



**HAL**  
open science

# Online Japanese Character Recognition Using Trajectory-Based Normalization and Direction Feature Extraction

Cheng-Lin Liu, Xiang-Dong Zhou

► **To cite this version:**

Cheng-Lin Liu, Xiang-Dong Zhou. Online Japanese Character Recognition Using Trajectory-Based Normalization and Direction Feature Extraction. Tenth International Workshop on Frontiers in Handwriting Recognition, Université de Rennes 1, Oct 2006, La Baule (France), France. inria-00108303

**HAL Id: inria-00108303**

**<https://inria.hal.science/inria-00108303>**

Submitted on 20 Oct 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Online Japanese Character Recognition Using Trajectory-Based Normalization and Direction Feature Extraction

Cheng-Lin Liu and Xiang-Dong Zhou  
National Laboratory of Pattern Recognition (NLPR)  
Institute of Automation, Chinese Academy of Sciences  
PO Box 2728, Beijing 100080, P.R. China  
Email: liucl@nlpr.ia.ac.cn

## Abstract

*This paper describes an online Japanese character recognition system using advanced techniques of pattern normalization and direction feature extraction. The normalization of point coordinates and the decomposition of direction elements are directly performed on online trajectory, and therefore, are computationally efficient. We compare one-dimensional and pseudo two-dimensional (pseudo 2D) normalization methods, as well as direction features from original pattern and from normalized pattern. In experiments on the TUAT HANDS databases, the pseudo 2D normalization methods yielded superior performance, while direction features from original pattern and from normalized pattern made little difference.*

*Keywords: Online character recognition; Trajectory-based normalization; Pseudo 2D normalization; Direction feature extraction.*

## 1. Introduction

Computer recognition of pen-input (online) handwritten characters is involved in various applications, like text editing, online form filling, note taking, pen interface, and so on. A great deal of research works have contributed to online character recognition since the 1960s and many effective methods have been proposed [1]. The methods include dynamic time warping, stroke segment matching, attributed graph matching, statistical feature matching, hidden Markov models (HMMs), artificial neural networks (ANNs), etc.

This paper describes an online character recognition system for handwritten Japanese characters and reports our results using trajectory-based normalization and direction feature extraction methods. As for all handwriting recognition problems, handwritten Japanese character recognition is difficult due to the wide variability of writing styles and the confusion between similar characters. Also, for Chinese and Japanese characters, the large number of characters (classes) poses a challenge in efficient classification. The methods of online Chinese/Japanese

character recognition can be roughly divided into statistical methods and structural methods [2]. Whereas structural matching is more relevant to human learning and perception, statistical methods are more computationally efficient. Taking advantage of learning from samples, statistical methods can give higher recognition accuracies.

We adopt a statistical classification scheme, wherein the recognition accuracy depends on the techniques of pre-processing, feature extraction, and classifier design. We use curve-fitting-based nonlinear normalization [3, 4] and pseudo two-dimensional (pseudo 2D) normalization methods [5], with slight modification to perform directly on online pattern trajectory. The local stroke direction of character pattern is decomposed into direction maps, which are blurred and sub-sampled to obtain feature values. We compare the direction feature from original pattern and that from normalized pattern. Feature extraction from original pattern incorporating coordinate normalization is also called normalization-cooperated feature extraction (NCFE) [6].

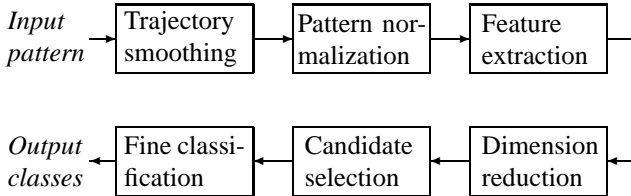
Our approach has two original features. First, we successfully applied curve-fitting-based nonlinear normalization and pseudo 2D normalization to online character recognition. Unlike the popular line density equalization method [7] that calculates stroke density on 2D image (for online recognition, the trajectory is mapped to a 2D image, e.g., [6, 8]), the curve-fitting-based method can be easily applied to online trajectory. Second, we apply the NCFE method directly to online trajectory, unlike the previous work that used NCFE on 2D image converted from trajectory [6]. Our normalization and feature extraction methods are both efficient and effective.

We have evaluated the recognition system on the TUAT HANDS databases (Kuchibue and Nakayosi) of online handwritten Japanese characters [9] and have obtained superior performance on the test set. The pseudo 2D normalization methods are shown to be superior to one-dimensional methods. On the other hand, direction features from original pattern and from normalized pattern made little difference.

