraction of a stroke.

4.2.3 MRF-based on-line character recognition

Markov random fields (MRFs) can effectively integrate the information between neighboring pen-tip points such as binary features and triple features, which have been successfully applied to off-line handwritten character recognition [74] and on-line stroke classification [75]. We use a linear-chair MRF model to match the feature points extracted from an on-line character pattern with the states of each character class.

The feature points from an input pattern is denoted by sites $S = \{s_1, s_2, \cdots, s_I\}$, where I is the number of feature points, and states of a character class C is denoted by labels $L = \{l_1, l_2, \cdots, l_J\}$. The mapping from S to L during character recognition is denoted as $F = \{s_1 = l_i, s_2 = l_j \cdots, s_I = l_k\}$. F is called a configuration.

The feature vector extracted from feature points of an input pattern is considered as the observation set $\boldsymbol{O}$, including unary feature and binary feature. According to Bayesian theorem, the recognized character class is given by

$$C^* = arg\max_C P(C|\boldsymbol{O}) = arg\max_C P(C)P(\boldsymbol{O}|C) \qquad (4\text{-}5)$$

where $P(C)$ is the a prior probability that the given pattern belongs to a character class C, $P(\boldsymbol{O}|C)$ is the likelihood function of the observation set $\boldsymbol{O}$ for a class C.

Since there are more than one configuration F, $P(\boldsymbol{O}|C)$ can be further given by

$$P(\boldsymbol{O}|C) = \sum_{all\,F} P(\boldsymbol{O}, F|C) \tag{4-6}$$

Making the matching under all F from S to L is intractable, so we just consider the best configuration obtained by Viterbi algorithm. That is

$$P(\boldsymbol{O}|C) = P(\boldsymbol{O}, F_{best}|C) = P(F|C)P(\boldsymbol{O}|F_{best}, C) \tag{4-7}$$

The Hammersley-Clifford theorem establishes the equivalence with the Markov random field [7].

$$P(F|C) = \frac{1}{Z} exp\big(-E(F|C)\big) \tag{4-8}$$

where $E(F|C) = \sum_{cl} V_{cl}^{F}(F|C)$ is the prior energy function and $V_{cl}^{F}(F|C)$ is prior clique potential function defined on the corresponding cl. $Z = \sum_{F} exp(-E(F|C))$ is the normalization factor called partition function.

Taking $P(\boldsymbol{O}, F|C)$ into consideration, we can obtain the global likelihood energy function given by Eq. (4-9). $E(\boldsymbol{O}|F_{best}, C)$ is computed by Eq. (4-10) where $V_{cl}^{O}(\boldsymbol{O}|F, C)$ is the likelihood clique potential function.

$$P(F|C)P(\boldsymbol{O}|F_{best}, C) = \frac{1}{Z} exp\big(-E(F|C) - E(\boldsymbol{O}|F_{best}, C)\big) \tag{4-9}$$

$$E(\boldsymbol{O}|F, C) = \sum_{cl} V_{cl}^{\boldsymbol{O}}(\boldsymbol{O}|F, C) \tag{4-10}$$

For simplicity, we consider only single-site cliques $cl_1 = \{s_i\}, (0 < i \ll I)$ and pair-site cliques $cl_2 = \left\{\{s_i, s_j\}\right\}, (0 < i < I, 1 < j \ll I)$ to construct linear-chain MRF. We can obtain Eq. (4-11) from above two equations (Eq. (4-9) and Eq. (4-10)), where l_{s_i} is the label of a class C assigned to s_i. O_{s_i} is the unary feature vector composed of X and Y coordinates of site s_i, and $O_{s_i s_j}$ is the binary feature vector composed of differences of x and y between sites s_i and s_j, namely, $dx\ (X\ coordinate\ of\ s_i - X\ coordinate\ of\ s_j)$ and $dy\ (Y\ coordinate\ of\ s_i - Y\ coordinate\ of\ s_j)$.

$$E(\boldsymbol{O}|F_{best},C)+E(F|C)=\sum_{cl}\left[V_{cl}^{\boldsymbol{O}}(\boldsymbol{O}|F,C)+\sum_{cl}V_{cl}^{F}(F|C)\right]$$
$$=\sum_{s_i\in cl_1}\left[V_{cl_1}^{O}\left(O_{s_i}\middle|l_{s_i},C\right)+V_{cl_1}^{F}\left(l_{s_i}\middle|C\right)\right]$$
$$+\sum_{\{s_i,s_j\}\in cl_2}\left[V_{cl_2}^{O}\left(O_{s_is_j}\middle|l_{s_i},l_{s_j},C\right)+V_{cl_2}^{F}\left(l_{s_i},l_{s_j}\middle|C\right)\right] \qquad (4\text{-}11)$$

To derive the likelihood clique potentials from the negative logarithm of the conditional probabilities, we get the Eq. (4-12) and Eq. (4-13) from Eq. (4-11), where $P\left(O_{s_1s_0}\middle|l_{s_1},l_{s_0},C\right)$ is set as 1.

$$V_{cl_1}^{O}\left(O_{s_i}\middle|l_{s_i},C\right)=-logP\left(O_{s_i}\middle|l_{s_i},C\right) \qquad (4\text{-}12)$$

$$V_{cl_2}^{O}\left(O_{s_is_j}\middle|l_{s_i},l_{s_j},C\right)=-logP\left(O_{s_is_j}\middle|l_{s_i},l_{s_j},C\right) \qquad (4\text{-}13)$$

Moreover, since a label just interacts with only the neighboring labels in the linear-chain MRF model, the state transition probability can be employed to derive the prior energy function instead of the prior clique potential.

$$E(F|C)=\sum_{i=1}^{I}\left[V_{cl_1}^{F}\left(l_{s_i}\middle|C\right)+V_{cl_2}^{F}\left(l_{s_i},l_{s_{i-1}}\middle|C\right)\right] \qquad (4\text{-}14)$$

$$=\sum_{i=1}^{I}-logP\left(l_{s_i}\middle|l_{s_{i-1}},C\right)$$

Therefore, the energy function is as follows:

$$E(O,F|C)=E(O|F_{best},C)+E(F|C)$$
$$=\sum_{i=1}^{I}\left[-logP\left(O_{s_i}\middle|l_{s_i},C\right)-logP\left(O_{s_is_{i-1}}\middle|l_{s_i},l_{s_{i-1}},C\right)\right.$$
$$\left.-\,logP\left(l_{s_i}\middle|l_{s_{i-1}},C\right)\right] \qquad (4\text{-}15)$$

From the Eq. (4-15), the smaller the energy becomes the larger the similarity between the input pattern and a character class *C*. Each character class has a linear-chain MRF model. Hence, the recognition system uses the Viterbi search to match feature points of the input pattern with states for the MRF model of each character class and to find the matching path with the smallest energy for each character class.

$P\left(O_{s_i}\middle|l_{s_i},C\right)$ and $P\left(O_{s_is_{i-1}}\middle|l_{s_i},l_{s_{i-1}},C\right)$ are estimated by Gaussian functions. $P\left(l_{s_i}\middle|l_{s_{i-1}},C\right)$

is calculated as follows.

$$P\left(l_{s_i}\middle|l_{s_{i-1}},C\right)=\frac{\#number\ of\ transitions\ from\ l_{s_{i-1}} to\ l_{s_i}}{\#number\ of\ sites\ assigned\ l_{s_{i-1}}} \tag{4-16}$$

$$P\left(l_{s_1}\middle|l_{s_0},C\right)=\frac{\#number\ of\ s_1\ assigned\ l_{s_1}}{\#number\ of\ s_1} \tag{4-17}$$

To train the MRF model for each character class, we firstly initialize the feature points of an arbitrary character pattern within the training patterns of the character class as states of the MRF. Second, we set each unary feature vector of each feature point as the mean of the Gaussian function for each single-state, and each binary feature vector between two adjacent feature points as the mean of the Gaussian function for each pair-state, with initializing the variances of those Gaussian functions and the state transition probabilities with 1. Then we use the Viterbi algorithm or the Baum-Welch algorithm to train the parameters of the MRF, i.e., the means and variances of Gaussian functions and the state transition probabilities. We repeat the training process until the optimal parameters are obtained.

4.3 Off-line character recognizer

Off-line character recognition is known as optical character recognition (OCR). It is mainly used to process off-line character patterns (two dimensional images), which are obtained usually by scanning the documents or from digital camera and so on. Since character image does not embed any information about the writing process like the writing order of pen-tip points, pen-up and pen-down and so on. Therefore, off-line character recognition recognizes character patterns usually by their shapes. Figure 4-5 shows an example of off-line character pattern.

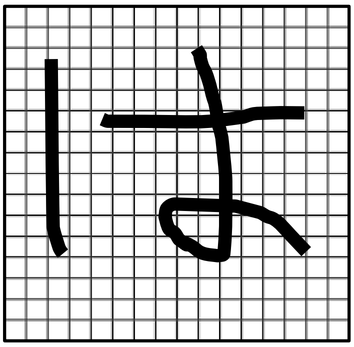


Figure 4-5 Off-line character pattern.

The major advantage of the off-line recognizers is to allow applying the image process technology, including nonlinear normalization and off-line feature. In recent years, nonlinear normalization (NLN) based on line density equalization, moment normalization (MN), bi-moment normalization (BMN), modified centroid-boundary alignment (MCBA), and their pseudo-two-dimensional (pseudo 2D) extensions all obtained good accuracy in handwritten character recognition. Moreover, as for off-line feature, the directional density feature and gradient feature extracted from character pattern also show more robust than feature points extracted directly from on-line pattern.

Since the directional features used for the off-line pattern are easily extracted from an on-line handwritten pattern by discarding temporal and structural information, we directly apply the off-line recognizer for the input on-line pattern, namely, it does not need to transform each on-line character pattern to an off-line character pattern.

Our off-line recognizer includes these components: nonlinear normalization, directional features extraction, dimensionality reduction and MQDF-based off-line character recognizer. Figure 4-6 shows the flow chart of an off-line character recognizer.

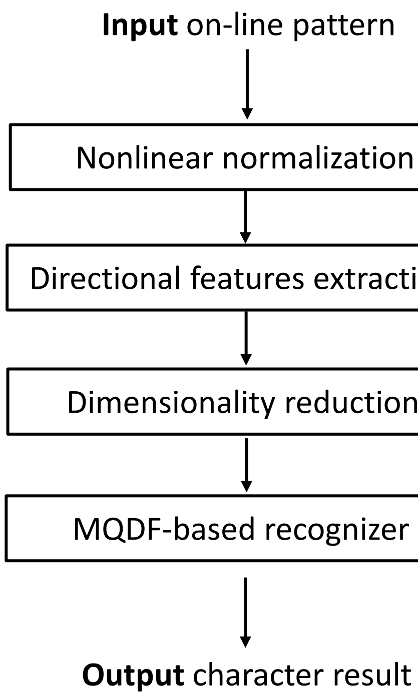


Figure 4-6 Flow chart for an off-line character recognizer.

4.3.1 Nonlinear normalization

Normalization regulates the size, position, and shape of character pattern, to reduce the shape variation between character pattern and the corresponding class. Some strategies were proposed to deform the character shape with aim to reduce the within-class variation. In this thesis, we apply the pseudo 2D bi-moment normalization (P2DBMN) [102] to normalize the input on-line pattern, which can be considered as an imaginary image.

Suppose the coordinate mapping functions $x'(x,y)$ and $y'(x,y)$ are obtained by linearly combining one-dimensional functions with the weight depending on another dimension as given by Eq. (4-18). The one-dimensional functions are obtained by applying 1D normalization to the projection functions of partial images.

$$\begin{cases} x'(x,y) = \sum_i w^{(i)}(y)\, x'^{(i)}(x) \\ y'(x,y) = \sum_i w^{(i)}(y)\, x'^{(i)}(x) \end{cases} \tag{4-18}$$

For an input on-line pattern which is considered as imaginary image $f(x,y)$ is partitioned into three horizontal soft strips by the weight function in *y*-axis:

$$f_x^i(x,y) = w^i(y) f(x,y), i = 1,2,3. \tag{4-19}$$

where $w^i(y)$ are weight functions as given by

$$\begin{cases} w^1(y) = w_0 \dfrac{y_c - y}{y_c}, y < y_c \\ w^2(y) = 1 - w^1(y), y < y_c \\ w^2(y) = 1 - w^3(y), y \geq y_c \\ w^3(y) = w_0 \dfrac{y_c - y}{H_1 - y_c}, y \geq y_c \end{cases} \tag{4-20}$$

where H_1 and y_c are boundary and coordinate in *y*-axis of centroid for the input pattern. w_0 is constant. Similarly, we obtain the three vertical soft strips $f_y^i(x,y) = w^i(x) f(x,y), i = 1,2,3$.

The three horizontal strips $f_x^i(x,y)$ $i = 1,2,3$ project onto the x-axis as in Eq. (4-21).

$$P_x^i(x) = \sum_y f_x^i(x,y)\, , i = 1,2,3 \tag{4-21}$$

The projection functions of the three strips on x-axis $P_x^i(x)$ $i = 1,2,3$, are used to compute three coordinate functions $x'^{(i)}(x)$, using the bi-moment normalization (BMN). The three 1-dimensional coordinate functions are then combined into a 2D coordinate function as given by

Eq. (4-22). The normalization composed of above two steps is commonly called P2DBMN.

$$x'(x,y) = \begin{cases} w^1(y)x'^{(1)}(x) + w^2(y)x'^{(2)}(x), y < y_c \\ w^3(y)x'^{(3)}(x) + w^2(y)x'^{(2)}(x), y \geq y_c \end{cases} \tag{4-22}$$

To obtain $x'^{(i)}(x)$ $i = 1,2,3$, by BMN, the second-order moments are split into two parts at the centroid: u_{20}^{i-} and u_{20}^{i+} in x-axis, u_{02}^{i-} and u_{02}^{i+} in y-axis. The bi-moments are computed from the projection of each strip as given by Eq. (4-23) and Eq. (4-24), respectively.

$$\begin{cases} u_{20}^{i+} = \dfrac{\sum_{x>x_c^i}\left(x - x_c^i\right)^2 P_x^i(x)}{\sum_{x>x_c^i} P_x^i(x)} \\ u_{20}^{i-} = \dfrac{\sum_{x\leq x_c^i}\left(x - x_c^i\right)^2 P_x^i(x)}{\sum_{x\leq x_c^i} P_x^i(x)} \end{cases} \tag{4-23}$$

$$\begin{cases} u_{02}^{i+} = \dfrac{\sum_{y>y_c^i}\left(y - y_c^i\right)^2 P_y^i(y)}{\sum_{y>y_c^i} P_y^i(x)} \\ u_{02}^{i-} = \dfrac{\sum_{y\leq y_c^i}\left(y - y_c^i\right)^2 P_y^i(y)}{\sum_{y\leq y_c^i} P_y^i(y)} \end{cases} \tag{4-24}$$

The centroid of each strip is computed by

$$\begin{cases} x_c^i = \dfrac{\sum_x xP_x^i(x)}{\sum_x P_x^i(x)} \\ y_c^i = \dfrac{\sum_y yP_y^i(y)}{\sum_y P_y^i(y)} \end{cases} \tag{4-25}$$

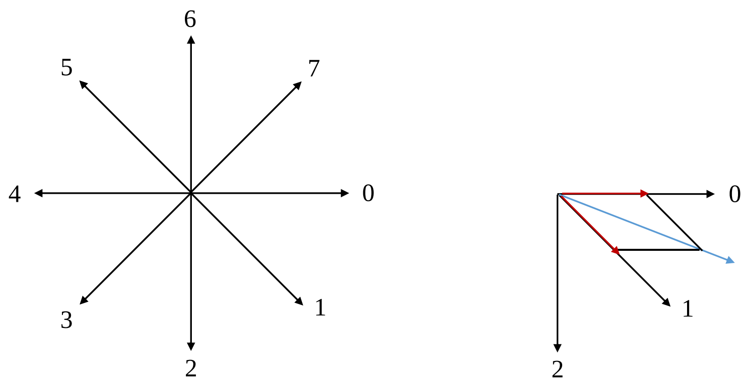
The boundaries of the input pattern are reset to $\left[x_c^i - 2\sqrt{u_{20}^{i-}},\ x_c^i + 2\sqrt{u_{20}^{i+}}\right]$ and $\left[y_c^i - 2\sqrt{u_{02}^{i-}}, y_c^i + 2\sqrt{u_{02}^{i+}}\right]$. For the x-axis, a quadratic function $u(x) = ax^2 + vx + c$ aligns three points $\left(x_c^i - 2\sqrt{u_{20}^{i-}},\ x_c^i, x_c^i + 2\sqrt{u_{20}^{i+}}\right)$ to normalized coordinate (0, 0.5, 1), and similarly, a quadratic function $v(y)$ used for the y-axis.

