PERGAMON

Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Pattern Recognition 37 (2004) 265–279

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/patcog

Handwritten digit recognition: investigation of normalization and feature extraction techniques

Cheng-Lin Liu*, Kazuki Nakashima, Hiroshi Sako, Hiromichi Fujisawa

Central Research Laboratory, Hitachi Ltd., 1-280 Higashi-koigakubo, Kokubunji-shi, Tokyo 185-8601, Japan

Received 13 November 2002; accepted 14 June 2003

Abstract

The performance evaluation of various techniques is important to select the correct options in developing character recognition systems. In our previous works, we have proposed aspect ratio adaptive normalization (ARAN) and have evaluated the performance of state-of-the-art feature extraction and classification techniques. For this time, we will propose some improved normalization functions and direction feature extraction strategies and will compare their performance with existing techniques. We compare ten normalization functions (seven based on dimensions and three based on moments) and eight feature vectors on three distinct data sources. The normalization functions and feature vectors are combined to produce eighty classification accuracies to each dataset. The comparison of normalization functions shows that moment-based functions outperform the dimension-based ones and the aspect ratio mapping is influential. The comparison of feature vectors shows that the improved feature extraction strategies outperform their baseline counterparts. The gradient feature from gray-scale image mostly yields the best performance and the improved NCFE (normalization-cooperated feature extraction) features also perform well. The combined effects of normalization, feature extraction, and classification have yielded very high accuracies on well-known datasets.

Keywords: Handwritten digit recognition; Normalization; Aspect ratio mapping; Direction feature; NCFE; Gradient feature

1. Introduction

In developing optical character recognition (OCR) systems, every module or step has multiple options of algorithmic implementation. Generally, a character recognizer involves three tasks: pre-processing, feature extraction, and classification. The options of pre-processing include the size and aspect ratio of normalized image, the interpolation technique of pixel values, etc. For feature extraction, a large variety of feature types and extraction techniques are available. In addition, a large number of classifiers are available for classification: parametric and nonparametric statistical classifiers, neural networks, support vector machines (SVMs), hybrid classifiers, etc. To select the correct option for each task, the available algorithms and feasible parameters should be evaluated in an environment relevant to practical application. To avoid exhaustive comparison, we may pre-select a subset of algorithms according to the requirements of application, the general characteristics of algorithms, and the previous evaluation results.

In pattern recognition community, numerous works have been contributed to the evaluation of classification algorithms. While in character recognition, a number of evaluation works have been accomplished to pre-processing and feature extraction as well. Our previous results of evaluation of feature extraction and classification techniques have provided some practical insights [1,2]. We have shown that some features and classifiers generally yield high performance at low complexity while some others are not preferable for practical applications due to the low performance or high complexity. For pre-processing, we have shown that

* Corresponding author. Tel: +81-42-323-1111; fax: +81-42-327-7778.
E-mail address: liucl@crl.hitachi.co.jp (C.-L. Liu).

0031-3203/$30.00
doi:10.1016/S0031-3203(03)00224-3

the normalization of character images is influential to the recognition performance and have proposed an aspect ratio adaptive normalization (ARAN) strategy to improve the performance [3].

This paper proposes some improved normalization and feature extraction strategies and will evaluate their performance with comparison to existing techniques. We focus on the diversity of implementation for ARAN and direction feature extraction. The performance of ARAN depends on the aspect ratio mapping function. The direction feature is most popularly used in character recognition and the performance largely depends on the representation of feature and the extraction technique. In this paper, we will give ten normalization functions (seven based on dimensions and three based on moments) and eight feature vectors (six existing ones and two new ones) for comparison to find out the good choices. The feature vectors represent the direction distributions of chaincodes or gradients. The new feature vectors are aimed to upgrade the performance of NCFE (normalization-cooperated feature extraction) [4]. Some related previous works are briefly reviewed as follows.

Normalization is considered to be the most important pre-processing factor for character recognition [5]. Normally, the character image is linearly mapped onto a standard plane by interpolation/extrapolation. The size and position of character is controlled such that the x/y dimensions of normalized plane are filled. The implementation of interpolation/extrapolation is influential to the recognition performance [6,7]. By linear mapping, the character shape is not deformed except the aspect ratio changes. Some strategies were proposed to deform the character shape with aim to reduce the within-class variation. The perspective transformation attempts to correct the imbalance of character width [8], the moment normalization attempts to rectify the rotation or slant [9], and the nonlinear normalization aims to equalize the line density [10,11]. For slant normalization, the slant can also be estimated from character field context instead of moments [12].

