

On-line Handwritten Chinese Character Recognition Based on Nested Segmentation of Radicals

Long-Long Ma, Cheng-Lin Liu

National Laboratory of Pattern Recognition,
Institute of Automation, Chinese Academy of Sciences
95 Zhongguancun East Road, Beijing 100190, P. R. China
{longma, liucl}@nlpr.ia.ac.cn

Abstract: This paper presents a radical-based on-line handwritten Chinese character recognition method, which integrates appearance-based radical recognition and geometric context into a principled framework using a character-radical dictionary to guide radical segmentation and recognition during path search. To solve the connection between radicals, we detect corner points to extract sub-strokes. Based on the hierarchical structure, the character pattern is over-segmented by three-layer nested pre-segmentation. For recognition, we use two dictionary representation schemes and accordingly different search algorithms. We have implemented the approach to Chinese characters of left-right and up-down structures. Experimental results on a sample set of 5,773 character classes consisting of 1,149 radicals demonstrate the effectiveness of our approach.

Key Words: On-line handwritten Chinese character recognition, hierarchical structure, pre-segmentation, path search, radical recognition

1 INTRODUCTION

With the emergence of hand-held devices, such as Pocket PCs and mobile phones, on-line handwritten Chinese character recognition (OLCCR) is gaining renewed interest. In the last decades, many approaches have been proposed and the recognition performance has advanced constantly [1]. To be implemented in hand-held devices with limited computation and storage capability, researchers are working towards high accuracy recognition methods with lower complexity.

The hierarchical nature of Chinese characters and Hangul characters has inspired radical-based recognition methods, which model a much smaller number of radicals instead of characters. Hierarchical character representation has been used in Hangul character recognition [2][3], where components (graphemes) and the relationships between them are represented statistically. A Kanji (Japanese Chinese) character recognition method uses stochastic context-free grammar (SCFG) to combine strokes into characters [4]. Such hierarchical representation-based methods have three benefits. First, model complexity is reduced by modeling radical shapes instead of holistic character shapes. Second, by focusing on radicals with simpler structures than characters, the recognition accuracy can be improved. Third, the classification of a small number of radicals needs a small set of training samples.

All radical-based recognition methods encounter the difficulty of radical segmentation, however. Rule-based radical detection using the prior knowledge of character structure and radical position [5] is likely to fail in cases of large shape variation. Based on a network representation of radical and ligature HMMs [6], radicals can be segmented by dynamically matching the radical models with sub-sequences of strokes. This approach does not tolerate stroke-order variations,

however. A method avoids radical segmentation by radical location detection and location-dependent radical classification using neural networks on whole images [7], but without radical segmentation, it suffers from the large number of location-dependent radical models and the low radical classification accuracy.

To overcome the difficulty of radical segmentation, we proposed a radical-based recognition approach for characters of left-right structure [8], which combines the merits of hierarchical structure and appearance-based classification of radicals. The approach is similar to character string recognition [9] in the senses of candidate radical segmentation and tree representation of character compositions. It uses radical recognition to guide radical segmentation and by using appearance-based radical models, is free of stroke order.

In this paper, we extend the method of [8] in several respects: adding sub-stroke extraction stage to handle the connection between radicals, three-layer nested hierarchical pre-segmentation and two representation schemes of character-radical dictionary. We have applied the radical-based approach to Chinese characters of left-right and up-down structures. In experiments on a sample set of 5,773 character classes consisting of 1,149 radicals, the extended method yielded higher accuracies than a holistic statistical method while consuming smaller storage space.

2 FORMATION OF RADICAL MODELS

Many Chinese characters share common sub-structures called radicals. Most radicals have semantic meanings and often a radical is also a single character. It is beneficial to use radicals as the units of classification because the number of radical classes is much smaller than the number of characters and the radicals have simpler structures. To recognize characters, the radical models are represented in a hierarchical

structure which guides radical segmentation. Such radical model-guided recognition scheme is helpful to overcome the ambiguity of radical segmentation caused by character shape variation and stroke connection.

