A Hierarchical Representation for the Reference Database of On-Line Chinese Character Recognition

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

[1] Institute of Computer Science and Information Engineering
National Chiao Tung University, Hsinchu, Taiwan 30050, R.O.C.
[2] Application Software Department, Computer & Communication Research Laboratories,
Industrial Technology Research Institute, Chutung, Hsinchu, Taiwan 31015, R.O.C.

Abstract. On-line handwritten Chinese characters 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. These variations make machine recognition difficult. There are multitudinous categories and complicated structures in Chinese characters. Therefore, for an OLCCR system, efficient storage techniques of the reference database are important especially for those portable computers with very limited computing resources. Basically, simple Chinese characters are constructed by basic strokes according to fixed structural rules, and complicated characters are composed of radicals or components based on fixed geometric configurations. In this paper, we propose a hierarchical representation for the reference database of an OLCCR system using a structural recognition method, in which both stroke-number and stroke-order variations can be tolerated in the recognition system. The major structural knowledge used in recognition includes stroke correspondence rules, spatial relationships between strokes, and character patterns. We utilize components and character structures to represent characters. Only the structural knowledge of components are stored, and the structural knowledge of each character can be retrieved based on its constituent components and its character structure. Both the representation method and the retrieving method of the structural knowledge are proposed in the paper. The storage requirement of the reference database is reduced to 1/4 of the amount without using hierarchical representation.

1 Introduction

On-line Chinese character recognition (OLCCR) systems are one kind of man-computer interfaces and is usually combined with various application software. The data size is then a concerned factor in the portable computing environment, especially for Chinese with large vocabularies.

In Chinese characters, most complicated characters are usually built from simple characters or simple patterns, named *components* or *radicals*, based on

fixed geometric configurations – *character structures*. Simple characters are constructed by basic strokes according to structural rules. Therefore, *structural recognition* is a promising approach. *Hierarchical representation* can reduce the requirement of storage space dramatically. Component decomposition is indispensable for retrieving information from a hierarchical reference databases.

Some researchers proposed various implementation methods of hierarchical representation for Chinese characters [1, 2]. Many researchers proposed various methods of component decomposition [1, 3, 4, 5, 6, 7, 8, 9]. However, some methods are under the constraint of fixed stroke number, some have time consuming computations, and some require large amount of storage space.

In this paper, we propose a hierarchical representation for the reference database of an OLCCR system without constraints on both stroke number and stroke order, and also propose a new method of component decomposition for data retrieving. The requirement of storage space in the hierarchical representation is about 1/4 of the amount without using hierarchical representation. The data retrieving from the hierarchical database requires $O(n)$ time, where n is the number of strokes in the retrieved character.

2 A Structural Recognition Approach

We adopt a structural recognition approach in the recognition system described in this paper. Although basic strokes are the character primitives, there may be some basic strokes connected together in handwritten scripts. Therefore, all *possible basic strokes* existing in an input script are all recognized such that connected strokes can be segmented apart.

When an input script is entered into the recognition system, it is processed through stages of preprocessing, basic stroke recognition, and preliminary classification. The candidate characters selected by the preliminary classification are taken to match against the input pattern pair by pair using a stroke-based matching. We utilize rules to accomplish the stroke correspondence between a candidate and an input pattern avoiding combinatorial exhaustion. Stroke correspondence rules contain the knowledge of basic stroke types and geometric features of strokes. Later, we use some structural knowledge to compute the similarity of the two characters, including the spatial relationships between constituent components, the spatial relationships between strokes within each component, and the relative length of lost strokes as well as superfluous strokes in the input script accompanied with adequate discriminant function. The relative length of strokes are computed 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 characters in the similar group [10, 11].

Stroke correspondence rules, spatial relationships between strokes, and character patterns occupy the great majority of the storage space. We utilize the following hierarchical representation to reduce the requirement of storage space.

3 Hierarchical Representation in Character Database

After analyzing the frequently used 5401 Chinese characters, we propose 622 component categories, such as "土", "木", "月", etc., and 208 types of character structures. The character structures, with occurrence frequencies in the first 10 ranks, are shown in Figure 1. Character structure 201 ranks first and there are 1309 characters in this character structure category. In the following, we analyze the space requirement of the 5401 character categories when a hierarchical representation is implemented.

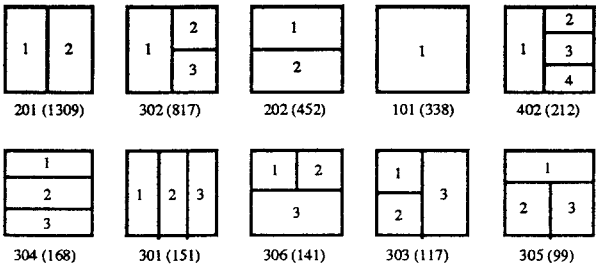

Fig. 1. Geometric configurations of the character structures with occurrence frequencies in the first 10 ranks, labeled with "structure codes (frequencies)."