The rest of this paper is organized as follows. Section 2 gives an overview of the recognition system. Section 3 describes the trajectory pattern normalization methods and Section 4 details direction feature extraction. Section 5 reports our experimental results and Section 6 provides concluding remarks.

## 2. System Overview

The online Japanese character recognition system is depicted diagrammatically in Fig. 1. It consists of two pre-processing steps (trajectory smoothing and pattern normalization), feature extraction, dimensionality reduction, and two classification stages (candidate selection and fine classification). Pre-processing is to regulate the pattern shape for reducing the within-class shape variation. Both dimensionality reduction and candidate selection are to accelerate classification.



**Figure 1.** Diagram of online character recognition system.

The input pattern trajectory, composed of the coordinates of sampled pen-down points, is smoothed by simply replacing the coordinates of each point with a weighted average of the current point and two neighboring points. We did not try sophisticated smoothing and trajectory re-sampling operations. In normalization, the coordinates are transformed such that the pattern size is standardized and the shape is regulated. We adopt curve-fitting-based normalization and pseudo 2D normalization methods, which need not converting the trajectory into 2D image.

For feature extraction, the local histogram of stroke direction (direction feature) has been proven very effective in character recognition. We use the directional decomposition technique of Kawamura et al. [10], which is particularly efficient for online trajectory. The stroke direction is either the one in original pattern or that in normalized pattern. We will compare both schemes.

We use the modified quadratic discriminant function (MQDF2) of Kimura et al. [11] for fine classification on candidate classes selected by a two-stage prototype classifier. The prototype classifier uses the class means as prototypes and the cluster centers of class means as group prototypes. The input pattern (feature vector) is first assigned to a number of nearest groups, and then compared with the class prototypes contained in the selected groups to select

a number of candidate classes. Further acceleration and as well, storage saving, are achieved by dimensionality reduction with Fisher linear discriminant analysis (FDA) [12].

We did not implement discriminatively trained dimensionality reduction and classification methods [13] since the aim of this work is to evaluate normalization and feature extraction methods.

## 3. Pattern Normalization

We briefly review the one-dimensional and pseudo 2D normalization methods and customize them to online character patterns. All these methods utilize the centroid information of original pattern.

### 3.1 Curve-fitting-based normalization

Curve-fitting-based normalization methods compute the coordinate mapping functions of  $x$ -axis and  $y$ -axis based on the projections of image intensity. For online patterns, the projections can be calculated directly from the trajectory without converting to a 2D image.

Three curve-fitting-based normalization methods using quadratic coordinate functions, called bi-moment normalization [3], centroid-boundary alignment (CBA), and modified CBA (MCBA) [4], have been proposed for off-line character recognition. The bi-moment method is an extension of the moment normalization method, which aligns the centroid of input image to the center of normalized image and re-scales the image according to 2nd-order moments:

$$\begin{aligned} x' &= \frac{W_2}{\delta_x} (x - x_c) + x'_c, \\ y' &= \frac{H_2}{\delta_y} (y - y_c) + y'_c. \end{aligned} \quad (1)$$

where  $(x_c, y_c)$  is the centroid of input image, and  $(x'_c, y'_c)$  is the center of normalized image with pre-specified width  $W_2$  and height  $H_2$ .  $\delta_x$  and  $\delta_y$  are the re-set width and height of input image from 2nd-order central moments:  $\delta_x = 4\sqrt{\mu_{20}}$ ,  $\delta_y = 4\sqrt{\mu_{02}}$ .

Unlike the moment normalization that sets the boundaries of input image equally distant from the centroid, the bi-moment method sets the four boundaries separately from bi-sected 2nd-order moments:  $\delta_x^-, \delta_x^+, \delta_y^-, \delta_y^+$ . Then quadratic functions are used to align  $(x_c - \delta_x^-, x_c, x_c + \delta_x^+)$  with  $(0, W_2/2, W_2)$  and  $(y_c - \delta_y^-, y_c, y_c + \delta_y^+)$  with  $(0, H_2/2, H_2)$ :

$$\begin{aligned} x' &= W_2 u(x), \\ y' &= H_2 v(y), \end{aligned} \quad (2)$$

where  $u(x)$  and  $v(y)$  are quadratic functions with output in  $[0,1]$ .