The definition of radical classes determines the way of radical segmentation and character recognition. Our previous method [8] simply uses radicals that are horizontally separable, and so, each character can be represented as a string of radicals. To better utilize the hierarchical structure of radical composition, we extend to use nested segmentation of radicals that are horizontally or vertically separable in recursion. We take the character “陞” as an example. In Fig.1, (a) denotes a character pattern, (b) shows the horizontally separable radicals as used in [8], and (c) shows the radicals used in nested segmentation.



Fig. 1: A charater pattern decomposed into different radicals.

We obtained radical models by embedded learning in two stages just as in [8]. First, character samples of each class are matched with a correct sequence of radicals (segmented by human interaction) using dynamic programming (DP) to segment into radicals. The class-specific radical templates are obtained by averaging the radicals (each represented by a feature vector) of each class. Second, the radical templates of all classes are clustered using agglomerative clustering to obtain shared radical models. Fig. 2 shows the extracted radicals of some character patterns.

Extracted radicals			
Character	Radicals	Character	Radicals
陞	阝 土 土	陞	阝 土 土
陞	阝 土 土	陞	阝 土 土
陞	阝 土 土	陞	阝 土 土
陞	阝 土 土	陞	阝 土 土
陞	阝 土 土	陞	阝 土 土
陞	阝 土 土	陞	阝 土 土
陞	阝 土 土	陞	阝 土 土
陞	阝 土 土	陞	阝 土 土

Fig. 2: Extracted radicals of some character patterns.

Currently, we have applied the radical extraction method to Chinese characters of left-right structure (4,284 classes) and

up-down structure (1,489 classes). By clustering the radical templates of 5,773 characters, we obtained 1,149 shared radical models (cluster centers).

3 SYSTEM OVERVIEW

The block diagram of our radical-based recognition system is shown in Fig. 3. Compared to the approach in [8], we add a stage of sub-stroke extraction to handle the connection between radicals due to cursive writing, and use three-layer nested pre-segmentation instead of one layer pre-segmentation.

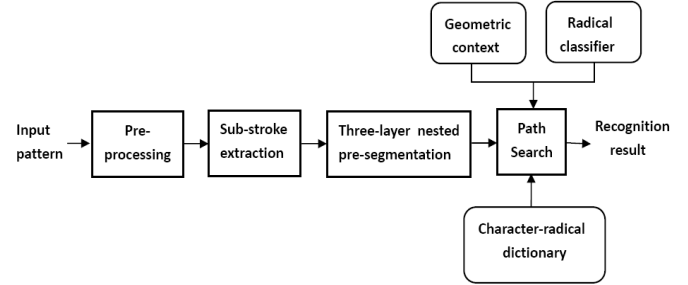


Fig. 3: Block diagram of radical-based recognition system.

The dictionary of character-radical compositions is represented in two schemes: sequential representation (Fig. 4(a)) and hierarchical representation (Fig. 4(b)).

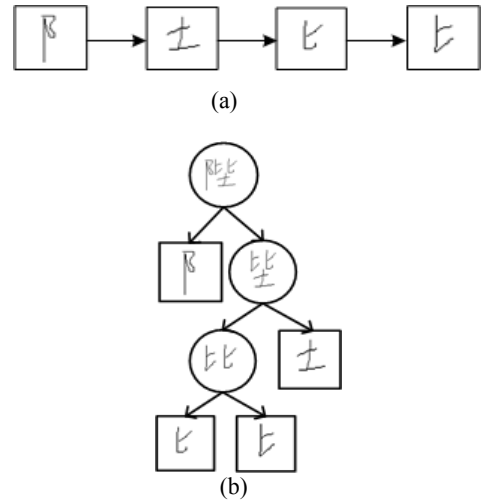


Fig. 4: Two representation schemes of character-radical dictionary

The purpose of sequential representation is to take advantage of radical string matching and tree (trie) structure of the whole dictionary as in [8]. For each character class, the order of radicals is determined according to writing order (we assume that the writing order of radicals is relatively stable but allow stroke-order variation within radicals). The trie structure of character-radical dictionary helps save both the storage of dictionary and the computation of recognition because the common prefix radicals are stored only once and matched once in path search. In our experimental case, the trie of 5,773 left-right and up-down characters has 7,645 nodes consisting of 1,149 distinct radicals. A portion of the trie is shown in Fig.5.

