Stroke Order and Stroke Number Free On-Line Chinese Character Recognition Using Attributed Relational Graph Matching

Jianzhuang Liu†, W. K. Cham† and Michael M. Y. Chang‡

†Department of Electronic Engineering, The Chinese University of Hong Kong, Hong Kong
‡Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong
E-mail: jzliu@ee.cuhk.edu.hk wkcham@ee.cuhk.edu.hk

Abstract

A structural method for on-line recognition of Chinese characters is proposed, which is stroke order and stroke number free. Both input characters and the model characters are represented with complete attributed relational graphs (ARGs). A new optimal matching measure between two ARGs is defined. Classification of an input character can be implemented by matching its ARG against every ARG of the model base. The matching procedure is formulated as a search problem of finding the minimum cost path in a state space tree, using the A algorithm. In order to speed up the search of the A*, besides a heuristic estimate, a novel strategy that utilizes the geometric position information of stroke segments of Chinese characters to prune the tree is employed. The efficiency of our method is demonstrated by the promising experimental results.*

1. Introduction

Today, rapid development of computer techniques has made personal computers (PCs) cheap enough for family use. A good on-line Chinese character recognition (OLCCR) system will provide a friendly interface for the use of Chinese and popularize PCs in China and some other areas. Although great progress has been made in OLCCR since the 1970's [7, 3], a number of researchers are still involved in this topic for achieving better performance of OLCCR. Recognition of handwritten Chinese characters is considered as a very hard problem because of large categories, complex structure, and widely variable and many similar shapes of Chinese characters. Researchers hope to develop efficient algorithms which are stroke order and stroke number free, and can run on general computers (e.g. PCs) within an acceptable computational time.

Both statistical and structural methods can be employed for recognition of Chinese characters [3]. Chinese characters are 2D pictographic characters, and intuitively, human beings classify Chinese characters by making use of their local structural information instead of their global statistical features. In this paper, we represent both the characters of our model base and the input characters with *complete* ARGs, and define a new measure for the optimal matching between two ARGs. In a complete ARG, the nodes describe stroke segments of characters and the arcs the relations between any two segments. In order to speed up the recognition procedure, the matching is formulated as a problem of search in a state space tree, and the A* algorithm is used to perform the heuristic search. A novel tree pruning strategy which uses the geometric position information of stroke segments of Chinese characters is proposed to assist the search of the A*.

The proposed method is an improvement of our previous work [5] which is stroke order free but easily obtains incorrect classification if there are three or more connected strokes in an input character. This is because the primitives used in [5] are strokes, hence connected strokes will make the stroke positions and relations change greatly. However, the stroke segment positions and relations are still stable even if there are many connected strokes.

2. Complete ARGs of Chinese characters

2.1. Stroke and stroke segment extraction

A stroke is defined as the writing from pen down to pen up when one writes on a digitizer with a stylus pen. A Chinese character consists of a set of standard strokes, and each standard stroke consists of from one to four segments, as shown in Table 1. On-line devices can capture the temporal information of the writing, such as the number, order and direction change of a stroke. To conveniently extract segments of an input character, we use straight lines to represent each stroke. A piecewise linear curve fitting procedure called the iterated endpoint fit [2] is suitable for our appli-

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

Table 1. Standard strokes

Type	Strokes	Type	Strokes
1		8	
2		9	
3		10	
4		11	
5		12	
6		13	
7		14	

Table 2. Some strokes

	Strokes		Strokes
1		6	
2		7	
3		8	
4		9	
5		10	

cation.

In nature handwriting, a stroke is called a connected stroke if it is the combination of two or more standard strokes. Some connected strokes generate extra segments but some do not. For example, the Chinese characters '' and '  ' may be written as '' and '', respectively. An extra segment '↖' appears in the former. By analyzing Chinese character handwriting, we can obtain some rules that may be used to detect extra segments in some kinds of connected strokes, and then delete them. Some of the rules we use are similar to those in [6] and [1]. For example, we consider that all the segments '↖' in characters are extra ones and should be deleted. Though there are such segments in some standard strokes (see Table 1), deleting them does not confuse a character with the others.

For the connected strokes which cause no extra segments and the standard strokes each with more than two segments, as shown in Table 2, we should use all the segments of these strokes to represent a character. An algorithm similar to dynamic programming [4] is employed to recognize these strokes. After the stroke and segment preprocessing, five segment types — (1) '→' (-20°,30°), (2) '↓' (250°,290°), (3) '↙' (180°,250°), (4) '↘' (290°,340°), and (5) '↗' (30°,90°) — are obtained.