The CBA method uses quadratic functions to align the centroid and the physical boundaries (ends of stroke projections), i.e., it aligns  $(x_{min}, x_c, x_{max})$  with  $(0, W_2/2, W_2)$  and  $(y_{min}, y_c, y_{max})$  with  $(0, H_2/2, H_2)$ . It corrects the skewness of centroid but cannot correct the

imbalance of inner/outer density. The MCBA method adjusts the inner density by adding a sine function to the quadratic function:

$$\begin{aligned} x' &= W_2[u(x) + \eta_x \sin(2\pi u(x))], \\ y' &= H_2[v(y) + \eta_y \sin(2\pi v(y))]. \end{aligned} \quad (3)$$

The amplitudes of sine functions,  $\eta_x$  and  $\eta_y$ , are estimated from the centroids of half images partitioned by the global centroid of input image [4].

To apply the above normalization methods to online patterns, the only different thing is to calculate the projection functions of character trajectory, from which the centroid and moments can be easily computed. To do this, we imagine that the character trajectory is posed on a grid. Each stroke (pen-down movement) of the trajectory is viewed as a sequence of line segments in an imaginary image (each pixel corresponding to a unit cell in the grid), each defined by a pair of consecutive sampled points. The sampled points have continuous coordinates, and the nodes of the grid have discrete coordinates. The two projection functions (histogram functions with unit intervals) are initially  $p_X(x) = 0$  and  $p_Y(y) = 0$ . Then for each line segment, its length is assigned to both projection functions. The length of line segment assigned to an interval is the portion of length falling in the interval. In Fig. 2, a line segment is projected onto two intervals in  $x$ -axis and three intervals in  $y$ -axis, and its length is assigned proportionally to these intervals.

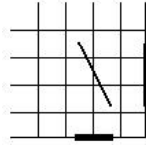


Figure 2. Projection of a line segment to two axes.

### 3.2 Pseudo two-dimensional normalization

In pseudo 2D normalization [5], coordinate mapping functions,  $x'(x, y)$  and  $y'(x, y)$ , are obtained by linearly combining one-dimensional (1D) functions with the weight depending on another dimension:

$$\begin{aligned} x'(x, y) &= \sum_i w^{(i)}(y)x'^{(i)}(x), \\ y'(x, y) &= \sum_i w^{(i)}(x)y'^{(i)}(y). \end{aligned} \quad (4)$$

The 1D coordinate functions are obtained by applying 1D normalization to the projection functions of partial images.

The character image  $f(x, y)$  (for online trajectory, the image is imaginary on a grid) can be reasonably partitioned into three horizontal soft strips:

$$f^{(i)}(x, y) = w^{(i)}(y)f(x, y), \quad i = 1, 2, 3, \quad (5)$$

where  $w^{(i)}(y)$  are weight functions (assuming the boundaries of  $y$ -axis are 0 and  $H_1$ ):

$$\begin{aligned} w^{(1)}(y) &= w_0 \frac{y_c - y}{y_c}, \quad y < y_c, \\ w^{(2)}(y) &= 1 - w^{(1)}(y), \quad y < y_c, \\ w^{(2)}(y) &= 1 - w^{(3)}(y), \quad y \geq y_c, \\ w^{(3)}(y) &= w_0 \frac{y - y_c}{H_1 - y_c}, \quad y \geq y_c, \end{aligned} \quad (6)$$

where  $0 \leq w_0 \leq 1$ . The projection functions of the three strips on  $x$ -axis,  $p_X^{(i)}(x)$ ,  $i = 1, 2, 3$ , are used to compute three coordinate functions,  $x'^{(i)}(x)$ , using any of the 1D normalization methods. The three 1D functions are then combined into a 2D coordinate function:

$$x'(x, y) = \begin{cases} w^{(1)}(y)x'^{(1)}(x) + w^{(2)}(y)x'^{(2)}(x), & y < y_c, \\ w^{(3)}(y)x'^{(3)}(x) + w^{(2)}(y)x'^{(2)}(x), & y \geq y_c. \end{cases} \quad (7)$$

Similarly, 1D coordinate functions,  $y'^{(i)}(y)$ ,  $i = 1, 2, 3$ , are computed from three vertical strips and combined into the 2D coordinate function  $y'(x, y)$ .

