A Study On the Use of 8-Directional Features For Online Handwritten Chinese Character Recognition *

Zhen-Long BAI and Qiang HUO
Department of Computer Science, The University of Hong Kong, Hong Kong, China
(Email: zlbai@cs.hku.hk, qhuo@cs.hku.hk)

Abstract

This paper presents a study of using 8-directional features for online handwritten Chinese character recognition. Given an online handwritten character sample, a series of processing steps, including linear size normalization, adding imaginary strokes, nonlinear shape normalization, equidistance resampling, and smoothing, are performed to derive a 64×64 *normalized online character sample. Then, 8-directional features are extracted from each online trajectory point, and 8 directional pattern images are generated accordingly, from which blurred directional features are extracted at* 8×8 *uniformly sampled locations using a filter derived from the Gaussian envelope of a Gabor filter. Finally, a 512-dimensional vector of raw features is formed. Extensive experiments on the task of recognizing 3755 level-1 Chinese characters in GB2312-80 standard are performed to compare and discern the best setting for several algorithmic choices and control parameters. The effectiveness of the studied approach is confirmed.*

1. Introduction

In this paper, we study the problem of how to extract so-called directional features from an online handwritten character sample to form a vector of raw features that can be used to construct a classifier for online Chinese character recognition (OLCCR) using any promising statistical pattern recognition approach. Previous works on this topic as reported in e.g., [4, 1, 5, 6], have demonstrated successfully the effectiveness of the 4-directional features, where 4 directions are defined naturally as vertical (|), horizontal (—), left-up (/) and right-down (\), as shown in Fig. 1(a). The primary motivations of this study are to extend the previous works to extracting 8-directional features with 8 directions as shown in Fig. 1(b), and to study their effectiveness for OLCCR.

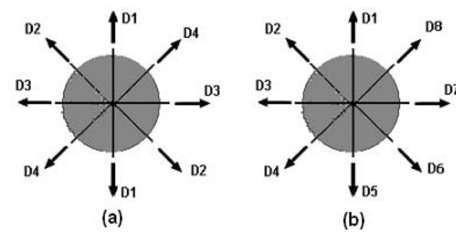

Figure 1. A notion of 4 vs. 8 directions.

It is noted that different ways of extracting directional features were used in the previous works. For example, in [4, 5], 4-directional features were extracted directly from the nonlinear shape normalized (NSN) online trajectory; while in [1, 6], 4-directional features were extracted from a bitmap using an "offline" approach. Furthermore, different ways of projecting a *direction vector* to relevant *directional axes*, and different strategies for grid-based feature blurring were used in [4] and [5], respectively. In addition to the above, there are actually other algorithmic choices in deriving directional features from an online handwritten character sample. We have conducted extensive experiments to study the behavior and performance implications of different choices with a hope of identifying the most promising scheme and discerning the best setting for the relevant control parameters. In this paper, we report those most important findings and recommend a promising approach of extracting 8-directional features for OLCCR.

The rest of the paper is organized as follows. Details of recommended approach are described in Section 2. Experimental results of comparative studies are reported in Section 3. Finally, our findings are summarized in Section 4.

2. Our Approach

The overall flowchart of our recommended approach is shown in Fig. 2. In the following subsections, we explain in detail how each module works.

*This work was supported by a grant from the RGC of the Hong Kong SAR (Project No. HKU7145/03E).

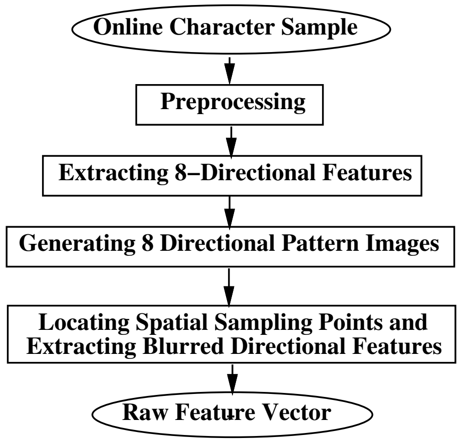

Figure 2. Overall flowchart.