Obviously, it is impossible to correctly detect all extra segments or obtain all segments which should remain for recognition, because of wide handwriting variations. However, our inexact ARG matching method discussed below can tolerate these preprocessing errors.

2.2. Complete ARG representation of Chinese characters

ARGs were first used to represent the structural information of patterns in [8]. Recognizing the structure of a given unknown pattern may be performed by transforming this pattern into an ARG and then matching the ARG with those which represent the structures of model patterns. Below we begin with Definition 1 for ARGs that is defined in [8].

Let V_N and V_A be sets of node labels and arc labels, respectively. Each element belonging to V_N or V_A is of the form (u, v), where u is a syntactic symbol denoting the structure of (u, v) and $v = (v_1, v_2, ..., v_m)$ is a semantic vector denoting m numerical and/or logical attributes of (u, v).

Definition 1. An attributed relational graph over $V = V_N \cup V_A$ is a 4-tuple $\omega = (N, A, \mu, \varepsilon)$, where

N is a finite nonempty set of nodes;

$A \subset N \times N$ is a set of distinct ordered pairs of distinct elements in N called arcs;

$\mu : N \to V_N$ is a function called node interpreter;

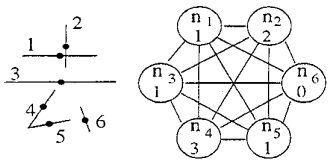
$\varepsilon : A \to V_A$ is a function called arc interpreter.

To represent the complex structure of a Chinese character with an ARG, a straightforward way is that the nodes of the ARG describe the segments of the character and the arcs describe the relations between any two different segments. Considering the computation rate and the wide stroke variations, we use simple but relatively stable segment relation features of Chinese characters. The complete ARG representation for a model character is given as follows.

(1) Nodes of an ARG—The syntactic symbol of a node has one of the five segment types. The semantic vector of a node is a binary value '1' or '0', where '1' denotes the segment is a long one and '0' a short one. A short segment in a handwritten Chinese character is a relatively unstable segment which may be written as one of the first four segment types.

(2) Relations of an ARG—The syntactic symbol of a relation, r_{ij}, between segment i and segment j is represented as a vector $r_{ij} = (a^1_{ij}, a^2_{ij}, a^3_{ij})$, where a^1_{ij}, a^2_{ij} and $a^3_{ij} \in \{0, 1, 2\}$. Let c_i and c_j be the geometric centers of segment i and segment j, respectively. Then $a^1_{ij} = 0, 1$, or 2 denotes c_i being 'below', 'above', or 'below or above' c_j. Also, $a^2_{ij} = 0, 1$, or 2 denotes c_i being on the 'right of', 'left of', or 'right of or left of' c_j. $a^3_{ij} = 0, 1$ or 2 denotes that segment i 'uncrosses', 'crosses', or 'uncrosses or crosses' segment j. The semantic attribute of the relations are not used.

In this ARG representation, a node is of the form (u, v), where $u \in \{1, 2, ..., 5\}$ and $v \in \{0, 1\}$. If a new segment type '0' is added which denotes a short segment, then the representation of a node will be of the simpler form u where



(a) (b)

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

(c)

Figure 1. (a) A Chinese character; (b) complete ARG of Figure 1(a); (c) generalized relation matrix of Figure 1(b).

$u \in \{0, 1, ..., 5\}$. This is beneficial to the programming and reducing computational time. The ARG representation of an input character is similar to that of a model character, but segment type 0 is not used because in handwriting short segments are easily written as long as some long segments, and vice versa. Moreover, a^1_{ij}, a^2_{ij} and $a^3_{ij} \in \{0, 1\}$ in a relation vector $r_{ij} = (a^1_{ij}, a^2_{ij}, a^3_{ij})$, between the input segments i and j.

Fig. 1 shows an example of the complete ARG representation for model character '  '. The points on the segments in Fig. 1(a) are their geometric centers. Fig. 1(b) is the ARG of '', where nodes n_{1-6} describe 6 segments and their types. The relations of the graph are represented by a generalized matrix $R = [r_{ij}]_{6\times 6}$ shown in Fig. 1(c). Note that there is some kind of symmetry between the elements $r_{ij} = (a^1_{ij}, a^2_{ij}, a^3_{ij})$ and $r_{ji} = (a^1_{ji}, a^2_{ji}, a^3_{ji})$. In order to tolerate daily handwriting variations, the relation matrix of a Chinese character in the model base must be designed carefully. For example, '  ' may be written as '', '' or ''. The relation 'above' or 'below' between the geometric centers of segment 1 and segment 2 is uncertain, and so a^1_{12} is set to 2. In addition, segment 6 belongs to a short unstable one, so its type is 0.