Depending on the method of computing 1D coordinate functions, we have variable pseudo 2D normalization methods: pseudo 2D moment normalization (P2DMN), pseudo 2D bi-moment normalization (P2DBMN), and pseudo 2D MCBA (P2DCBA)<sup>1</sup>. To apply these methods to online trajectory, the remaining task is to compute the projection functions of three strips from the input trajectory. Again, the trajectory is viewed as line segments in an imaginary image. On computing the global centroid  $(x_c, y_c)$  from the global projection functions  $p_X(x)$  and  $p_Y(y)$ , the weight functions  $w^{(i)}(y)$  and  $w^{(i)}(x)$  can be obtained. Then, the line segments can be directly assigned to the strip projection functions incorporating the weight functions. Say, when a line segment is assigned to an interval of  $p_X^{(i)}(x)$ , the length falling in the interval is also weighted by  $w^{(i)}(y)$  depending on the vertical position of the line segment.

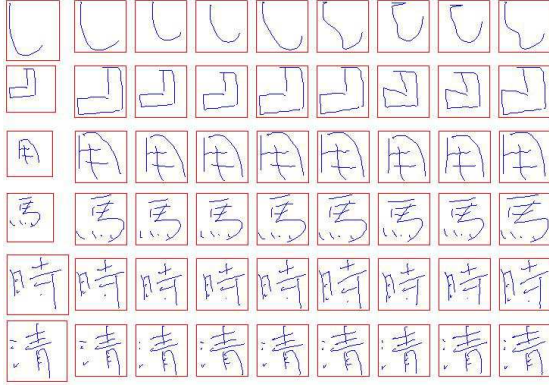
The coordinate mapping functions, obtained by either one-dimensional or pseudo 2D normalization methods, are used to transform the coordinates of all sampled points of character trajectory. The points with transformed coordinates form the trajectory of normalized pattern.

Fig. 3 shows six online character patterns and their normalized ones by eight methods: linear normalization (linear boundary alignment), moment, bi-moment, CBA, MCBA, P2DMN, P2DBMN, and P2DCBA. The weight in (6) was set to  $w_0 = 0.75$ . We can see that while pseudo 2D normalization methods can generate better normalized shapes than 1D methods on Kanji characters, they yield excessive deformation on characters of simple structures.

## 4. Feature Extraction

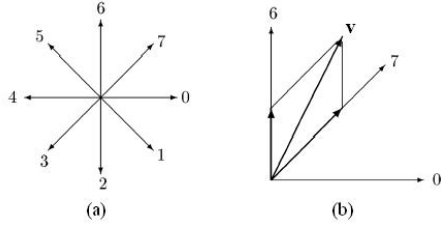
Direction feature extraction is performed in two steps: directional decomposition and blurring. The local stroke

<sup>1</sup>We did not extend the CBA method to pseudo 2D, and instead refer to the pseudo 2D extension of MCBA as P2DCBA.



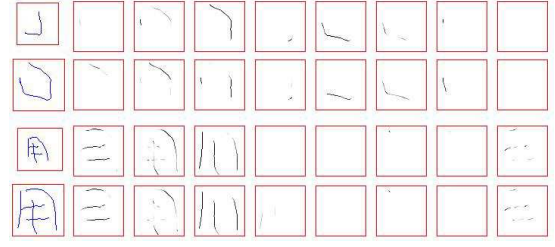
**Figure 3.** Online pattern and normalized patterns by eight methods (from left: original, linear, moment, bi-moment, CBA, MCBA, P2DMN, P2DBMN, and P2DCBA).

segments are assigned to a number of direction planes, then each plane is blurred (low-pass filtered) and the sampled values are taken as feature values. We use eight direction planes, corresponding to eight chaincode directions (Fig. 4 (a)). Following the method of Kawamura et al. [10], each line segment, defined by two consecutive pen-down points, is decomposed into two components in two neighboring chaincode directions (Fig. 4 (b)).



**Figure 4.** Eight chaincode directions (a) and the directional decomposition of a line segment (b).

The direction of the line segment depends on whether the coordinates of the two end points are normalized or not. The direction based on normalized coordinates (normalized direction) is conventionally used. In the case of nonlinear normalization or pseudo 2D normalization, however, the normalized direction may be deformed considerably. This can be alleviated by normalization-cooperated feature extraction (NCFE) [6], in which the local direction in original pattern is decomposed while the transformed coordinates are used to shift the direction elements in direction planes. Unlike previous NCFE methods performed on 2D images mapped from online patterns, we decompose the line segments of input trajectory directly into normalized direction planes. Further, our direction planes have continuous pixel values (continuous NCFE [14]), to account for the partial overlap of line segments with pixels.