Feature extraction is at the core of character recognition and a large variety of feature types and extraction strategies have been proposed [13]. The distribution of local stroke direction (direction feature) is most popularly used due to the high performance and the ease of implementation. The local stroke direction can be measured from skeleton [14], chaincode [15,16], or gradient [17–19]. The chaincode feature is widely adopted, whereas the gradient feature is applicable to gray-scale images as well as binary images. Some complementary features were proposed to enhance the discriminating power of direction feature, such as the structural and concavity features [17], profile shape feature [20], and curvature feature [21]. On the other hand, the so-called normalization-cooperated feature extraction (NCFE) method extracts direction feature from the original image other than the normalized image [4].

We have evaluated previously a number of statistical and neural classifiers in handwritten character recognition [1,2] and have found out that some classifiers give high accuracies at low complexity. On the other hand, the support vector machine (SVM) [21,22] is superior in classification accuracy but very computationally expensive. In this paper, for testing the performance of normalization and feature extraction techniques, we use three classifiers that give high accuracies: the polynomial classifier [23,24], the discriminative learning quadratic discriminant function (DLQDF) [25], and the support vector classifier with radial basis function kernel (SVC-rbf).

Among the ten normalization functions that we test, the basic strategies, namely, linear normalization, nonlinear normalization, and moment normalization, have been long used in character recognition. Our contribution is to diversify the aspect ratio mapping to improve the recognition performance. Among the eight feature vectors, the basic feature types (chaincode feature, profile shape feature, NCFE feature, and gradient feature) were proposed previously and have been evaluated in our previous works. This paper proposes two improved versions to the NCFE: the enhanced NCFE with profile shape feature and the continuous NCFE. The improved versions of chaincode feature and gradient feature were proposed in our previous works [2] and will be further evaluated in this study.

We test the performance of handwritten digit recognition on three databases: CENPARMI [26], NIST [27], and Hitachi. The databases have different characteristics and hence show different tendencies of performance. The digit images of CENPARMI database were collected from live mail images of USPS, the NIST database was collected from specially designed forms filled by US Census Bureau employees and high school students. The Hitachi database contains a huge number of digit images collected from sampling sheets and real form images and is divided into several subsets of varying qualities. From each of three databases, a dataset is specified for training classifiers, and the other dataset(s) are tested with the trained classifiers. On a test dataset, a specific classifier gives eighty classification accuracies based on the combination of ten normalization functions and eight feature vectors. Further, each feature vector has two variations depending on the resolution of direction: 4-orientation and 8-direction, which are evaluated separately.

The rest of this paper is organized as follows. Section 2 describes the normalization strategies; Section 3 describes the feature extraction techniques. Section 4 presents the experimental results and Section 5 provides concluding remarks.

2. Normalization techniques

2.1. Implementation of normalization

For ease of feature extraction and classification, the x/y dimension (size) of normalized image plane (standard

plane) is fixed. In aspect ratio adaptive normalization (ARAN), however, the dimensions of the standard plane are not necessarily filled [3]. Depending on the aspect ratio, the normalized image is centered in the plane with one dimension filled. Assume the standard plane is square and the side length is denoted by L. Denote the width and height of the normalized character image as W_2 and H_2, the aspect ratio is defined by

$$R_2 = \begin{cases} W_2/H_2, & \text{if } W_2 < H_2, \\ H_2/W_2, & \text{otherwise.} \end{cases} \quad (1)$$

If the normalized image fills one dimension, then $\max(W_2, H_2) = L$. In moment-based normalization with the centroid of character image aligned to the center of standard plane, however, the normalized image does not necessarily fill one dimension and may go beyond the plane. In this case, W_2 and H_2 are not decided by L, and the image part outside the standard plane is cut off.

In the implementation of ARAN, the normalized character image is filled into another plane of flexible size $W_2 \times H_2$, and then this flexible plane is shifted to overlap the standard plane by aligning boundaries or centroid. In the following, we will illustrate the transformation of a character image of size $W_1 \times H_1$ into a normalized image of size $W_2 \times H_2$. The transformation can be accomplished by forward mapping or backward mapping. Denote the original image and the normalized image as $f(x, y)$ and $g(x', y')$, respectively, the normalized image is generated by $g(x', y') = f(x, y)$ based on coordinate mapping. The forward mapping and backward mapping are given by

$$x' = x'(x, y),$$

$$y' = y'(x, y),$$

and

$$x = x(x', y'),$$

$$y = y(x', y'),$$

respectively. In the following, we first describe the coordinate mapping of various normalization methods, and then address the interpolation of pixels.

The forward mapping and backward mapping of linear normalization, moment normalization, slant normalization, and nonlinear normalization are tabulated in Table 1. In the table, α and β denote the ratios of transformation, given by

$$\alpha = W_2/W_1,$$

$$\beta = H_2/H_1.$$

While for nonlinear normalization, h_x and h_y denote the normalized accumulated line density histograms [10,11].

