

# A Hidden Markov Model Based Segmentation and Recognition Algorithm for Chinese Handwritten Address Character Strings

Qiang Fu, X.Q. Ding, C.S. Liu, Yan Jiang

Dept. of Electronic Engineering, Tsinghua University, P.R. China

State Key Laboratory of Intelligent Technology and Systems

{fuq, dxq, lcs, jyan}@ocrserv.ee.tsinghua.edu.cn

## Abstract

*An efficient method of Chinese handwritten address character string segmentation and recognition is presented. First, an address string image is pre-segmented into several radicals using stroke extraction and stroke merge. Next, the radical series obtained by pre-segmentation merge into different character image series according to different merging paths. After that, the optimal merging path is selected using recognition and semantic information. The recognition information is given by the character classifier. The semantic information is obtained from address database which contains 180,000 address items. Finally, the optimal recognition results of the character image series which are combined by radical series according to the optimal merging paths are obtained. In experiments on 897 mail images, the proposed method achieves correct rate of 85 percent while the error rate is 15 percent.*

## 1. Introduction

Segmentation and recognition of Chinese handwritten address character strings has significant utilities on the automatic mail-sorting system. Because most classifiers are designed for recognizing isolated character, a character string should be segmented into isolated characters so that the character classifiers could be applied to give the recognition results. The problem of handwritten character strings segmentation is hard because of adjacent characters' touching or overlap; large variance of character size, style, distance and so on. Many morphologic features based methods have been proposed to solve Chinese handwritten character strings segmentation [1, 2]. However, it seems difficult to get satisfying segmentation results merely using morphological features. Therefore, one doable scheme is over-segmenting a string image into fragment series first, and then using manifold information, i.e. geometric, recognition and semantic

information, to find optimal merging path [3, 4]. The merging path expresses the way according to which the fragment series merge into character image series.

The basic element of a Chinese character is stroke. A stroke is a relative straight line. Different strokes may have different directions. According its directions, strokes could be labeled as four direction type, i.e. vertical, horizontal, right slanting and left slanting [5]. Those strokes which are very closely adjacent could combine into highly structured components, called radicals [6]. A Chinese character could consist of one or several those radicals. To get correct segmentation result, those radicals should merge into correct characters according to the proper merging path.

## 2. Algorithm outline

The algorithm proposed by this paper first pre-segments an address character string image into radical series. This pre-segmentation processing contains two steps: (a) Extracting strokes of characters in the string. (b) Merging those strokes which are closely adjacent to each other into radicals. After pre-segmentation, the radical series are obtained. Next, the radical series merge into different character image series according to different merging paths. Finally, the optimal merging path and optimal recognition result is obtained. The algorithm's outline is shown as Fig.1.

## 3. Pre-segmentation

### 3.1 Stroke extraction

The address characters in actual handwritten envelopes often have small space between. Further, they often have touching or overlap strokes between adjacent characters. It is difficult for those common methods based on connected component analysis or candidate segmentation point analysis to solve stroke touching or overlap problems [5]. Therefore, stroke extraction method is adopted so that characters with

touching or overlap strokes could be separated correctly. The stroke extraction algorithm's principle is to build one stroke using those pixels which are connected and in the same relative straight line-figure region. Using this method, the address character string image is over-segmented into many strokes. Different strokes are separated from each other, even for those touching or overlap strokes. Further, each stroke keeps its integrity even for it containing cross point. The details of the stroke extraction could be found in [7].

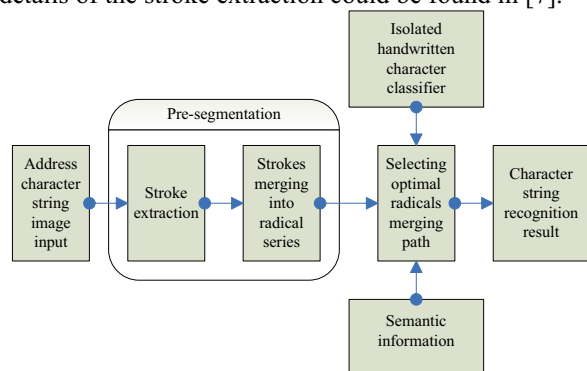
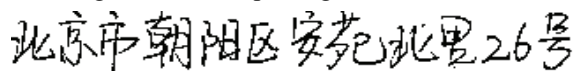


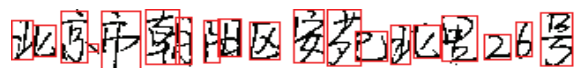
Figure 1. Algorithm flowchart

### 3.2 Strokes merge into radicals

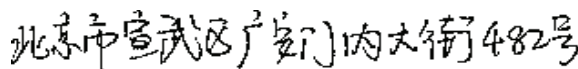
The strokes obtained by stroke extraction combine into radicals according to the method mentioned in [8]. After this step, the stroke series have merged into radical series. The radical series is pre-segmentation result. Fig.2 shows two pre-segmentation results.