Finally, the coordinate functions are given by the following equation.

$$\begin{cases} x'^{(i)}(x) = u(x)x^{(i)}(x) \\ y'^{(i)}(x) = v(y)y^{(i)}(x) \end{cases} \tag{4-26}$$

4.3.2 Directional features extraction

We extract directly direction features from each on-line character pattern. The implementation of direction features extraction is various depending on the directional element decomposition, the sampling of feature values, the resolution of direction and feature plane, etc. Considering that the stroke segments of Japanese characters can be approximated into four orientations: horizontal, vertical, left-diagonal and right-diagonal, early works used to decompose the stroke (or contour) segments into these four orientations. Further, Liu et al. [76] proposed to decompose the stroke into eight, even 12 and 36 directions. Generally speaking, four and eight directional features are widely used. It is obvious that decomposing the contour pixels into eight directions instead of four orientations (a pair of opposite directions merged into one orientation) significantly improved the recognition accuracy. This is because separating the two sides of a stroke edge can better discriminate the parallel strokes.



(a) (b)

Figure 4-7 Eight chaincode directions (a) and the directional decomposition of a blue line segment (b).

In the off-line recognition, we use the eight directions-based decomposition for each line segment, defined by two consecutive pen-down points, and extract the directional features [103], which are histograms of normalized stroke direction. The eight direction planes, corresponding to eight chaincode directions, as shown in Figure 4-7 (a). And each line segment is decomposed into two components in two neighboring chaincode directions, as shown in Figure 4-7 (b).

4.3.3 Blurring and sampling

Each direction plane, with the standard size as the normalized image, need to be reduced to extract feature values of moderate dimensionality. A simple way is to partition the direction plane into a number of block zones and take the total or average value of each zone as a feature value. Partition of variable-size zones was proposed to overcome the non-uniform distribution of stroke density [77]. Overlapping blocks alleviate the effect of stroke-position variation on the boundary of blocks [78], yet a more effective way involves partitioning the plane into soft zones, which follows the principle of low-pass spatial filtering and sampling [79].

In implementation of blurring, the impulse response function (IRF) of spatial filter is approximated into a weighted window, also called a blurring mask. The IRF is often a Gaussian function given by

$$h(x,y) = \frac{1}{2\pi\sigma_x^2} exp\left(-\frac{x^2+y^2}{2\sigma_x^2}\right) \tag{4-27}$$

According to the Sampling Theorem, the variance parameter σ_x relates to the sampling frequency (the reciprocal of sampling interval). On truncating the band-width of Gaussian filter, an empirical formula was given in [80]:

$$\sigma_x = \frac{\sqrt{2}t_x}{\pi}, \tag{4-28}$$

where t_x is the sampling interval. At a location (x_0, y_0) of image $f(x,y)$, the convolution gives a sampled feature value

$$F(x_0,y_0) = \sum_x \sum_y f(x,y)h(x-x_0, y-y_0) \tag{4-29}$$

For ease of implementation, partition a direction plane into a mesh of equal-size blocks and set the sampling points to the center of each block. Assume to extract $K \times K$ values from a plane, the size of plane is set to $Kt_x \times Kt_x$. From N_d direction planes, the total number of extracted feature values is $N_d \times K^2$.

In the above-mentioned normalization and feature extraction, we set the size of normalized plane (and direction planes) to 24×24 pixels. Since we directly assign the line segments to eight direction planes, and the direction planes have continuous pixel values, a moderately small size of direction plane does not sacrifice the recognition accuracy. Each direction plane is then blurred and sub-sampled with sampling interval 3. Hence, we obtain 64 (8×8) feature values from each direction plane and the final dimensionality of feature vector is 512 (8×8×8).

The extracted feature values are causal variables. Power transformation can make the density function of causal variables closer to Gaussian [80]. This helps improve the classification performance of statistical classifiers. Power transformation is also called variable transformation [80] or Box-Cox transformation [81]. Power 0.5 is employed to transform the variables or feature vector.

4.3.4 Dimensionality reduction

In order to reduce the computation complexity, we use fisher discriminant analysis (FDA) to reduce the dimensionality of feature vectors. In the process of FDA, we need between-class scatter covariance S_b and within-class scatter covariance S_w of training samples. Suppose there are C character classes $(\omega_1, \omega_2, \cdots, \omega_C)$ and the j-th class with N_j training samples. The total training samples is N. Then, S_w and S_b are defined as:

$$S_w = \sum_{j=1}^{C} \sum_{i=1}^{N_j} (x_j^i - \bar{x}_j)(x_j^i - \bar{x}_j)^T \tag{4-30}$$

$$S_b = \sum_{j=1}^{C} N_j (\bar{X}_j - \bar{X})(\bar{X}_j - \bar{X})^T \tag{4-31}$$

where $X = \{x_j^i\}\ (j = 1,2,\cdots, C, i = 1,2,\cdots, N_j)$ is set of samples with n-dimensions. $\bar{X}_j = (\frac{1}{N_j}) \sum_{i=1}^{N_j} x_j^i$ and $\bar{X} = (\frac{1}{N}) \sum_{j=1}^{C} \sum_{i=1}^{N_j} x_j^i$ are the mean vector of the j-th class and all classes, respectively.

Based on the Fisher discriminant criterion [101], the process of working out the transformation matrix is to find out the optimal ratio which makes the S_b as large as possible while making the S_w as small as possible, which is described as

$$W_{opt} = arg \max_W \left| \frac{W^T S_b W}{W^T S_w W} \right| = [W_1\ W_2 \ldots W_m], \tag{4-32}$$

where $\{W_i | i = 1,2,..,m\}$ are m n-dimensional eigenvectors of $S_w^{-1} S_b$ corresponding to the m largest eigenvalues. W_{opt} is the $n \times m$ matrix composed of the m n-dimensions eigenvectors. By the transformation matrix W_{opt}, we can reduce the dimensionality of feature vectors from n-dimensions to m-dimensions.

4.3.5 MQDF-based off-line character recognition

The modified quadratic discriminant function (MQDF) [82] is the smoothed version of QDF, which performs Bayesian classification under the assumptions of multivariate Gaussian density for each class and equal a priori probabilities for all class. For an input pattern $X = (x_1, \cdots, x_n)^T$, the quadratic discriminant function (QDF) for the class ω_i $(i = 1, \cdots, M)$ has the form

$$g_0(X, \omega_i) = (X - u_i)^T \sum\nolimits_i^{-1} (X - u_i) + log\, |\Sigma_i| \tag{4-33}$$

where u_i and $\sum_i$ denote the mean vector and the covariance matrix of the class ω_i, respectively. QDF is actually a distance metric in the sense that the class of minimum distance is assigned to the input pattern.

QDF can be re-written in the form of eigenvectors and eigenvalues:

$$g_1(X, \omega_i) = \sum\nolimits_{j=1}^{d} \frac{1}{\lambda_{ij}} \left[\varphi_{ij}(X - u_i)^T\right]^2 + \sum\nolimits_{j=1}^{d} log\lambda_{ij} \tag{4-34}$$

where, $j = 1,2, \dots, d$, denote the eigenvalues of the class ω_i sorted in decreasing order, and φ_{ij}, $j = \lambda_{ij} 1,2, \dots, d$, are the corresponding eigenvectors.

By replacing the minor eigenvalues with a larger constant, the modified quadratic discriminant function (MQDF) is obtained as

$$\begin{aligned} g_2(X, \omega_i) = \sum\nolimits_{j=1}^{k} \frac{1}{\lambda_{ij}} \left[\varphi_{ij}(X - u_i)^T\right]^2 + \frac{1}{\delta_i} D_c(X) + \sum\nolimits_{j=1}^{k} log\, \lambda_{ij} \\ + (n-k)\, log\, \lambda_{ij} \end{aligned} \tag{4-35}$$

where λ_{ij} and φ_{ij}, $j = 1,2, \cdots, k$, denote the eigenvalue of the class ω_i sorted in decreasing order and the corresponding eigenvectors, respectively. k denotes the number of principal components and $D_c(X)$ is the square Euclidean distance in the complement subspace shown in Eq. (4-36). The parameter δ_i can be set as a class-independent constant as proposed by Kimura et al. [82] and $tr(\sum_i)$ denotes the trace of covariance.

$$D_c(X) = \|X - u_i\|^2 - \sum_{j=1}^{k} \left[\varphi_{ij}(X - u_i)^T\right]^2 \tag{4-36}$$

$$\delta_i = \frac{tr(\Sigma_i) - \sum_{j=1}^{k} \lambda_{ij}}{(n-k)} = \frac{1}{(n-k)} \sum_{j=k+1}^{n} \lambda_{ij} \tag{4-37}$$

4.4 Recognizer combination

The on-line and off-line character recognizers are combined by a linear function [33]. Suppose a character pattern x_i is recognized as a character class c_i by the on-line recognizer and off-line one with their similarity scores $f_{on}^{c_i}$ and $f_{off}^{c_i}$, respectively. Then, the confidence of the combined recognizer $f_c^{c_i}$ by the sum rule with class-independent linear combining parameters is given by the following formula:

$$f_{com}^{c_i} = \lambda_1 f_{on}^{c_i} + \lambda_2 f_{off}^{c_i} \tag{4-34}$$

where λ_1 and λ_2 are parameters. We use the minimum classification error (MCE) criterion to optimize the parameters, which will be described in chapter 5.

5. Character-position-free on-line handwritten text recognition

In this chapter, we describe proposed recognition methods for character-position-free on-line handwritten Japanese and Chinese text patterns by two segmentation methods to allow a user to overlay characters freely without confirming previously written characters. In the over-segmentation step, we consider two segmentation methods to solve the character segmentation problem. The first one is candidate segmentation method, which classifies off-strokes into segmentation point, non-segmentation point, and undecided point according the output of SVM model. The second one is undecided segmentation method, which sets each off-stroke as undecided point. Both of these two segmentation methods evaluate the character segmentation probability by SVM model. Then, the optimal segmentation-recognition path can be effectively found by Viterbi search in the candidate lattice, combining the scores of character recognition, geometric features, linguistic context, as well as the segmentation scores by SVM classification.

5.1 System overview

The character-position-free on-line handwritten text recognition method has three major steps: over-segmentation, candidate lattice construction and handwritten text recognition by optimal path search, as shown in Figure 5-1. Each step is described in detail in the following sections.

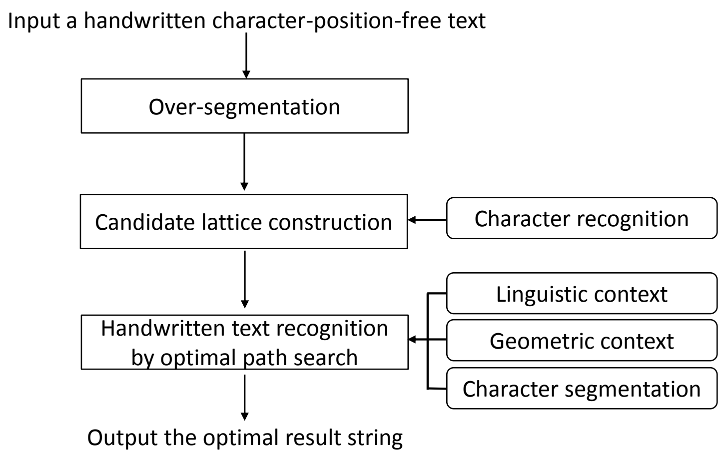


Figure 5-1 Flow chart of recognition process.

5.2 Over-segmentation

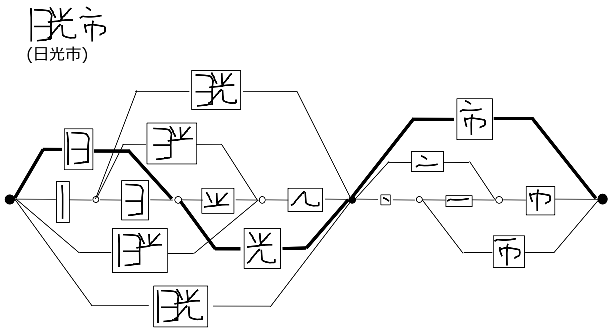
A handwritten text pattern is composed of many characters with a sequence of strokes, In Japanese, different kinds and complexities of characters: Kanji, Hiragana, Katakana, numeric characters and others are mixed. An input text pattern should be correctly segmented into each character as far as possible. It is difficult, however, due to the facts that spaces between characters are not obvious, many characters include multiple radicals with internal gaps and some characters are connected in writing. To solve these problems, a text pattern is over-segmented into a sequence of primitive segments so as to segment true segmentation points surely but may segment single character patterns into pieces, which could be combined in the later text recognition stage. Zhu et al. employ two-stage segmentation scheme [5]. In the first stage, each off-stroke (a vector from the last point of a previous stroke to the first point of the next stroke) is classified into non-segmentation point (NSP) and hypothetical one based on geometric features. Then, in the second stage, each hypothetical point is classified into segmentation point (SP) and undecided point (UP) using SVM model according to 20-dimensional features extracted from an off-stroke, where a SP separates two characters at the off-stroke, an NSP indicates the off-stroke is within a character and a UP is interpreted either as a SP or an NSP. When it is interpreted as a SP, it is used to extract candidate character patterns beside it with nearest neighbor SPs or UPs interpreted as SPs. When it is interpreted as an NSP, it is considered within a character pattern and does not play a role for segmentation. We call a sequence of strokes delimited by SP or UP as a primitive segment.

In a character-position-free handwritten text pattern, however, spaces between characters are very unstable. We can't directly use the conventional handwritten text recognition model. The first stage in the above-mentioned recognizer may combine two characters since the space between them disappears. Therefore, we remove the first stage and only employ the second stage.

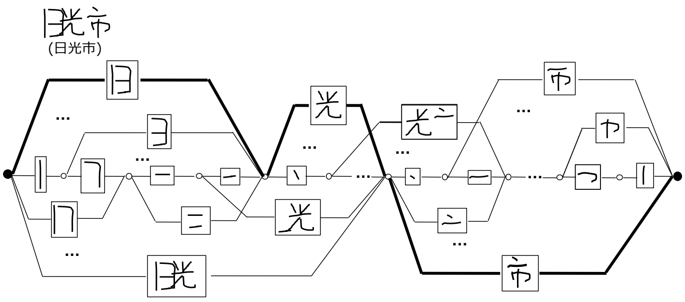
The next concern is the classification of off-strokes into NSP, SP or UP. We may follow this scheme or change the scheme. In this paper, we compare two segmentation methods. The first one is the conventional method to classify off-strokes into NSP, SP or UP although all the parameters and thresholds are retrained according to the new training patterns. We call this method “candidate segmentation method”. On the other hand, we set every off-stroke as UP in the alternative method although we employ the output of SVM model in the text recognition stage. We call it “undecided segmentation method”.