In our experiments, the moment normalization refers to linear transformation without rotation, with the center and size of normalized image determined by moments. (x_c, y_c) denotes the center of gravity (centroid) of the original image, given by

$$x_c = m_{10}/m_{00},$$

$$y_c = m_{01}/m_{00},$$

where m_{pq} denotes the geometric moments:

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y),$$

and (x_c', y_c') denotes the geometric center of the normalized plane, given by

$$x_c' = W_2/2,$$

$$y_c' = H_2/2.$$

While in slant normalization based on moments, the angle of slant is calculated from second-order moments:

$$\tan\theta = \frac{\mu_{11}}{\mu_{02}},$$

where

$$\mu_{pq} = \sum_x \sum_y (x - x_c)^p (y - y_c)^q f(x, y).$$

Table 1
Coordinate mapping of various normalization methods

Method	Forward mapping	Backward mapping
Linear	$x' = \alpha x$ $y' = \beta y$	$x = x'/\alpha$ $y = y'/\beta$
Moment	$x' = \alpha(x - x_c) + x_c'$ $y' = \beta(y - y_c) + y_c'$	$x = (x' - x_c')/\beta + x_c$ $y = (y' - y_c')/\beta + y_c$
Slant	$x' = x - (y - y_c)\tan\theta$ $y' = y$	$x = x' + (y - y_c)\tan\theta$ $y = y'$
Nonlinear	$x' = W_2 h_x(x)$ $y' = H_2 h_y(y)$	$x = h_x^{-1}(x'/W_2)$ $y = h_y^{-1}(y'/H_2)$

In forward mapping, x and y are discrete but $x'(x, y)$ and $y'(x, y)$ are not necessarily discrete; while in backward mapping, x' and y' are discrete but $x(x', y')$ and $y(x', y')$ are not necessarily discrete. Further, in forward mapping, the mapped coordinates (x', y') do not necessarily fill all pixels in the normalized plane. Therefore, coordinate discretization or pixel interpolation is needed in the implementation of normalization.

In our experiments, all the (original) character images have binary gray levels. We use coordinate discretization to generate binary normalized images and use pixel interpolation to generate gray-scale normalized images. By discretization, the mapped coordinates (x', y') or (x, y) are approximated with the closest integer numbers, $([x'], [y'])$ or $([x], [y])$. In the discretization of backward mapping, the discrete coordinates (x', y') scan the pixels of the normalized

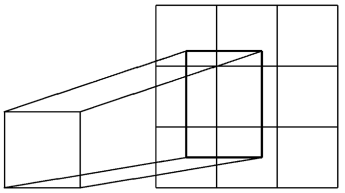

Fig. 1. Forward mapping of unit square pixel.

plane and the gray level $f([x],[y])$ is assigned to $g(x',y')$. In the discretization of forward mapping, the discrete coordinates (x,y) scan the pixels of the original image and the gray level $f(x,y)$ is assigned to all the pixels ranged from $([x'(x)],[y'(y)])$ to $([x'(x+1)],[y'(y+1)])$.

In the interpolation of backward mapping for generating gray-scale image, the mapped position (x,y) is surrounded with four discrete pixels. The gray level $g(x',y')$ is a weighted combination of the four pixel values. In the interpolation of forward mapping for generating gray-scale image, each pixel in the original image and the normalized image is viewed as a square of unit area. By coordinate mapping, the unit square of original image is mapped to a rectangle in the normalized plane. As illustrated in Fig. 1, in the normalized plane, each unit square overlapped with the rectangle is given a gray level proportional to the overlapping area.

2.2. Aspect ratio mapping

To implement the normalization, the width and height of the normalized image, W_2 and H_2, are to be determined. We set $\max(W_2,H_2)$ equal to the side length L of the standard plane, while $\min(W_2,H_2)$ is determined by the aspect ratio according to Eq. (1). The aspect ratio of the normalized image is adaptable to that of the original image. Hence the aspect ratio mapping function determines the size and shape of the normalized image. In the following, we will give ten normalization functions with varying aspect ratio mapping, among which seven are based on image dimensions and three are based on moments.

For dimension-based normalization, the actual image width and height are taken as the dimensions. While for moment-based normalization, the original image is viewed to be centered at the centroid and the boundaries are re-set to $[x_c-2\sqrt{\mu_{20}},x_c+2\sqrt{\mu_{20}}]$ and $[y_c-2\sqrt{\mu_{02}},y_c+2\sqrt{\mu_{02}}]$. Accordingly, the dimensions are re-set to $W_1=4\sqrt{\mu_{20}}$ and $H_1=4\sqrt{\mu_{02}}$. The image plane is expanded or trimmed so as to fit this range. The aspect ratio of the original image is then calculated by

$$R_1=\begin{cases} W_1/H_1, & \text{if } W_1<H_1,\\ H_1/W_1, & \text{otherwise.}\end{cases}$$

Varying the normalization strategy and the aspect ration mapping function, the ten normalization functions are listed as follows.