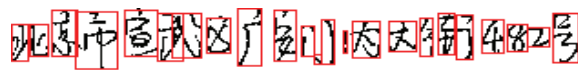
(a) Input Chinese handwritten address character string



(b) Radical series extracted by Presegmentation



(c) Input Chinese handwritten address character string



(d) Radical series extracted by Presegmentation

Figure 2. Pre-segmentation results

### 4. Selection of optimal merging path and recognition result

An input address character string is over-segmented into radical series using the pre-segmentation processing. The radical series could merge into different character image series according to different merging path. The merging path is the way according to which radicals combine into characters. For example, the radical series in Fig.2 (d) has 23 radicals. It could merge into character image series shown in Fig.3 (a) according to the merging path1 (1-2,3,4,5,6-7,8,9,10,11-13,14-15,16,17-19,20,21,22,23). In fact, this path is the correct path. It also could merge into character image series shown in Fig.3 (c) according to path2 (1-2,3,4,5,6-7,8,9,10,11-13,14-15,16,17,18-19,20,21-22,23). Numbers in the bracket are the sequence indexes of radicals. Every possible merging path should satisfy the condition that only those radicals having adjacent indexes could merge with each other. Each merging path corresponds to its character image series. The segmentation problem is converted to optimal merging path selection. The correct radicals merging path would be selected according to the proper evaluation function which will be mentioned in the following paragraphs.

Suppose that there are total  $N$  radicals after pre-segmentation, then the total number of possible merging paths is  $O(2^{N-1})$ . It means that the sum of possible paths will increase exponentially with the  $N$  increasing. In order to reduce the computational complexity, the rough evaluation is applied first in order to select a small candidate merging path set. Then the optimal merging path is selected from the candidate merging path set according to the accurate evaluation.

#### 4.1 Selection of candidate merging path set

The rough evaluation only uses radicals' geometric feature to evaluate every possible character image series. Each character image has a score. A character image's score consists of the following ones. The details could be found in [8].

- Score of character image's width. Compute it by comparing the character image width with the estimation of average width of all character images in the string.
- Score of width-height ratio. Compute it by comparing the character image width-height ratio with its estimation of the string.
- Score of distance of radicals inside the character.

Each character image's score is the sum of those scores above with proper weight. Each character image series' rough evaluation score is the score summation of every character image in the series. The 100 best

candidate character image series, also corresponding to 100 best candidate merging paths, are selected according to the rough evaluation score using dynamic programming method, "Eppstein's K shortest paths Algorithm".

## 4.2 Optimal merging path selection and recognizing corresponding character image series

The relative smaller set of candidate merging paths is obtained according to the rough evaluation. Different merging paths correspond to different character image series. That leads to different recognition results of the character string. Even for the same merging path, so the same character image series, the recognition results of character string may be different because classifier would give multi-candidate interpretations for each character image. For example, supposing that the correct segmentation has been found, as Fig.3 (a), the candidate recognition results of each character are shown in Fig.3 (b). It is clear that the string's correct recognition result is not certain to be composed of every character's first candidate recognition result.

In this part, the precise evaluation based on HMM would be used to select optimal merging path and optimal recognition results of the corresponding character image series.

Suppose that  $N$  radicals are extracted by the pre-segmentation processing.  $RadicalImg_i$  denotes the  $i$ th radical image,  $1 \leq i \leq N$ .  $S$  denotes a merging path. Suppose  $M$  character images are obtained according to this merging path  $S$ .  $CharImg_j$  denotes the  $j$ th character image,  $M \leq N, 1 \leq j \leq M$ .  $Char_j$  denotes a recognition result of  $CharImg_j$ . According to MAP criterion, the optimal path  $\hat{S}$  and string recognition result  $\hat{Char}_1, \dots, \hat{Char}_M$  satisfy the expression (1).