Namely, the first method classifies off-strokes into NSP, SP or UP, but the second method treats every off-stroke as UP. Both of the two methods, however, transform the output of SVM to segmentation probability value. Moreover, the segmentation probability value is combined into the optimal path evaluation in candidate segmentation-recognition paths.

Figure 5-2(a) shows segmentation by the candidate segmentation method and Figure 5-2(b) shows that by the undecided segmentation method, respectively, where a node is a segmentation point and a rectangle is a candidate character pattern. Rectangles between adjacent segmentation points are primitive segments. A segmentation path connects candidate character patterns following segmentation points from the start to the end. The thickly marked path is the correct segmentation sequence. We delete candidate character patterns if their widths are longer than the threshold. It is clear that the undecided segmentation method has more segmentation points than the candidate segmentation method.



(a) Segmentation by the candidate segmentation method.



(b) Segmentation by the candidate segmentation method.

Figure 5-2 Over-segmentation.

In both segmentation methods, we need to extract more geometric features from an off-stroke in order to enhance the reliability of over-segmentation. Through investigation into related literatures, we employ all the useful geometric features proposed so far, i.e. 56-dimensional features, to train SVM model. The detail will be described in the next subsection.

5.3 SVM model

Support vector machines (SVMs) developed from statistical learning theory [83] for pattern recognition, have been successful applied to the handwriting segmentation task. Sun et al. [84]

compared different supervised classifiers for classifying gaps between pieces of handwritten text to inter-word and intra-word classes, and found that SVMs outperform the other classifiers. Zhu et al. [85] employed SVM to determine segmentation point candidates for improving on-line freely written Japanese text recognition. Moreover, they showed that the character recognition rate by SVM-based segmentation are better than that by the three-layers neural network, although SVM method takes more training time than the neural network. Harbi et al. [86] also employed a linear kernel-based SVM classifier with temporal and spatial features for clock drawing segmentation, and showed this method outperforms the current state-of-the-art method on two collected datasets.

As for the character-position-free on-line handwritten text segmentation, we continue employ SVM classifier to segment each off-stroke with more geometric features.

5.3.1 Support Vector Machine (SVM)

Suppose we are given a training set $D_T = \{(x_i, y_i)|i = 1, \cdots, N\} \epsilon (X \times Y)^N$, where $x_i \epsilon X = R^n$ stands for the feature vector of a training pattern *i*, and $y_i \epsilon Y = \{-1, 1\}$ is an associated class label of a training pattern *i*, N is the number of training patterns, respectively.

Then, by mapping from the space of R^n to the high dimension space H:

$$\Phi\colon \begin{array}{c} \mathrm{X} = R^n \longrightarrow H \\ x \mapsto \mathrm{x} = \Phi(x)^n \end{array} \tag{5-1}$$

D_T is mapped as:

$$D'_T = \{(\mathrm{x}_i, y_i)|i = 1, \cdots, N\} = \{(\Phi(x_i), y_i)|i = 1, \cdots, N\} \tag{5-2}$$

The key idea of SVM is to learn the parameters of the hyperplane in space H that has maximum margin to classify two classes on training set.

To find the hyperplane $\omega x_i + b = 0$, it can be translated into the following optimization problem:

$$\begin{cases} min\colon \dfrac{1}{2}\|\omega\|^2 + C\displaystyle\sum_{i=1}^{N}\xi_i \\ s.t\colon\ \xi_i \geq 0, y_i(\omega x_i + b) \geq 1 - \xi_i \end{cases} \tag{5-3}$$

where $\frac{1}{2}\|\omega\|^2$ stands for the maximum margin, ξ_i is the learning error of a training pattern *i*, *C* is the trade-off between learning error and maximum margin, respectively.

Then, the feature vectors are mapped into an alternative space choosing kernel function

$K(x_i, x_j) = \varphi(x_i)\varphi(x_j)$ for nonlinear discrimination. Consequently, it leads to the following quadratic optimization problem:

$$\begin{cases} min: W(\alpha) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i\, y_j \alpha_i \alpha_j K(x_i, x_j) + \sum_{i=1}^{N} \alpha_i \\ s.t: \sum_{i=1}^{N} y_i \alpha_i = 0, \forall i: 0 \le \alpha_i \le C \end{cases} \tag{5-4}$$

where α is a vector of N variables and each element α_i corresponds to a training pattern (x_i, y_i).

The solution of the optimization problem as shown in Eq. (5-4) is to find a vector α^* to let $W(\alpha)$ is the minimum and the constraints are fulfilled. The classification of an unknown pattern x is made based on the sign of the following function $f(x)$, where SV stands for support vector as shown in Figure 5-3.

$$f(x) = \sum_{i:SV} \alpha_i^*\, y_i K(x, x_i) + b^* \tag{5-5}$$

In this thesis, we set the target value of segmentation points as 1, and that of non-segmentation points as -1. We use SVM^{light} [87] to obtain the separating hyperplane by solving this optimization problem as shown in Eq. (5-4) on training patterns. This software efficiently solves classification problem with many thousand support vectors, and converge with fast optimization algorithm.

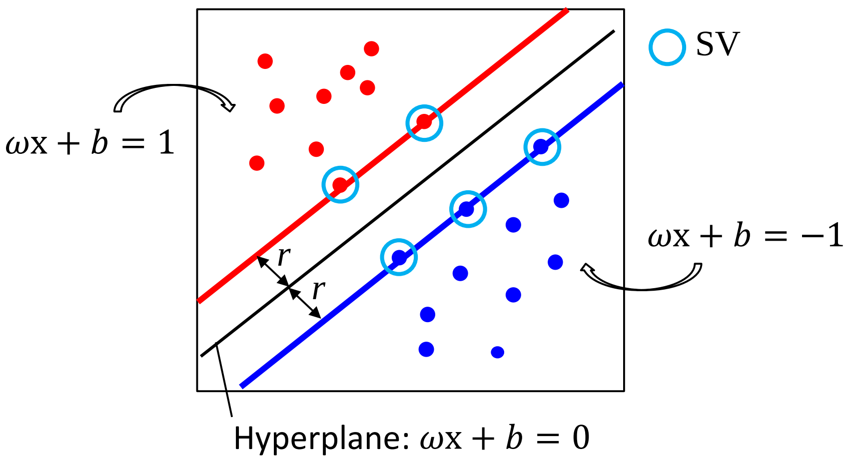


Figure 5-3 Example of Support vectors.

5.3.2 Features for SVM

An off-stroke is evaluated by SVM model. We cover all the useful geometric features from the literature [54] and [85], and extract 56 features from each off-stroke. Table 1 shows the features and table 2 shows terms to derive the 56 features. Most of the features are normalized by an average character size (*acs*). The average character size is estimated by measuring the length of the longer side of the bounding box for each stroke, sorting the lengths from all the strokes and taking the average of the larger 1/3 of them.

We just use order information for the on-line character recognition engine, but we do not use time information. Time information can be used to separate characters intentionally, but it causes mis-segmentation when a user stops writing in a character pattern. In automobile environment and even in ordinary environment, a user may stop writing or resume writing.

Table 5-1 Features extracted from an off-stroke.

No.	Definition	No.	Definition
f_1	DB_x / acs	f_{29}	D_{rl} / *acs*
f_2	O_{all} / (acs)^2	f_{30}	*width of* S_p / acs
f_3	D_{bx}/ width of B_p	f_{31}	height of S_p / acs
f_4	D_{bx} / width of B_s	f_{32}	width of S_s / acs
f_5	D_{bx} / acs	f_{33}	height of S_s / acs
f_6	D_{by} / height of B_p	f_{34}	log(width/height of S_p)
f_7	D_{by} / height of B_s	f_{35}	log(width/height of S_s)
f_8	D_{by} / acs	f_{36}	L_p /*acs*
f_9	O_b / area of B_p	f_{37}	L_s /*acs*
f_{10}	O_b / area of B_s	f_{38}	square root of B_p / acs
f_{11}	O_b / (acs)^2	f_{39}	square root of B_s / acs
f_{12}	D_{bsx} / acs	f_{40}	x-coordinate of P_{1e} /*acs*
f_{13}	D_{bsy} / acs	f_{41}	y-coordinate of P_{1e} /*acs*
f_{14}	D_{bs} / acs	f_{42}	x-coordinate of P_{2s}/ *acs*
f_{15}	Df_b / acs	f_{43}	y-coordinate of P_{2s}/ *acs*
f_{16}	L_{off} / acs	f_{44}	DP_x / *acs*
f_{17}	sin(L_{off})	f_{45}	DP_y / *acs*
f_{18}	cos(L_{off})	f_{46}	D_{1elp} / acs
f_{19}	f_1 / max(f_1) in text	f_{47}	D_{1ebp} / acs
f_{20}	*x-center of* S_p / acs	f_{48}	D_{2srp} / acs
f_{21}	*y-center of* S_p / acs	f_{49}	D_{2sbp} / acs
f_{22}	*x-center of* S_s / acs	f_{50}	D_{2els} / acs
f_{23}	*y-center of* S_s / acs	f_{51}	D_{2ebs} / acs
f_{24}	D_l / *acs*	f_{52}	*width of* S_{ps} / acs
f_{25}	D_r / *acs*	f_{53}	*height of* S_{ps} / acs
f_{26}	D_t / *acs*	f_{54}	log(width/height of S_{ps})
f_{27}	D_b / *acs*	f_{55}	*x-center of* S_{ps} / acs
f_{28}	D_{bt} / *acs*	f_{56}	*y-center of* S_{ps} / acs

We examined the distribution of each feature using all the training patterns. We have found 20 features are clearly essential such as f_{49} as shown in Figure 5-4(a) where the two classes (NSP and

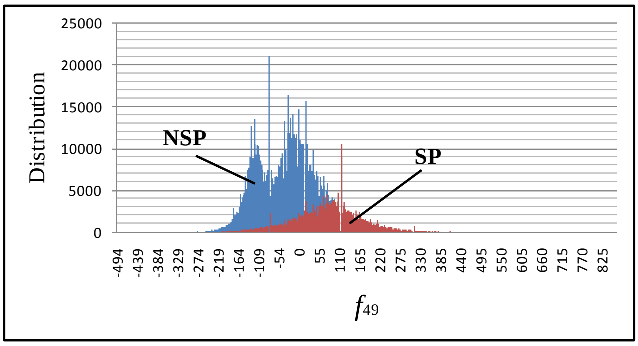
SP) are divided to some extent, while other features such as f_{21} cannot divide the two classes as shown in Figure 5-4(b). Table 3 shows the 20 essential features. We still keep using all 56 features, however, so that they contribute to the segmentation of character-position-free on-line handwritten Japanese text using SVM.

Table 5-2 Terms to derive features.

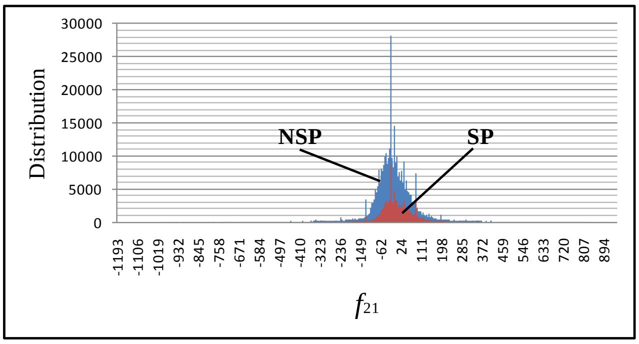
Term.	Description
acs	Average character size of text line
B_p	Bounding box of immediately preceding stroke
B_s	Bounding box of immediately succeeding stroke
B_{p_all}	Bounding box of all preceding strokes
B_{s_all}	Bounding box of all succeeding strokes
DB_x	Distance between B_{p_all} and B_{s_all} in x-axis
D_{bx}	Distance between B_p and B_s in x-axis
D_{by}	Distance between B_p and B_s in y-axis
O_b	Overlap area between B_p and B_s
O_{all}	Overlap area between $B_{p\text{-}all}$ and $B_{s\text{-}all}$
D_{bsx}	Distance between centers of B_p and B_s in x-axis
D_{bsy}	Distance between centers of B_p and B_s in y-axis
D_{bs}	Absolute distance of centers of B_p and B_s
Df_b	Vertical distance between B_{p_all} and B_s
L_{off}	Length of off-stroke
S_p	Immediately preceding stroke
S_s	Immediately succeeding stroke
S_{ps}	Union of S_p and S_s
P_{1e}	End point of S_p
P_{2s}	Start point of S_s
P_{2e}	End point of S_s
L_p	Length of S_p
L_s	Length of S_s
D_l	Distance between left bounds of B_p and B_s
D_r	Distance between right bounds of B_p and B_s
D_t	Distance between top bounds of B_p and B_s
D_b	Distance between bottom bounds of B_p and B_s
D_{bt}	Distance between bottom-top bounds of B_p and B_s
D_{rl}	Distance between right-left bounds of B_p and B_s
DP_x	Distance between P_{1e} and P_{2s} in x-axis
DP_y	Distance between P_{1e} and P_{2s} in y-axis
D_{1elp}	Distance between P_{1e} and left bound of B_p
D_{1ebp}	Distance between P_{1e} and bottom bound of B_p
D_{2srp}	Distance between P_{2s} and right bound of B_p
D_{2sbp}	Distance between P_{2s} and bottom bound of B_p
D_{2els}	Distance between P_{2e} and left bound of B_s
D_{2ebs}	Distance between P_{2e} and bottom bound of B_s

Table 5-3 Essential features.

Essential features	Number
$f_1, f_2, f_5, f_8, f_{12}, f_{13}, f_{14}, f_{16}, f_{24}, f_{25}, ,f_{26}, f_{27}, f_{28}, f_{29}, f_{44}, f_{45} ,f_{48} ,f_{49}, f_{52}, f_{53}$	20



(a) Distributions of the 49th feature.



(b) Distribution of the 21th feature.

Figure 5-4 Distributions of two features in training patterns.

Furthermore, in order to standardize the feature values, we normalize the values of each feature based on the mean μ_f and standard deviation σ_f of that feature. The normalized feature value is then calculated as follows:

$$v'_f = \frac{v_f - \mu_f}{\sigma_f} \tag{5-6}$$

The mean and standard deviation are calculated for each feature over the training patterns. These values are stored and used for normalizing the training and test patterns.

Using these normalized features of training patterns, we train the SVM model by setting NSP as negative class and SP as positive class. SVM model actually classifies each off-stroke into NSP, SP or UP based on its output for the candidate segmentation method, but it does not classify for the undecided segmentation method. For both of them, however, it produces probability of an off-stroke as NSP or SP.

5.3.3 SVM-based classification

In the candidate segmentation method, we classify each off-stroke into SP, NSP or UP. The trained SVM model, however, classifies off-strokes into SP and NSP. Therefore, we need to think how to judge undecided point.

We followed the judgment method mentioned in [85]. Based on the distribution of the outputs of the SVM model on training patterns, as shown in Figure 5-5. We can set the concatenation threshold th_c and the segmentation threshold th_s for both the sides of th and judge values smaller than th_c as concatenation (non-segmentation) points, values larger than th_s as segmentation points, and the others as undecided points to obtain the higher segmentation rate.

Moreover, the widths $(th - th_c)$ and $(th_s - th)$ are not need to equal, due to the unbalanced distribution of the outputs for two classes of segmentation points and non-segmentation points. We employ th_c and th_s determined on the training patterns for the testing patterns.

