High Accuracy Handwritten Chinese Character Recognition by Improved Feature Matching Method

Cheng-Lin Liu, In-Jung Kim, and Jin H. Kim
AI Lab., Department of Computer Science, Korea Advanced Institute of Science and Technology
373-1 Kusong-dong Yusong-gu, Taejon 305-701, South Korea
Fax: +82-42-869-3510, E-mail: {liucl, ijkim}@ai.kaist.ac.kr

Abstract

In this paper, we propose some strategies to improve the recognition performance of feature matching method for handwritten Chinese character recognition (HCCR). Favorable modifications are given to all stages throughout the recognition. In preprocessing, we devised a modified nonlinear normalization algorithm and a connectivity-preserving smoothing algorithm. For feature extraction, an efficient directional decomposition algorithm and a systematic approach to design blurring mask are presented. Finally, a modified LVQ3 algorithm is applied to optimize the reference vectors for classification. The integrated effect of these strategies significantly improves the recognition performance. Recognition results on databases ETL8B2 and ETL9B are promising.

1. Introduction

Handwritten Chinese character recognition (HCCR) has long been regarded extremely difficult in pattern recognition community. The major difficulties of HCCR are: large vocabulary, complicated structure, wide writing variation, and mutual shape similarity between different categories. To solve this problem, various approaches have been proposed. To cope with the large vocabulary, hierarchical classification strategy is adopted. The shape variation is absorbed by efficient feature extraction or description techniques. To discriminate similar characters, specific modules incorporating relevant knowledge maybe built. The complicated structure makes the description difficult. However, it provides rich discriminative information favorable for recognition.

Efficient feature extraction/description and classification are essential to achieve high performance for HCCR. Now that the classification strategy is closely dependent on description scheme, the variety of recognition methods can be categorized in terms of description. So far, two approaches have attracted much attention, which are feature matching and structural matching [1][2]. The feature matching approach improves recognition performance by extracting local structural feature. It can absorb shape variation in certain degree. Feature matching is easy to be implemented and as the matter of fact, has given rise to promising recognition results. Structural matching, on the other hand, is somehow sophisticated yet potentially more flexible to tolerate large variation. Among structural matching methods, relaxation [3] maybe the most attractive one. Structural matching is regarded more consistent to human writing generation and understanding and has indeed produced desirable results. However, the technique is far from maturity and has lots of problems to solve.

The purpose of this paper is to propose a comprehensively improved feature matching method for HCCR. Improvements are given to all stages throughout the recognition, including preprocessing, feature extraction and classification. It is proved by experiments that feature matching can produce very high accuracy under elaborately devised algorithms. Before the detailed description of our method, the development and main ideas of feature matching are briefly outlined.

The success of feature matching is owing to some emergent techniques: feature blurring, directional feature extraction and non-linear normalization [4], each of which produced significant performance gain in HCCR. Blurring technique is efficient to absorb displacement. Directional feature [5][6], as well as other effective features [1], take into account the local structure of neighboring points, thereby improve the discrimination and deformation tolerance. The emergence of non-linear normalization (NLN) is another milestone. This technique is useful to restore shape deformation of handwritten characters. The main idea of NLN is to reallocate the position of strokes in order that they are distributed uniformly. Two NLN

0-8186-7898-4/97 $10.00 © 1997 IEEE

methods, presented by Yamada [7] and Tsukumo [8], respectively, are the representatives of this idea.

Recent advances in feature matching are mainly in improvement of classification, including sophisticated classifier design, prototype optimization for minimum distance classifier [9][10], etc. For the moment, it is hard to propose any new innovative idea to feature matching. However, we may refine each stage of recognition to further improve the performance.

In our recognition system, we adopt the flow of non-linear normalization followed by directional feature extraction and minimum distance classification. To improve the performance, we modify Yamada's NLN method to mitigate excessive peripheral shape distortion and computational complexity. The normalized image is smoothed by a connectivity-preserving smoothing algorithm. In feature extraction stage, we present an efficient directional decomposition algorithm and a systematic approach to deign blurring mask. For classification, we build two hierarchies by using lower resolution feature for coarse classification and higher resolution feature for fine classification. Furthermore, the reference vectors of fine classification are optimized by a modified LVQ3 algorithm. The details of the techniques will be given through section 2 to section 4. Experimental results are given in section 5. The paper is closed by concluding remarks in section 6.