3. ARG matching via state space search

3.1. Definitions of costs and matchings

Determining whether two Chinese characters are similar or not can be formulated as a graph matching problem. In the matching procedure there are costs with respect to node mappings and arc mappings. We use function $C_1(i, s_1; k, s_2)$ to denote the cost of mapping node i with segment type s_1 in ARG1 to node k with segment type s_2 in ARG2. Because of variations in handwriting, different segment type mappings may have different cost values. The costs of mapping input segments to primitive segments are defined in Table 3.

Table 3. Costs associated with segment mapping

Input segments	Segments in model base					
	0	1	2	3	4	5
1	1	0	7	7	2	2
2	1	7	0	2	2	7
3	1	7	2	0	7	7
4	1	2	2	7	0	7
5	7	2	7	7	7	0

Let i and j be nodes in an ARG, then the ordered pair (i, j) is called a (directed) arc of the ARG. The cost of mapping arc (i, j) in ARG1 to arc (k, l) in ARG2 is defined as

$$C_2(i, j; k, l) = \sum_{m=1}^{3} w_m d_m(a^m_{ij}, a^m_{kl}), \qquad (1)$$

where w_{1-3} are weighting factors, and

$$d_m(a^m_{ij}, a^m_{kl}) = \begin{cases} 0 & \text{if } (a^m_{ij} = a^m_{kl}) \text{ or if } (a^m_{ij} \text{ or } a^m_{kl} = 2) \\ 1 & \text{otherwise} \end{cases}$$

$m = 1, 2, 3$.

As mentioned above, the segment number of an input character may differ from that of its model because of handwriting variations. Therefore, we use the inexact graph matching that allows matching between two ARGs with different numbers of nodes. A cost $C_3(k, s; t)$ is introduced, with which node k with segment type s in the graph having more nodes is mapped to a *null* node in the other graph, where t is the segment number of the graph with fewer nodes. If we regard all mappings between any type of segment and a null segment as the same, then $C_3(k, s; t)$ may be written as $C_3(k; t)$.

Definition 2. Let $|N|$ be the cardinal of a set N and $NULL$ the set of null nodes. Also let $\omega_1 = (N_1, A_1, \mu_1, \varepsilon_1)$ and $\omega_2 = (N_2, A_2, \mu_2, \varepsilon_2)$ be two ARGs, with $|N_1| \leq |N_2|$. An inexact matching between ω_1 and ω_2 is defined as the function $f : N_1 \cup NULL \rightarrow N_2$, where $|N_1| + |NULL| = |N_2|$ (in the case of $|N_1| = |N_2|$, $NULL = \emptyset$), if the following condition is satisfied:

$$n \neq m \Rightarrow f(n) \neq f(m), \forall n, m \in N_1 \cup NULL \text{ and } f(n), f(m) \in N_2.$$

Definition 3. Let ω_1 and ω_2 be two ARGs and $f : N_1 \cup NULL \rightarrow N_2$ be an inexact matching. The cost of the matching, $cost(f, \omega_1, \omega_2)$, is defined as

$$cost(f, \omega_1, \omega_2) = \sum_{\substack{f(i)=k \\ i \in N_1}} C_1(i, s_1; k, s_2) +$$

$$\sum_{\substack{f(i)=k \\ f(j)=l \\ i,j \in N_1, i \neq j}} C_2(i, j; k, l) + \sum_{\substack{f(i)=k \\ i \in NULL}} C_3(k, s; t). \quad (2)$$

Definition 4. The optimal matching between ω_1 and ω_2, with $|N_1| \leq |N_2|$, is a mapping function $f^* \in M(f)$ such that

$$cost(f^*, \omega_1, \omega_2) = \min_{f \in M(f)} \{cost(f, \omega_1, \omega_2)\}, \quad (3)$$

where $M(f)$ is the set of all possible mappings.