- F0: linear normalization with fixed aspect ratio, $R_2=1$.
- F1: linear normalization with aspect ratio preserved, $R_2=R_1$.
- F2: linear normalization with square root of aspect ratio, $R_2=\sqrt{R_1}$.
- F3: linear normalization with cubic root of aspect ratio, $R_2=\sqrt[3]{R_1}$.
- F4: linear normalization with piecewise linear aspect ratio,
$$R_2=\begin{cases} 0.25+1.5R_1, & \text{if } R_1<0.5\\ 1, & \text{otherwise}\end{cases}$$
- F5: linear normalization with square root of sine of aspect ratio,
$$R_2=\sqrt{\sin\left(\frac{\pi}{2}R_1\right)}.$$
- F6: nonlinear normalization with aspect ratio mapping F5. The line density histograms are calculated using the method of [10].
- F7: moment normalization with aspect ratio preserved.
- F8: moment normalization with square root of aspect ratio.
- F9: moment normalization with cubic root of aspect ratio.

The aspect ratio mapping functions F0–F5 are plotted in Fig. 2. Among the functions, F0, F4 and F5 have been tested in [3], wherein F4 and F5 were shown to outperform F0.

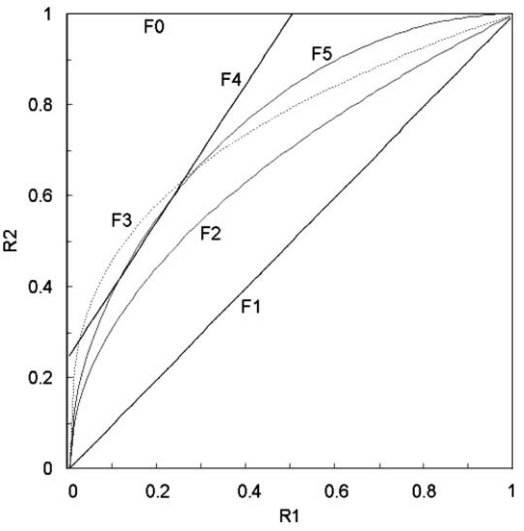

Fig. 2. Curves of aspect ratio mapping functions.

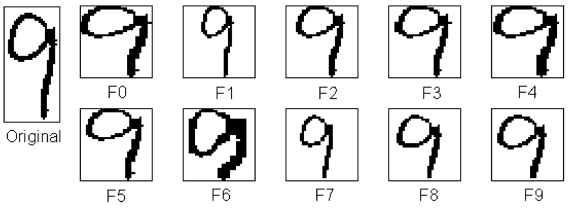

Fig. 3. Normalized binary images of 10 normalization functions.

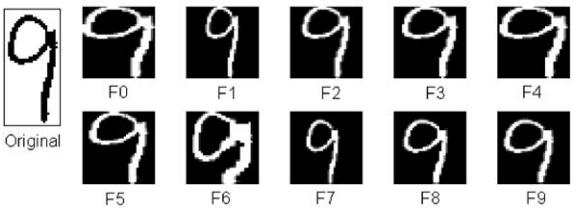

Fig. 4. Normalized gray-scale images of 10 normalization functions.

In our experiments, all the normalization functions are implemented by forward mapping. Figs. 3 and 4 show the normalized binary images and gray-scale images of the same original image corresponding to 10 normalization functions, respectively. The size of standard normalized plane is 35×35. We can see that when the aspect ratio of normalized image deviates much from the original aspect ratio (e.g., F0 and F5), the deformation of character shape is also considerable. For moment-based normalization (F7, F8 and F9), the normalized image is centered at the centroid and neither dimension of the standard plane is filled.

3. Feature extraction techniques

We extract three types of direction features: chaincode feature, NCFE feature, and gradient feature. The chaincode feature has two variations depending on whether slant normalization is applied or not. For chaincode feature and gradient feature extraction, the normalized image is decomposed into four orientation planes or eight direction planes. While in NCFE, the edge pixels of the original image are directly mapped to 4-orientation or 8-direction planes. For feature measuring, each feature plane is partitioned into uniform zones and the intensity of each zone is accumulated as a measurement, or blurring masks are convolved with the plane to give measurements. The convolution with blurring masks is equivalent to low-pass filtering and sampling. Usually, a Gaussian mask is used:

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

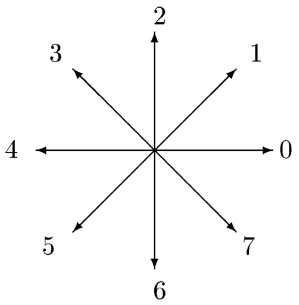

Fig. 5. Eight directions of chaincodes.