2. Preprocessing

2.1. Nonlinear normalization (NLN)

Lee and Park [11] experimentally compared some NLN methods and revealed that all of them perform fairly well while Yamada's method [7] is a little better in sense of recognition accuracy but more computationally intensive. We adopt this method and give some modifications. Firstly, this method can be implemented more efficiently with moderate computing effort. The computation of line density histogram is accelerated by run-length coding of image. With all black and white runs in horizontal and vertical directions recorded, the line density histogram can be computed from the runs without scanning the image repeatedly.

In NLN algorithms, Tsukumo [8] defined the line density directly to be the reciprocal of line interval (a run in background) in horizontal or vertical direction. When the pixel is in stroke area, the density is set to be a small value. This method is straightforward in implementation but its handling of stroke and peripheral area is rough. In contrast, Yamada [7] computed the line density more carefully where the line interval is defined according to the configuration of nearby stroke edges and the normalized shape is more desirable.

In experiments, we found that in NLN by Yamada's method, peripheral shape in four boundaries are distorted unduly. This is due to the definition of line density in peripheral area. In Yamada's method, the line interval is defined as $L_x = 2W$ (in horizontal direction, W is the character width) in peripheral area. As the reciprocal of L_x, the line density will be very small, thereby the peripheral shape shrink excessively. This effect may degrade some useful features and so that confuse similar characters.

In order to mitigate the excessive peripheral shape distortion, we modify the definition of line interval for peripheral area. Our definition is $L_x = W + D_x$ instead of $L_x = 2W$, where D_x is the distance of the peripheral stroke edge to the boundary of enclosing rectangle. The fact that $W + D_x < 2W$ makes the shape shrinkage milder. Moreover, the effect of D_x will compress shorter protrusion less than longer one. It is noteworthy that the modified definition is hybrid of Tsukumo's and Yamada's method.

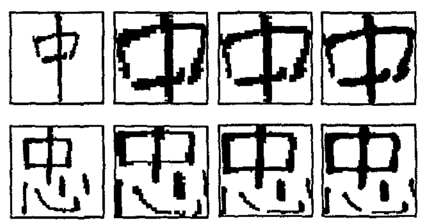

Fig. 1. Non-linear normalization and smoothing

Fig. 1 shows some examples of NLN. In each row, the leftmost image is the original image, the 2nd one is normalized by Yamada's method, and the 3rd one is normalized by the modified NLN method. It is revealed that the modified algorithm do mitigate the peripheral shape distortion and meanwhile preserve the power of shape restoration.

2.2. Connectivity-preserving smoothing

Since normalization may produce staircases, the smoothing after normalization is necessary for recognition based on contour feature. In some normalized characters, there are thin strokes which may be broken or eliminated by smoothing. To preserve thin strokes, one of the authors proposed a connectivity-preserving smoothing algorithm [12], which has been successfully used in handwritten numeral recognition. The basic idea is as follows. For each pixel p, the black number $b(p)$ and 8-connectivity number $N_c(p)$ in 3×3 neighborhood are computed. $N_c(p)$ is related to the connectivity of the pixel, if $N_c(p)$ is more than or equal to 2, changing the value of p will destroy the connectivity. The rules to decide whether to retain or change a pixel are designed so as to preserve the connectivity. The procedure is similar to an iteration in pixel-based thinning algorithm. In Fig. 1, the rightmost image is the result of connectivity-preserving smoothing.

3. Feature extraction

3.1. Directional decomposition

The feature extracted for HCCR is the local stroke direction distribution. Generally, each contour pixel is assigned a direction code and direction histogram in local area is counted. The assignment of direction is effectively a directional decomposition of image. The direction of a pixel can be determined by various means, like template matching, gradient, and contour tracing. Contour tracing is straightforward to get direction code but unnecessarily troublesome. However, it is advantageous in sense that multiply connected pixels can be exploited multiple times. To exploit both two sides of contour while avoid tracing, reference [12] presented a directional decomposition technique, which produces same effect as contour tracing but be accomplished by only one raster scanning. The algorithm is outlined in the following.

