On-Line Handwriting Recognition of Chinese Characters via a Rule-Based Approach

Ju-Wei Chen[†‡] and Suh-Yin Lee[†]

[†]Institute of Computer Science and Information Engineering
National Chiao Tung University, Hsinchu, Taiwan 30050, R.O.C.

[‡]Application Software Department, Computer & Communication Research Laboratories,
Industrial Technology Research Institute, Chutung, Hsinchu, Taiwan 31015, R.O.C.

Abstract

This paper presents a rule-based approach for on-line Chinese character recognition without writing constraints on both stroke number and order. Stroke correspondence is accomplished based on rules predefined such that combinatorial exhaustion can be avoided for character matching. Stroke correspondence rules contain the basic stroke types and geometric features of strokes. Because rules are constituted by invariant structural features, wide handwriting variations can be accommodated. Utilizing the structural characteristics in Chinese characters, we represent the reference database hierarchically to reduce its data size. It only requires $O(n \log n)$ time to accomplish stroke correspondence between a template of n strokes and an input script using the hierarchical reference database. This approach is very suitable for those systems with very limited computing resource.

1 Introduction

On-line handwritten Chinese character recognition (OLCCR) is the key technology for Chinese pen-based systems. Handwriting may vary in stroke shapes, character configuration, stroke order, and the number of strokes. The issue of both stroke order and stroke number free recognition is very important and has been investigated by many researchers [1, 2, 3, 4, 5]. However, some have time consuming computations, and most of them are still under some writing constraints.

Stroke re-ordering is a strategy to cope with stroke-order variations. In the methods proposed previously, some cannot deal with connected strokes [6, 7], and some may have unstable re-ordering results for handwritten characters with wide variations [5, 8].

In this paper, we propose a rule-based approach to recognize on-line Chinese handwriting without constraints on both stroke number and stroke order. Before stroke correspondence, all possible basic strokes in an input script are recognized. Therefore, connected strokes can be segmented apart if exist. Stroke correspondence rules contain the knowledge of possible types of basic strokes allowed in handwriting and invariant geometric features of strokes. Using the rule approach, a template of n strokes has n associated rules, and requires $O(n)$ time to accomplish the stroke correspondence. To apply the proposed approach to portable systems, we represent the reference database hierarchically. Although the time complexity of stroke correspondence will increase to $O(n \log n)$, the requirement of storage space is dramatically reduced. Experiments were performed to verify the effectiveness of the proposed approach.

2 Basic Concepts

Chinese characters possess abundant structural knowledge. We utilize it and propose a rule-based approach for on-line Chinese character recognition. Each handwritten input character is first processed by the stage of preprocessing, in which the line-segment representation can be acquired from the series of point data with noises. After input strokes are recognized, the candidate characters are selected by the preliminary classification. The distances between these candidates and the input script will be calculated in the stage of structural analysis. The candidate with minimum distance is the recognition result. The structural analysis includes three processing steps: *stroke correspondence*,

1015-4651/96 $5.00 © 1996 IEEE
Proceedings of ICPR '96

Table 1. Basic strokes.

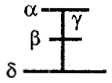
Code	Shape	Code	Shape
0		7	
1		8	
2		9	
3		10	
4		11	
5		12	
6		13	

distance calculation, and *detailed recognition*, where the stroke correspondence is accomplished based on rules predefined to decrease the computation time.

The strokes of Chinese characters in block style can be classified into 14 basic stroke types, as shown in Table 1. Figure 1(a) shows the standard pattern of character "王" and its basic strokes. Two example cursive patterns of character "王", varying in both stroke order and stroke connection, are illustrated in Figure 2(a). A connected stroke in hasty writing may not be classified into any of the 14 types. Therefore, segmenting a connected stroke apart is an essential task.

In our proposed method, all *possible basic strokes* existing in a cursive input stroke are recognized first. They constitute the candidate strokes of stroke correspondence. Figure 2(b) shows all possible basic strokes of the cursive pattern in the right side of Figure 2(a). It contains only one input stroke, but has 11 possible basic strokes.

To depict the process of rule-based stroke correspondence easily, we analyze and designate character primitives as follows. The strokes actually appearing in a character pattern are named *fore strokes*. Along the pen track of writing, the pseudo segment connecting two consecutive fore strokes is called *back stroke*. A back stroke in a template may appear as a fore stroke in an input pattern because of stroke connection, such as stroke c' in Figure 2(a), or may degenerate into a point as the intersection point of a' and b'. These handwriting variations should be accommodated in character matching. The "null" type should also be included for describing the relation between template and input primitives. Therefore, character primitives are classified into four types: *fore strokes*, *back strokes*, *degenerate points*, and *null*. The task of stroke correspondence is to find a binary relation between template primitives and input primitives.