The variance parameter σ_x is related to the interval between blurring masks, which can be viewed as the sampling interval and is the reciprocal of sampling frequency. An empirical formula was given in Ref. [20]:

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

where t_x is the interval between blurring masks in x-axis and y-axis. At the center (x_0, y_0) of blurring mask, the convolution gives a measurement

$$F(x_0, y_0) = \sum_x \sum_y f(x, y)h(x - x_0, y - y_0).$$

In chaincode feature extraction, the contour pixels of the normalized image are assigned 8-direction codes (Fig. 5) and the contour pixels of each direction are assigned to the corresponding direction plane. The assignment of chaincodes can be accomplished in a raster scan of the image without contour tracing [20]. In this procedure, a pixel of multiple connectivity can be assigned multiple chaincodes. If 4-orientation feature is to be extracted, the direction planes of each pair of opposite directions are merged into one feature plane, and blurring is performed on four planes.

For NCFE, each chaincode in the original image is viewed as a line segment, which is mapped to another line segment in a standard direction plane. In the direction plane, each grid cell crossed by the line segment in the main (x or y) direction is given a unit of direction contribution. Since the coverage of line segment to the grid cells is not discrete, we hereby propose an improved NCFE method based on continuous feature plane. In the continuous plane, a pixel is viewed as a unit square and the strength of direction is proportional to the length of line segment falling in this square. An example is illustrated in Fig. 6, where a line segment mapped from a chaincode covers four grid cells A, B, C and D. By discrete NCFE, the cells A and C are assigned a direction unit, whereas by continuous NCFE, all the four cells are assigned direction strengths proportional to the in-cell line segment length. Note that for NCFE, sometime smoothing the character image is necessary so that the chaincodes better account for stroke direction.

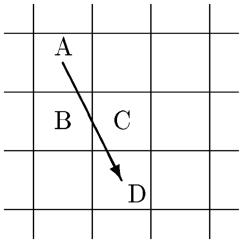

Fig. 6. NCFE on continuous feature plane.

-1	0	1
-2	0	2
-1	0	1

1	2	1
0	0	0
-1	-2	-1

Fig. 7. Sobel masks for gradient.

For gradient feature extraction, we use the Sobel operator to compute the x/y components of gradient and the gradient image is decomposed into four orientation planes or eight direction planes. The Sobel operator has been used by other researchers [17,18], and some researchers have also used the Roberts operator [19] and Kirsh operator [28].

The Sobel operator has two masks to compute the gradient components in two axes. The masks are shown in Fig. 7 and the gradient $\mathbf{g}(x, y) = [g_x, g_y]^T$ at location (x, y) is computed by

$$g_x(x, y) = f(x+1, y-1) + 2f(x+1, y) + f(x+1, y+1) - f(x-1, y-1) - 2f(x-1, y) - f(x-1, y+1),$$

$$g_y(x, y) = f(x-1, y+1) + 2f(x, y+1) + f(x+1, y+1) - f(x-1, y-1) - 2f(x, y-1) - f(x+1, y-1).$$

The gradient strength and direction can be computed from the vector $[g_x, g_y]^T$. For character feature extraction, the gradient of every pixel on the normalized image is computed. The range of gradient direction is partitioned into a number (say, eight or sixteen) of regions and each region corresponds to a direction plane. Each pixel is assigned to a direction region and the gradient strength contributes to the intensity of the corresponding plane. This strategy has been adopted in previous works [17,19]. In our experiments, however, we adopt another strategy, which decomposes each gradient vector into components in standard directions. This strategy was previously used in feature extraction of on-line character recognition [29]. We decompose the gradient vectors into eight chaincode directions. If a gradient direction lies between two standard direction, it is decomposed into two components in the two standard directions, as shown in Fig. 8.

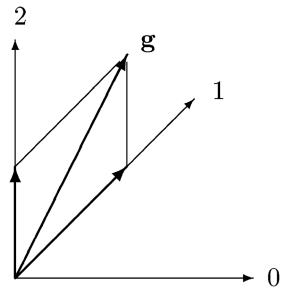

Fig. 8. Decomposition of gradient direction.