$$\hat{Char}_1, \dots, \hat{Char}_M, \hat{S} = \underset{Char_1, \dots, Char_M, S}{\operatorname{argmax}} p(Char_1, \dots, Char_M | RadicalImg_1, \dots, RadicalImg_N, S) \quad (1)$$

Among it,  $p(Char_1, \dots, Char_M | RadicalImg_1, \dots, RadicalImg_N, S)$  expresses recognition credibility of  $Char_1, Char_2, \dots, Char_M$  under merging path  $S$ . Next, we show how to compute it.

$$\begin{aligned} & p(Char_1, Char_2, \dots, Char_M | S, RadicalImg_1, \dots, RadicalImg_N) \\ &= p(Char_1, \dots, Char_M | CharImg_1, \dots, CharImg_M) \end{aligned} \quad (2)$$

Among (2),  $CharImg_1, \dots, CharImg_M$  are merged by  $RadicalImg_1, \dots, RadicalImg_N$  according to merging path

$S$ . The right of equation (2) could be rewritten as following.

$$\begin{aligned} & p(Char_1, \dots, Char_M | CharImg_1, \dots, CharImg_M) \\ &= p(Char_1 | CharImg_1, CharImg_2, \dots, CharImg_M) \\ & \times \prod_{k=2}^M p(Char_k | Char_{k-1}, Char_{k-2}, \dots, Char_1, CharImg_1, CharImg_2, \dots, CharImg_M) \end{aligned} \quad (3)$$

Now, we make assumption that

$$\begin{aligned} & p(Char_k | Char_{k-1}, Char_{k-2}, \dots, Char_1, CharImg_1, CharImg_2, \dots, CharImg_M) \\ &= p(Char_k | Char_{k-1}, CharImg_k) \quad 2 \leq k \leq M \end{aligned} \quad (4)$$

The assumption means that one character image's recognition result is only dependent on its own image and the previous one character image's recognition result. Substitute (3) with (4), we get

$$\begin{aligned} & p(Char_1, \dots, Char_M | CharImg_1, \dots, CharImg_M) \\ &= p(Char_1 | CharImg_1) \times \prod_{k=2}^M p(Char_k | Char_{k-1}, CharImg_k) \end{aligned} \quad (5)$$

Now, we make another two assumptions that

$$p(CharImg_k | Char_{k-1}) = p(CharImg_k) \quad 2 \leq k \leq M \quad (6)$$

$$p(CharImg_k | Char_k, Char_{k-1}) = p(CharImg_k | Char_k) \quad 2 \leq k \leq M \quad (7)$$

These assumptions are based on the simplification that one character's image is independent on the previous one character; it is only dependent on its own character. With assumption (6) and (7), we get (8).

$$\begin{aligned} & p(Char_k | Char_{k-1}, CharImg_k) \\ &= \frac{p(Char_{k-1}) \times p(Char_k | Char_{k-1}) \times p(CharImg_k | Char_k, Char_{k-1})}{p(Char_{k-1}) \times p(CharImg_k | Char_{k-1})} \\ &= \frac{p(Char_k | Char_{k-1}) \times p(CharImg_k | Char_k)}{p(CharImg_k)} \quad 2 \leq k \leq M \end{aligned} \quad (8)$$

$$\text{Because } \frac{p(CharImg_k | Char_k)}{p(CharImg_k)} = \frac{p(Char_k | CharImg_k)}{p(Char_k)},$$

substituting for (8), we get (9).

$$\begin{aligned} & p(Char_k | Char_{k-1}, CharImg_k) \\ &= \frac{p(Char_k | Char_{k-1}) \times p(Char_k | CharImg_k)}{p(Char_k)} \quad 2 \leq k \leq M \end{aligned} \quad (9)$$

According to (2), (5) and (9), we get that

$$\begin{aligned} & p(Char_1, \dots, Char_M | S, SubCharImg_1, \dots, SubCharImg_N) \\ &= p(Char_1 | CharImg_1) \times \prod_{k=2}^M p(Char_k | Char_{k-1}, CharImg_k) \\ &= p(Char_1 | CharImg_1) \times \left[ \prod_{k=2}^M \frac{p(Char_k | Char_{k-1}) \times p(Char_k | CharImg_k)}{p(Char_k)} \right] \\ &= \frac{\left[ p(Char_1) \times \prod_{k=2}^M p(Char_k | Char_{k-1}) \right] \times \left[ \prod_{k=1}^M p(Char_k | CharImg_k) \right]}{\prod_{k=1}^M p(Char_k)} \end{aligned} \quad (10)$$