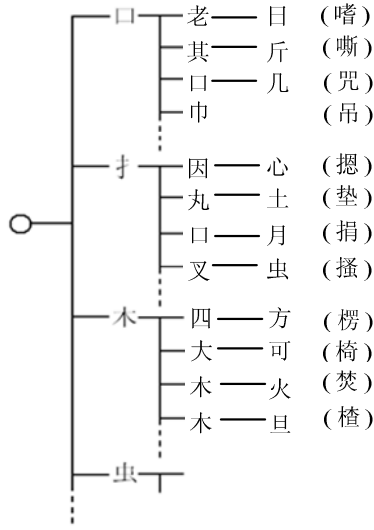


Fig. 5: A portion of the trie structure for sequential representation.

By hierarchical representation, each character class is represented as a tree structure indicating the radicals and the relationships between them. Hierarchical representation is more informative and is totally free of radical order. However, it is not easy to unify the tree structures of different characters into a single data structure as the trie for sequential representation. In recognition, the input pattern is matched the character models one by one.

4 RADICAL-BASED RECOGNITION

For character recognition, we first split the input strokes into sub-strokes to handle the connection between radicals. Radicals are generated in pre-segmentation, and are matched with the character models to determine the final radical segmentation and character identity.

4.1 Sub-Stroke Extraction

Each input stroke is split into sub-strokes at corner points, which are detected according the change of local direction angles. For each sampled point at the stroke, the moving direction angle from its preceding point is calculated. To overcome the fluctuation of stroke trajectory, the angles of the sequence of points are smoothed by Gaussian filtering. For a point p with direction angle θ_p , the average angle α_p of its t preceding points is calculated. If the abstract difference between θ_p and α_p is greater than a threshold (empirically set as $\pi/4$ in our experiments), the point p is detected as a corner point. We detect corner points from the third point to the third from the last, and $t=2$. Fig. 6 shows some examples of detected corner points, at which the strokes are split into sub-strokes



Fig. 6: Examples of corner point detection.

4.2 Three-Layer Nested Pre-Segmentation

Sub-strokes are grouped into blocks using three-layer nested pre-segmentation, and the blocks are combined into radicals during model matching by path search. At each layer, sub-strokes or blocks are merged according to the horizontal or vertical overlapping degree between them. The overlapping degree is computed incorporating the span, overlap length and distance between the bounding boxes (as in [8]). Two dictionary representation schemes have the same pre-segmentation process, except that for sequential representation, pre-segmentation is completed before path search, while for hierarchical representation, the grouping of blocks at a layer depends on the matching result of its upper layer. The grouping process is as follows.

The horizontal grouping of sub-strokes into blocks at the first layer is the same as the way in [8]. At the second layer, each block of the first layer is further segmented into smaller blocks vertically. And at the third layer, the blocks generated at the second layer are further segmented horizontally. Most Chinese characters can be segmented into radicals by this three-layer segmentation. Fig.7 shows an example of three-layer radical segmentation.

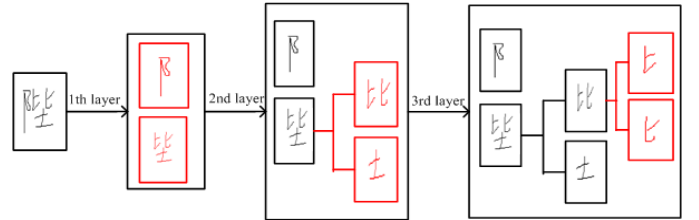


Fig. 7: Three-layer pre-segmentation process.

4.3 Path Search

Depending on the representation scheme of character-radical dictionary, we use two different search algorithms in character recognition. In both cases, we integrate geometric information as well as radical recognition scores into path evaluation.