**Figure 5.** Input pattern and decomposition of original direction (1st and 3rd rows), normalized pattern and decomposition of normalized direction (2nd and 4th rows).

Assume that the size of direction planes is  $W_2 \times H_2$  pixels, and each pixel is viewed as a square of unit area in a grid. A line segment in the input pattern trajectory,  $\mathbf{v} = ((x_1, y_1), (x_2, y_2))$ , is mapped to another segment in the normalized pattern by coordinate transformation:  $\mathbf{v}' = ((x'_1, y'_1), (x'_2, y'_2))$ . The vector  $\mathbf{v}$  (original direction) or  $\mathbf{v}'$  (normalized direction) is decomposed into two components as in Fig. 4 (b), with lengths  $l_1$  (direction 1) and  $l_2$  (direction 2). The corresponding two direction planes are given weights  $w_1 = l_1/l$  and  $w_2 = l_2/l$  ( $l$  is the length of  $\mathbf{v}$  or  $\mathbf{v}'$ ), respectively. The mapped line segment  $\mathbf{v}'$  overlaps with some pixels in two direction planes. In the plane of direction 1, each pixel overlapping with  $\mathbf{v}'$  is given a value, which equals the overlapping length in the unit area times the weight  $w_1$ . Similarly, in the plane of direction 2, each pixel overlapping with  $\mathbf{v}'$  is given a value of overlapping length times  $w_2$ .

Fig. 5 shows the direction planes of two online patterns by decomposition of original and normalized directions. We can see that in stroke regions of large deformation, the direction planes of original direction and normalized direction show difference.

On directional decomposition, each direction plane is blurred using a low-pass Gaussian filter. The pixel values are sampled uniformly, and according to the Sampling Theorem, the Gaussian filter parameter (variance of spherical Gaussian) was shown to be related to the sampling interval [15]:

$$\sigma_x = \frac{\sqrt{2}t_x}{\pi}, \quad (8)$$

where  $t_x$  is the sampling interval (reciprocal of sampling frequency).

In coordinate normalization and direction feature extraction, we set the size of normalized plane (and direction planes) to  $24 \times 24$  pixels. Since we directly assign the line segments (of original direction or normalized direction) to eight direction planes, and the direction planes have continuous pixel values, a moderately small size of direction plane does not sacrifice the recognition accuracy. Each direction plane is then blurred and sub-sampled with sampling interval 3. As result, we obtain 64 feature values

from each direction plane and the final dimensionality of feature vector is 512. To improve the Gaussianity of feature distribution, each value is transformed by Box-Cox transformation (also called variable transformation [12]). We set the power of variable transformation to 0.5 without attempt to optimize it.

## 5. Experimental Results

To evaluate the performance of the proposed methods, we have experimented on the TUAT HANDS databases, kuchibue\_d-97-06 (kuchibue\_d or Kuchibue in brief) and nakayosi\_t-98-09 (nakayosi\_t or Nakayosi in brief), of on-line handwritten Japanese characters [9]. The Kuchibue database contains the handwritten samples of 120 writers, 11,962 patterns per writer covering 3,356 character classes. Excluding the JIS level-2 Kanji characters, there are 11,951 patterns for 3,345 classes (2,965 Kanji characters and 380 symbols) per writer. The Nakayosi database contains the samples of 163 writers, 10,403 patterns covering 4,438 classes per writer, in which there are 9,309 patterns for the JIS level-1 Kanji characters and the 380 symbols (3,345 classes). As many previous works did [2, 16], we experimented with 3,345 classes, and used the samples of Nakayosi for training classifiers and the samples of Kuchibue for testing. Therefore, there are totally  $9,309 \times 163 = 1,517,367$  training samples and  $11,951 \times 120 = 1,434,120$  test samples.

As well as recognizing 3,345 classes of Kanji and symbols, we also carried out experiments of recognizing 2,965 JIS level-1 Kanji characters only. The Kuchibue database and Nakayosi database contain 766,915 and 675,840 samples of JIS level-1 Kanji characters, respectively. Again, the samples of Nakayosi database were used for training and the samples of Kuchibue database for testing.

