Joint Conference of 1996 International Computer Symposium
December 19~21, Kaohsiung, Taiwan, R.O.C.

An On-line Handwriting Recognition Approach of Chinese Characters for Personal Digital Assistants

Ju-Wei Chen[1,*], Suh-Yin Lee[2]

[1] Advanced Technology Center
Computer & Communication Research Laboratories
Industrial Technology Research Institute
Chutung, Hsinchu, Taiwan 31015, R.O.C.
E-mail: jwchen@autumn.ccl.itri.org.tw

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

Abstract

This paper presents an on-line Chinese character recognition approach for personal digital assistants without writing constraints on both stroke order and stroke number. Stroke correspondence is accomplished via a rule-based approach such that combinatorial exhaustion can be avoided. Then, character distance is computed for character discrimination. For portable information processing systems, such as personal digital assistants (PDAs), the computing resource is very limited. We propose a hierarchical representation to reduce the storage requirement of the reference database, which only occupies about 1/4 of the amount without using hierarchical representation. While using the hierarchical reference database, it only requires $O(n \log n)$ time to accomplish stroke correspondence between an input script and a template of n strokes.

Keywords:

Personal Digital Assistants; On-line Chinese character recognition (OLCCR); Rule-based recognition approach; Hierarchical representation.

1 Introduction

On-line Chinese character recognition (OLCCR) is the key technology for Chinese pen-based systems. Chinese is a large-vocabulary language. Complicated structures and handwriting variations, such as variations in stroke order and stroke connection, further increase the difficulty of machine recognition. Many researchers have elaborated on developing OLCCR methods releasing writing constraints. However, most of previous methods are still under some constraints [1, 2, 3, 4, 5]. In portable information processing systems, such as personal digital assistants (PDAs), the computing resource is very limited. Therefore, the data size of a recognition system is also a concerned factor.

In this paper, we propose a rule-based approach to accomplish stroke correspondence for on-line handwriting with wide variations. Before stroke correspondence, all possible basic strokes in an input script are first recognized. Connected strokes can be segmented apart if exist. Stroke correspondence rules contain the knowledge of possible types of basic strokes allowed in handwriting and geometric features of strokes. Therefore, the variations in stroke order and stroke number can both be accommodated. A template of n strokes has n rules, and only requires $O(n)$ time to accomplish the stroke correspondence.

In Chinese characters, simple characters are constructed by basic strokes according to fixed structural rules. Most complicated characters are usually built from simple characters or simple patterns, named *components* or *radicals*, based on fixed geometric configurations – *character structures*. Therefore, we can utilize a *hierarchical representation* to reduce the requirement of storage space. The procedure of component decomposition is indispensable for retrieving information from the hierarchical reference databases. Many researchers proposed various methods of component decomposition [6, 7, 8, 9, 10]. However, some are under the constraint of fixed stroke number, some have time consuming computations, and some require large amount of storage space.

In this paper, for retrieving data from the hierarchical database, we propose a new method of component decomposition. Using the hierarchical database, although the time complexity of the stroke correspondence will increase to $O(n \log n)$ for a template char-

*Responsible for correspondence.

acter of n strokes, the requirement of storage space in the hierarchical representation is reduced to 1/4 of the amount without using hierarchical representation. Experiments were performed to verify the effectiveness of the proposed approach for PDAs.

2 A Rule-based Recognition Approach

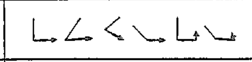
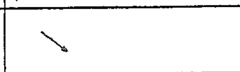
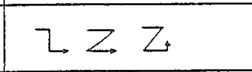
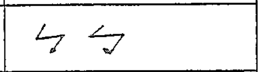
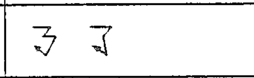
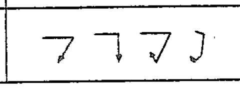
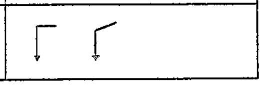
Chinese characters possess abundant structural characteristics, which are important for machine recognition. Utilizing the structural knowledge, we propose a rule-based recognition approach for OLCCR without constraints on both stroke number and stroke order. In the processing flow of our recognition system, each handwritten input character is processed by the stage of preprocessing first, in which the line-segment representation can be acquired from the series of sampled point data with noises. Then, basic strokes included in the input script are all recognized. Next, the candidate characters are selected by the preliminary classification utilizing both the estimated range of possible numbers of input strokes [11] and statistical features [12]. The distances between these candidates and the input script will be calculated in the following stage of structural analysis. The candidate with minimum distance is the recognition result. The structural analysis includes three processing steps: *stroke correspondence*, *distance calculation*, and *detailed recognition*, where the stroke correspondence is accomplished based on rules predefined for decreasing the computation time.