The gradient feature can be extracted from either binary or gray-scale normalized image. From a binary character image, the gray-scale normalized image is generated by forward binary-to-gray pixel mapping as illustrated in Section 2. We also call the gray-scale image generated this way as pseudo-gray image [2]. The gradient feature from pseudo-gray image is an improved version of the gradient feature from binary image.

To enhance the discriminating power of direction feature, we have used structural and profile shape features as complementary features. The structural feature is represented as the horizontal crossing counts while the profile shape feature is represented as the distances between the outmost edge and the convex hull [20]. The sequences of crossing counts and profile distances are also blurred by 1D Gaussian masks to give feature measurements. The length of sequences is 35 since the normalized plane is set to 35×35 pixels. We extract 11 blurred horizontal crossing counts and 22 (left and right) profile shape measurements. These measurements are added to the blurred chaincode feature to form an enhanced feature vector.

The NCFE feature can also be enhanced with complementary features. In the manner of NCFE, we calculate the crossing counts and profile shape distances from the original image and map them into the standard plane. They are then blurred to give $11 + 22$ measurements to add to the NCFE direction feature. The enhanced NCFE feature as well as the continuous NCFE feature can be viewed as improvements to the existing NCFE method.

In our experiments, we test eight direction features or enhanced feature vectors. Each feature vector has a 4-orientation version and an 8-direction version. The feature vectors are listed in the following.

- Feature vector **blr**: blurred chaincode feature, 100D for 4-orientation and 200D for 8-direction. On each feature plane, 5×5 Gaussian masks are uniformly placed to compute 25 measurements.

- Feature vector **des**: deslant chaincode feature, 100D for 4-orientation and 200D for 8-direction. The original image is deslanted and then normalized to extract chaincode feature.
- Feature vector **mul**: enhanced chaincode feature with complementary features, 133D for 4-orientation and 233D for 8-direction.
- Feature vector **ncf**: discrete NCFE direction feature, 100D for 4-orientation and 200D for 8-direction.
- Feature vector **ncf-p**: discrete NCFE direction feature plus complementary features. 133D for 4-orientation and 233D for 8-direction.
- Feature vector **ncf-c**: continuous NCFE direction feature, 100D for 4-orientation and 200D for 8-direction.
- Feature vector **grd**: gradient feature from binary images, 100D for 4-orientation and 200D for 8-direction.
- Feature vector **grd-g**: gradient feature from pseudo gray-scale images, 100D for 4-orientation and 200D for 8-direction.

To modify the feature distribution for improving the classification performance, all the measurements in the feature vectors are transformed by variable transformation $y = x^p$ [30]. This transformation is also well known as Box-Cox transformation [31]. For ease of calculation, the power parameter is not optimized in our experiments but set to p=0.5.

4. Experimental results

4.1. Results on CENPARMI database

The handwritten digit database of CENPARMI (Concordia University, Canada) contains 6000 digit images (600 images from each of 10 classes) collected from live mails of USPS, scanned at 166DPI. In the database, 4000 images are specified for training and the remaining 2000 images are for testing. Some images of the test dataset are shown in Fig. 9.

We use two classifiers in the recognition of CEPARMI database: the polynomial classifier (PC) and the support vector classifier with radial basis function kernel (SVC-rbf). In our previous experiments, the SVC-rbf were shown to perform best while the PC is the best among non-SV classifiers [2]. The SVC-rbf is far more expensive than the PC in training and classification, but since the CENPARMI dataset is small, the computation for testing various normalization functions and features is allowable. More details of the classifiers can be found in Ref. [2]. The PC uses 60 principal components of the pattern space and their binomial expansions as the inputs of learnable layer. In the implementation of SVC-rbf, the variance parameter of the kernel is proportional to the estimated sample variance and the upper bound of coefficient is set to 10.

After training the classifiers on the training dataset, each classifier is used to classify the test dataset. The classification error rates of 4-orientation features and 8-direction features by two classifiers are listed in Table 2. For the results of each classifier, the lowest error rate of each feature vector among ten normalization functions is highlighted. The average error rate of each feature vector is calculated over the normalization functions while the average error rate of each normalization function is calculated over the feature vectors. The normalization functions and features can be rank ordered according to the average error rates. To better account for the confidence of difference of performance, we calculate a relative performance measure (RPM). Among the normalization functions or feature vectors, set the RPM of the method with minimum average error rate to 100, the RPM of other methods is calculated by

$$RPM = \frac{100\,AER}{\min(AER)}, \tag{2}$$

where AER denotes the average error rate. The RPM is insensitive to the abstract error rates and hence serves as a good measure of performance relative to the best method. It provides more information than the rank order in that the difference of RPM between neighboring rank positions is variable.