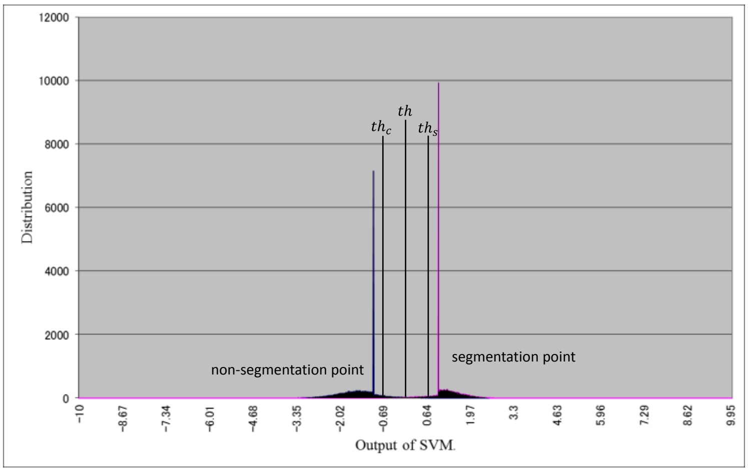


Figure 5-5 Distribution of the outputs of the SVM model on training patterns.

5.4 Candidate lattice construction

Each candidate character pattern is associated with a number of candidate classes with confidence scores from character classification. The combination of all candidate character patterns and candidate classes construct a segmentation-recognition candidate lattice, where each arc denotes a segmentation point and each node denotes a character class assigned to a candidate pattern. Figure 5-6 shows a part of the candidate lattice of an example as shown in Figure 5-2 where the correct path is thickly marked.

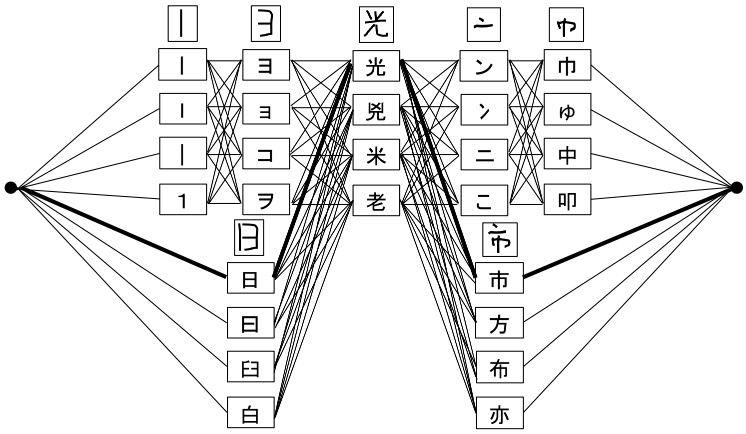


Figure 5-6 Candidate lattice with two segmentation paths.

5.5 Handwritten text recognition by optimal path search

The original path evaluation model was proposed in [5], and then formulated in [34]. We utilize the same criterion to evaluate paths in the candidate lattice and search for the optimal text result by the Viterbi algorithm.

5.5.1 Path evaluation criterion

Given a handwritten text pattern, which is segmented into a sequence of candidate character patterns $X = x_1 x_2, \cdots, x_n$, and a candidate character pattern x_i is recognized as a character class c_i, the probability of forming a recognized text string $C = c_1 c_2, \cdots, c_n$ is calculated by the following evaluation function:

$$f(\mathbf{X},\mathbf{C}) = \sum_{i=1}^{n} \left(\begin{array}{l} (\lambda_{11} + \lambda_{12}(k_i - 1)) \log P(c_i \mid c_{i-2}, c_{i-1}) + \\ (\lambda_{21} + \lambda_{22}(k_i - 1)) \log P(b_i \mid c_i) + (\lambda_{31} + \lambda_{32}(k_i - 1)) \log P(q_i \mid c_i) + \\ (\lambda_{41} + \lambda_{42}(k_i - 1)) \log P(x_i \mid c_i) + (\lambda_{51} + \lambda_{52}(k_i - 1)) \log P(p_i^u \mid c_i) + \\ (\lambda_{61} + \lambda_{62}(k_i - 1)) \log P(p_i^b \mid c_{i-1}, c_i) + \\ \lambda_{71} \log P(g_{j_i} \mid Sb) + \lambda_{72} \sum_{j=j_i+1}^{j_i+k_i-1} \log P(g_j \mid Sw) \end{array} \right) + \lambda n \tag{5-7}$$

where n is the number of candidate character patterns in a path, k_i is the number of primitive segments within a candidate character pattern x_i, λ_{h1}, λ_{h2} (h=1~7) and λ are weighting parameters.

$P(c_i|c_{i-2}, c_{i-1})$ in Eq. (5-7) is trigram linguistic context probability detailed in Chapter 6.

The term b_i, q_i, p_i^u and p_i^b in Eq. (5-7) stand for bounding box feature, inner-gap feature, and unary position feature of a candidate character pattern, and binary position feature between candidate character patterns, respectively. They are together called geometric context, their detail information are introduced in Chapter 6.

$\mathrm{P}(x_i|c_i)$ in Eq. (5-7) is given by a character recognizer detailed in Chapter 4.

The term g_j is a spacing feature vector concerning segmentation point, which is extracted from an off-stroke. Both of the probabilities are approximated by the SVM model introduced in Sect. 6.3. Here, SP is always treated as *Sb* and NSP is always treated as *Sw*. UP is interpreted as either *Sb* or *Sw*. When it is within a character pattern, it is treated as *Sw*, when it is between character patterns, it is treated as *Sb*.

$P\left(g_{j_i}|Sb\right)$ is the probability that the spacing between candidate character patterns (*Sb*) appears as g_{j_i} and $P\left(g_j|Sw\right)$ is the probability that spacing within a candidate character pattern (*Sw*) appears as g_j in Eq. (5-7). We approximate these two probabilities using the SVM model introduced in Sect. 6.3. The output values of SVM is warped to obtain probabilities $P(o_i|Sb)$ and $P(o_i|Sw)$, where o_i is the output value of SVM for g_i. The warping function is obtained from the distribution of all output values of SVM on the training dataset. We set $P(o_1|Sb)$ always as 1.

To warp all output values of SVM, we first obtain the histograms of $P(o_i|Sb)$ and $P(o_i|Sw)$, then take the cumulative probabilities $P'(o_i|Sb)$ and $P'(o_i|Sw)$ as follows:

$$
\begin{aligned}
P'(o_i|Sb) &= \sum\nolimits_{l=-\infty}^{o_i} p(l|Sb) \\
P'(o_i|Sw) &= \sum\nolimits_{l=o_i}^{\infty} p(l|Sw)
\end{aligned}
\qquad (5\text{-}8)
$$

Then, $P'(o_i|Sb)$ and $P'(o_i|Sw)$ are fitted by two sigmoidal functions, with the parameters estimated by minimizing squared errors, which is similar to Platt's method [88].

5.5.2 Parameter optimization

We train all the weighting parameters λ_{h1}, λ_{h2} (h=1~7) and λ in Eq. (5-7) by the minimum classification error (MCE) criterion [40] or the genetic algorithm (GA), using training data of character-position-free text patterns to maximize the recognition rate on this training data.

(a) MCE criterion

Liu et al. [89] have applied this criterion on handwritten numeral string recognition to improve recognition performance.

In the character-position-free handwritten text recognition, the weighting parameters Λ are trained on a training set $D = \{X^i, C^i | i = 1, \cdots, N\}$, where C^i denotes the ground-truth text class label of a training sample X^i, and N is the number of training samples. Each class C is assigned a discriminant score $g(X^i, C, \Lambda)$. Following Juang et al. [40], the misclassification measure on a training sample from class C^i is given by:

$$d(X^i, C^i, \Lambda) = -g(X^i, C^i, \Lambda) + \log\left(\frac{1}{N-1} \sum_{C \neq C^i} e^{-\eta g(X^i, C, \Lambda)}\right) \quad (5\text{-}9)$$

where η is a positive number. When $\eta \to \infty$,

$$d(X^i, C^i, \Lambda) = -g(X^i, C^i, \Lambda) + g(X^i, \bar{C}^i, \Lambda) \quad (5\text{-}10)$$

where $\bar{C}^i$ is the class label with the highest discriminant score in the closest rival class, namely,

$$\mathrm{g}(\mathrm{X}^i, \bar{C}^i, \Lambda) = \max_{C \neq C^i} g(\mathrm{X}^i, \mathrm{C}, \Lambda) \quad (5\text{-}11)$$

The loss of misclassification using sigmoid function is computed by,

$$l(\mathrm{X}^i, \mathrm{C}^i, \Lambda) = \frac{1}{1 + e^{-\xi d(X^i, C^i, \Lambda)}} \quad (5\text{-}12)$$

where ξ is a parameter. Then, the loss of misclassification based on training set is defined as:

$$L(\Lambda, \mathrm{D}) = \frac{1}{N} \sum_{i=1}^{N} l(X^i, C^i, \Lambda) \quad (5\text{-}13)$$

We use the stochastic gradient descent [41] to learn the optimal parameters in Eq. (5-13). The parameters are updated on each training sample by

$$\Lambda(t+1) = \Lambda(t) - \varepsilon(t) U \nabla l(X^i, C^i, \Lambda)|_{\Lambda = \Lambda(t)} \quad (5\text{-}14)$$

where $\Lambda(t)$ denotes the parameters on time t, $\varepsilon(t)$ is the learning step, U is related to the inverse of Hessian matrix and is usually approximated to be diagonal.

As for the character-position-free handwritten text recognition, MCE is to find the optimal parameters in Eq. (5-13) by minimizing difference between the scores of the most confusing text class and that of the correct one. The discriminant function is the path evaluation criterion defined in Eq. (5-7). The rival segmentation-recognition path, which is the most confusable one with the correct one, is obtained by beam search. Assume the discriminant functions f_c and f_r for the correct path and rival one, respectively. The parameters are updated iteratively by:

$$\Lambda(t+1) = \Lambda(t) - \varepsilon(t)\xi l(X^i, C^i, \Lambda(\mathrm{t}))(1 - l(X^i, C^i, \Lambda(\mathrm{t}))(f_r - f_c) \qquad (5\text{-}15)$$

(b) GA

Zhu et al. [5] have reported that the GA-based parameter optimization method yields better recognition performance than MCE-based method for on-line handwritten Japanese text recognition. The GA-based method, however, takes more training times than MCE.

The parameters are estimated by a GA on the training text patterns as follows:

Step1 (initialization): Initialize N chromosomes with random values from 0 to 1, average fitness of the N chromosomes f_{old} as 0 and time t as 1.

Step 2 (crossover): Select two chromosomes at random from N chromosomes. Cross the elements between two random positions to produce two new chromosomes. Repeat until obtaining M new chromosomes.

Step 3 (mutation): Change each element of $N+M$ chromosomes with a random value from −1 to 1 at a probability P_{mut}.

Step 4 (fitness evaluation): Evaluate fitness in terms of the recognition rate on training data with the weight values encoded in each chromosome.

Step 5 (selection): Decide the roulette probability of each chromosome according to its fitness. First select two chromosomes with the highest fitness, and then select chromosomes using the roulette until obtaining N new chromosomes. Replace the old N chromosomes with the new ones.

Step 6 (iteration): Obtain the average fitness of the new N chromosomes f_{new}. If ($f_{new} - f_{old} <$ threshold) occurs n_{stop} times or $t > T$, return the chromosome of the highest fitness. Otherwise, set f_{new} to f_{old}, increment t, and go to step 2.

For evaluating the fitness of a chromosome, each training sample is searched for the optimal path evaluated using the weight values in the chromosome. To save computation, we first set each weight value as 1 and select the top 100 recognition candidates (segmentation-recognition paths) for each training sample.

6. Linguistic context and geometric context

In this chapter, we describe the linguistic context and geometric context, both of them play an important role in the path evaluation criterion for character-position-free on-line handwritten Japanese and Chinese text recognition.

Due to the characters in a handwritten text cannot be segmented unambiguously before recognition, over-segmentation-based method is commonly employed to solve this problem. Therefore, this method may produces many candidate character patterns in the candidate lattice. For each candidate pattern, the character recognizer usually provides not only a unique similar class with the corresponding score, but also top N ($N \geq 1$) candidate classes with scores. The linguistic context can provide valuable information for selecting the optimal class from the top N candidate ones. Moreover, combining the linguistic knowledge, the geometric context and character recognition results, it can verify the candidate patterns, and so improve the text recognition rate.

6.1 Linguistic context

In handwritten text recognition, the linguistic processing of character recognition results after character segmentation is usually referred to as postprocessing [1]. Due to the character recognizer provides several candidate classes for a candidate character pattern, the selection of the optimal class from the set of candidate classes is based on the linguistic knowledge model. The linguistic knowledge models are usually represented in word dictionaries and statistical language models, such as character-based n-gram [91], and word-based n-gram [21], [92], [93]. The word-based n-gram language model is generally based on the syntactic/semantic classes (e.g., parts of speech) of words. Its use in linguistic processing involves the segmentation of text into words, usually by morphological analysis using a lexicon [92], [93]. Moreover, the adaptation of writer-specific linguistic dictionaries is beneficial for writer dependent handwritten character recognition [94]. Using the linguistic processing, the error rate of off-line handwritten English text is reduced by about 50 percent for single writer data and by about 25 percent for multiple writer data [98].

Recently, the unsupervised language model adaption is proposed for unconstrained off-line handwritten Chinese text patterns, and improves the recognition performance impressively, especially for the ancient domain documents [95]. Li et al. [96] applied the recurrent neural network language model (RNNLM), which is superior to the n-gram language models due to its capability to capture long-span history by discriminative leaning using the recurrent neural network, to improve the recognition of off-line handwritten Arabic documents.

Due to the word-based n-gram language models need an additional word segmentation tasks, the simple and effective character-based n-gram model is widely used for handwritten Japanese and Chinese text recognition [5], [38]. Figure 6-1 gives an example to shown character-based and word-based unigram (n = 1), bigram (n = 2) and trigram (n = 3) language model.

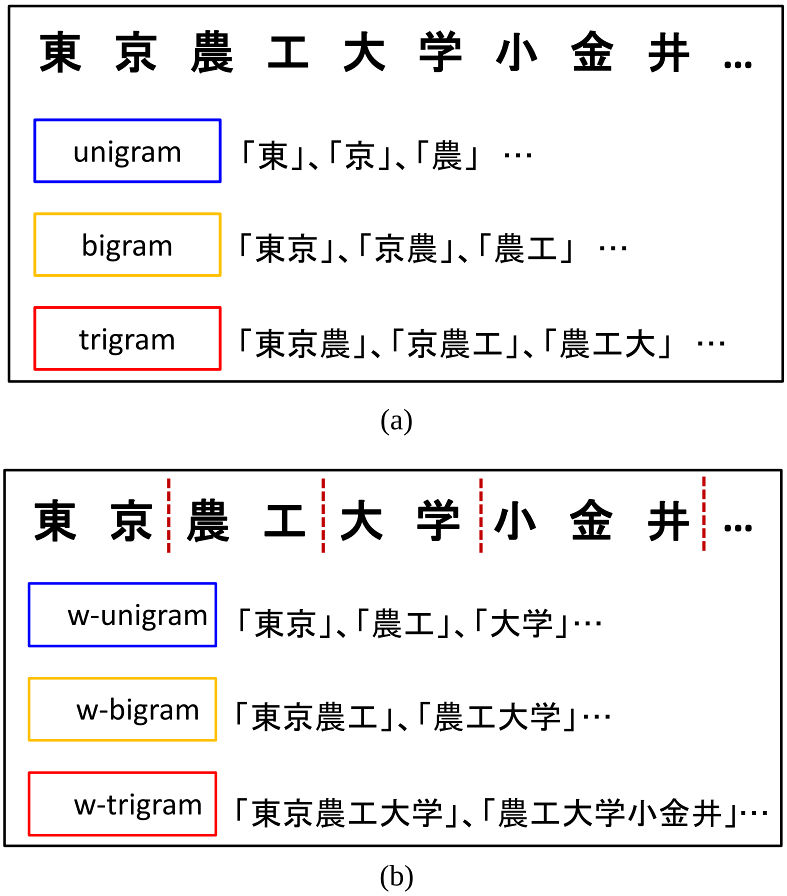


Figure 6-1 Examples of character-based (a) and word-based (b) unigram, bigram and trigram model.