The contour points of normalized character image are assigned into four subimages corresponding to four directions (horizontal, vertical and two diagonal directions). Initially, the four directional subimages $f_d(x,y)$, d=0,1,2,3 are all empty, i.e., $f_d(x,y) = 0$. Then for each black pixel in character image $f(x,y)$, denote its 8-neighbours by x_i, $i = 0, 1, \ldots, 7$ in clockwise order with the east neighbor as x_0. If $f(x,y) = 1$, then for i=0,1,2,3,

if $x_{2i} = 0$

  then if $x_{2i+1} = 1$

    then $d = (2i+1) mod\, 8$, $f_d(x,y) = f_d(x,y) + 1$

  else if $x_{2i+2} = 1$

    then $d = (2i+2) mod\, 8$, $f_d(x,y) = f_d(x,y) + 1$

  end if

end if

By this procedure, the resulted subimages $f_d(x,y)$, d=0,1,2,3 are the directional decomposition of $f(x,y)$.

3.2. Feature extraction

Features are extracted from four directional subimages and constitute a feature vector. Features from a subimage are extracted by counting pixel values in local fields. To do this, the image frame can be partitioned into rectangular grids regularly or dynamically. The dynamic partitioning [5] was said to be robust against displacement. But in our experience, blurring was proved to be more efficient, hence it is adopted in this paper.

The feature for coarse classification is extracted by partitioning image frame into 4×4 grids in each directional subimage without blurring. The feature vector extracted has low dimension (4×4 features in 4 directions, totally $4 \times 4 \times 4 = 64$D), which is beneficial to fast search of candidates in large vocabulary.

To extract feature for fine classification, firstly 16×16 values are counted in 16×16 grids in each subimage. The 16×16 values are viewed as another coarse image and blurred into 8×8 values by Gaussian blurring mask. For Gaussian mask, to determine the parameter of standard deviation σ is not trivial. Instead of trial and error, we apply signal processing principle to do it systematically. In [12], it was shown that the parameter is related to the sampling rate and is determined by sampling theorem when the blurring is viewed as low-pass filtering. The blurring mask is a Gaussian filter, whose impulse response function is given by

$$h(x,y) = \frac{1}{2\pi\sigma_x^2} exp(-\frac{x^2+y^2}{2\sigma_x^2}) \tag{1}$$

By analyzing the power spectrum of this filter and preserving majority of energy in filtering when truncating the band-width, the optimal value of parameter is given by

$$\sigma_x = \frac{\sqrt{2}t_x}{\pi} \tag{2}$$

where t_x is the sampling interval. When we extract 8×8 blurred features from 16×16 values, the sampling interval is $t_x = 2$. We approximate the filter into a 4×4 blurring mask. Substituting equation (2) into equation (1) and set $x, y \in \{-1.5, -0.5, 0.5, 1.5\}$, we obtain the mask as in Fig. 2. From each directional subimage, the 8×8 features are extracted by convoluting the 16×16 values with 8×8 shifted masks. The resulted feature vector has dimension $8 \times 8 \times 4 = 256$D. This is used in fine classification.

4. Classifier design

For recognition of large character set, the simplest classifier, minimum distance classifier (MDC) is frequently used to reduce both dictionary storage and training effort. To improve the recognition accuracy

0.0144	0.0456	0.0456	0.0144
0.0456	0.1444	0.1444	0.0456
0.0456	0.1444	0.1444	0.0456
0.0144	0.0456	0.0456	0.0144

Fig. 2. Blurring mask for feature extraction

of MDC, learning vector quantization (LVQ) is used to optimize the reference vectors. Different versions of LVQ are reviewed by Kohonen [13]. LVQ3, the most efficient one, is outlined in the following.

Given learning vectors with class label, initial code vectors for each class are selected from learning vectors or built by clustering. Then, the code vectors are adjusted by iterative comparing learning vectors with code vectors and adjusting code vectors. At time t, input learning vector $\mathbf{x}(t)$, match it with all code vectors, search the two closest code vectors, and adjust according to the rules:

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) - \alpha(t)[\mathbf{x}(t) - \mathbf{m}_i(t)] \qquad (3)$$

$$\mathbf{m}_j(t+1) = \mathbf{m}_j(t) + \alpha(t)[\mathbf{x}(t) - \mathbf{m}_j(t)] \qquad (4)$$

where $\mathbf{m}_i$ and $\mathbf{m}_j$ are the two closest code vectors to $\mathbf{x}$ with matching distance d_i and d_j respectively, $\mathbf{x}$ and $\mathbf{m}_j$ belong to the same class, while $\mathbf{x}$ and $\mathbf{m}_i$ belong to different classes; furthermore, $\mathbf{x}$ must fall into the window:

$$min(\frac{d_i}{d_j}, \frac{d_j}{d_i}) > s \qquad s \in (0,1)$$

And if $\mathbf{x}$, $\mathbf{m}_i$ and $\mathbf{m}_j$ belong to the same class,

$$\mathbf{m}_k(t+1) = \mathbf{m}_k(t) + \epsilon\alpha(t)[\mathbf{x}(t) - \mathbf{m}_k(t)]$$

for $k \in i, j$, $\epsilon \in [0.1, 0.5]$, being smaller for narrower windows.

LVQ3 is insufficient that it does not treat the case that both $\mathbf{m}_i$ and $\mathbf{m}_j$ are in different classes as $\mathbf{x}$. Lee & Song [9] improved LVQ3 by drawing code vectors apart when both $\mathbf{m}_i$ and $\mathbf{m}_j$ belong to different classes as $\mathbf{x}$. In out system of HCCR, we built only one reference for each class. The reference is initialized by averaging all training vectors of same class. Because the reference is initially the center of a class, it is not adjusted when both the two closest code vectors belong to the same class as the input. The modified LVQ3 algorithm is summarized as follows:

1) if the closest code vector $\mathbf{m}_i$ belongs to different class as $\mathbf{x}$, draw $\mathbf{m}_i$ apart from $\mathbf{x}$ as equation (3) regardless of the second nearest neighbor.
2) Under the first condition, if the second nearest neighbor $\mathbf{m}_j$ belongs to the same class as $\mathbf{x}$, draw $\mathbf{m}_j$ nearer to $\mathbf{x}$ as equation (4); otherwise, draw $\mathbf{m}_j$ apart from $\mathbf{x}$.
3) if the closest code vector $\mathbf{m}_i$ belongs to same class as $\mathbf{x}$, yet the second closest code vector $\mathbf{m}_j$ belongs to different class, and they fall into the window, then draw $\mathbf{m}_i$ nearer and $\mathbf{m}_j$ apart.

The modified algorithm is a little different from that of [9]. By the modified LVQ3 with references initialized by averaging, we built one reference vector of 256D for each class for fine classification. The references for coarse classification are built simply by averaging.

5. Experimental results

The proposed recognition method was experimented on large vocabulary databases ETL8B2 and ETL9B. In ETL8B2 there has 881 classes of Chinese characters, each with 160 variations, whereas in ETL9B there has 2965 classes of Chinese characters, each with 200 variations. In either of the databases, a group of characters with one sample from each class is referred to as a dataset or simply, a set. In accordance with previous works, from ETL8B2, 80 sets of odd number order are used in training and that of even number order are used in recognition test. From ETL9B, 100 sets of odd number order are used in training and 100 sets with even number order are used in test.

The classifier has two hierarchies, the accuracy of coarse classification is 93.20% on ETL8B2 and 89.07% on ETL9B. The accumulative accuracy of 20 candidates in ETL8B2 is 99.63% and that of 50 candidates in ETL9B is 99.26%. The overall recognition rates on test data of ETL8B2 and ETL9B are given in Table 1 and Table 2 respectively, where both the classification with and without LVQ are given, also included some previous results for comparison.

Table 1. Recognition rates on test data of ETL8B2 (%)

Method	80 sets	set no.2	set no.34
Averaging	96.68	99.21	92.28
MLVQ3	97.84	99.66	93.53
Yamada90		99.1	91.6
Yamamoto92		99.0	93.4

When the references for fine classification are designed simply by averaging, the recognition rate is 96.68% for ETL8B2 and 93.87% for ETL9B. The application of modified LVQ3 does significantly improve the

Table 2. Recognition rates on test data of ETL9B (%)

Method	Averaging	MLVQ3	Tsukumo92	Kazuki96
100 sets	93.87	95.73	95.05	95.84

recognition rate, from 96.68% to 97.84% for ETL8B2 and from 93.87% to 95.73% for ETL9B.