The reciprocal of RPM corresponds to the ratio of error rate between the best method and an alternative method. When the ratio of error rate is greater than 0.9 (correspondingly, the RPM is less than 111.1), we consider that the performance of the alternative method is competitive to the best one. Otherwise, if the ratio is greater than 0.8 (the RPM is less than 125), we consider that the performance of the alternative method is good. The ratio between 0.7 and 0.8 (RPM between 125 and 142.9) indicates that the alternative method is fair, and the ratio less than 0.7 (RPM greater than 142.9) signifies poor performance of the alternative method.

On the results of CENPARMI database, we compare the performances of normalization functions and feature vectors in respect of the average error rates. Comparing the

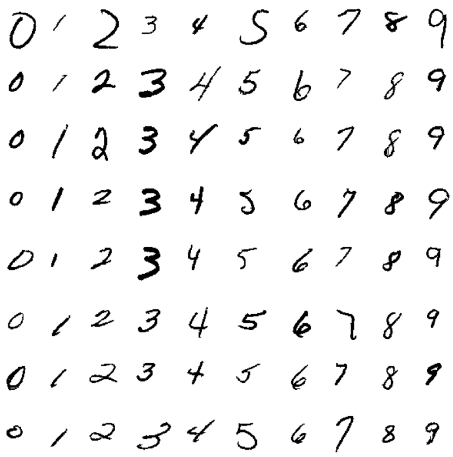

Fig. 9. Sample images of CENPARMI database.

Table 2
Error rates (%) of CENPARMI test dataset

Method	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	Aver	Relative
4-orientation, PC												
blr	1.60	1.50	1.40	1.45	1.65	1.55	1.95	1.70	**1.20**	**1.20**	1.520	128.3
des	1.05	1.55	1.10	1.00	1.00	**0.95**	1.35	1.50	1.15	1.20	1.185	100
mul	1.45	1.40	1.40	1.45	1.50	1.40	1.75	1.55	**1.10**	1.15	1.415	119.4
ncf	1.60	1.70	1.60	1.50	1.60	1.45	1.65	1.70	1.40	**1.35**	1.555	131.2
ncf-p	1.45	1.35	1.35	1.35	1.40	1.40	**1.20**	1.60	1.30	1.25	1.365	115.2
ncf-c	1.50	1.70	1.45	1.30	1.40	1.45	1.50	1.50	1.30	**1.25**	1.435	121.1
grd	1.45	1.80	1.35	**1.30**	1.45	1.65	1.80	1.80	1.40	1.40	1.540	130.0
grd-g	1.45	1.55	1.25	1.30	1.45	1.25	1.65	1.50	**1.20**	**1.20**	1.380	116.5
Aver	1.444	1.569	1.362	1.331	1.431	1.388	1.606	1.606	1.256	1.250		
Relative	115.5	125.5	109.0	106.5	114.5	111.0	128.5	128.5	100.5	100		
4-orientation, SVC-rbf												
blr	**1.10**	1.35	1.20	**1.10**	1.25	**1.10**	1.80	1.85	1.30	1.35	1.340	125.8
des	1.00	1.25	0.95	**0.85**	0.95	**0.85**	1.25	1.50	1.10	0.95	1.065	100
mul	1.00	1.10	1.20	1.10	**0.95**	1.00	1.60	1.70	1.15	1.00	1.180	110.8
ncf	**1.20**	1.40	1.30	1.30	1.30	1.35	**1.20**	1.55	1.40	1.25	1.325	124.4
ncf-p	1.10	1.30	**1.05**	1.20	1.10	1.15	1.15	1.40	1.25	1.15	1.185	111.3
ncf-c	**1.00**	1.10	1.20	1.10	1.10	1.20	1.10	1.55	1.30	1.40	1.205	113.1
grd	1.25	1.55	1.25	**1.20**	**1.20**	1.30	1.80	1.65	1.50	1.35	1.405	131.9
grd-g	1.15	1.35	**1.05**	1.15	1.15	1.10	1.45	1.75	1.30	1.25	1.270	119.2
Aver	1.100	1.300	1.150	1.125	1.125	1.131	1.419	1.619	1.288	1.212		
Relative	100	118.2	104.5	102.3	102.3	102.8	129.0	147.2	117.1	110.2		
8-direction, PC												
blr	1.35	1.35	**1.05**	1.25	1.35	1.30	1.70	1.60	**1.05**	1.10	1.310	121.8
des	0.95	1.45	1.05	0.95	**0.90**	**0.90**	1.15	1.20	1.15	1.05	1.075	100
mul	1.30	1.30	1.15	1.20	1.35	1.20	1.65	1.50	1.10	1.05	1.280	119.1
ncf	1.35	1.55	1.30	1.50	1.45	1.55	1.40	1.80	1.45	**1.25**	1.460	135.8
ncf-p	1.15	1.30	1.15	1.35	**1.10**	1.35	1.40	1.60	1.20	1.25	1.285	119.5
ncf-c	**1.30**	1.55	1.35	**1.30**	1.35	1.40	**1.30**	1.50	1.35	**1.30**	1.370	127.4
grd	1.20	1.50	**1.10**	1.15	1.25	1.15	1.55	1.75	1.15	1.20	1.300	120.9
grd-g	1.25	1.25	**1.10**	1.20	1.30	1.20	1.45	1.50	1.20	1.25	1.270	118.1
Aver	1.231	1.406	1.156	1.237	1.256	1.256	1.450	1.556	1.206	1.181		
Relative	106.5	121.6	100	107.0	108.7	108.7	125.4	134.6	104.3	102.2		
8-direction, SVC-rbf												
blr	1.15	1.35	1.10	1.15	1.10	1.20	1.40	1.60	1.10	**1.05**	1.220	120.8
des	0.90	1.10	1.05	0.90	**0.85**	0.90	1.10	1.40	1.00	0.90	1.010	100
mul	1.10	1.15	1.10	1.15	1.10	**0.95**	1.45	1.35	1.05	1.00	1.140	112.9
ncf	1.15	1.15	**1.00**	1.15	1.15	1.20	1.15	1.50	1.25	1.10	1.180	116.8
ncf-p	1.10	1.10	**1.00**	1.10	1.05	1.10	1.15	1.45	1.15	1.10	1.130	111.9
ncf-c	**1.10**	**1.10**	1.15	**1.10**	**1.10**	1.20	1.30	1.40	1.20	**1.10**	1.175	116.3
grd	**1.10**	1.40	1.15	1.15	1.15	1.15	1.45	1.65	1.25	1.15	1.260	124.8
grd-g	1.00	1.30	1.05	**0.90**	1.05	0.95	1.30	1.50	1.10	1.15	1.130	111.9
Aver	1.075	1.206	1.075	1.075	1.069	1.081	1.288	1.481	1.138	1.069		
Relative	100.6	112.8	100.6	100.6	100	101.1	120.5	138.5	106.5	100		
Overall	105.7	119.5	103.5	104.1	106.4	105.9	125.9	137.2	107.1	103.1		