On pre-processing and feature extraction of each sample, the feature dimensionality is reduced to 160 by Fisher linear discriminant analysis (FDA), whose parameters are estimated on the feature vectors of training samples. On the reduced vector, we select 100 candidate classes according to the Euclidean distance to class means by two-stage classification, where we used 200 group prototypes (cluster centers of class means). On the candidate classes, the modified quadratic discriminant function (MQDF2) [11] is computed for fine classification:

$$g_2(\mathbf{x}, \omega_i) = \sum_{j=1}^k \frac{1}{\lambda_{ij}} [\phi_{ij}^T (\mathbf{x} - \mu_i)]^2 + \frac{1}{\delta_i} r_i(\mathbf{x}) + \sum_{j=1}^k \log \lambda_{ij} + (d - k) \log \delta_i, \quad (9)$$

where  $\mu_i$  is the mean vector of class  $\omega_i$ ,  $\lambda_{ij}$  ( $j = 1, \dots, k$ ) are the largest eigenvalues of the covariance matrix and  $\phi_{ij}$  are the corresponding eigenvectors,  $k$  denotes the number of principal axes and  $r_i(\mathbf{x})$  is the residual of subspace projection:  $r_i(\mathbf{x}) = \|\mathbf{x} - \mu_i\|^2 - \sum_{j=1}^k [(\mathbf{x} - \mu_i)^T \phi_{ij}]^2$ .  $\delta_i$  is set to a class-independent constant and

its value is optimized by holdout cross-validation on the training data set. Specifically,  $\delta_i$  was selected to maximize the classification accuracy on 1/5 of training samples on estimating class means, principal eigenvalues and eigenvectors on the remaining 4/5 of training samples. We used  $k = 50$  principal eigenvectors for each class.

Table 1 shows the test accuracies (correct rates on test set) of 3,345-class recognition, and Table 2 shows the test accuracies of 2,965-class Kanji recognition. We give the accuracies of Euclidean distance (nearest class mean) as well as MQDF2. We can see that for both class sets, all the centroid-based normalization methods (moment, bi-moment, CBA, MCBA, P2DMN, P2DBMN, P2DCBA) yield higher accuracies than linear normalization.

**Table 1.** Test accuracies (%) of 3,345-class recognition.

3,345 classes	Original direct.		Normalized direct.	
	Euclid	MQDF2	Euclid	MQDF2
Linear	77.60	89.07	77.67	88.78
Moment	82.97	90.51	82.95	90.44
Bi-moment	82.69	90.57	82.78	90.53
CBA	82.08	90.17	81.94	90.07
MCBA	82.12	90.35	81.90	90.05
P2DMN	83.54	90.81	83.20	90.88
P2DBMN	<b>83.97</b>	<b>90.94</b>	<b>83.53</b>	<b>90.89</b>
P2DCBA	83.49	90.87	83.33	90.64

**Table 2.** Test accuracies (%) of 2,965-class Kanji recognition.

2,965 classes	Original direct.		Normalized direct.	
	Euclid	MQDF2	Euclid	MQDF2
Linear	87.02	95.91	87.12	95.99
Moment	93.86	97.90	93.82	97.84
Bi-moment	93.77	97.85	93.76	97.82
CBA	92.97	97.53	93.01	97.57
MCBA	93.56	97.77	93.60	97.74
P2DMN	94.70	98.17	94.64	<b>98.24</b>
P2DBMN	<b>94.74</b>	<b>98.18</b>	<b>94.71</b>	98.22
P2DCBA	94.50	98.06	94.56	98.11

In both 3,345-class and Kanji recognition, the accuracies of centroid-based normalization methods can be roughly divided into three grades: {CBA}, {moment, bi-moment, MCBA}, {P2DMN, P2DBMN, P2DCBA}. The methods in each of the three sets perform comparably, though the accuracies of MCBA are slightly lower than the moment and bi-moment methods and the accuracies of P2DCBA are slightly lower than P2DMN and P2DBMN. The moment and bi-moment methods perform comparably, and their pseudo 2D extensions yield best performance. The difference of accuracy between one-

dimensional normalization methods and pseudo 2D methods is evident for both 3,345-class recognition and Kanji recognition.

Despite that the decomposition of original direction and that of normalized direction may give different feature values (Fig. 5), the experimental results show that their recognition accuracies are comparable. We expect that the two types of direction features are complementary on some characters.

To compare with previous results [2], we found only one work that has used all the samples of 3,345 classes of Nakayosi in training and all the samples of Kuchibue in testing. In this experimentation scheme, Kitadai and Nakagawa reported a recognition rate 87.2% [16]. Thus, our recognition rates of over 90.8% are superior.