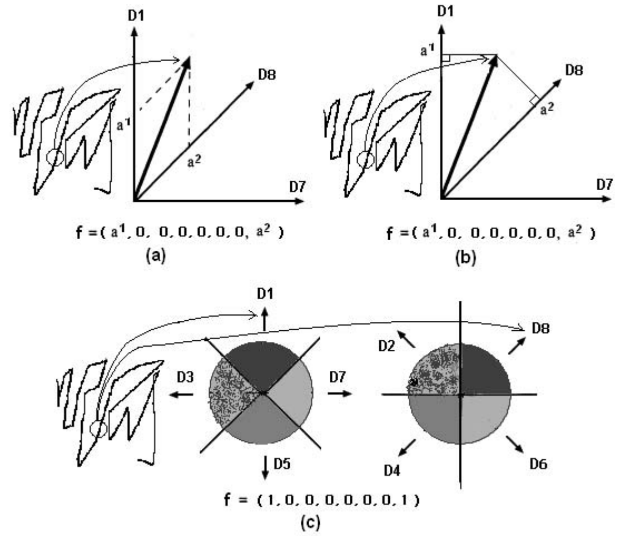

Figure 3. Different ways of projecting a direction vector onto directional axes and the corresponding directional "feature values": (a) adapted from [4]; (b) adapted from [5]; (c) our proposal.

2.1. Preprocessing

The main objective of a series of preprocessing steps is to remove certain variations among character samples of the same class that would otherwise reduce recognition accuracy. This module includes the following steps:

(1) *Linear size normalization*: Given a character sample, it is normalized to a fixed size of 64×64 using an aspect-ratio preserving linear mapping.

(2) *Adding imaginary strokes*: Imaginary strokes are those pen moving trajectories in pen-up states that are not recorded in the original character sample. We define an imaginary stroke as a straight line from the end point of a pen-down stroke to the start point of its next pen-down stroke. All such constructed imaginary strokes are added into the stroke set of a character sample.

(3) *Nonlinear shape normalization (NSN)*: NSN is used to normalize shape variability. The online character sample after the above two steps is first transformed into a bitmap that is then normalized by a *dot density equalization* approach reported originally in [7]. Using the derived NSN warping functions, the online character sample after the above step (2) is transformed into a new sample such that the temporal order of the original points is maintained.

(4) *Re-sampling*: Re-sampling is purposed to reduce distance variation between two adjacent online points and the variance of number of points in a stroke. The sequence of online points in each stroke (including all imaginary strokes) of a character is re-sampled by a sequence of equidistance points (a distance of 1 unit length is used in our approach).

(5) *Smoothing*: Smoothing can reduce stroke shape variation in a small local region. In a stroke, besides the start point and end point, we replace the position of every other point by the mean value of that of its 2 neighbors and itself.

2.2. Extracting 8-Directional Features

Given a stroke point P_j, its *direction vector* $\overrightarrow{V_j}$ is defined as follows:

$$\overrightarrow{V_j} = \begin{cases} \overrightarrow{P_j P_{j+1}} & \text{if } P_j \text{ is a start point} \\ \overrightarrow{P_{j-1} P_{j+1}} & \text{if } P_j \text{ is a non-end point} \\ \overrightarrow{P_{j-1} P_j} & \text{if } P_j \text{ is an end point} \end{cases} \qquad (1)$$

For a non-end point P_j, if its two neighbors P_{j-1} and P_{j+1} are in the same position, the point P_j is ignored and no directional features are extracted at this point.

Given $\overrightarrow{V_j}$, its normalized version, $\overrightarrow{V_j}/||\overrightarrow{V_j}||$, can be projected onto two of the 8 directional axes, as shown in Fig. 3, one is from the direction set of $\{D_1, D_3, D_5, D_7\}$ and denoted as d_j^1, and the other is from the set of $\{D_2, D_4, D_6, D_8\}$ and denoted as d_j^2. If we define the coordinate system for the original online point $P_j = (x_j, y_j)$ as follows: the x-axis is from left to right, and the y-axis is from top down; then d_j^1 and d_j^2 for a non-end point P_j can

Proceedings of the 2005 Eight International Conference on Document Analysis and Recognition (ICDAR'05)
1520-5263/05 $20.00 © 2005 IEEE

be identified as follows:

$$d_j^1 = \begin{cases} D_7 & \text{if } x_{j-1} \le x_{j+1} \;\&\; |y_{j+1} - y_{j-1}| \le |x_{j+1} - x_{j-1}| \\ D_3 & \text{if } x_{j-1} > x_{j+1} \;\&\; |y_{j+1} - y_{j-1}| \le |x_{j+1} - x_{j-1}| \\ D_5 & \text{if } y_{j-1} \le y_{j+1} \;\&\; |y_{j+1} - y_{j-1}| > |x_{j+1} - x_{j-1}| \\ D_1 & \text{if } y_{j-1} > y_{j+1} \;\&\; |y_{j+1} - y_{j-1}| > |x_{j+1} - x_{j-1}| \end{cases}$$

$$d_j^2 = \begin{cases} D_6 & \text{if } x_{j-1} \le x_{j+1} \;\&\; y_{j-1} \le y_{j+1} \\ D_8 & \text{if } x_{j-1} \le x_{j+1} \;\&\; y_{j-1} > y_{j+1} \\ D_2 & \text{if } x_{j-1} > x_{j+1} \;\&\; y_{j-1} > y_{j+1} \\ D_4 & \text{if } x_{j-1} > x_{j+1} \;\&\; y_{j-1} \le y_{j+1} \end{cases}$$