Segment order: α -> β -> γ -> δ

Basic strokes:

Stroke no.	Shape	String	Stroke type
1		<α>	0
2		<β>	0
3		<γ>	2
4		<δ>	0

(a)

(1) Rule of stroke α: the stroke with stroke type 0 or 3, and its center point with the minimum Euclidean to the left-top corner point of the charcater.

(2) Rule of stroke γ: the stroke with stroke type 2 and the top boundary of its MBR with the maximum y coordinate .

(3) Rule of stroke δ: the stroke with stroke type 0 or 4, and its center point with the minimum Euclidean to the left-bottom corner point of the charcater.

(4) Rule of stroke β: the stroke with stroke type 0 and the top boundary of its MBR with the maximum y coordinate .

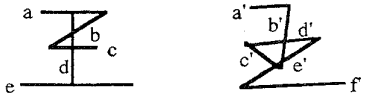
(b)

Figure 1. (a) Standard pattern of character "王" and its constituent basic strokes; (b) its stroke correspondence rules.

Definition 2.1 A stroke matching is a binary relation $q : X \rightarrow Y$ from set X to Y, where X denotes the set of primitives of a template character and Y is the set of primitives of an input character. For any element $x_i \in X$, if the mapped image $y_i (y_i = q(x_i),\ y_i \in Y)$ exists, then there exists only one image y_i.

There are eight possible types of matching pairs: $fore \rightarrow fore$, $back \rightarrow back$, $back \rightarrow fore$, $back \rightarrow point$, $back \rightarrow null$, $null \rightarrow back$, $fore \rightarrow null$, and $null \rightarrow fore$, where $x \rightarrow y$ indicates a matched pair; *fore* indicates a *fore stroke* primitive, *back* indicates a *back stroke* primitive, *point* indicates a *degenerate point* primitive, and *null* indicates no matched primitive.

Each character category has a set of rules. Figure 1(b) shows the stroke correspondence rules of char-



(a)

Segment order: a' -> b' -> c' -> d' -> e' -> f'.

Possible basic strokes:

Stroke no.	Shape	String	Stroke type
1		<a', b', c'>	6
2		<d', e', f'>	10
3		<a', b'>	6
4		<b', c'>	9
5		<d', e'>	6
6		<e', f'>	7
7		<a'>	0
8		<b'>	2
9		<d'>	0
10		<e'>	1
11		<f'>	0

(b)

Steps of stroke correspondence:

Rule no.	Mapping			Remaining pattern
(1)	α	->	a'	
(2)	γ	->	b'	
(3)	δ	->	f'	
(4)	β	->	d'	
	(γ, β)	->	c'	
	(β, δ)	->	e'	

(c)

Figure 2. (a) Two example cursive patterns of character "王" with different stroke orders accompanied with stroke connections; (b) all possible basic strokes included in the cursive pattern in the right side of (a); (c) stroke correspondence process of the cursive pattern.

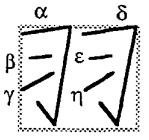
acter "王". The sequence of rules indicates the sequence of stroke matching of a template. The stroke correspondence rules consist of basic stroke types allowed in handwriting and invariant geometric features of strokes. A handwritten stroke may consist of one or more basic strokes due to stroke connections. The matching of a template stroke is accomplished by selecting a possible basic stroke from the input script based on the rule. Then, the stroke is decomposed from the input pattern. The next stroke correspondence rule is applied onto the remaining strokes of the input pattern until all rules of the template are used up.

The process of stroke correspondence can be explained by Figure 2(c). The fore strokes of the template are first matched with the fore strokes in the input script. The back strokes of the template pattern are obtained by tracing the matched pairs following the input stroke order. A back stroke in the template character may be mapped to a back stroke, a degenerate point, or a fore stroke in an input script. The matching of back strokes can also be determined by the matched pairs of fore strokes following the input stroke order. In Figure 2(c), matched pairs $\alpha \rightarrow a'$, $\gamma \rightarrow b'$, $\delta \rightarrow f'$, and $\beta \rightarrow d'$ are the matched fore strokes. The back stroke (i, j) is between fore strokes i and j. The back strokes (α, γ), (γ, β), and (β, δ) can be obtained after the above matched pairs are found. By tracing the matched pairs of fore strokes following the input stroke order, back strokes (α, γ), (γ, β), and (β, δ) can be mapped to the intersection point of a' and b', c', as well as e', respectively.