Statistical language modeling involves attempts to capture regularities of natural language in order to improve the performance of various natural language applications, such as machine translations. N-gram models as the most successful statistical language mode have been applied in handwriting recognition, since it can be easily integrated with the character recognition. We will introduce it in the next section.

6.1.1 N-gram language model

The most widely used language models is n-gram language models, where *n* is called the order of the model. Such model estimates the statistical dependency between *n* characters or words. Considering the complexity of language models, the order *n* usually takes1, 2 or 3, namely unigram, bigram and trigram langugae model, respectively.

Given a handwritten Japanese/Chinese text or sentence with *l* characters $W = w_1 w_2 \cdots w_l$, based on the statistical langugae model, the priori probability of this sentence $P(C)$ can be decomposed as follows:

$$P(W) = P(w_1)P(w_2|w_1)P(w_3|w_1 w_2) \cdots P(w_l|w_1 \cdots w_{l-1})$$

$$= \prod_{i=1}^{l} P(w_i|w_1 w_2 \cdots w_{i-1}) \quad (6\text{-}1)$$

Here, we assume that the probability of character w_i being written depends only on the previous characters $(w_1 \cdots w_{i-1})$ of the sentence.

In n-gram language models, Eq. (6-1) is transformed into the Eq. (6-2) with changing the probability of character w_i being written depends only on the previous $(n-1)$ characters of the sentence.

$$P(W) = \prod_{i=1}^{l} P(w_i|w_1 w_2 \cdots w_{i-1}) = \prod_{i=1}^{l} P\left(w_i \middle| w_{i-n+1}^{i-1}\right) \quad (6\text{-}2)$$

Where *n* is called the order of the model. w_i^j denotes the characters sequence $w_i \cdots w_j$. Even for low orders, the number of equivalence classes becomes quickly intractable. In practice, the unigrams, bigrams and trigrams are commonly used. They are shown in Eq. (6-3), Eq. (6-4), and Eq. (6-5), respectively.

$$P(W) \approx \prod_{i=1}^{l} P(w_i) \quad (6\text{-}3)$$

$$P(W) \approx \prod_{i=1}^{l} P(w_i|w_{i-1}) \quad (6\text{-}4)$$

$$P(W) \approx \prod_{i=1}^{l} P(w_i|w_{i-2} w_{i-1}) \quad (6\text{-}5)$$

The probabilities $P\left(w_i \middle| w_{i-n+1}^{i-1}\right)$ are estimated from a corpus of training texts using Maximum Likelihood (ML) estimation, namely, by counting the number of times a certain sequence of *n* characters appears in the corpus of training texts, given by

$$P\left(w_i \middle| w_{i-n+1}^{i-1}\right) = \frac{count\left(w_{i-n+1}^{i}\right)}{count\left(w_{i-n+1}^{i-1}\right)} = \frac{C\left(w_{i-n+1}^{i}\right)}{C\left(w_{i-n+1}^{i-1}\right)} \tag{6-6}$$

where $C(\cdot)$ denotes the number of times the argument is counted from the given training corpus.

The model resulting from Eq. (6-6) maximizes the likelihood of the training corpus, which used to obtain the language models. The n-gram statistics language models have several advantages, such as the quick speed due to probabilities of n-gram are stored in pre-computed tables, simple calculation, and generality due to models can be applied to any domain or language, as long as there exists some training corpus.

For the character-position-free on-line handwritten text recognition, we choose the trigram language model which combined not only trigram, but also bigram and unigram models, with considering the computation complexity and effectiveness.

Following the ML estimation, however, the n-gram models face an important problem due to no corpus is large or wide enough to contain all possible n-grams, namely, all the texts of n characters not appearing in the training corpus have zero probability. Moreover, many n-grams appear too few times to allow a good statistical estimation of their probability $P\left(w_i \middle| w_{i-n+1}^{i-1}\right)$. In order to solve this problem, the smoothing techniques is applied. We will introduce it in the next section.

6.1.2 Smoothing algorithms

As many of n-gram probability estimates are going to be zero due to it is impossible that all words are seen in the training text corpus. Whenever a character string W with $P(W) = 0$ during a text recognition task, that is, the character string should not occur, which is too hard discrimination for handwritten text recognition, a recognition error will be made. It helps prevent errors to assign all character strings in non-zero probabilities for handwritten text recognition.

Smoothing technique is used to overcome this problem, i.e. zero probabilities of the unseen n-grams in the given text corpus by redistributing probabilities between seen and unseen events. Smoothing techniques produce more accurate probabilities by adjusting the maximum likelihood estimate of probabilities. Typically, smoothing methods prevent any probability from being zero, but they also attempt to improve the accuracy of the model as a whole. The name smoothing comes from the fact that these techniques tend to make distribution more uniform, which can be viewed as making them smoother. Especially for the very low probabilities such as zero probabilities are adjusted upward, and high probabilities are adjusted downward.

One simple way of smoothing technique used in practice is the additive smoothing [97], also called Laplace smoothing, which is to pretend each n-gram occurs slightly more often than it actually does for avoiding zero probabilities. Eq. (6-6) is then transformed by following this additive smoothing as follows:

$$P_{add}\left(w_i \middle| w_{i-n+1}^{i-1}\right) = \frac{C\left(w_{i-n+1}^{i}\right) + \delta}{C\left(w_{i-n+1}^{i-1}\right) + \delta|V|} \tag{6-7}$$

where δ is a constant, and subjected to $0 < \delta \leq 1$. V is the vocabulary, or set of all characters considered.

The δ is generally considered as 1, and called add-one smoothing. Let us consider the application of add-one smoothing to bigram probabilities, Eq. (6-7) is simplified as follows:

$$P_{+1}(w_i|w_{i-1}) = \frac{C(w_{i-1}w_i) + 1}{C(w_{i-1}) + |V|} \tag{6-8}$$

Many other smoothing techniques have been introduced in the literature [97]. Such as the simple interpolation, Katz smoothing, Backoff Kneser-Ney smoothing and Interpolated Kneser-Ney smoothing. We will describe the simple interpolation algorithm.

The simple interpolation is a combine techniques in language modeling to simply interpolate them together. For instance, if one has a trigram model, a bigram model, and a unigram model, then

$$P_{Interpolation}(w_i|w_{i-2}w_{i-1}) = \lambda_1 P(w_i|w_{i-2}w_{i-1}) + \lambda_2 P(w_i|w_{i-1}) + \lambda_3 P(w_i) \tag{6-9}$$

where λ_1, λ_2 and λ_3 are parameters with constraint that $0 \leq \lambda_1, \lambda_2 \text{ and } \lambda_3 \leq 1$.

In practice, to ensure no word is assigned zero probability, we commonly interpolate with the uniform distribution $P(w_i) = 1/size\ of\ vocabulary$, we also need to deal with the case when, for instance, the trigram context $c_{i-2}c_{i-1}c_i$ has never been seen, namely $C(w_{i-2}w_{i-1}w_i) = 0$. In this case, we use an interpolated bigram model, etc. Given its simplicity, simple interpolation works surprisingly well, but other techniques, such as Katz smoothing, work even better, but need much more training corpus.

In our study, we use this smooth method for trigram language model, which combines the unigram, bigram and trigram with parameters, where the parameters subject to $\lambda_1 + \lambda_2 + \lambda_3 = 1$. It is reduced to unigram or bigram when w_i is the first or second character of a sentence. Moreover, to reduce the model size, we set empirically a threshold to prune the low trigrams probabilities.

6.2 Geometric context

In handwritten Japanese/Chinese text recognition, over-segmentation-based method is commonly employed to overcome the character segmentation problem, due to it is infeasible to segment character reliably prior to recognition. The geometric context, which includes the compatibility of character size, position and between-character relationship with respect to the text layout, can help disambiguate the uncertainty in character segmentation. Especially in Japanese language, the position of the small Kana such as “tsu” is obviously different with the normal Kanji. Furthermore, the recognition accuracy will be improved by incorporating the geometric context with character recognition and linguistic context in the candidate segmentation and recognition path evaluation.

In on-line handwritten Japanese text recognition, Nakagawa et al. [23] incorporated the likelihood of geometric features into the path scores, but only simple features are used, such as character size, inter-character and between-character gap. Zhou et al. [27] incorporated the geometric context with character recognition and linguistic context into a united framework to overcome the effect of string length variability and improve the recognition performance. Here, the geometric context models are made by a statistical method, including class-dependent unary and binary geometric features with more features. Recently, Zhu et al. [5] combined the more geometric context with the character recognition, linguistic context and character segmentation to improve recognition accuracy. The geometric features include the size feature, character inner-gap feature, and class-dependent unary and binary position features.

In on-line handwritten Chinese text recognition, the employed geometric context including more features (class-dependent unary and binary geometric features, and class-independent unary and binary geometric features) [36], [38]. Compared to [5], the mainly difference is that it uses class-independent unary and binary geometric features, using simple SVM model. Yin et al. [99] integrated the geometric features to improve the performance of text alignment in Chinese annotation system. Wu et al. [100] proposed an improved binary geometric model that combines single-character and between-character features to improve significantly the numeral string recognition performance on the NIST special database 19.

We will introduce the geometric features used in the character-position-free handwritten text recognition system, including the character size feature, character inner-gap feature, and unary position feature of a candidate character pattern, and binary position feature between candidate character patterns.

The character size feature (or shape feature), namely the term b_i in Eq. (5-7), is composed of the height and width of the bounding box of a candidate character pattern. Figure 6-2 shows an example of size features of candidate character patterns.

The character inner-gap feature of a candidate pattern, namely the term q_i in Eq. (5-7), is obtained by projecting the candidate character pattern into the vertical and horizontal directions, splitting each of their histograms into 3 slices, finding a gap or gaps in each slice, and summing total lengths of gaps. Hence, the inner-gap feature vector includes 6 values. Figure 6-3 shows an example of the inner-gap feature of a character pattern.

The class-dependent unary position feature, namely the term p_i^u in Eq. (5-7), consists of two vertical distances from the horizontal center of a text line to the top and bottom of the bounding box of a candidate character pattern, as shown in Figure 6-4.

The class-dependent binary position feature, namely the term p_i^b in Eq. (5-7), is composed of a vertical distance between the top edges of the bounding boxes of two adjacent candidate character patterns in a text line and that between the bottom edges of the bounding boxes, as shown in Figure 6-5.

Due to Japanese and Chinese language are large character set including thousands of character classes, it is almost impossible to get sufficient training samples covering every class pair. A feasible method is that using cluster method to reduce the number of classes. The character classes are then clustered into six super-classes by grouping the mean vectors of the unary geometric features of all character classes on a training text set using the k-means algorithm. Hence, a pair of successive characters belong to one of 36 binary super-classes. The training text character samples, re-labeled to six unary super-classes, are used to estimate the Gaussian probability density functions of 36 binary super-classes. Then, the binary geometric probability $P\left(p_i^b \middle| c_{i-1}, c_i\right)$ is substituted by $P\left(p_i^b \middle| \tilde{c}_{i-1}, \tilde{c}_i\right)$, where $\tilde{c}_{i-1}$ and $\tilde{c}_i$ are the unary supper-classes of c_{i-1} and c_i.

We normalize the above 4 features (b_i, q_i, p_i^u, and p_i^b) by the *acs* (average character size). Then, we assume $P(b_i|c_i)$, $P(q_i|c_i)$, $P(p_i^u|c_i)$, and $P\left(p_i^b \middle| \tilde{c}_{i-1}, \tilde{c}_i\right)$ to be normal distributions and model their logarithms by a quadratic discriminant function (QDF).

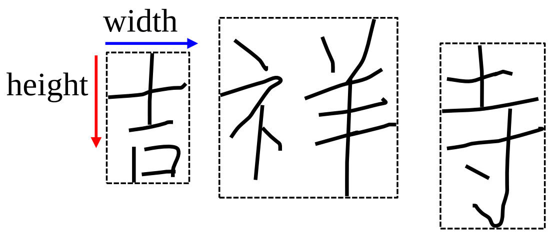


Figure 6-2 Shape features of character patterns.

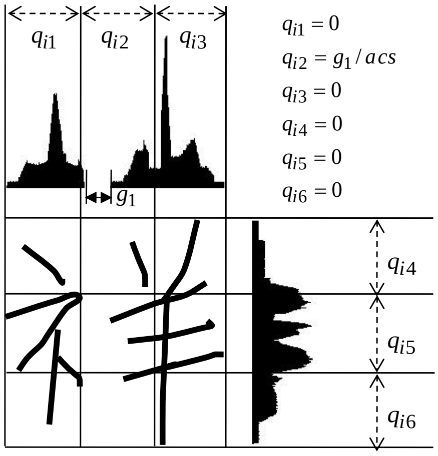


Figure 6-3 Inner-gap feature of a character pattern.

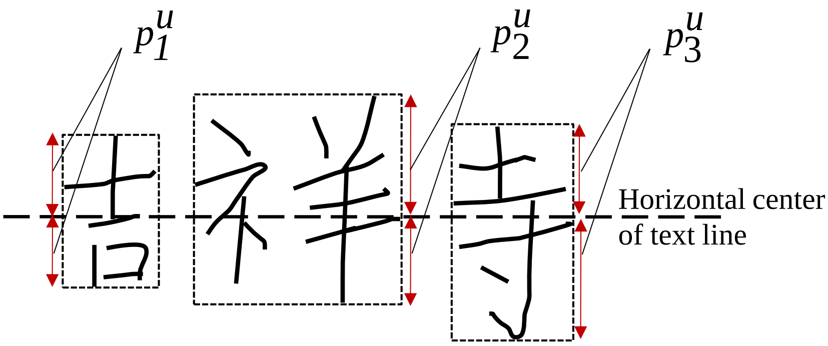


Figure 6-4 Unary position features of character patterns.

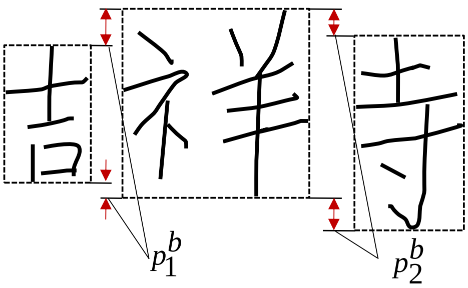


Figure 6-5 Binary position features between adjacent character patterns.

7. Experiments

In this chapter, we describe several experiments on generated character-position-free on-line handwritten Japanese and Chinese text datasets, and collected handwritten Japanese text patterns, as well as normally handwritten Japanese text patterns. Moreover, we compare the results of proposed segmentation methods with the original recognizer, and give some analyses of recognition performance.

7.1 Datasets

The character-position-free on-line handwritten Japanese and Chinese text datasets are generated by following the generation models described in Chapter 3. We call Japanese and Chinese text datasets as Dataset *M* (*M*=1 to 4) and ch_Dataset *M* (*M*=1 to 4), respectively.

The collected Japanese text patterns written without supported by wrist or elbow and without visual feedback from 10 participants as described in Chapter 3. We call this set of collected sample patterns as Dataset 5. The statistics of these datasets is shown in Table 7-1.

Table 7-1 Statistics of datasets.

Dataset	#Writers	#Text lines	#Character patterns	#Character categories
Kondate_h	100	13,685	139,779	1,161
Dataset *M* (*M*=1 to 4)	100	15,389	129,076	1,123
Dataset 5	10	470	3,580	198
Training set of ch_Dataset *M* (*M*=1 to 4)	240	44,903	293,198	2,175
Test set of ch_Dataset *M* (*M*=1 to 4)	60	11,246	73,301	2,173

For Dataset 1 to 4 of handwritten Japanese text patterns, we use a 4-fold cross-validation method to evaluate the performance of recognizers. For Dataset 5, however, we employ a simple method since it is too small to make a cross validation.