The strokes of Chinese characters in block style can be classified into 14 basic stroke types, as shown in Table 1. A connected stroke in hasty writing may not be classified into any of the 14 types. 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).

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

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

Table 1: Basic strokes.

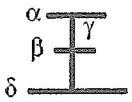
Code	Shape	Code	Shape
0		7	
1		8	
2		9	
3		10	
4		11	
5		12	
6		13	

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'. The "null" type should also be included for describing the relation without matching primitive. 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.

Definition 2.1 A stroke matching is a binary relation $r : X \rightarrow Y$ 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 $y_i (y_i = r(x_i),\ y_i \in Y)$ 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.

Chinese characters are constituted by basic strokes based on certain geometric configurations. To cope with both stroke-order and stroke-number variations, we utilize basic stroke types and 27 types of geometric features of strokes in designing stroke correspondence rules. All of them are proposed for obtaining stable stroke correspondence.



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 .

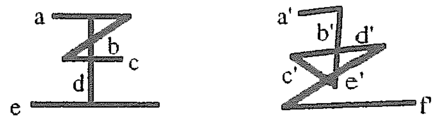
(3) Rule of stroke δ: the stroke with stroke type 0 or 4, and its center point with the minimum Euclidean to the left-bottom corner point of the charcater.

(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.

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



(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.

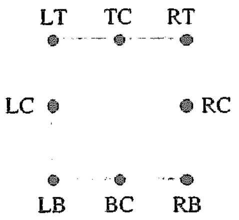


Figure 3: Eight reference points on the boundaries of the bounding rectangle of a decomposed component.

reference points on the boundaries of the bounding rectangle of the decomposed component, illustrated in Figure 3. 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.

Stroke correspondence rules used in our recognition system are classified into *type one* and *type two*, Both of them contain the information of possible types of basic strokes allowed considering handwriting variations. The information is used to eliminate those strokes impossible to be the matching stroke, of which the stroke types violates those predefined in the rule. It also has the function of segmenting connected strokes apart. A *type one* rule utilizes one geometric feature to find the matching stroke for simple characters, and a *type two* rule utilizes two for complicated characters. The order of stroke correspondence is based on the predefined rule sequence.

Each stroke in a character has an associated stroke correspondence rule. Therefore, each character has a set of stroke correspondence rules. Figure 1(b) shows the stroke correspondence rules of character "王". The matching process 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. 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. Following the input stroke order and tracing the matched pairs of fore strokes, 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.

After stroke correspondence, the matching relation between template strokes and input strokes has been acquired. Based on the relation, we can acquire the information of stroke connection. For character discrimination, we utilize structural knowledge accompanied with appropriate discriminant functions to calculate the distance of the two characters. The structural knowledge includes the spatial relationships between components, the spatial relationships between strokes within each constituent component, the lengths of lost strokes and superfluous strokes, and the information of stroke connections. The lengths of strokes can be retrieved from character patterns. When the number of candidate characters with minimum distance is more than one, these candidate characters constitute a similar group. The input character is further identified by special structural features of the similar group [13].

Theoretically, the 27 features are computed for each possible basic stroke 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 respective array. Each stroke correspondence rule would indicate that the matching stroke should have maximum or minimum value of certain type of geometric features in the remaining strokes. During stroke correspondence, the matching stroke can be determined directly based on the data in these arrays. The stroke correspondence requires $O(n)$ time for a template of n strokes based on the n rules.

Because we use hierarchical reference database, each component need to be decomposed from a character. The 14 auxiliary features of each component cannot be computed until each component is decomposed. Then, the decomposed strokes are sorted based on these 14 features, respectively. Therefore, the time complexity of stroke correspondence increases to $O(n \log n)$. Although the time complexity increases, the reduced space is considerable.