3 Stroke Correspondence Rules

Chinese characters are constituted by basic strokes based on certain geometric configurations. To cope with stroke-order and stroke-number variations, we utilize both basic stroke types and geometric features of strokes in designing stroke correspondence rules.

Stroke correspondence rules used in our recognition system are classified into *type one* and *type two*, denoted by $R1$ and $R2$, respectively. We explain stroke correspondence rules using Figure 3. Figure 3(a) illustrates the standard pattern of character "羽". Figure 3(b) lists its stroke correspondence rules. The order of stroke correspondence is based on the predefined rule sequence. All rules contain the information of possible types of basic strokes considering handwriting variations. The information is used to eliminate those strokes with stroke types violating those predefined in the rule. It also has the function of segmenting connected strokes apart. As in Figure 3(b), a *type one*



(a)

Rule no.	Stk. label	Rule type	Stk. types	Geometric features of strokes
(1)	α	R2	6	F25 S1 C2, F7 S0
(2)	β	R1	0, 1, 2, 3	F20 S0
(3)	γ	R1	0, 1, 2, 3	F5 S0
(4)	δ	R1	6	F25 S1
(5)	ε	R1	0, 1, 2, 3	F6 S1
(6)	η	R1	0, 1, 2, 3	F6 S0

(b)

Figure 3. (a) Standard pattern of character "羽" with stroke labels; (b) its stroke correspondence rules.

rule, denoted by $(Fi, Sj)(1 \leq i \leq 27$ *and* $j = 0$ *or* $1)$, utilizes one geometric feature numbered i to find the matching stroke from the possible input basic strokes based on minimum or maximum of a certain feature Fi, which is denoted by $S0$ or $S1$, respectively. A *type two* rule utilizes two geometric features for complicated characters, which is denoted by $(Fi, Sj, Ck; Fi', Sj')$. Ck indicates that k candidate strokes are selected from the remaining input possible basic strokes based on the first feature. The second feature is used to determine the matching strokes from the k candidate strokes.

We use 27 types of geometric features of strokes in designing stroke correspondence rules. All of them are proposed for obtaining stable stroke correspondence. Each stroke is considered to be bounded by a minimum bounding rectangle (MBR). The x and y coordinates of the four boundaries and the center point of the MBR of a stroke are designated as geometric features numbered 1 to 6, respectively. The x and y coordinates of the start point and end point of a stroke are geometric features, numbered 7 to 10. In our work, we adopt a hierarchical representation in the reference database. Each character is described by its constituent component code(s) and its character structure. For each component, stroke correspondence rules are stored. During matching, constituent components need to be decomposed from the character one by one. Strokes of neighbor components may be erroneously included in a decomposed component. To exclude these erroneous strokes, we define eight reference points on the boundaries of the bounding rectangle of the decomposed component, illustrated in Figure 4. Based on the eight reference points, we propose 14 auxiliary geometric features numbered 11 to 24. They are the Euclidean distances from the start and end point of a stroke as well as the center point of a stroke's MBR to the eight reference points, respectively. The geometric feature numbered 25 is the length of a stroke; the feature numbered 26 indicates the city block distance from the left-bottom corner point of a character to the left-bottom corner point of a stroke's MBR; the feature numbered 27 indicates the city block distance from the left-bottom corner point of a character to the right-top corner point of a stroke's MBR.

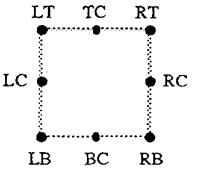


Figure 4. Eight reference points on the boundaries of the bounding rectangle of a decomposed component.

Without any hierarchical representation, the proposed geometric features of each possible basic stroke can be computed at once after all possible basic strokes in an input character are identified. All of them are sorted based on each feature in increasing order and the sorted sequence is recorded in one array, respectively. During stroke correspondence, the matching stroke can be determined directly based on the data in these arrays. The stroke correspondence only requires $O(n)$ time for a template of n strokes based on the n rules. When we use hierarchical reference database, components need to be decomposed from a character one by one. The 14 auxiliary features of each component need be computed after decomposition. The decomposed strokes are sorted based on each feature, respectively. Therefore, the time complexity of stroke correspondence increases to $O(n \log n)$.