Definition 5. Let $\omega = (N, A, \mu, \varepsilon)$ be the ARG of an input character and $W = \{\omega_1, \omega_2, ..., \omega_p\}$ be the ARG set of p models. Partition W into two subsets W_1 and W_2 such that $W = W_1 \cup W_2, W_1 \cap W_2 = \emptyset, \omega_i = (N_i, A_i, \mu_i, \varepsilon_i) \in W_1$ with $|N_i| \leq |N|$, and $\omega_j = (N_j, A_j, \mu_j, \varepsilon_j) \in W_2$ with $|N_j| > |N|$. Define

$$cost(f_k^*, \omega_k, \omega) = \min_{\omega_i \in W_1} \{cost(f_i^*, \omega_i, \omega)\} \quad (4)$$

and

$$cost(f_l^*, \omega, \omega_l) = \min_{\omega_j \in W_2} \{cost(f_j^*, \omega, \omega_j)\}. \quad (5)$$

If $cost(f_k^*, \omega_k, \omega) \leq cost(f_l^*, \omega, \omega_l)$ and $cost(f_k^*, \omega_k, \omega) < T$, the input is called the most similar to model k; if $cost(f_l^*, \omega, \omega_l) \leq cost(f_k^*, \omega_k, \omega)$ and $cost(f_l^*, \omega, \omega_l) < T$, the input the most similar to model l; otherwise the input not similar to any model. T is a pre-defined upper limit which may vary with the node number of ω.

It is clear that exhaustive search for obtaining $cost(f^*, \omega_1, \omega_2)$ is heavily time-consuming. The computational complexity of that search is $O(|N_1|!)$ when $|N_1| = |N_2|$. Therefore, fast search strategies are necessary.

3.2. State space search with A* algorithm

We convert the optimal ARG matching into a search problem of finding the minimum cost path from the initial state to a goal state in a state space tree, with the A* algorithm. The search approach is similar to that of [5]. For the limitation of space, we omit this description here. As the A* has exponential complexity in many cases, acceptable recognition time cannot be obtained if heuristic functions good enough are not available. We, therefore, propose a novel tree pruning strategy which uses the geometric position information of segments of Chinese characters to assist the search of the A*.

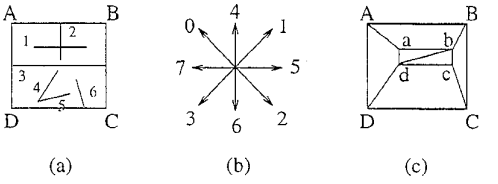


Figure 2. (a) A character and rectangle $ABCD$; (b) 8 directions; (c) geometric illustration of $D_{0-3}(i)$.

4. A pruning strategy

In on-line recognition, a Chinese character as a whole can be regarded as no rotation variety. Hence a lot of information about the geometric positions of segments in the character may be used to assist A* algorithm's search for the minimum cost path. Fig. 2(a) shows the Chinese character '十' and the smallest rectangle $ABCD$ that surrounds it. What are the stable geometric position features of segments of '十' in daily writing style? Intuitively, segments 1, and 2 are written near the upper side of $ABCD$; segment 3 in the middle; segments 4 and 5 near the lower side; segment 6 near the lower side or the lower-right corner. In the following we formulate these character-dependent features by using a set of segment geometric position features.

Let $abcd$ be the smallest rectangle that surrounds segment i of an input character with s segments. Eight directions in Fig. 2(b), are used to denote the directions of eight distances $D_{0-7}(i)$, where $D_{0-3}(i)$ are distances from a to A, b to B, c to C, and d to D, respectively, as shown in Fig. 2(c), and $D_{4-7}(i)$ are distances from the geometric center of $abcd$ to the respective four sides of $ABCD$. Besides, a notation $od(D_q(i))$, $q \in \{0, 1, ..., 7\}$, is used to denote that $D_q(i)$ is the $od(D_q(i))$-th smallest distance among $\{D_q(1), D_q(2), ..., D_q(s)\}$.

Definition 6. The set of segment geometric position features of a model Chinese character with s segments is defined as a set of s 3-tuples

$$GPF = \{(d_i, x_i, y_i) | i = 1, 2, ..., s\}, \quad (6)$$

where $d_i \in \{0, 1, ..., 7\}, x_i \leq od(D_{d_i}(i)) \leq y_i$, and $x_i, y_i \in \{1, 2, ..., s\}$.

When searching the state space tree for the optimal matching between ARG1 of the input and ARG2 of a model, the A* algorithm runs with a pruning procedure inserted. The generation of a successor node in the tree means that a segment (say, segment i) in ARG1 is mapped to a segment (say, segment k) in ARG2. Let the kth elements of GPF of the model be (d_k, x_k, y_k). If $x_k \leq od(D_{d_k}(i)) \leq y_k$, i.e., the geometric position of segment i is subject to the

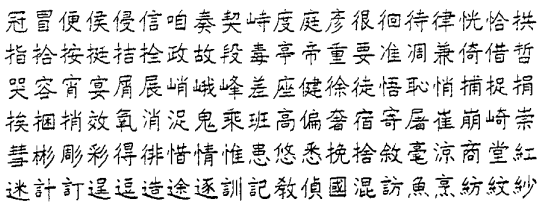


Figure 3. Some models.