We divide each Dataset *M* (*M*=1 to 4) into 4 groups: Dataset_*M_D1* (patterns by writer 1 ~ writer 25), Dataset_*M_D2* (writer 26 ~ writer 50), Dataset_*M_D3* (writer 51 ~ writer 75), and Dataset_*M_D4* (writer 76 ~ writer 100), each group includes 25 peoples' patterns, and let 3 groups (75 peoples' patterns) as training patterns, the remaining 1 group (25 peoples' patterns) as testing patterns. Moreover, to let our proposed model recognize patterns of all the datasets with the same set of parameters, we combine all the training patterns (75×4) as the total training patterns. That is, the combined training patterns are shared among the following experiments but the testing patterns are used from each Dataset. If we use the training patterns and testing patterns in each dataset, we get slightly better results but it seems unfair since we cannot predict which model is appropriate.

7.2 Settings

For character-position-free handwritten Japanese text patterns, we compare our candidate segmentation method (CSM) and undecided segmentation method (USM) with the original recognizer. It was developed for normally handwritten horizontal Japanese text patterns based on the method [5] and reduced in size by feature selection, LDA, vector quantization and data type transformation. This comparison is made by changing over-segmentation and replacing the candidate character segmentation probability while succeeding the character recognition, unary and binary position features, size and inner-gap features, and linguistic context to evaluate the candidate segmentation-recognition paths.

As for the training the parameters in the path evaluation function defined in Eq. (5-7), we choose the MCE criterion to train them on the training text patterns combined from dataset 1 to dataset 4. We initially set all parameters as 1, the initial learning step as 0.004, and the parameter of ξ as 0.9 in Eq. (5-14). The parameters are updated iteratively following Eq. (5-14). The beam width of beam search is being set as 2.

As for the character recognition, which combines on-line and off-line character recognizers by a linear function as described in Chapter 4, where the combing parameters are trained by Nakayosi database [90], we keep top 10 candidate character classes for each candidate character pattern. We also use Nakayosi database to train geometric feature functions: character unary and binary position features, character size and inner-gap features.

As for the linguistic context model, it is trained on the year 1993 volume of the ASAHI newspaper and the year 2002 volume of the NIKKEI newspaper. We estimate the smoothing parameters $(\beta_1 = 0.7, \beta_2 = 0.2, \beta_3 = 0.1)$ in Eq. (6-7) by Nakayosi database.

As for the SVM model, we train it on off-strokes of the combined training text patterns. However, the number of off-strokes is so large (more than a million), we use 1/10 of them as training data. Moreover, we choose the following radial basis as kernel function:

$$K(x_i, x_j) = exp\frac{\left\|x_i - x_j\right\|^2}{2\sigma^2} \tag{7-1}$$

We obtained σ and C as shown in Eq. (5-3) by examining several values in experiments using training data. We then obtained the parameters of the separating hyperplanes for the SVM model using the same training data again.

On the other hand, to evaluate the effectiveness of the segmentation probability by the SVM model, we consider the cases when the segmentation scores are not combined in the path evaluation function for Viterbi search, namely, by setting 0 for λ_{71} and λ_{72} in Eq. (5-7), for candidate segmentation method and undecided segmentation method. Then we adjust the other parameters in Eq. (5-7) on the combined training patterns by MCE. We call them the candidate segmentation method without segmentation scores (CSM_w/o *Ss*) and the undecided segmentation method without segmentation scores (USM_w/o *Ss*), respectively.

For character-position-free handwritten Chinese text patterns, we also compare the candidate segmentation method and undecided segmentation method, using the similar method as for handwritten Japanese text patterns.

We trained the on-line and off-line Chinese character recognizers, the combination weighting parameters of on-line and off-line character recognizers, and geometric scoring functions using the on-line handwriting Chinese database CASIA-HWDB1.0-1.2 [39], which have 7,356 classes, including 7,184 Chinese characters and 172 symbols.

As for the Chinese linguistic context model, it is trained on a corpus of People's Daily. For the SVM model, we train it on off-strokes of the combined training text patterns from training set of ch_Dataset 1 to 4. However, the number of off-strokes is so large (more than a million), we use 1/100 of them as training data. Moreover, we use the radial basis as kernel function.

As for the weight parameters of the path evaluation, we train them by MCE criterion on the text patterns combining all training set of ch_Dataset 1 to 4. We test the performance of the character-position-free Chinese text recognizer on the text lines from test set of ch_Dataset 1 to that of ch_Dataset 4, respectively.

7.3 Results of Experiments

We use the recognition rate (R_c) defined in Eq. (7-2), the segmentation measure (F) defined in Eq. (7-3), which combines recall and precision rates, and the average recognition time cost per character pattern (T_c), to evaluate the text recognizers.

All the experiments are made on a PC with Intel(R) Core(TM) i7-3770 CPU @3.40GHz 3.40GHz (2 processers) and 8 GB memory.

$$R_c = \frac{\#number\ of\ correctly\ recognized\ characters}{\#number\ of\ all\ characters} \tag{7-2}$$

$$F = \frac{2}{1/R + 1/P}$$

$$R = \frac{\#number\ of\ correctly\ detected\ segmentation\ points}{\#number\ of\ true\ segmentation\ points}$$

$$P = \frac{\#number\ of\ correctly\ detected\ segmentation\ points}{\#number\ of\ true\ segmentation\ points} \tag{7-3}$$

7.3.1 Character-position-free Japanese text patterns

We abbreviate the candidate segmentation method as CSM, and the undecided segmentation method as USM in the following tables. Table 7-2 to Table 7-5 shows the recognition performance for Dataset 1 to Dataset 4, respectively.

Table 7-2 Recognition performance on Dataset 1.

Performance Method	R_c (%)	F	T_c (s)
Original recognizer	23.72	0.4902	0.012
CSM	89.95	0.9671	0.089
USM	91.83	0.9751	0.139
CSM_w/o *Ss*	89.72	0.9646	0.086
USM_w/o *Ss*	91.37	0.9693	0.125

Table 7-3 Recognition performance on Dataset 2.

Performance Method	R_c (%)	F	T_c (s)
Original recognizer	39.61	0.6627	0.013
CSM	90.99	0.9728	0.078
USM	92.23	0.9771	0.121
CSM_w/o *Ss*	90.81	0.9707	0.075
USM_w/o *Ss*	91.80	0.9722	0.100

Table 7-4 Recognition performance on Dataset 3.

Method \ Performance	R_c (%)	F	T_c (s)
Original recognizer	0.00	0.00	
CSM	91.31	0.9784	0.299
USM	92.34	0.9795	0.585
CSM_w/o *Ss*	91.09	0.9760	0.283
USM_w/o *Ss*	91.85	0.9733	0.557

Table 7-5 Recognition performance on Dataset 4.

Method \ Performance	R_c (%)	F	T_c (s)
Original recognizer	0.04	0.1600	
CSM	83.26	0.9273	0.111
USM	84.97	0.9325	0.133
CSM_w/o *Ss*	82.94	0.9274	0.111
USM_w/o *Ss*	84.01	0.9294	0.128

For Dataset 5 (collected sample patterns), we firstly prepare the combined text patterns: {Dataset_*M_D1*, Dataset_*M_D2*, Dataset_*M_D3*} (*M*=1 to 4), to train both the segmentation methods and to tune the parameters of the path evaluation function by MCE; then, we increase the number of candidate character classes for each candidate character pattern from 10 to 15 (top 15), further update the parameters of the path evaluation function with adding 3/5 of collected text patterns by the genetic algorithm, and test the performance to the remaining 2/5. Table 7-6 shows the recognition performance for Dataset 5.

Table 7-6 Recognition performance on 2/5 of Dataset 5.

Method \ Performance	R_c (%)	F	T_c (s)
Original recognizer	50.84	0.7670	0.015
CSM	74.86	0.9033	0.119
USM	76.19	0.9046	0.159

The final experiment is to confirm the performance of CSM and USM on normally handwritten text patterns. Table 7-7 shows the results. For comparison, we also show the performance of the original recognizer (which is CSM) and that with CSM being replaced by USM, i.e., all off-strokes being set as UP (named Original recognizer with USM). Note that parameters for these two recognizers have been tuned for normally handwritten text.

Table 7-7 Recognition performance on Kondate_h.

Method \ Performance	R_c (%)	F	T_c (s)
Original recognizer	93.40	0.9917	0.012
Original recognizer with USM	90.04	0.9542	0.050
CSM	90.38	0.9711	0.050
USM	92.04	0.9814	0.070

Table 7-7 shows interesting results. When the parameters are tuned for normally handwritten text, CSM (Original recognizer) is significantly more accurate and quicker than USM (Original recognizer with USM). On the other hand, when the parameters are tuned for character-position-free handwritten text, USM is more accurate with slightly higher time cost. This will be discussed in the next subsection.

7.3.2 Character-position-free Chinese text patterns

We investigated the recognition performances of the two segmentation methods (CSM and USM) on the character-position-free on-line handwritten Chinese text patterns, which generated from the test set of CASIA-OLHWDB2.1 database. Due to the existing Chinese text recognizer is designed for normally handwritten text pattern, and it is obvious that it cannot resolve the text patterns with partly or fully overlapped characters from the previous section, hence, we did not evaluate the recognition performance of the existing text recognizer on these generated character-position-free text patterns. Table 7-8 to Table 7-11 shows the recognition performance for ch_Dataset 1 to ch_Dataset 4, respectively.

Table 7-8 Recognition performance on ch_Dataset 1.

Method \ Performance	R_c (%)	F	T_c (s)
CSM	79.73	0.8921	0.178
USM	83.42	0.9163	0.336

Table 7-9 Recognition performance on ch_Dataset 2.

Performance / Method	R_c (%)	F	T_c (s)
CSM	81.41	0.9025	0.149
USM	84.09	0.9213	0.267

Table 7-10 Recognition performance on ch_Dataset 3.

Performance / Method	R_c (%)	F	T_c (s)
CSM	82.80	0.9174	0.559
USM	83.60	0.9220	0.687

Table 7-11 Recognition performance on ch_Dataset 4.

Performance / Method	R_c (%)	F	T_c (s)
CSM	74.34	0.8407	0.274
USM	77.15	0.8662	0.372

7.4 Analysis and Discussion

We will mainly give the analysis and discussion about the character-level recognition accuracy and recognition speed for character-position-free handwritten Japanese and Chinese text datasets, respectively.

7.4.1 For character-position-free Japanese text patterns

From the results presented in the previous section, both the candidate segmentation method (CSM) and the undecided segmentation method (USM) produce far better recognition rate and segmentation measure for all the datasets than the original recognizer, as shown in Figure 7-1.

Since the average recognition rate by the original recognizer on normally handwritten text patterns is 93.40%, both the candidate segmentation method and the undecided segmentation method have realized almost the similar recognition rate even for character-position-free handwritten text except Dataset 4.

In fact, it is notable that both the candidate segmentation method and the undecided segmentation method achieve 90.38% and 92.04% recognition rate for Kondate_h, respectively, as shown in Table 7-7. This implies that many factors in the evaluation function (Eq. (5-7)) may relax one or two constraints, especially, the segmentation constraint at the sacrifice of the time complexity.

The original recognizer is the integrated segmentation and recognition system incorporating several factors into the evaluation function. It misclassifies true segmentation points as non-segmentation points in the over-segmentation step for the character-position-free on-line handwritten text recognition, however, due to the fully or partially overlaid characters. While the undecided segmentation method sets each off-stroke as undecided point to avoid the misclassification and the candidate segmentation method keeps the true segmentation points as much as possible by adjusting the thresholds for the output of SVM. That is why these two methods can overcome the problem of the original recognizer for the character-position-free on-line handwritten text recognition.

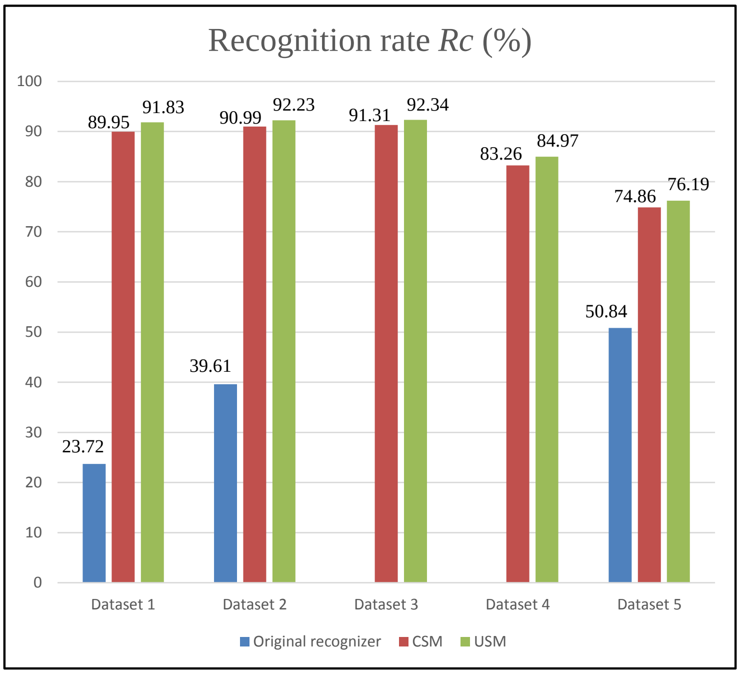
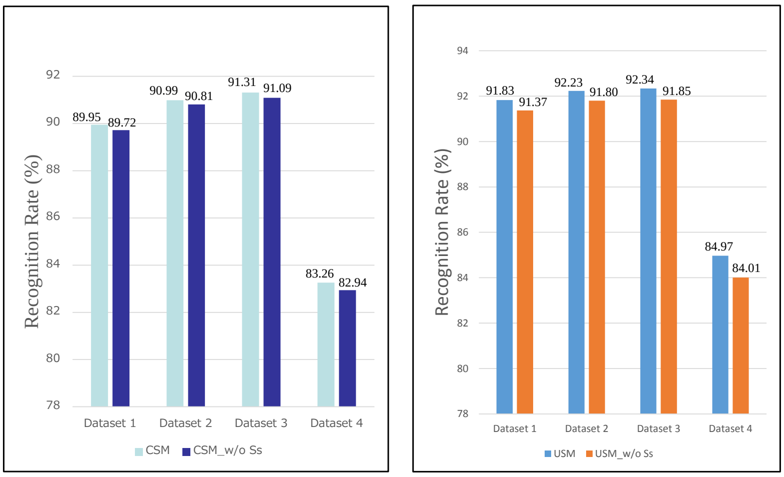


Figure 7-1 Comparison the recognition rates of CSM and USM with the original recognizer for Dataset 1 to 5.

As for the effectiveness of the segmentation probability by the SVM model, it is clarified that unemployment of the segmentation scores decreases the recognition rate of the undecided segmentation method by 0.46 point, 0.43 point, 0.49 point and 0.96 point for Dataset 1 to Dataset 4, respectively, and that of the candidate segmentation method by 0.23 point, 0.18 point, 0.22

point and 0.32 point for Dataset 1 to Dataset 4, respectively, as shown in Figure 7-2. The recognition rate of the candidate segmentation method decreases less than the undecided segmentation method, because the former classifies off-strokes into non-segmentation point, segmentation point, and undecided one based on the output of the SVM model. Hence, it is better to combine the segmentation scores in the path evaluation for Viterbi search.



(a) (b)

Figure 7-2 Effectiveness of the character segmentation probability for CSM (a) and USM (b).