After stroke correspondence, the matching relation between template strokes and input strokes has been acquired. Based on the relation, we can further acquire the information of stroke connection. For character discrimination, we utilize structural knowledge accompanied with appropriate discriminant functions to calculate their distance. The structural knowledge used in recognition occupies the great majority of the storage space. We therefore utilize a hierarchical representation such that a storage space of about 260 Kbytes is enough for storing the structural knowledge of a char-

acter set of 5401 categories.

4 Experiments and Results

To verify the effectiveness of our proposed approach, we performed two experiments. The first is for verifying the performance of the structural analysis in the recognition process, including recognition accuracy and speed. Each character category has 11 testing samples taken from the *ITRI OLCCR* database [9]. In this experiment, we used the estimated range of the number of input strokes alone in preliminary classification. For 1225 character categories randomly selected from the 5401 categories, the average number of candidate characters was 274. The average first rank recognition rate was 93.51%, and the average 5th rank cumulative recognition rate reached 98.16%. The average recognition speed was about 0.8 second per character on a PC/AT-486 and was about 0.5 second on a Pentium-100 based PC.

The number of candidates selected by the preliminary classification will influence the recognition speed. When the recognition system is extended to recognize 5401 or more categories, more information should be included in the preliminary classification to avoid the number of candidates increasing. The second experiment is to reveal the feasibility of using both the estimated range of possible numbers of input strokes and statistical features in the preliminary classification for a large character set of 5401 categories. We used 23 samples per character category in testing. On the average, using statistical features alone, the number of candidate characters was 1480; using the estimated range of the numbers of input strokes alone, the number of candidate characters was 1040. When both of them are used, the number of candidate characters decreased to 353. Therefore, the real-time recognition would be realized via the proposed rule-based approach even for a large character set.

5 Conclusions

This paper presents an on-line recognition method of Chinese characters via a rule-based approach without constraints on both stroke number and stroke order. The stroke correspondence requires $O(n)$ time for a template of n strokes, and requires $O(n \log n)$ time when the stroke correspondence rules are represented hierarchically. For a character set of 5401 characters, a storage space about 260 Kbytes is enough for storing the structural knowledge used in recognition, which is about 1/4 of the amount without using hierarchical representation. The experimental results reveal that the proposed approach can be applied in portable systems with very limited computing resource for recognizing a large character set. The architecture of the proposed recognition system could also be applied in recognizing characters written in various styles by using the rules suitable for the style to be recognized.

Acknowledgements

The author would like to thank the research grant supported by the Intelligent Man/Machine Interface Application Project (project no. 35N7100) sponsored by the Minister of Economic Affairs, Taiwan, R.O.C.

References

[1] T. Wakahara and M. Umeda, "Stroke-number and stroke-order free on-line character recognition by selective stroke linkage method," *Proc. 4th ICTP,* pp. 157-162, Oct. 1983.

[2] T. Wakahara and M. Umeda, "On-line cursive script recognition using stroke linkage rules," *Proc. 7th ICPR,* pp. 1065-1068, 1984.

[3] T. Wakahara, "On-line cursive script recognition using local affine transformation," *Proc. 9th ICPR,* Nov. 1988, pp. 1133-1137.

[4] C. K. Lin, K. C. Fan, and F. T. P. Lee, "On-line recognition by deviation-expansion model and dynamic programming matching," *Pattern Recognition,* .Vol. 26, No. 2, pp. 259-268, 1993.

[5] Y. J. Liu and J. W. Tai, "A method of stroke order arrangement for on-line Chinese character recognition," *Acta Automatica Sinica,* Vol.14, No.3, pp. 207-214, May, 1988.

[6] Y. Hidai, K. Ooi, and Y. Nakamura, "Stroke reordering algorithm for on-line handwritten character recognition," *Proc. 8th ICPR,* 1986, pp. 934-936.

[7] T. Morishita, M. Ooura, and Y. Ishii, "A Kanji recognition method which detects writing errors," *Computer Processing of Chinese and Oriental Languages,* Vol. 3, pp. 351-365, Mar. 1988.

[8] P. J. Ye, H. Hugli, and F. Pellandini, "Techniques for on-line Chinese character recognition with reduced writing constraints," *Proc. 7th ICPR,* 1984, pp. 1043-1045.

[9] J. W. Chen and S. L. Shiau, "Database of on-line handwritten Chinese character samples," Ministry of the Interior Copyright (54214) in the R.O.C., Oct. 1987.