3.1 Hierarchical Representation of Character Patterns

In the proposed hierarchical representation, only component patterns are stored. Each character is described by its constituent component code(s) and character structure. During retrieving patterns, one character pattern can be rebuilt by the constituent component pattern(s) based on the character structure predefined, and can be illustrated by Figure 2. In Figure 2(a), character "吃" has six strokes and is composed of components "口" and "乞". Character "杏" has seven strokes and is composed of components "木" and "口". Therefore, 13 stroke patterns have to be stored for characters "吃" and "杏" without hierarchical representation. The spatial relationships between strokes of character "吃" are denoted by symbols *l, m, n, o, p,* and those of character "杏" are denoted by symbols *q, r, s, t, l, m.* Each character pattern can be described by constituent line segments. In Figure 2(b), the total number of strokes is ten for the three components "口", "乞", and "木". Many components are frequently used to represent the 5401 character categories. Although extra space needs to store constituent component codes and the synthesis rules for each character, the requirement of storage space still decreases in total.

Without hierarchical representation, the data size of 5401 character patterns requires at least 250 Kbytes. By using the hierarchical representation, the total data size of character patterns is about 135 Kbytes. The reduction rate is about 1/2 of the original.

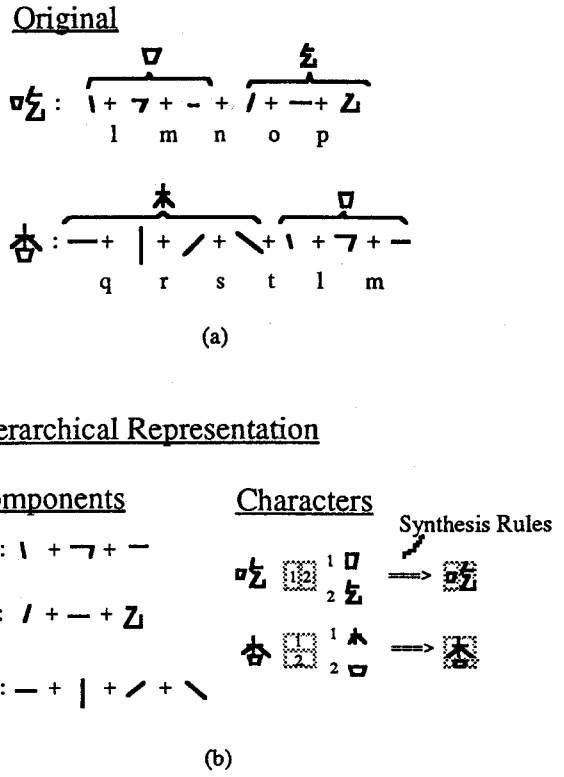

Fig. 2. Representations of character patterns "吃" and "杏". (a) Representation without hierarchical strategy; (b) hierarchical representation.

As to spatial relationships between strokes of characters, the effectiveness of the hierarchical representation is also proved in a similar way, and is illustrated by Figure 3. The character description of character patterns can also be used in retrieving the spatial relationships between strokes of characters, the duplication is eliminated. The space requirement for storing the spatial relationships between strokes for 622 components is about 13Kbytes, and is about 6% of the original.

3.2 Hierarchical Representation of Stroke Correspondence Rules

In a similar way, only the stroke correspondence rules of components are stored in the reference database in the hierarchical representation. When an input character is matched with a template character, each component included in that template would be decomposed from the input character one by one, and each decomposed component is further decomposed stroke by stroke according to the stroke correspondence rules. Figure 4 illustrates the representation of stroke correspondence rules by two characters "吃" and "杏". Characters "吃" and

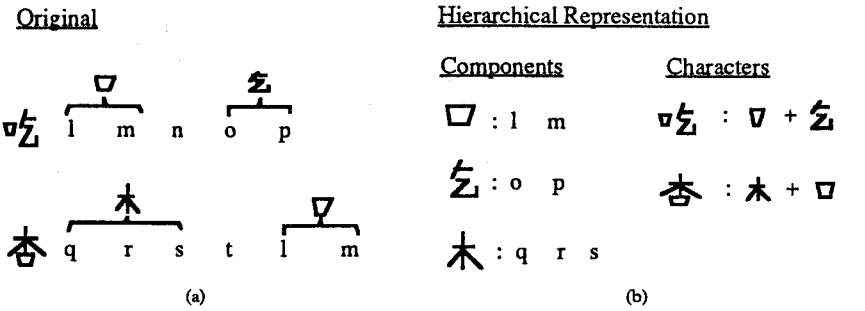

Fig. 3. Representations of spatial relationships between strokes of characters "吃" and "杏". (a) Representation without hierarchical strategy; (b) hierarchical representation.