For any character,  $p(Char)$  or  $p(Char_i | Char_j)$  is a statistic which is calculated from the address database containing 180,000 address items. The address items in the database are constrained in Beijing address, so it demands that input images should be Beijing address strings'. The classifier used in the paper is trained for

classifying all Chinese character, Arab digits and English characters. It is not trained specially for address character set. For any character image, classifier could give 10 candidate recognition result characters, each with a recognition distance. We use these distance to calculate  $p(Char|CharImg)$ . In more detail, if the character is the  $j$ th candidate recognition results, formula (11) is used to calculate; if the character is not a candidate recognition result,  $p(Char|CharImg)$  is set to a very small constant. Among (11),  $Char_i^{cand}$  denotes the  $i$ th candidate recognition result of  $CharImg$ ;  $Distance_i^{cand}$  denotes recognition distance of  $Char_i^{cand}$ ;  $1 \leq i \leq 10$ .

$$p(Char_j^{cand} | CharImg) = \frac{p(Char_j^{cand}) \times \exp(-Distance_j^{cand})}{\sum_{i=1}^{10} p(Char_i^{cand}) \times \exp(-Distance_i^{cand})} \quad 1 \leq j \leq 10 \quad (11)$$

According to the formula (10), it is clear that the hidden Markov model could be used. Among it, the character image series, i.e.  $CharImg_1, \dots,$

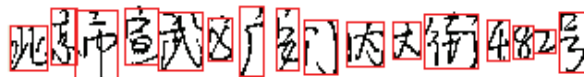
$CharImg_M$ , are considered as observation series of HMM, and the recognitions of these images, i.e.  $Char_1, \dots, Char_M$  are state series of HMM. For each specified merging path  $S$ , the corresponding character image series is determinate. Next, Viterbi algorithm is used to find optimal state series, i.e. recognition results of the image series under path  $S$ . This optimal recognition results' score computed by (10) is seemed as accurate score of path  $S$ . Using this method, every candidate merging path could get its optimal recognition results using Viterbi algorithm and its accurate score computed by (10). The optimal merging path  $\hat{S}$  is one with the highest accurate score; and the optimal string recognition result,  $\hat{Char}_1, \dots, \hat{Char}_M$ , is the optimal recognition results of character image series merged by radical series according to  $\hat{S}$ .

Take Fig.2 (d) as an example: Fig.3 (a) corresponding to merging path1, its optimal recognition results obtained through Viterbi algorithm are characters which are connected by line in Fig.3 (b). Meanwhile, they are also the correct recognition result of the string. Fig.3 (c) corresponding to merging path2, its optimal recognition results are characters connected by line in Fig.3 (d). Path1's optimal recognition result's score computed by (10) is much more than that of path2, so path1 is better than path2. In fact, path1's score is the highest one in all candidate merging paths.

So, its optimal recognition result is the final result of the whole string.

### 4.3 Summary of section 4

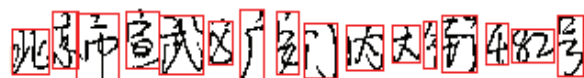
First, use rough evaluation to select 100 best merging paths as candidate paths from all possible merging paths. Among it, there is the optimal merging path. Next, for every specified merging path  $S$ , the hidden Markov model and Viterbi algorithm are applied to find optimal recognition results of character image series which are merged by radicals according to path  $S$ . The formula (10) is used to calculate the score of this optimal recognition results. This score will serve as the final score of the path  $S$ . Every path in 100 candidate paths will get its final score. The path with highest score will be considered as optimal merging path. The optimal recognition results of the optimal merging path will be considered as the interpretation of the whole address character string.



(a) Character image series merged according to merging path1

此	忘	市	宣	武	区	广	安	门	肉	丈	衙	4	8	2	号
北	怎	吊	宜	试	医	户	妥	刁	肉	宋	衙	千	吴	z	号
兆	忍	布	壹	式	八	户	多	刁	闪	文	衙	午	告	Z	言
站	京	而	寅	我	匿	产	妄	习	丙	夫	衙	电	鲁	乙	穹
靴	恋		寅	或	迟	厂	妄	汀	均	夫	衙	壬	哥	I	星