4.3.1 Beam Search for Sequential Representation

By sequential representation, the radical sequences of all characters are stored in a lexicon of trie structure. The sequence of stroke blocks (primitive segments) of the input pattern is matched with all the radical sequences in the lexicon simultaneously by beam search, in a similar way to lexicon-driven character string recognition [9].

In lexicon-driven matching, the stroke blocks of input pattern are dynamically combined into candidate radicals, which are matched with the radical models corresponding to the offspring nodes of a node in the trie. Each candidate radical is formed by at most six consecutive blocks of strokes. By beam search, nodes in the search space with low radical matching scores are pruned to speed up search. Finally, the optimally matched string of radicals gives the result of radical segmentation and character recognition. More details of the search process can be found in [8].

4.3.2 DP Search for Hierarchical Representation

In hierarchical representation, each class is represented as a tree structure with up to three layers (excluding the root), each layer being a sequence of radicals (terminal nodes) or meta-radicals (non-terminal nodes). For recognition, the input pattern is matched with the tree structures one by one, and the matching process with one class has up to three layers. At each layer, the input pattern (or a block of strokes) is matched with a layer of the tree by dynamic programming (DP) to segment into radicals or meta-radicals. When matching a block with a non-terminal node, a sub-procedure of DP is called to segment the block into radicals/meta-radicals of the next layer and give the matching score of meta-radical. Algorithm 1 depicts the matching process of input pattern with the tree structure template. Assume $r_1^{(i)}, r_2^{(i)}, \dots, r_{m_p}^{(i)}$ and $s_1^{(i)}, s_2^{(i)}, \dots, s_{n_p}^{(i)}$ respectively denote the model sequences and pre-segmentation results at the i^{th} layer, $1 \leq i \leq 3$. $D(i)$ is used to record the minimum partial cost up to the i^{th} layer. $D(i, p, m_p, n_p)$ denotes the optimal matching cost during m_p model sequences and n_p primitive segments at the i^{th} layer.

Algorithm 1 Nested DP

Input: the pre-segmentation results $s_1^{(1)}, s_2^{(1)}, \dots, s_{n_1}^{(1)}$.

(1) Initiation. $D(0) = 0, i=1$.

(2) Recursion. $D(i) = D(i-1) + \sum_{p=1}^k D(i, p, m_p, n_p)$, where k

denotes the number of meta-radicals at the $(i-1)^{\text{th}}$ layer.

(3) Terminate until there aren't meta-radicals at the i^{th} layer. Postscript:

The accumulated partial cost of $D(i, p, m_p, n_p)$ is:

$$D(s, t, i, m_p, n_p) = \min_{k_i} [D(s - k_i, t - 1, i, m_p, n_p) + d(< s_{t-k_i+1}^{(i)} \dots s_t^{(i)} >, r_t^{(i)})],$$

where $s = 1, \dots, m_p, t = 1, \dots, n_p, k_i$ denotes the number of primitive segments in candidate pattern.

$d(< s_{t-k_i+1}^{(i)} \dots s_t^{(i)} >, r_t^{(i)}) = -\log P(< s_{t-k_i+1}^{(i)} \dots s_t^{(i)} >, r_t^{(i)})$ (to be referred in formula (2)).

4.3.3 Geometric Context Modeling

The likelihood of geometric feature of radicals (unary geometry) or the relationship between neighboring radicals (binary geometry) is measured using Gaussian probability density functions (PDF) on attributes of single radical patterns or pairs of radicals. For sequential representation, we model the relationship between consecutive radicals, and for hierarchical representation, model the relationship between neighboring radicals/meta-radicals at each layer. A PDF is built for each radical class or a pair of radical classes, and the negative log-likelihood is taken as the cost of matching.

The unary geometric features extracted from single radical pattern are: width, height, (x,y) coordinates of the center of

bounding box, normalized with respect to the character size. The binary geometric features extracted from a pair of radical patterns are: the signed differences of width and height, and of x- and y-coordinates of the centers of bounding boxes.