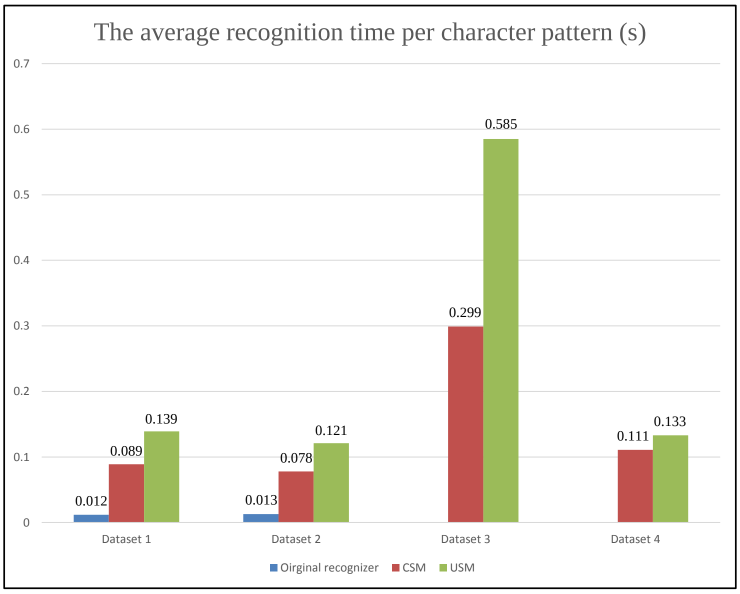


Figure 7-3 Comparison the average recognition time per character pattern of CSM and USM with the original recognizer for Dataset 1 to 4.

As for the recognition speed, except Dataset 3, the average recognition time cost per character pattern is 0.093 second and 0.131 second, by the candidate segmentation method and the undecided segmentation method, respectively, whereas the original method recognizer takes 0.012 second for on normally handwritten text. This is because most off-strokes are classified into undecided points in the candidate segmentation method, and all off-strokes are set to undecided points, so that the constructed candidate lattice becomes so large.

Moving to the comparison between the candidate segmentation method and the undecided segmentation method, the recognition rate by the latter is superior to the former by 1.88 point, 1.24 point, 1.03 point, and 1.71 point, for Dataset 1, 2, 3, and 4, respectively, as shown in Figure 7-1. On the other hand, the average time cost per character by the latter increases 0.050 second, 0.043 second, 0.286 second, and 0.022 second, for Dataset 1, 2, 3, and 4, respectively, which is about 1.20~1.96 times the former, as shown in Figure 7-3, but it is not a problem for real-time recognition.

Dataset 3 is a set of text patterns where characters are completely overlaid. The overlaid handwritten character recognition has been proposed for small surface devices. Our experiments show that it can be treated uniformly by the character-position-free handwritten text recognition.

The recognition rate is even slightly higher for Dataset 3. The off-strokes between characters move generally from bottom-right to top-left, so that character segmentation reliability could be enhanced and the character recognition rate is improved slightly from other cases.

On the other hand, the time cost is about 4 times larger than the others. The reason is as follows: candidate character patterns whose widths are longer than the threshold are deleted from the candidate lattice as described in Chapter 5, but no candidate is deleted for Dataset 3. Even in this case, however, the time is about 0.3 or 0.6 second for a character. This is not a serious problem for practical applications since it is far quicker than handwriting by people.

For Dataset 5 of collected handwritten text samples, the recognition rate is improved from the original recognizer but still lower than the others. This would be partially because the dataset is too small and partially because some strokes in character patterns are so largely displaced or deformed so that those patterns are not character-position-free but stroke-position-free. The amount of those patterns is not large but not negligible. Removing all the position information could be considered but would damage total character recognition considerably. To cope with them is our next challenge.

As for the comparison between the undecided segmentation method and the candidate segmentation method in general, the undecided segmentation method is more accurate with slightly higher time cost for character-position-free handwritten text, but the undecided segmentation method is inferior to the candidate segmentation method for normally handwritten text patterns in both accuracy and speed as shown in Table 7-7. This was exactly the reason that we employed the candidate segmentation method for ordinary text input. When the character position information is reliable, it must be exploited.

7.4.2 For character-position-free Chinese text patterns

For the results of character-position-free Chinese text recognition presented in the previous section 7.3, the undecided segmentation method (USM) produces better recognition and segmentation measure for all the ch_Datasets than the candidate segmentation method (CSM), as shown in Figure 7-4.

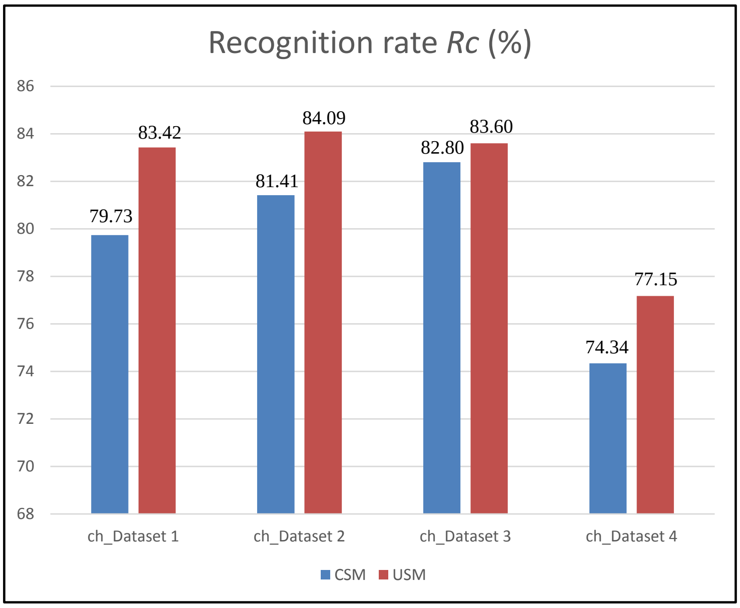


Figure 7-4 Comparison the recognition rates of CSM with USM for all ch_Datasets.

Since the recognition rate by the existing original recognizer on normally handwritten Chinese text patterns is 83.62%, both the candidate segmentation method and the undecided segmentation method have realized almost the similar recognition rate even for character-position-free handwritten text except ch_Dataset 4.

8. Conclusion and future work

In this chapter, we draw the conclusion of this thesis, and give several directions for the future works.

8.1 Conclusion

In this thesis, we have proposed a method to recognize character-position-free on-line handwritten text patterns, and investigated the recognition performance on generated character-position-free handwritten Japanese and Chinese text patterns. We have considered two segmentation methods, one classifies each off-stroke into non-segmentation point, segmentation point, or undecided point according to the output of SVM model, the other directly sets each off-stroke as undecided point. The results of experiments confirmed that the proposed method achieves the best recognition performance for character-position-free text patterns by the undecided segmentation method, approaching the performance of the original recognizer on normally handwritten text patterns. The undecided segmentation method for Japanese text patterns has been employed for the EMIRAI 3 xDAS assisted-driving concept car [104].

We have also considered the case that characters are completely overlaid in this framework, and have shown that the proposed method works well as for other cases.

8.2 Future work

Although we have realized the character-position-free on-line recognition for handwritten Japanese and Chinese text patterns, it yet needs to improve the recognition performance including not only the recognition accuracy but also the recognition speed, especially for applications on the pen-based or touch-based handheld devices with low power.

First, we need to collect a large set of real patterns in automobile environment. Although we simulated physical conditions to write characters, mental conditions while driving a car with watching the frontal view and predicting the other objects' movement could not be reflected to simulated patterns. Real patterns must be also collected from small-surface environment.

Second, from the results of experiments, the character segmentation probability by the SVM model plays an important role in the handwritten text recognition. In candidate segmentation method, however, based on the output of SVM model, it sometimes misclassifies the true segmentation points as non-segmentation points, and true non-segmentation points as segmentation points. We need to try other methods to get a more robust classification result, such as Recurrent Neural Networks (RNNs) with the long short-term memory (LSTM) architecture [105], which are able to effectively utilize the contextual information.

Third, we should reduce the recognition speed time by refining the redundant candidate lattice. There are many obvious non-character candidate patterns in the candidate lattice, we may employ effective geometric features to remove them reliably.

Finally, we should enhance the method to recognize stroke-position-free character patterns to some extent by extracting more geometric and linguistic features.

References

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

[2] M. Nakagawa, "Non-keyboard input of Japanese text—on-line recognition of handwritten characters as the most hopeful approach," Journal of Information Processing, vol. 13, no. 1, pp. 15-34, 1990.

[3] "Anoto," [Online]. Available: http://www.anoto.com

[4] Japan Automobile Manufacturers Association, Inc. (JAMA), "Guidelines for in-vehicle display systems-version 3.0," 2004.

[5] B. Zhu, X.D. Zhou, C.L. Liu and M. Nakagawa, "A robust model for on-line handwritten Japanese text recognition," International Journal on Document Analysis and Recognition, vol. 13, no. 2, pp. 121-131, 2010.

[6] R. Plamondon and S.N. Srihari, "On-line and off-line handwriting recognition: a comprehensive survey," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 22, no. 1, pp. 63-84, 2000.

[7] C.L. Liu, S. Jaeger and M. Nakagawa, "Online recognition of Chinese characters: the state-of-the-art," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 26, no. 2, pp. 198-213, 2004.

[8] C.L. Liu, H. Sako and H. Fujisawa, "Effects of classifier structures and training regimes on integrated segmentation and recognition of handwritten numeral strings," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 26, no. 11, pp. 1395-1407, 2004.

[9] B. Zhu and M. Nakagawa, "Recent trends in online handwritten character recognition," (in Japanese) IEICE Trans. Information and Systems, vol. J95-D, no. 4, pp. 335-340, 2012.

[10] M. Cheriet, N. Kharma, C.L. Liu and C.Y. Suen, "Character recognition systems - A guide for students and practitioners," John Wiley & Sons, Inc., Hoboken, New Jersey, 2007.

[11] M. Mohamed and P. Gader, "Handwritten word recognition using segmentation-free hidden Markov modeling and segmentation-based dynamic programming techniques," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 18, no. 5, pp. 548-554, 1996.

[12] H. Murase, T. Wakahara and M. Umeda, "Online writing-box free character string recognition by candidate character lattice method," (in Japanese) IEICE Trans. Information and Systems, vol. J68-D, no. 4, pp.765-772, 1985.

[13] L.Y. Tseng and R.C. Chen, “Segmenting handwritten Chinese characters based on heuristic merging of stroke bounding boxes and dynamic programming,” Pattern Recognition Letters, vol. 19, no. 10, pp. 963-973, 1998.

[14] Y. Lu, C.L. Tan, P.F. Shi, and K.H. Zhang, “Segmentation of handwritten Chinese characters from destination addresses of mail pieces,” International Journal of Pattern Recognition and Artificial Intelligence, vol. 16, no. 1, pp. 85-96, 2002.

[15] S. Zhao, Z. Chi, P. Shi and H. Yan, “Two-stage segmentation of unconstrained handwritten Chinese characters,” Pattern Recognition, vol. 36, no. 1, pp. 145-156, 2003.

[16] X. Wei, S. Ma, and Y. Jin, “Segmentation of connected Chinese characters based on genetic algorithm,” Proceedings of the 8th International Conference on Document Analysis and Recognition, Seoul, Korea, pp. 645-649, 2005.

[17] Z. Liang and P. Shi, “A metasynthetic approach for segmenting handwritten Chinese character strings,” Pattern Recognition Letters, vol. 26, no. 10, pp. 1498-1511, 2005.

[18] N. Furukawa, J. Tokuno and H. Ikeda, “Online character segmentation method for unconstrained handwriting strings using off-stroke features”, Proceedings of the 10th International Workshop on Frontiers in Handwriting Recognition, La Baule, France, 2006.

[19] X. Gao, P.M. Lallican and C. Viard-Gaudin, “A two-stage online handwritten Chinese character segmentation algorithm based on dynamic programming,” Proceedings of the 8th International Conference on Document Analysis and Recognition, Seoul, Korea, pp. 735-739, 2005.

[20] Q. Fu, X.Q. Ding, T. Liu, Y. Jiang and Z. Ren, “A novel segmentation and recognition algorithm for Chinese handwritten address character strings,” Proceedings of the 18th International Conference on Pattern Recognition, pp. 974-977, 2006.

[21] Q.F. Wang, F. Yin and C.L. Liu, “Handwritten Chinese text recognition by integrating multiple contexts,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 34, no. 8, pp. 1469-1481, 2012.

[22] S. Senda and K. Yamada, “A maximum-likelihood approach to segmentation-based recognition of unconstrained handwriting text,” Proceedings of the 6th International Conference on Document Analysis and Recognition, Seattle, USA, pp. 184-188, 2001.

[23] M. Nakagawa, B. Zhu and M. Onuma, “A model of on-line handwritten Japanese text recognition free from line direction and writing format constraints,” IEICE Trans. Information and Systems, vol. E88-D, no. 8, pp. 1815-1822, 2005.

[24] N.X. Li and L.W. Jin, “A Bayesian-based method of unconstrained handwritten offline Chinese text Line recognition,” International Journal on Document Analysis and Recognition, vol. 16, no. 1, pp. 17-31, 2013.

[25] R.O. Duda, P.E. Hart and D.G. Stork, "Pattern classification," John Wiley & Sons, 2012.
[26] X.D. Zhou, C.L. Liu, and M. Nakagawa, "Online handwritten Japanese character string recognition using conditional random fields," Proceedings of the 10th International Conference on Document Analysis and Recognition, Barcelona, Spain, pp. 521-525, 2009.
[27] X.D. Zhou, J.L. Yu, C.L. Liu, T. Nagasaki and K. Marukawa, "Online handwritten Japanese character string recognition incorporating geometric context," Proceedings of the 9th International Conference on Document Analysis and Recognition, Curitiba, Brazil, pp. 48-52, 2007.
[28] T.M. Breuel, "A system for the off-line recognition of handwritten text," Proceedings of the12th International Conference on Pattern Recognition, vol. 2, Jerusalem, Israel, pp. 129-134, 1994.
[29] M.S. Kim, S. Ryu, K.T. Cho, T.H. Rhee, H.I. Choi and J.H. Kim, "Recognition-based digitalization of Korean historical archives," Proceedings of the Asia Information Retrieval Symposium, Beijing, China, pp. 281-288, 2004.
[30] Q. Fu, X.Q. Ding, C.S. Liu, Y. Jiang and Z. Ren, "A hidden Markov model based segmentation and recognition algorithm for Chinese handwritten address character strings," Proceedings of the 8th International Conference on Document Analysis and Recognition, Seoul, Korea, pp. 590-594, 2005.
[31] X.Q. Ding and H.L. Liu, "Segmentation-driven offline handwritten Chinese and Arabic script recognition," Proceedings of the Summit on Arabic and Chinese Handwriting, College Park, USA, pp. 61-73, 2006.
[32] S. Tulyakov and V. Govindaraju, "Probabilistic model for segmentation based word recognition with lexicon," Proceedings of the 6th International Conference on Document Analysis and Recognition, Seattle, USA, pp. 164-167, 2001.
[33] B. Zhu, J. Gao and M. Nakagawa, "Objective function design for MCE-based combination of on-line and off-line character recognizers for on-line handwritten Japanese text recognition," Proceedings of the 11th International Conference on Document Analysis and Recognition, Beijing, China, pp.594-598, 2011.
[34] J. Gao, B. Zhu and M. Nakagawa, "Development of a robust and compact on-line handwritten Japanese text recognizer for hand-held devices," IEICE Trans. Information and Systems, vol. E96-D, no. 4, pp. 927-938, 2013.
[35] J. Gao, B. Zhu and M. Nakagawa, "Building compact recognizer with recognition rate maintained for on-line handwritten Japanese text recognition," Pattern Recognition Letters, vol. 35, pp. 169-177, 2014.

[36] D.H. Wang, C.L. Liu and X.D. Zhou, “An approach for real-time recognition of online Chinese handwritten sentences,” Pattern Recognition, vol. 45, no. 10, pp. 3661-3675, 2012.

[37] T.H. Su, T.W. Zhang and D.J. Guan, “Corpus-based HIT-MW database for offline recognition of general-purpose Chinese handwritten text,” International Journal of Document Analysis and Recognition, vol. 10, no. 1, pp. 27-38, 2007.