"杏" have 13 rules to be stored without hierarchical representation, but the three components "口", "乞", and "木" have only ten in the hierarchical representation. Because many components are used frequently to represent the 5401 character categories, the reduced space is considerable. However, the decomposition of components is an essential procedure for retrieving information from the hierarchical representation.

Decomposition of components (or *radical separation*) 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 decomposed stroke belonging to neighbor components. The fewer erroneous strokes are, the easier the design of stroke correspondence rules is.

The topic of component decomposition has been investigated by many people, but their proposed methods have the drawback mentioned previously and can not be used for retrieving data in our recognition system. Therefore, we propose a new method of component decomposition, and utilize the knowledge of character structures, geometric features of strokes and decomposition sequences of components in order 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** Assume that each stroke is bounded by a minimum bounding rectangle (*MBR*). The *Xmin, Xmax, Ymin, Ymax* of the MBR the coordinates of the MBR's center point are the 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.

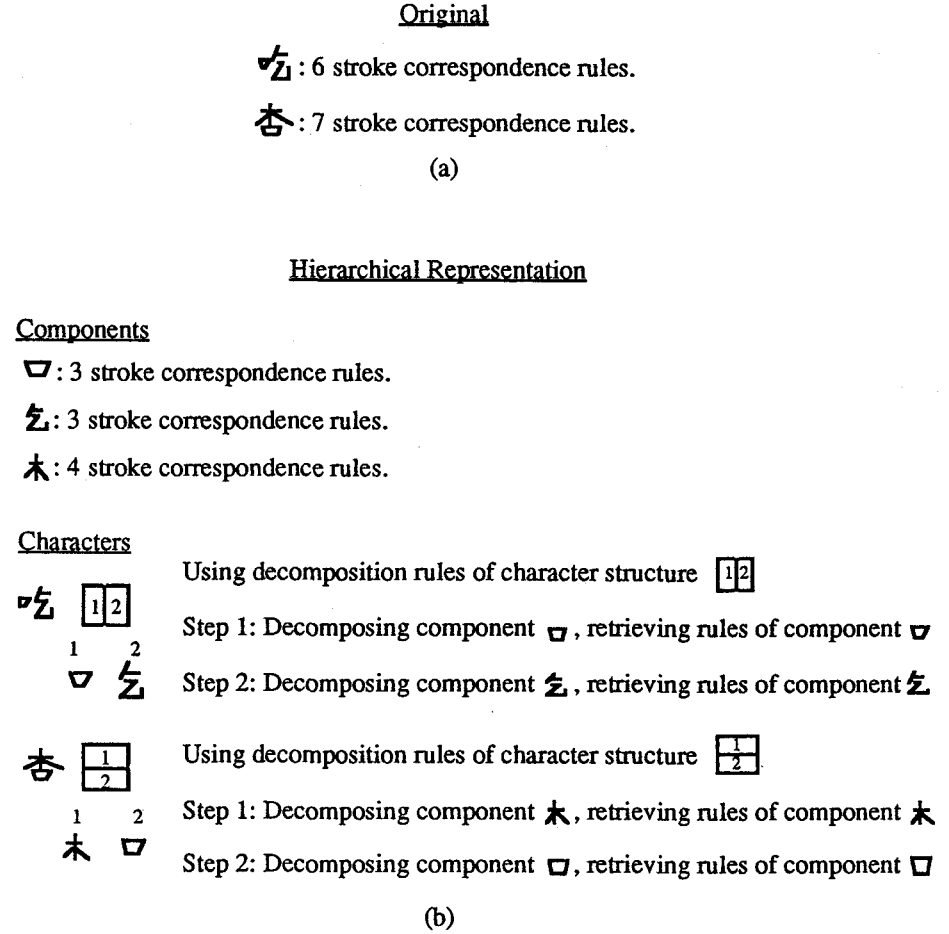

Fig. 4. Representations of stroke correspondence rules of characters "吃" and "杏". (a) Representation without hierarchical strategy; (b) hierarchical representation and retrieving process.

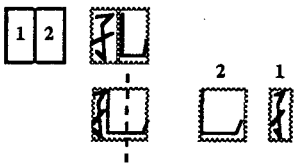

Fig. 5. The decomposition result of character "孔" will be stable if component 2 is decomposed before component 1.