4.3.4 Path Evaluation

In beam search with sequential representation, a path in the search space corresponds to a segmentation of input pattern into candidate radicals. In nested DP with hierarchical representation, a path at each layer corresponds to the segmentation of a block of strokes into radicals/meta-radicals. In either case, the evaluation of paths is crucial to the result of optimal path search.

To match a sequence X of blocks with a string $Q = q_1 \dots q_{T_0}$ of radicals, X can be segmented into various strings $S = s_1 \dots s_{T_0}$ of candidate radicals. The optimal segmentation path S^* and string class Q^* are defined by

$$(Q^*, S^*) = \arg \max_{(Q, S)} P(Q, S) = \arg \min_{(Q, S)} -\log P(Q, S) \quad (1)$$

Incorporating radical class likelihood and geometric context likelihood, $\log P(Q, S | X)$ can be approximately factorized into (referring to [10])

$$\log P(Q, S) = \sum_{i=1}^{T_0} (\lambda_1 \log P(S_i^c | q_i) + \lambda_2 \log P(S_i^{g_1} | q_i) + \lambda_3 \log P(S_i^{g_2} | q_{i-1}, q_i)) \quad (2)$$

where $\lambda_1 + \lambda_2 + \lambda_3 = 1$, T_0 is the length of candidate radical string path, $P(S_i^c | q_i)$ is the likelihood of radical pattern S_i^c

with respect to class q_i , $P(S_i^{g_1} | q_i)$ and $P(S_i^{g_2} | q_{i-1}, q_i)$ measure the unary and binary geometric context, respectively. S_i^c , $S_i^{g_1}$ and $S_i^{g_2}$ correspond to the feature vectors characterizing radical shape, unary geometry and binary geometry of S_i , respectively.

After path search, we keep many candidate results with the accumulated cost. The optimal radical string is given with the smallest average cost using T_0 to normalize the accumulate cost.

5 EXPERIMENTAL RESULTS

We evaluated the performance of the proposed radical-based recognition approach on a database of online handwritten Chinese character database of 6,763 classes (the characters in GB2312-80), each class with 60 samples produced by 60 writers. The samples of 5,773 classes of left-right and up-down structures were used in our experiments. We used 50 samples per class for training classifiers for radical and character recognition, and the remaining 10 samples per class for evaluating the performance.

Each radical pattern undergoes the same procedures of normalization, feature extraction and classification as done in a holistic character recognition method [11]. Specifically, a moment normalization method is used to normalize the coordinates of pen trajectory points, and direction histogram features are extracted directly from pen trajectory using a

normalization-cooperated feature extraction (NCFE) method [12]. The resulting 512-dimensional feature vector is reduced to 160 by Fisher linear discriminant analysis (LDA), and a modified quadratic discriminant function (MQDF) classifier [13] with 20 principal eigenvectors per class is used to assign the radical pattern to 10 top-rank radical classes. Radical patterns extracted from training samples were also used for estimating the Gaussian PDFs for geometric context. The parameters in path evaluation were empirically selected as $\lambda_1 = 0.1$, $\lambda_2 = 0.2$ and $\lambda_3 = 0.7$.

We evaluated two methods of radical-based recognition: sequential representation with beam search and hierarchical representation with DP search. The test accuracies of these two methods and the holistic MQDF method are listed in Table 1. We can see that radical-based recognition yields comparable character recognition accuracy to holistic recognition.

The main advantage of radical-based recognition is the smaller number of radical classes (1,149) than the number of character classes (5,773). Radical-based recognition, however, does not save computation time because it involves multiple candidates of radical segmentation. Nested DP with hierarchical representation is even more computationally intensive than beam search with sequential representation (about three times) because the character classes are matched one by one.

Table 1: Test accuracies of radical-based and holistic methods.

Method		#Class	Accuracy
Radical-based	Sequential representation + beam search	1,149	97.22%
	Hierarchical representation + nested DP		97.18%
Holistic		5,773	97.14%

For two radical-based methods, recognition errors mainly come from pre-segmentation error or confusion between similar characters of the same structure type. In pre-segmentation, though we use corner point detection to handle the connection between radicals, errors still remain with skewed characters or heavily overlapping radicals. The confusion between similar characters (Fig. 8) often lies in similar radicals.