We observed a big gap between the accuracies of 3,345-class recognition and Kanji recognition. This is because there are too many confusing characters among the 380 symbols and between symbols and Kanji characters. Our results of either 3,345-class or Kanji recognition can serve new benchmarks for evaluating future works.

## 6. Conclusion

We have implemented an online handwritten Japanese character recognition system using efficient trajectory-based normalization and direction feature extraction methods. We compared one-dimensional and pseudo 2D normalization methods, and direction features from original pattern and from normalized pattern. Our recognition results are superior to previous ones. The pseudo 2D normalization methods yield higher accuracies than one-dimensional ones, whereas direction features from original and normalized patterns perform comparably.

It is our future work to modify the pseudo 2D normalization methods for better transforming patterns of simple structures. On the other hand, the recognition accuracy can be improved by better dimensionality reduction and classification methods.

## Acknowledgements

This work is supported by Central Research Laboratory, Hitachi, Ltd. (Tokyo, Japan). The authors thank Katsumi Marukawa and Takeshi Nagasaki for cooperation, and Prof. Masaki Nakagawa for providing the HANDS databases.

## References

- [1] C.C. Tappert, C.Y. Suen, T. Wakahara, The state of the art in on-line handwriting recognition, *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(8): 787-808, 1990.
- [2] C.-L. Liu, S. Jaeger, M. Nakagawa, Online recognition of Chinese characters: the state-of-the-art, *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(2): 198-213, 2004.
- [3] C.-L. Liu, H. Sako, H. Fujisawa, Handwritten Chinese character recognition: alternatives to nonlinear normalization, *Proc. 7th ICDAR*, Edinburgh, Scotland, 2003, pp.524-528.
- [4] C.-L. Liu, K. Marukawa, H. Sako, H. Fujisawa, Global shape normalization for handwritten Chinese character recognition: a new method, *Proc. 9th IWFHR*, Tokyo, 2004, pp.300-305.
- [5] C.-L. Liu, K. Marukawa, Pseudo two-dimensional shape normalization methods for handwritten Chinese character recognition, *Pattern Recognition*, 38(12): 2242-2255, 2005.
- [6] M. Hamanaka, K. Yamada, J. Tsukumo, On-line Japanese character recognition experiments by an off-line method based on normalization-cooperated feature extraction, *Proc. 3rd ICDAR*, Tsukuba, Japan, 1993, pp.204-207.
- [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] Z.-L. Bai, Q. Huo, A study on the use of 8-directional features for online handwritten Chinese character recognition, *Proc. 8th ICDAR*, Seoul, Korea, 2005, pp.262-266.
- [9] K. Matsumoto, T. Fukushima, M. Nakagawa, Collection and analysis of on-line handwritten Japanese character patterns, *Proc. 6th ICDAR*, Seattle, 2001, pp.496-500.
- [10] A. Kawamura, et al., On-line recognition of freely handwritten Japanese characters using directional feature densities, *Proc. 11th ICPR*, The Hague, 1992, Vol.2, pp.183-186.
- [11] F. Kimura, K. Takashina, S. Tsuruoka, Y. Miyake, Modified quadratic discriminant functions and the application to Chinese character recognition, *IEEE Trans. Pattern Anal. Mach. Intell.*, 9(1): 149-153, 1987.
- [12] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd edition, Academic Press, 1990.
- [13] C.-L. Liu, High accuracy handwritten Chinese character recognition using quadratic classifiers with discriminative feature extraction, *Proc. 18th ICPR*, Hong Kong, 2006.
- [14] C.-L. Liu, K. Nakashima, H. Sako, H. Fujisawa, Handwritten digit recognition: investigation of normalization and feature extraction techniques, *Pattern Recognition*, 37(2): 265-279, 2004.
- [15] C.-L. Liu, Y.-J. Liu, R.-W. Dai, Preprocessing and statistical/structural feature extraction for handwritten numeral recognition, in *Progress of Handwriting Recognition*, A.C. Downton and S. Impedovo (Eds.), World Scientific, 1997, pp.161-168.
- [16] A. Kitadai, M. Nakagawa, A learning algorithm for structural character pattern representation used in on-line recognition of handwritten Japanese characters, *Proc. 8th IWFHR*, Ontario, Canada, 2002, pp.163-168.