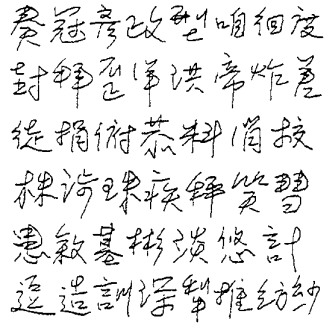


Figure 4. Some test data.

constraint upon the geometric position of segment k, then the newly-generated node will be put in the open list of the A*; otherwise, the node is pruned away.

5. Experimental results and conclusions

In our experiment, 200 frequently used Chinese characters each with stroke number between 9 and 11 (segment number between 9 and 15) are selected for testing the performance of our method. (A Chinese character has an average of 9 strokes [7]). The model data base was manually set up according to standard Chinese character patterns such as those shown in Fig. 3. We choose the weighting factors w_1, w_2 and w_3 in (1) to be 7,7 and 3, respectively, and the cost $C_3(k, s; t)$ to be 3. As some segments of many Chinese characters are easily written to cross each other while they are not supposed to do so in standard writing, so w_3 is assigned a smaller value.

The test data consist of more than 3000 Chinese characters written by 7 people. The subjects were allowed to write the characters in any stroke orders and having connected (or split) strokes. Fig. 4 shows some of the test data which are classified correctly. As can be seen, our method may tolerate wide stroke variations and many connected strokes. The recognition rate varies with the number of connected strokes in characters. When the model characters having stroke number between 9 and 11 were written as their deformed characters having stroke number between 6 and 8, the recognition rate was about 97%. When these models were written having only 3 or 4 strokes, the recognition rate was about 91%. These tentative results are very satisfactory. The average time for classifying an input character is about 1 second on a 50MHz PC/486.

In this paper, a structural method for on-line recognition of Chinese characters is proposed, which is stroke order and stroke number free. Both input characters and models are represented as complete ARGs. A minimum cost measure for matching between two ARGs is defined. The recognition of an input is implemented by matching its ARG with those of the model base. The matching is formulated as the search for the minimum cost path in a state space tree, using the heuristic algorithm A*. In order to further speed up the search procedure of the A*, a novel strategy that utilizes the geometric position information of segments in a Chinese character is employed for pruning the search tree. The experimental results obtained are very promising. To further improve the performance of our method, future research efforts may include the study of (1) preclassification and stroke preprocessing approaches, (2) more efficient tree search algorithms, and (3) the use of more geometric features of segments of Chinese characters.

Acknowledgment

We are thankful for the financial support provided by the Hong Kong RGC Earmarked Research Grant CUHK67/92E.

References

[1] K. S. Chou, K. C. Fan, T. I. Fan, C. K. Lin, and B. S. Jeng. Knowledge model based approach in recognition of on-line chinese characters. *IEEE J. Selected Areas Communi.*, 12:1566–1574, 1994.

[2] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. Wiley-Interscience, New York, 1973.

[3] V. K. Govindan and A. P. Shivaprsad. Character recognition — a review. *Pattern Recognition*, 23:671–683, 1990.

[4] F. S. Hillier and G. J. Lieberman. *Introduction to Operations Research*. McGraw-Hill, New York, 1990.

[5] J. Liu, W. K. Cham, and M. Y. Chang. On-line chinese character recognition with attributed relational graph matching. In R. T. Chin, H. H. S. Ip, A. C. Naiman, and T. C. Pong, editors, *Image Analysis Applications and Computer Graphics*, pages 189–196. Springer, 1995.

[6] Y. J. Liu and J. W. Tai. An on-line chinese character recognition system for handwritten in chinese calligraphy. In *From Pixel to Features III — Frontiers in Handwriting Recognition*, pages 87–99. Elsevier Science Publishers B. V., 1992.

[7] C. C. Tappet, C. Y. Suen, and T. Wakahara. The state of the art in on-line handwriting recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12:787–808, 1990.

[8] W. H. Tsai and K. S. Fu. Error-correcting isomorphisms of attributed relational graphs for pattern analysis. *IEEE Trans. Syst. Man Cybern.*, 9:757–768, 1979.