2. **Windows**: A rectangular window is utilized to represent the occupied area of the component to be decomposed. This area is covered by the geometric 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**: When a character structure consists of k components, there are $k!$ possible sequences of component decomposition. 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 feature measures 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 $W(i,j)$. When a surrounding tolerance zone of width δ added to $W(i,j)$ forms a larger window $W'(\delta,i,j)$, it would be used in deriving rules of component decomposition such that more variations in component size could be accommodated. 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

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

For all possible decomposition sequences and geometric features of strokes, the minimum number of erroneous strokes $\gamma_A(\delta)$ can be obtained by the following equation

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

For character A, the rule of component decomposition utilizes the j'th set of geometric features and the k'th decomposition sequence.

For 208 types of character structures, when one component is decomposed from an input script, there are 16 possible spatial relationships between the 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

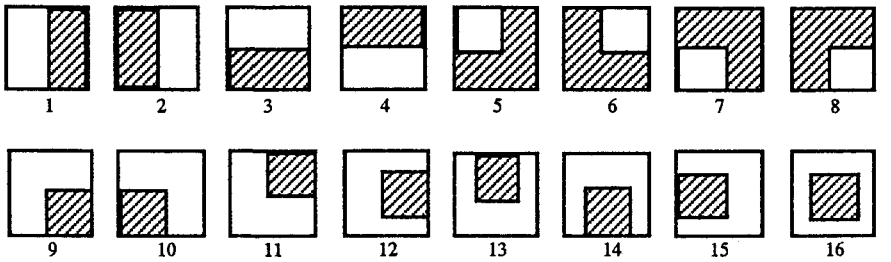

Fig. 6. Sixteen types of spatial relationship between a decomposed component and its neighbor component(s), labeled with *position codes*.

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 in boundary areas of the component, as shown in Figure 7. For the same component category, different rules are needed in different cases. This fact will increase the requirement of storage space.

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

In order to further reduce the number of rule sets, we add some constraints in deriving the rule of component decomposition for a character. 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 the natural sequence in the rule. In

order to minimize the number of erroneous strokes, we choose a set of geometric features of strokes according to the following equation

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

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.

In this way, the number of rule sets will not be more than three times the number of component categories. A space of 115 Kbytes is enough for storing the stroke correspondence rules of all components when the hierarchical representation is adopted. Table 1 shows the comparison of storage space between the amount of the hierarchical representation and the amount without hierarchical representation. The total space using hierarchical representation in Section 3.1 and 3.2 is about 1/4 of the amount not using the hierarchical representation.

Table 1. Comparison of storage space between the amount of hierarchical representation and the amount without hierarchical representation.

Structural Information	Hier. Rep.	Without Hier. Rep.
Character Patterns	135	> 250
Spatial Relationships between Strokes	13	250
Stroke Correspondence Rules	< 115	461
Total Space (Kbytes)	< 263	> 961

4 Conclusions

In this paper, we propose a hierarchical representation for the reference database of an OLCCR system without constraint on both stroke number and stroke order. Three major parts: stroke correspondence rules, spatial relationships between strokes, and character patterns are represented hierarchically. Only the structural knowledge of components are stored, and the structural knowledge of each character can be retrieved based on its constituent components and its character structure. For retrieving stroke correspondence rules from the hierarchical representation, we propose a new method of component decomposition, and also propose an algorithm which can automatically find the rules of component decomposition. The storage requirement of the reference database is reduced dramatically, and only occupies about 1/4 of the amount without using hierarchical representation.

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. 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.
2. S. L. Shiau, J. W. Chen, A. J. Hsieh, and S. J. Kung, "On-line handwritten Chinese character recognition by string matching," *Proc. the International Conference on Computer Processing of Chinese and Oriental Languages,* Toronto Canada, 1988, pp. 76-80.
3. 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.
4. 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.
5. N. Tanaka, H. Aota, M. Shiono, H. Sanada, and Y. Tezuka, "A method of subpattern extraction from handprinted KANJI characters," *IEICE Technical Report,* (in Japanese), Vol. PRL83-26, 1983.
6. B. S. Chan, T, H. Lin, J. U. Yan, and J. G. Lin, "Automatic radical extraction system of Chinese characters," *Proc. National Computer Symposium, R.O.C.*, Dec. 1985, pp. 1216-1224.
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. H. Ogawa and K. Taniguchi, "Extraction of partial pattern in handprinted Chinese character by relaxation," *Trans. IEICE Technical Report,* (in Japanese), Vol. PRL80-40, 1980.
9. 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.
10. Ju-Wei Chen and Suh-Yin Lee, "A new stroke spatial relationship representation for Chinese characters," *Proc. the International Conference on Computer Processing of Chinese and Oriental Languages*, Florida U.S.A., 1992, pp. 333-340.
11. Ju-Wei Chen, "Spatial Relationship and Structure-based Representation and Recognition of On-line Chinese Characters," PhD thesis, National Chiao Tung University in Taiwan, 1995.