[38] X.D. Zhou, D.H. Wang, F. Tian, C.L. Liu and M. Nakagawa, “Handwritten Chinese/Japanese text recognition using semi-Markov conditional random fields,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 35, no. 10, pp. 2413-2426, 2013.

[39] C.L. Liu, F. Yin, D.H. Wang, and Q.F. Wang, “CASIA Online and Offline Chinese Handwriting Databases,” Proceedings of the 11th International Conference on Document Analysis and Recognition, Beijing, China, pp. 37-41, 2011.

[40] B.H. Juang, W. Hou and C.H. Lee, “Minimum classification error rate methods for speech recognition,” IEEE Trans. Speech and Audio Processing, vol. 5, no. 3, pp. 257-265, 1997.

[41] H. Robbins and S. Monro, “A stochastic approximation method,” The Annals of Mathematical Statistics, pp.400-407, 1951.

[42] H. Ney and S. Ortmanns, “Progress in dynamic programming search for LVCSR,” Proceedings of the IEEE, vol. 88, no. 8, pp. 1224-1240, 2000.

[43] C.L. Liu, M. Koga and H. Fujisawa, “Lexicon-driven segmentation and recognition of handwritten character strings for Japanese address reading,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 24, no. 11, pp. 1425-1437, 2002.

[44] B. Zhu and M. Nakagawa, “Trie-lexicon-driven recognition for on-Line handwritten Japanese disease names using a time-synchronous method,” Proceedings of the 11th International Conference on Document Analysis and Recognition, Beijing, China, pp. 1130-1134, 2011.

[45] T.H. Su, T.W. Zhang, H.J. Huang and Yu Zhou, “HMM-based recognizer with segmentation-free strategy for unconstrained Chinese handwritten text,” Proceedings of the 9th International Conference on Document Analysis and Recognition, Curitiba, Brazil, vol. 1, pp. 133-137, 2007.

[46] T.H. Su, T.W. Zhang, D.J. Guan and H.J. Huang, “Off-line recognition of realistic Chinese handwriting using segmentation-free strategy,” Pattern Recognition, vol. 42, no. 1, pp. 167-182, 2009.

[47] R. Messina and J. Louradour, "Segmentation-free handwritten Chinese text recognition with LSTM-RNN." Proceedings of the 13th International Conference on Document Analysis and Recognition, Nancy, France, pp. 171-175, 2015.

[48] H. Shimodaira, T. Sudo, M. Nakai, and S. Sagayama, "On-line overlaid handwriting recognition based on substroke HMMs," Proceedings of the 7th International Conference on Document Analysis and Recognition, Edinburgh, Scotland, pp.1043-1047, 2003.

[49] Y. Tonouchi and A. Kawamura, "Text input system using online overlapped handwriting recognition for mobile devices," Proceedings of the 9th International Conference on Document Analysis and Recognition, Curitiba, Brazil, pp.754-758, 2007.

[50] A. Bharath and S. Madhvanath, "FreePad: a novel handwriting-based text input for pen and touch interfaces," Proceedings of the 13th International Conference on Intelligent User Interfaces, Canary Island, Spain, pp.297-300, 2008.

[51] W. Kienzle and K. Hinckley, "Writing handwritten messages on a small touchscreen," Proceedings of the 15th International Conference on Human-Computer Interaction with Mobile Devices and Services, Germany, pp. 179-182, 2013.

[52] Y. Zou, Y. Liu, Y. Liu and K. Wang, "Overlapped handwriting input on mobile phones," Proceedings of the 11th International Conference on Document Analysis and Recognition, Beijing, China, pp.369-373, 2011.

[53] X. Wan, C. Liu and Y. Zou, "On-line Chinese character recognition system for overlapping samples," Proceedings of the 11th International Conference on Document Analysis and Recognition, Beijing, China, pp.799-803, 2011.

[54] Y.F. Lv, L.L. Huang, D.H. Wang and C.L. Liu, "Learning-based candidate segmentation scoring for real-time recognition of online overlaid Chinese handwriting," Proceedings of the 12th International Conference on Document Analysis and Recognition, Washington, DC, pp.74-78, 2013.

[55] Z. Zhang, "Microsoft Kinect Sensor and its Effect," IEEE on Multimedia, vol. 19, no. 2, pp. 4-10, 2012.

[56] "Leap motion," [Online]. Available: http://www.leapmotion.com/

[57] X. Zhang, Z. Ye, L. Jin, Z. Feng, "A new writing experience: finger writing in the Air using a Kinect sensor," IEEE on Multimedia, vol. 20, no. 4, pp.85-93, 2013.

[58] M. Chen, G. AlRegib and B.H. Juang, "Air-writing recognition—Part I: Modeling and recognition of characters, words and connecting motions," IEEE Trans. Human-Machine Systems, (Accepted) to be published.

[59] M. Chen, G. AlRegib and B.H. Juang, "Air-writing recognition—Part II: Detection and Recognition of writing activity in continuous stream of motion data," IEEE Trans. Human-Machine Systems, (Accepted) to be published.

[60] Privilege Insurance, "Alert over unsafe use of navigation equipment," 2006. [Online]. Available: http://www.carpages.co.uk/motoring-news/satellite-navigation-22-02-06.asp

[61] O. Tsimhoni, D. Smith and P. Green, “Destination entry while driving: speech recognition versus a touch-screen keyboard,” Tech. Rep. UMTRI-2001-24, the University of Michigan Transportation Research Institute, 2002.

[62] J. Maciej and M. Vollrath, “Comparison of manual vs. speech-based interaction with in-vehicle information system,” Accident Analysis and Prevention, vol. 41, no. 5, pp. 924-930, 2009.

[63] N. Hataoka, Y. Obuchi, T. Nakano and T. Kobayashi, “Speech interface evaluation on car navigation system: many undesirable utterances and severe noisy speech,” Chapter 10 of Speech and Language Technologies, 2011.

[64] J. F. Kamp, C. Marin-Lamellet, J. F. Forzy and D. Causeur, “HMI aspects of the usability of internet services with an in-car terminal on a driving simulator,” IATSS research, vol. 25, no. 2, pp. 29-39, 2001.

[65] G. E. Burnett, S. M. Lomas, B. Mason, J. M. Porter and S. J. Summerskill, “Writing and driving: An assessment of handwriting recognition as a means of alphanumeric data entry in a driving context,” Advances in Transportation Studies, Special Issue, pp. 59-72, 2005.

[66] D. Kern, A. Schmidt, J. Arnsmann, T. Appelmann, N. Pararasasegaran and B. Piepiera, “Writing to your car: handwritten text input while driving,” In CHI 2009 Extended Abstracts on Human Factors in Computing Systems, USA, pp. 4705-4710, 2009.

[67] J. Liang, B. Zhu, T. Kumagai and M. Nakagawa, “Character-Position-Free On-line Handwritten Japanese Text Recognition,” Proceedings of the 3rd IAPR Asian Conference on Pattern Recognition, Kuala Lumpur, Malaysia, pp.225−229, 2015.

[68] L. Xu, F. Yin, Q.F. Wang and C.L. Liu, “Touching character separation in Chinese handwriting using visibility-based foreground analysis,” In Proceeding of the 11th International Conference on Document Analysis and Recognition, Beijing, China, pp. 859-863, 2011.

[69] T. Matsushita and M. Nakagawa, “A database of on-line handwritten mixed objects named “Kondate”,” Proceedings of the 14th ICFHR, Crete, Greece, pp.369-374, 2014.

[70] U. Ramer, “An iterative procedure for the polygonal approximation of plane curves,” Computer graphics and image processing, vol. 1, pp. 244-256, 1972.

[71] G. Srikantan, D.S. Lee and J.T. Favata, “Comparison of normalization methods for character recognition,” Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, Canada, vol. 2, pp. 719-722, 1995.

[72] J.J. de Oliveira Jr, L.R. Veloso and J.M. de Carvalho, “Interpolation/decimation scheme applied to size normalization of characters images,” Proceedings of the 15th International Conference on Pattern Recognition, Barcelona, Spain, vol. 2, pp. 577-580, 2000.

[73] C.L. Liu, M. Koga, H. Sako and H. Fujisawa, “Aspect ratio adaptive normalization for handwritten character recognition,” Lecture Notes in Computer Science, vol. 1948, pp. 418-425, 2000.

[74] J. Zeng and Z.Q. Liu, “Markov random fields for handwritten Chinese character recognition,” Proceedings of the 8th International Conference on Document Analysis and Recognition, Seoul, Korea, vol. 1, pp. 101-105, 2005.

[75] X.D. Zhou and C.L. Liu, “Text/non-text ink stroke classification in Japanese handwriting based on Markov random fields,” Proceedings of the 9th International Conference on Document Analysis and Recognition, Curitiba, Brazil, vol. 1, pp. 377-381, 2007.

[76] C.L. Liu, K. Nakashima, H. Sako and H. Fujisawa, “Handwritten digit recognition: benchmarking of state-of-the-art techniques,” Pattern Recognition, vol. 36, no. 10, pp.2271-2285, 2003.

[77] Y. Yamashita, K. Higuchi, Y. Yamada and Y. Haga, “Classification of handprinted Kanji characters by the structured segment matching method,” Pattern Recognition Letters, vol. 1, no. 5, pp. 475-479, 1983.

[78] C.L. Liu, Y.J. Liu and R.W. Dai, “Preprocessing and statistical/structural feature extraction for handwritten numeral recognition,” Progress of Handwriting Recognition, World Scientific, pp. 161-168, 1997.

[79] R.J. Carroll and D. Ruppert, “On prediction and the power transformation family,” Biometrika, vol. 68, no. 3, pp. 609-615. 1981.

[80] T. Wakabayashi, S. Tsuruoka, F. Kimura and Y. Miyake, “On the size and variable transformation of feature vector for handwritten character recognition,” (in Japanese) IEICE Trans. Information and Systems, J76-D-II, no. 12, pp. 2495-2503, 1993.

[81] R. Van Der Heiden and F. Groen, “The box-cox metric for nearest neighbour classification improvement,” Pattern Recognition, vol. 30, no. 2, pp. 273-279, 1997.

[82] F. Kimura, K. Takashina, S. Tsuruoka and Y. Miyake, “Modified quadratic discriminant functions and the application to Chinese character recognition,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 9, no. 1, pp. 149-153, 1987.

[83] V.N. Vapnik, “Statistical Learning Theory,” John-Wiley Press, 1998.

[84] Y. Sun, T.S. Butler, A. Shafarenko, R. Adams, M. Loomes and N. Davey, “Segmenting handwritten text using supervised classification techniques,” Proceedings of IEEE International Joint Conference on Neural Networks, vol. 1, pp. 657-662, 2004.

[85] B. Zhu and M. Nakagawa, “Segmentation of on-line freely written Japanese text using SVM for improving text recognition,” IEICE Trans. Information and Systems, vol.E91-D, no.1, pp.105-113, 2008.

[86] Z. Harbi, Y. Hicks, R. Setchi and A. Bayer, "Segmentation of Clock Drawings Based on Spatial and Temporal Features," Procedia Computer Science 60, pp.1640-1648, 2015.

[87] T. Joachims, "Making large-scale SVM learning practical", Universität Dortmund, 1999.

[88] J.C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," In Advances in Large Margin Classifiers, MIT Press, Cambridge, 1999.

[89] C.L. Liu and K. Marukawa, "Handwritten numeral string recognition: character-level vs. string-level classifier training," Proceedings of the 17th International Conference on Pattern Recognition, Cambridge, UK, vol. 1, pp. 405-408, 2004.

[90] M. Nakagawa and K. Matsumoto, "Collection of on-line handwritten Japanese character pattern databases and their analysis," International Journal on Document Analysis and Recognition, vol. 7, no. 1, pp. 69-81, 2004.

[91] C.Y. Suen, "N-gram statistics for natural language understanding and text processing," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 1, no. 2, pp. 164-172, 1979.

[92] K. Takeuchi and Y. Matsumoto, "Japanese OCR correction using stochastic morphological analyzer and probabilistic N-gram word model," International Journal of Computer Processing of Languages, vol. 13, no. 1, pp. 69-82, 2000.

[93] P.K. Wong, and C. Chan, "Postprocessing statistical language models for handwritten Chinese character recognizer," IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 29, no. 2, pp. 286-291, 1999.

[94] N. Iwayama and K. Ishigaki, "Adaptive context processing in on-line handwritten character recognition," Proceedings of the 7th International Workshop Frontiers in Handwriting Recognition, pp. 469-474, 2000.

[95] Q.F. Wang, F. Yin and C.L. Liu, "Unsupervised language model adaptation for handwritten Chinese text recognition," Pattern Recognition, vol. 47, no. 3, pp.1202-1216, 2014.

[96] N. Li, J. Chen, H. Cao, B. Zhang and P. Natarajan, "Applications of Recurrent Neural Network Language Model in Offline Handwriting Recognition and Word Spotting," Proceedings of the 14th International Conference on Frontiers in Handwriting, Crete, Greece, pp. 134-139, 2014.

[97] S.F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," Computer Speech and Language, vol. 13, pp. 359-394, 1999.

[98] A. Vinciarelli, S. Bengio and H. Bunke, "Offline recognition of unconstrained handwritten texts using HMMs and statistical language models," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 26, no. 6, pp.709-720, 2004.

[99] F. Yin, Q.F. Wang, and C.L. Liu, "Transcript mapping for handwritten Chinese documents by integrating character recognition model and geometric context," Pattern Recognition, vol. 46, no. 10, pp. 2807-2818, 2013.

[100] Y.C. Wu, F. Yin and C.L. Liu, "Evaluation of Geometric Context Models for Handwritten Numeral String Recognition," Proceedings of the 14th International Conference on Frontiers in Handwriting Recognition, Crete, Greece, pp. 193-198, 2014.

[101] M. Loog, R.P.W. Duin and R. Haeb-Umbach, "Multiclass linear dimension reduction by weighted pairwise Fisher criteria," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 23, no. 7, pp.762-766, 2001.

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

[103] C.L. Liu and X.D. Zhou, "Online Japanese character recognition using trajectory-based normalization and direction feature extraction," Proceedings of the 10th International Conference on Frontiers in Handwriting Recognition, pp. 217-222, 2006.

[104] Mitsubishi electric, "Mitsubishi electric introduces EMIRAI 3 xDAS assisted-driving concept car," the 44th Tokyo motor show, Tokyo big sight, Japan, 2015. [Online]. Available: http://www.mitsubishielectric.com/news/2015/pdf/1008.pdf

[105] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735-1780, 1997.

Appendix – Ⅰ: List of tables

Appendix – Ⅱ : List of figures

Appendix –Ⅲ: Author's publications

Journals

[1] Jianjuan Liang, Bilan Zhu, Taro Kumagai and Masaki Nakagawa, "Character-position-free on-line handwritten Japanese text recognition by two segmentation methods," *to appear in IEICE Trans. on Information and Systems,* vol. E99-D, no. 4, Apr. 2016.

International Conferences

[2] Jianjuan Liang, Bilan Zhu, Taro Kumagai and Masaki Nakagawa, "Character-position-free on-line handwritten Japanese text recognition," *Proceedings of the 3rd IAPR Asian Conference on Pattern Recognition,* Kuala Lumpur, Malaysia, pp.225–229, Nov. 2015. (Poster)

Domestic Conferences

[3] Jianjuan Liang, Bilan Zhu, Taro Kumagai and Masaki Nakagawa, "Position-free on-line handwritten text recognition," (in Japanese) *IEICE Techinal Report,* vol. 115, no. 98, pp.53–58, June 2015.
梁建娟, 朱碧蘭, 熊谷太郎, 中川正樹, "文字位置自由オンライン手書き文字列認識方式," 電子情報通信学会技術研究報告, vol.115, no. 98, pp.53-58, 2015.6.