3 A Hierarchical Representation of the Reference Database

The great majority of the reference database is occupied by the structural knowledge, including stroke correspondence rules, spatial relationships between strokes, and character patterns. We propose a hier-

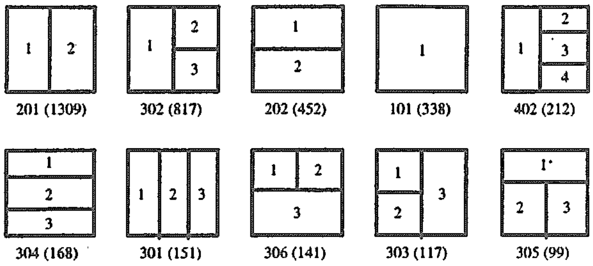


Figure 4: Geometric configurations of the character structures with occurrence frequencies in the first 10 ranks, labeled with "structure codes (frequencies)."

archical representation to reduce the requirement of storage space.

In the hierarchical reference database, each character is described by its constituent component code(s) and character structure. After analysis, we propose 622 component categories, such as "土", "木" and "月", and 208 types of character structures to describe the frequently used 5401 Chinese characters. The character structures, with occurrence frequencies in the first 10 ranks, are shown in Figure 4. Character structure 201 ranks first and there are 1309 characters in this character structure category.

We take character patterns as the example to analyze the requirement of storage space when we use the proposed hierarchical representation. Instead of storing the whole character patterns, we only store component patterns in the reference database. During data retrieval, one character pattern can be rebuilt by its constituent component pattern(s) based on the character structure predefined. For example, character "吃" has six strokes, consisting of components "口" and "乞". Character "杏" has seven strokes, which is composed of components "木" and "口". Therefore, 13 stroke patterns have to be stored for characters "吃" and "杏". If we adopt a hierarchical representation, that is, the two characters are described by the three components "口", "乞", and "木". The total number of strokes of the three components is ten. The hierarchical representation can save a large amount of data duplicates of components. Although extra space is needed to store constituent component codes and the synthesis rules for each character, the total requirement of storage space still decreases dramatically. The analyses of the space requirement about the spatial relationships between strokes and stroke correspondence rules are similar to the above analysis. The reduced storage space of the three parts is about 3/4 of the amount not using the

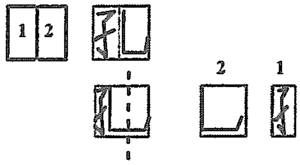


Figure 5: The decomposition result of character "孔" will be stable if component 2 is decomposed before component 1.

hierarchical representation.

Using the hierarchical representation, the decomposition of components is an essential procedure. It is to determine which strokes belong to target components predefined. *Correct strokes* are the decomposed strokes really belonging to the target component, and *erroneous strokes* are those belonging to neighbor components. Fewer erroneous strokes would make the design of stroke correspondence rules easier.

The topic of component decomposition has been investigated by many people, but their proposed methods have drawbacks mentioned previously and are not suitable for retrieving data in our recognition system. Therefore, we propose a new method of component decomposition, utilizing the knowledge of character structures, geometric features of strokes and decomposition sequences of components to decrease erroneous strokes.

The concept of this method can be illustrated by Figure 5. For character "孔", no matter what ratio in size the two components has, component 2 with one constituent stroke is always located in the extreme right of the character. If we utilize maximum x coordinate of the strokes to decompose component 2 before component 1, the erroneous stroke can be avoided even for wide handwriting variations. Some terminologies are given first for explaining the details of our proposed method.

1. **Geometric features of strokes** The four features $Xmin$, $Xmax$, $Ymin$ and $Ymax$ of the MBR of a stroke and the x, y coordinates of the MBR's center point are the six possible geometric features used in component decomposition. They can be classified into x and y coordinate measures. For character structures with left-right relations, only x measures are used in component decomposition, and for top-bottom relations, only y measures are used.
2. **Windows**: A rectangular window is utilized to represent the occupied area of the component to be decomposed. This area is covered by the ge-

ometric features of the constituent strokes. The four boundaries of the window is derived from learning samples of characters.

3. **Tolerance zone**: A *tolerance zone* of width δ is set around the derived window such that more variations in component size could be tolerated.
4. **Sequence of component decomposition**: A character structure of k components has $k!$ possible sequences of component decomposition in total. When a decomposition is carried out from surrounding components to central components, the erroneous decomposition rate would usually be less than those *vice verse*. Under the constraint, the number of sequences allowed would be much less than $k!$. People usually write Chinese characters based on the stroke order from top to bottom, from left to right, or from surrounding to central. For each character structure, we define one as the *natural sequence* of component decomposition based on such order.