Even though we didn't recognize Japanese characters (hiragana and katagana) in the databases, our result is comparable to previous results, because Japanese characters are very few and their shape is much different from Chinese characters. We cite the results of Yamada [7] and Yamamoto [14]. They only gave recognition rates on datasets No.2 and No.34 of ETL8B2, which are typical of high and low quality data. Their results is comparable to our result of averaging, but when we optimize the references by LVQ, our result is much better. Considering that the feature dimension of Yamada is $16 \times 16 \times 4 = 1024$D and Yamamoto applied relaxation in recognition, our method is much more efficient in computation.

Our result on ETL9B is compared to Tsukumo [15] and Kazuki [16]. Tsukumo [15] achieved high recognition rate of 95.05% on 100 test sets of ETL9B by nonlinear normalization and flexible pattern matching, which is comparable to our result but the recognition method is more complicated. Recently, Kazuki et al [16] achieved accuracy of 95.84% by combined neural network on directional feature. Their result is a little better than ours, but they store the template of each class in a multi-layer neural network, which consumes much more storage space. In our method, each class has only one template and the distance measure is unweighted. This scheme is much more economical in storage and classification than that of Kazuki [16] as well as [10].

6. Conclusion

In this paper, we give favorable modifications to all stages throughout the recognition of feature matching for HCCR, and the integrated effect of these modifications significantly improves the overall recognition performance. We tested the recognition method on large vocabulary databases and compared our result with that of previous works. Our result is much better or comparable to the previous results. Because our method is implemented under much smaller complexity in computation and storage, it is safe to say that our method is valuable.

References

[1] S.Mori, K.Yamamoto, M.Yasuda, Research on machine recognition of handprinted characters, *IEEE Trans. PAMI*, 6(4), 386-405, 1984.

[2] T.H.Hilderbrand, W.Liu, Optical recognition of Chinese characters: advances since 1980, *Pattern Recognition*, 26(2), 205-225, 1993.

[3] K.Yamamoto, A.Rosenfeld, Recognition of handprinted Kanji characters by a relaxation method, *Proc. 6th ICPR*, 395-398, 1982.

[4] M.Umeda, Advances in recognition methods for handwritten Kanji characters, *IEICE Trans. Inf. & Syst.*, 79-D(5), 401-410, 1996.

[5] Y.Yamashita, et al, Classification of handprinted Kanji characters by the structured segment matching method, *Pattern Recognition Letters*, 1, 475-479, 1983.

[6] F.Kimura, et al, Modified quadratic discriminant functions and the application to Chinese character recognition, *IEEE Trans. PAMI*, 9(1), 149-153, 1987.

[7] H.Yamada, K.Yamamoto, T.Saito, A nonlinear normalization method for handprinted Kanji character recognition – line density equalization, *Pattern Recognition*, 23(9), 1023-1029, 1990.

[8] J.Tsukumo, H.Tanaka, Classification of handprinted Chinese characters using nonlinear normalization and correlation methods, *Proc. 9th ICPR*, 168-171, 1988.

[9] S.-W.Lee, H.-H.Song, Optimal design of reference models using simulated annealing combined with improved LVQ3, *Proc. 2nd ICDAR*, 244-249, 1993.

[10] Y.S.Huang, et al, A novel neural net-based system and process for constructing optimized prototypes for character recognition, *Pre-Proc. IWFHR5*, Colchester, England, 103-108, 1996.

[11] S.-W.Lee, J.-S.Park, Nonlinear shape normalization methods for the recognition of large-set handwritten characters, *Pattern Recognition*, 27(7), 895-902, 1994.

[12] C.-L.Liu, Y.-J.Liu, R.-W.Dai, Multiresolution statistical and structural feature extraction for handwritten numeral recognition, *Pre-Proc. IWFHR5*, Colchester, England, 61-66, 1996.

[13] T.Kohonen, The self-organizing map, *Proc. IEEE*, 78(9), 1464-1480, 1990.

[14] K.Yamamoto, H.Yamada, Recognition of handprinted Chinese characters and the Japanese cursive syllabary, in *Structural Document Image Analysis*, Springer-Verlag, pp.250-260, 1992.

[15] J.Tsukumo, Handprinted Kanji character recognition based on flexible template matching, *Proc. 11th ICPR*, II, 483-486, 1992.

[16] S.Kazuki, et al, High accuracy recognition of ETL9B using exclusive learning neural network-II (ELNET-II), *IEICE Trans. Inf. & Syst.*, 79-D(5), 516-521, 1996.