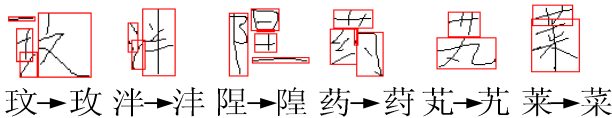


Fig. 8: Examples of confusion between similar characters.

6 CONCLUSION

We proposed a radical-based approach with nested hierarchical representation for online handwritten Chinese character recognition. For characters of left-right and up-down structures, three-layer nested pre-segmentation can ensure a correct segmentation of radicals. For character recognition, we use two representation schemes of character-radical dictionary and accordingly different search algorithms. Our experimental results show that radical-based recognition yields comparable character recognition accuracies with a state-of-the-art holistic

statistical recognition method. Future works will be done to improve pre-segmentation and similar radical discrimination.

ACKNOWLEDGEMENTS

This work was supported in part by Microsoft Research Asia under the program of Mobile Computing in Education and the National Natural Science Foundation of China (NSFC) under grants no.60775004 and no.60825301.

REFERENCES

- [1] C.-L. Liu, S. Jaeger, M. Nakagawa, Online handwritten Chinese character recognition: The state of the art, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(2): 198-213, 2004.
- [2] J. Kwon, B. Sin, J.H. Kim, Recognition of on-line cursive Korean characters combining statistical and structural methods, *Pattern Recognition*, 30(8): 1225-1263, 1997.
- [3] K.-W. Kang, J.H. Kim, Utilization of hierarchical stochastic relationship modeling for Hangul character recognition, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(9): 1185-1196, 2004.
- [4] I. Ota, R. Yamamoto, T. Nishimoto, S. Sagayama, Online handwritten Kanji recognition based on inter-stroke grammar, *Proc. 9th ICDAR*, Curitiba, Brazil, 2007, pp.1188-1192.
- [5] Y.J. Liu, L.Q. Zhang, J.W. Dai, A new approach to on-line handwriting Chinese character recognition, *Proc. 2nd ICDAR*, Tsukuba, Japan, 1993, pp.192-195.
- [6] M. Nakai, N. Akira, H. Shimodaira, S. Sagayama, Substroke approach to HMM-based on-line Kanji handwriting recognition, *Proc. 6th ICDAR*, Seattle, WA, 2001, pp.491-495.
- [7] K. Chellapilla, P. Simard, A new radical based approach to offline handwritten East-Asian character recognition, *Proc. 10th IWFHR*, 2006, La Baule, France, pp.261-266.
- [8] L.-L. Ma, C.-L. Liu, A new radical-based approach to online handwritten Chinese character recognition. *Proc. 19th ICPR*, Tampa, FL, 2008.
- [9] C.-L. Liu, M. Koga, H. Fujisawa, Lexicon-driven segmentation and recognition of handwritten character strings for Japanese address reading, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(11): 1425-1437, 2002.
- [10] X.-D. Zhou, J.-L. Yu, C.-L. Liu, T. Nagasaki, K. Marukawa, Online handwritten Japanese character string recognition incorporating geometric context, *Proc. 9th ICDAR*, Curitiba, Brazil, 2007, pp.48-52.
- [11] C.-L. Liu, X.-D. Zhou, Online Japanese character recognition using trajectory-based normalization and direction feature extraction, *Proc. 10th IWFHR*, La Baule, France, 2006, pp.217-222.
- [12] M. Hamanaka, K. Yamada, J. Tsukumo, On-line Japanese character recognition experiments by an off-line method based on normalized-cooperated feature extraction, *Proc. 2nd ICDAR*, Tsukuba, Japan, 1993, pp.204-207.
- [13] F. Kimura, K. Takashina, S. Tsuruoka, Y. Miyake, Modified quadratic discriminant functions and the application to Chinese character recognition, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 9(1): 149-153, 1987.