For the example shown in Fig. 3, these two directions are $d_j^1 = D_1$, $d_j^2 = D_8$ for the highlighted direction vector. Given the above identified directions, an 8-dimensional feature vector can be formed with non-zero *directional feature values* a_j^1 and a_j^2 corresponding to d_j^1 and d_j^2 respectively. Feature values corresponding to other 6 directions are set as 0s. Using the same example, such a feature vector is $(a_j^1, 0, 0, 0, 0, 0, 0, a_j^2)^t$. Apparently, there are different ways to calculate a_j^1 and a_j^2. We have studied three methods as described in the following.

The first method ("Method-1") is adapted from [4] and shown in Fig. 3(a). For a non-end online point P_j, a_j^1 and a_j^2 are calculated as follows:

$$a_j^1 = \frac{|d_x - d_y|}{s} \quad (2)$$

$$a_j^2 = \frac{\sqrt{2} \cdot \min(d_x, d_y)}{s} \quad (3)$$

where $d_x = |x_{j+1} - x_{j-1}|$, $d_y = |y_{j+1} - y_{j-1}|$ and $s = \sqrt{d_x^2 + d_y^2}$. The second method ("Method-2") is adapted from [5] and shown in Fig. 3(b). For a non-end online point P_j, a_j^1 and a_j^2 are calculated as follows:

$$a_j^1 = \frac{\max(d_x, d_y)}{s} \quad (4)$$

$$a_j^2 = \frac{\sqrt{2}}{2} \cdot \frac{(d_x + d_y)}{s} \,. \quad (5)$$

The third method ("Method-3") is proposed here and shown in Fig. 3(c). We just simply set $a_j^1 = 1$, $a_j^2 = 1$.

If P_j is an end point of a stroke, we replace (x_{j-1}, y_{j-1}) with (x_j, y_j) for the start point, and replace (x_{j+1}, y_{j+1}) with (x_j, y_j) for the end point in the above discussions.

2.3. Generating 8 Directional Pattern Images

After extracting 8-directional features from all online points of a character, 8 *directional pattern images* $\{B_d = [f_d(x, y)],\ x, y = 1, 2, \cdots, 64;\ d = D_1, D_2, \cdots, D_8\}$ can be generated as follows: set $f_{d_j^1}(x_j, y_j) = a_j^1$ and $f_{d_j^2}(x_j, y_j) = a_j^2$; and set the values for all the other $f_d(x, y)$'s as 0s. For each of the above directional pattern images, the following *thickening* processing is further performed: for each non-zero pixel $f_d(x, y) = a$, the value of each of its 8 neighboring pixels is set as $f(x+m, y+n) = max\{f(x+m, y+n), a\}$, where $m = -1, 0, 1$ and $n = -1, 0, 1$.

2.4. Locating Spatial Sampling Points and Extracting Blurred Directional Features

Each directional pattern image is divided uniformly into 8×8 grids whose centers are treated as locations of 8×8 spatial sampling points. At each sampling point (x_i, y_j), a blurred directional feature is extracted as follows:

$$F_d(x_i, y_j) = \sum_{x=-N}^{x=N} \sum_{y=-N}^{y=N} f_d(x_i + x, y_j + y) G(x, y) \quad (6)$$

where $G(x, y)$ is a Gaussian filter whose size is determined by a parameter N. Based on our experience in using Gabor features for Chinese OCR and offline Chinese character recognition (e.g. [2, 3]), we decided to use the following Gaussian envelop derived from the Gabor filter to serve as the Gaussian filter $G(x, y)$:

$$G(x, y) = \frac{\kappa^2}{\sigma^2} exp[-\frac{\kappa^2(x^2 + y^2)}{2\sigma^2}] = \frac{4}{\lambda^2} exp[-\frac{2(x^2 + y^2)}{\lambda^2}] \quad (7)$$

where $\sigma = \pi$, $\kappa = \frac{2\pi}{\lambda}$, and λ is the wavelength of the plane wave of the original Gabor filter. According to our past experience (e.g. [2, 3]), for an image with a size of 64×64, a spatial sampling resolution of 8×8 and a wavelength $\lambda = 8$ should offer a "good" setting for these two control parameters. Interestingly, this is also confirmed to be a "good setting" for OLCCR in our experiments in this project. Given the wavelength, the control parameter N is set as $N = 2\lambda$. It is noted that the optimal setting of the above control parameters will be different for images with different sizes.