Assume that character category A has character structure α. In structure α, there are L components, N possible decomposition sequences, and M possible sets of geometric features of strokes used in component decomposition. A set of handwritten samples is used in deriving the rule of component decomposition of character A. For all learning samples of character A, the minimum area, occupied by the jth set of geometric features of all strokes in the ith component, is denoted by $W(i,j)$. A surrounding tolerance zone of width δ added to $W(i,j)$ forms a larger window $W'(\delta,i,j)$, which would be used in deriving rules of component decomposition for tolerating wider variations in component size. The total number of erroneous decomposition strokes in the ith component is denoted by $e_A(\delta,i,j,k)$, where the kth decomposition sequence and the jth set of geometric features are used in component decomposition. The total number of erroneous strokes in the whole character denoted by $E_A(\delta,j,k)$ is described by the following equation

$$E_A(\delta,j,k) = \sum_{i=1}^{L} e_A(\delta,i,j,k). \quad (1)$$

The minimum number of erroneous strokes $\gamma_A(\delta)$ can be obtained by the following equation

$$\gamma_A(\delta) = \min_{1\le j\le M, 1\le k\le N} E_A(\delta,j,k) = E_A(\delta,j',k'). \quad (2)$$

For 208 types of character structures, there are 16 possible spatial relationships between a decomposed component and its neighbor components remained in the script, labeled by *position codes*, as shown in Figure 6. Each white area in Figure 6 indicates the location of the component being decomposed and each marking area indicates the occupied area of its neighbor components. A component category may have more than one position code when it is decomposed from different character categories. During decomposition, the erroneous strokes belonging to neighbor components may be included and located in the different boundary areas of the component, as shown in Figure 7. For the same component category, more than one set of rules may be needed to cope with the different cases. This fact will increase the requirement of storage space.

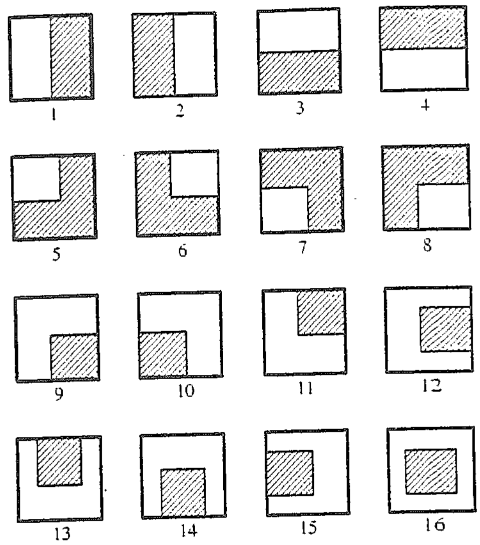


Figure 6: Sixteen types of spatial relationship between a decomposed component and its neighbor component(s), labeled with *position codes*.

To reduce the number of rule sets, we add a constraint in deriving the decomposition rule. If a component category occurs in both positions of code 1 and code 2 in the 5401 characters, we can restrict the decomposition sequence on the *natural sequence* such that the component category has only one position code, either code 1 or 2. Similarly, a decomposition sequence can be restricted such that a component category can have only either position code 3 or 4; can have only one of code 5, 6, 7, or 8. Therefore, if the minimum number of erroneous strokes $\gamma_A(\delta)$ derived from equation 2 is not equal to zero, we adopt a *predefined sequence - natural sequence* in the rule, and choose a set of geometric features of strokes to minimize the number of erroneous strokes based on the

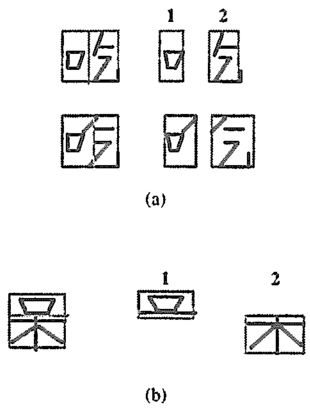


Figure 7: Strokes of neighbor components may be erroneously included in the decomposed component. (a) An erroneous stroke appears in the right of decomposed component "口"; (b) an erroneous stroke appears in the bottom of decomposed component "口".

following equation

$$\gamma_{A,n}(\delta) = \min_{1 \leq j \leq M} E_A(\delta, j, n) = E_A(\delta, j'', n), \qquad (3)$$

where n denotes the natural sequence and the j''th set of geometric features of strokes, being the best result, is adopted in the decomposition rule of character category A. Using the hierarchical reference database, the total space is about 1/4 of the amount not using the hierarchical representation. A storage space about 260 Kbytes is enough for storing the three major parts of the structural knowledge such that the proposed approach can be adopted by PDAs.