(b) Candidate and optimal recognition results of path1



(c) Character image series merged according to merging path2

此	忘	市	宣	武	区	广	安	门	肉	丈	衙	4	8	2	号
北	怎	吊	宜	试	医	户	妥	刁	肉	宋	I	钉	千	肌	号
兆	忍	布	壹	式	八	户	多	刁	闪	文	L	封	午	虹	言
站	京	而	寅	我	匿	产	妄	习	丙	夫	I	钗	电	缸	穹
靴	恋		寅	或	迟	厂	妄	汀	均	夫	i	钗	壬	缸	星

(d) Candidate and optimal recognition results of path2

Figure 3. Example of different ways for segmentation and recognition

## 5. Experiments and results

To evaluate the performance of the algorithm, experiments are carried out using 897 Chinese handwritten address character strings. The correct rate of individual character is show in table 1. The

character error is caused by two reasons: one is caused by character string segmentation error which is indicated by segmentation error rate in table 1; the other is caused by individual character recognition error which is indicated by recognition error rate in table 1.

Table 1. Experiment on 897 samples

Address string number	Total character number	Character correct rate	Segmentation error rate	Recognition error rate	Total error rate
897	12,207	85.40 %	11.74 %	2.86 %	14.60%

Here are some examples of segmentation and recognition results for Chinese handwritten address character string.

北京市朝阳区安苑北里26号

(a) Input Chinese handwritten address character string

北京市朝阳区安苑北里26号

(b) Segmentation result using proposed algorithm

(c) recognition result using proposed algorithm:  
北京市朝阳区安苑北里26号

北京市宣武区广安门内大街482号

(d) Input Chinese handwritten address character string

北京市宣武区广安门内大街482号

(e) Segmentation result using proposed algorithm

(f) recognition result using proposed algorithm:  
北京市宣武区广安门内大街482号

北京市丰台区东大街5号

(g) Input Chinese handwritten address character string

北京市丰台区东大街5号

(h) Segmentation result using proposed algorithm

(i) recognition result using proposed algorithm:  
北京市丰台区东大街5号

Figure 4. Segmentation and recognition result using proposed algorithm

## 6. Conclusion

The experiments show that the proposed algorithm is effective on segmentation and recognition of unconstraint Chinese handwritten address character strings. There are two key points of the algorithm:

(a) The robust and efficient pre-segmentation algorithm.

(b) The reasonable function to evaluate merging paths and recognition results. The evaluation function in this paper integrates multi-kind information. Especially, the semantic information is very important to segment and recognize address character strings. To simplify the semantic information, hidden Markov model is applied.

In further work, the recognition results which are obtained by proposed algorithm would be used to match the address database items. Using matching procedure, the better final address interpretation would be got.

## 7. References

- [1] Chiang, C.C., Yu, S.S., "An iterative character segmentation method for irregularly formatted Chinese documents", Proceedings of the Optical Character Recognition and Document Analysis, Taiwan, 1996, pp. 61-67.
- [2] Lu, Y., Shridhar, M., "Character segmentation in handwritten words - An overview", Pattern Recognition, vol.29, no.1, 1996, 77-96.
- [3] Casey, R.G., and Lecolinet, E., "A survey of methods and strategies in character segmentation", IEEE Transactions Pattern Analysis and Machine Intelligence, vol.18, no.7, 1996, pp. 690-706.
- [4] Cheng-Lin Liu, Masashi Koga, "Lexicon-Driven Segmentation and Recognition of Handwritten Character Strings for Japanese Address Reading", IEEE Transactions On Pattern Analysis and Machine Intelligence, vol.24, no.11, November 2002, pp. 1425-1437.
- [5] Lin Yu Tseng, Rung Ching Chen, "Segmenting Handwritten Chinese characters based on heuristic merging of stroke bounding boxes and dynamic programming", Pattern Recognition Letters, vol.19, 1998, pp. 963-973.
- [6] Jiang Gao, Xiaoqing Ding, and Youshou Wu, "A segmentation algorithm for handwritten Chinese character strings", International Conference on Document Analysis and Recognition, Bangalore, Sept. 1999, pp. 633-636.
- [7] Tseng, L.Y., Chuang, C.T., "An efficient knowledge based stroke extraction method for multi-font Chinese characters", Pattern Recognition, vol.25, no.12, 1992, pp. 1445-1458.
- [8] WANG Rong, DING Xiaoqing, and LIU Changsong, "Handwritten Chinese address segmentation and recognition based on merging strokes", J Tsinghua Univ (Sci & Tech), vol.44, no.4, 2004, pp. 498-502.