Because there are totally 8 directional pattern images, each with 8×8 sampling points, an $8 \times 8 \times 8 = 512$ dimensional vector of raw features can finally be formed by using the nonlinear transformed features, $\{\sqrt{F_d(x_i, y_j)}, d = D_1, D_2, \cdots, D_8; i, j = 1, 2, \cdots, 8\}$.

3. Experiments and Results

3.1. Experimental Setup

In order to evaluate the efficacy of the above techniques, a series of experiments are conducted on the task of the recognition of isolated online handwritten Chinese characters. A subset of the corpus of Chinese handwritings developed in our lab was used. The data set contains 300 writers'

Table 1. A comparison of character recognition accuracies (in %) of using 4 vs. 8 directional features under the condition of using vs. not using imaginary strokes (no thickening operation, no nonlinear feature transformation).

"Top-N" Recognition Results	Single-Prototype Classifier				1-NN Classifier			
	No imaginary strokes		With imaginary strokes		No imaginary strokes		With imaginary strokes	
	4-direction	8-direction	4-direction	8-direction	4-direction	8 direction	4-direction	8-direction
N=1	71.48	80.43	74.84	84.57	74.04	79.19	77.26	83.86
N=5	87.84	93.20	89.40	94.92	93.86	95.70	94.54	96.82
N=10	91.56	95.60	92.50	96.62	97.04	97.95	97.26	98.44
N=50	96.72	98.44	96.85	98.69	99.72	99.80	99.69	99.81

Table 2. A comparison of character recognition accuracies (in %) of using three different direction vector projection methods (with thickening operation, no nonlinear feature transformation).

"Top-N"	Single-Prototype Classifier			1-NN Classifier		
Recognition Results	Method-1	Method-2	Method-3	Method-1	Method-2	Method-3
N=1	85.55	83.54	83.26	85.02	84.59	84.33
N=5	95.43	94.62	94.49	97.17	97.07	96.99
N=10	97.00	96.48	96.41	98.62	98.57	98.51
N=50	98.89	98.77	98.73	99.83	99.82	99.82

samples of 3755 categories of level 1 (GB1) Chinese characters in GB2312-80 standard. Each writer contributed one sample for each character category. To collect the handwriting samples, each writer was asked to write naturally, using a stylus pen, a set of Chinese characters on the touch screen of a PDA or a Pocket PC. No other restriction is imposed on the writing styles. For each character class, we use 200 samples randomly selected from the above data set for training and the remaining 100 samples for testing. The following two simple character classifiers are used for performance evaluation on the testing set:

- A maximum discriminant function based classifier with a single prototype. The prototype is the mean of training feature vectors, and the discriminant function is the negative Euclidean distance between a testing feature vector and the prototype feature vector;
- A 1-NN classifier with all the training feature vectors as prototypes and the Euclidean distance as a dissimilarity measure.

3.2. A Comparison of 4 vs. 8 Directional Features

The first set of experiments are designed to compare the performance of using 4 or 8 directional features under two different choices of adding and not adding imaginary strokes in the preprocessing stage. The directional feature values are extracted using Eqs. (2) and (3). No thickening processing is performed in generating directional pattern images. Nonlinear transformation is not applied when 512-dimensional feature vector is formed. The "Top-N" character recognition accuracies (in %) are summarized in Table 1. From the results, two conclusions can be drawn: 1) Using 8-directional features achieves a much better performance than using 4-directional features; 2) Adding imaginary strokes gives a better performance than not adding imaginary strokes. Therefore, in the later experiments, we always use 8-directional features and add imaginary strokes in the preprocessing steps.

3.3. A Comparison of Different Projection Methods

The second set of experiments are designed to compare the performance of using three different direction vector projection methods for extracting directional features at each on-line trajectory point. We refer to them as Method-1, Method-2, and Method-3, respectively according to the descriptions in subsection 2.2. Thickening processing is performed in generating directional pattern images. Nonlinear transformation is not applied when 512-dimensional feature vector is formed. The "Top-N" character recognition accuracies (in %) are summarized in Table 2.