4 Experiments and Results

To verify the effectiveness of our proposed approach, we performed two experiments. The first is for verifying recognition accuracy and speed of the structural analysis. Each character category has 11 testing samples taken from the *ITRI OLCCR* database [14]. We used the estimated range of the number of input strokes alone in preliminary classification. For 1225 categories randomly selected from the 5401, the average number of candidate characters was 274. On the average, the first rank recognition rate was 93.51%; the 5th rank cumulative recognition rate reached 98.16%; the recognition speed was about 0.5 second per character on a Pentium-100 based PC. The recognition rate will be further improved by incorporating more variations into the reference database.

When the recognition system is extended to recognize 5401 or more categories, the number of candidates selected by the preliminary classification will influence the recognition speed. 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. For 5401 categories, we used 23 testing samples per character category, also taken from the *ITRI OLCCR* database. 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. The recognition speed would further be improved by utilizing other effective features in preliminary classification.

5 Conclusions

This paper presents an OLCCR approach for PDAs without constraints on both stroke number and stroke order. The stroke correspondence requires $O(n \log n)$ time for a template of n strokes when the reference database is 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 PDAs and other portable systems with very limited computing resource for recognizing a large character set. The proposed architecture of the recognition system could also be applied in recognizing characters written in various styles by using the rules suitable for the style to be recognized.

Stroke correspondence rules used in experiments were inducted from character samples by a semi-automatic approach. Authors designed stroke correspondence rules utilizing programs to speed up the induction. The training of the reference database is usually the major and the most important work for developing a recognition system. Therefore, there is a great demand for developing automatic learning environment. If cursive handwriting with variations to very high degrees were recognized through machines, the stroke correspondence rules might contain more features and should be further investigated. These issues are worthy of future research.

Acknowledgements

The author would like to thank the research grant supported by projects numbered 35N7100 and 3N22400, 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] 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.

[5] 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.

[6] F. H. Cheng and W. H. Hsu, "Partial pattern extraction and matching algorithm for Chinese characters," *Proc. 6-th IASTED International Symposium, ROBOTICS & AUTOMATION'85*, Santa Barbara, U.S.A., May 1985, pp. 18-22.

[7] F. H. Cheng and W. H. Hsu, "Radical extraction by background thinning method for handwritten Chinese Characters," *Proc. 1987 Int. Conf. on Chinese Computing*, Jun. 1987, pp. 175-182.

[8] K. P. Chan and Y. S. Cheung, "Fuzzy-attribute graph and its application to the Chinese character recognition," *Computer Processing of Chinese and Oriental Languages*, Vol. 4, Nos. 2 & 3, pp. 85-98, 1989.

[9] S. W. Lu and C. Y. Suen, "Hierarchical attributed graph representation and recognition of handwritten Chinese characters," *Pattern Recognition*, Vol. 24, No. 7, pp. 617-632, 1991.

[10] S. L. Shiau, S. J. Kung, A. J. Hsieh, J. W. Chen, and M. C. Kao, "Stroke-order free on-line Chinese character recognition by structural decomposition method," *2nd International Workshop on Frontiers in Handwriting Recognition*, pp. 21-31, Sep. 1991, Bonas France.

[11] J. W. Chen and S. Y. Lee, "On-line handwritten Chinese character recognition via a fuzzy attribute representation," *Image and Vision Computing*, Vol. 12, No. 10, pp.669-681, 1994.

[12] L. T. Tu, Y. S. Lin, C. P. Yeh, I. S. Shyu. J. L. Wang, K. H. Joe, and W. W. Lin, "Recognition of handprinted Chinese characters by feature matching," *Proc. 1st National Workshop on Character Recognition*, R.O.C, 1991, pp. 166-175.

[13] J. W. Chen, "Spatial relationship and structure-based representation and recognition of on-line Chinese characters," PhD thesis, National Chiao Tung University in Taiwan, 1995.

[14] 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,