By comparing the results of two columns labeled as "8-direction" and "With imaginary strokes" in Table 1 with the ones of corresponding columns labeled as "Method-1" in Table 2, it is observed that the thickening operation offers

Table 3. A comparison of character recognition accuracies (in %) of using vs. not using the nonlinear feature transformation.

"Top-N" Recognition Results	Single-Prototype Classifier		1-NN Classifier	
	No	Yes	No	Yes
N=1	85.55	88.19	85.02	88.35
N=5	95.43	96.64	97.17	98.24
N=10	97.00	97.91	98.62	99.16
N=50	98.89	99.33	99.83	99.91

useful improvement of the performance. By comparing the results of using different projection methods in Table 2, it is observed that "Method-1" performs better than "Method-2" and "Method-3". Therefore, both thickening operation and "Method-1" are used in the later experiments.

3.4. The Effect of Using Nonlinear Feature Transformation

The third set of experiments are designed to compare the performance of using and not using the nonlinear feature transformation when 512-dimensional feature vector is formed, and the results are summarized in Table 3. A significant performance improvement is achieved by using the nonlinear feature transformation. In addition to the *square root function*, the following convex nonlinear transformation functions are also tried:

$$\begin{aligned} f(x) &= x^{\alpha},\ 0 < \alpha < 1; \\ f(x) &= \sin(\alpha x),\ 0 < \alpha x < \frac{\pi}{2}; \\ f(x) &= \tanh(x); \\ f(x) &= \log(1+x); \\ f(x) &= \frac{1-e^{-x}}{1+e^{-x}}\ . \end{aligned}$$

In a preliminary study of using a reduced vocabulary with 100 GB1 Chinese characters, it was observed that all of the above nonlinear transformations was useful in improving the performance, yet the *square root function* gave the best performance.

4. Summary

In this paper, we have presented a study on how to extract 8-directional features for OLCCR. A promising approach has been identified after a careful study of the effects of the following algorithmic choices on the recognition performance: 1) to use 4 or 8 directional features; 2) whether to add imaginary strokes; 3) the effects of different methods for direction vector projection; 4) whether to use a thickening operation in generating directional pattern images; 5) whether to apply a nonlinear transformation in deriving the final features. From the above results, the following conclusions can be drawn: 1) Among all the strategies we tried, using 8-directional features gives the greatest performance improvement in comparison with that of using 4-directional features; 2) adding imaginary strokes is useful in upgrading the performance; 3) thickening operation is helpful in improving the performance; 4) the first method for direction vector projection performs better than the other two methods; 5) the *square root function* is a good choice for nonlinear feature transformation that offers a significant performance improvement. As a future work, we will use the raw features identified in this study to construct a classifier for OLCCR by using the proven technology as described in [3]. It would be interesting to see what kind of performance could be achieved by combining these state-of-the-art technologies.

References

[1] M. Hamanaka, K. Yamada, and J. Tsukumo, "On-Line Japanese Character Recognition Experiments by an Off-Line Method Based on Normalization-Cooperated Feature Extraction," in *Proc. 3rd Int'l Conf. Document Analysis and Recognition*, 1993, pp.204-207.

[2] Q. Huo, Z.-D. Feng, and Y. Ge, "A Study on the Use of Gabor Features for Chinese OCR," in *Proc. ISIMP-2001* (2001 International Symposium on Intelligent Multimedia, Video & Speech Processing), 2001, pp.389-392.

[3] Q. Huo, Y. Ge, and Z.-D. Feng, "High Performance Chinese OCR based on Gabor Features, Discriminative Feature Extraction and Model Training," in *Proc. ICASSP-2001*, May 2001, pp.III-1517-1520.

[4] A. Kawamura *et al.*, "On-Line Recognition of Freely Handwritten Japanese Characters Using Directional Feature Densities," in *Proc.11th Int'l Conf. Pattern Recognition*, 1992, Vol. 2, pp.183-186.

[5] M. Nakagawa *et al.*, "Robust and Highly Customizable Recognition of On-Line Handwritten Japanese Characters," in *Proc. 13th Int'l Conf. Pattern Recognition*, 1996, Vol.3, pp.269-273.

[6] M. Okamoto and K. Yamamoto, "On-line Handwriting Character Recognition Using Direction-change Features That Consider Imaginary Strokes," *Pattern Recognition*, Vol.32, pp.1115-1128, 1999.

[7] Y. Yamashita *et al.*, "Classification of Handprinted Kanji Characters by the Structured Segment Matching Method," *Pattern Recognition Letters*, Vol.1, pp.475-479, 1983.