normalization functions, F9, F0, F2, and F4 perform best in the four cases (variations of feature resolution and classifier), respectively. F2 and F9 also compete with the best in all the four cases. F0 and F4 compete with the best except in the case of 4-orientation feature with PC. In addition, F2 and F5 compete with the best in all the four cases, and F8 compete with the best in three cases. According to the overall RPM (averaged over four cases) shown at the bottom of Table 2, seven normalization functions F0, F2, F3, F4, F5, F8, and F9 perform comparably well. The three inferior functions are F1 and F7 that preserve aspect ratio and the nonlinear normalization F6.

Comparing the feature vectors, the deslant chaincode feature (*des*) performs best in all the four cases. This is due to the fact that the slant of the digit samples is significant and slant normalization can largely reduce the within-class shape variation. In respect of the RPM, no other features compete with the deslant feature except that the enhanced chaincode feature (*mul*), the enhanced NCFE feature (*ncf-p*), the continuous NCFE feature (*ncf-c*) and the gray-scale gradient feature (*grd-g*) perform fairly well. It is evident that the enhanced chaincode feature (*mul*) outperforms the baseline chaincode feature (*blr*), the improved NCFE features (*ncf-p* and *ncf-c*) outperform the baseline NCFE feature (*ncf*), and the gray-scale gradient feature outperforms the binary gradient feature (*grd*).

Comparing the 4-orientation features and 8-direction features, it is clear that the performance of 8-direction features is superior, at the cost of higher dimensionality. It is also meaningful to compare the results on CENPARMI dataset with those reported in the literature. A number of high accuracies were collected in [2], wherein a record of high accuracy, 98.90% was reported. In this study, by varying normalization and feature extraction, we achieve a new record of 99.15% on the test dataset of CEPARMI database.

4.2. Results on NIST database

As introduced in Ref. [1], we have made an experiment database from the CD of NIST Special Database 19 (SD19). The training dataset is composed of the digit samples of 600 writers (no.0–399 and no.2100–2299), and the test dataset is composed of the digit samples of 400 writers (no. 500–699 and no.2400–2599). The total numbers of samples in the training dataset and test dataset are 66,274 and 45,398, respectively. The images were scanned at 300DPI. Some samples of the test data are shown in Fig. 10.

The character images of NIST database have lot of contour noises, so smoothing is necessary for NCFE. Two classifiers are used in the recognition of NIST database: PC and discriminative learning quadratic discriminant function (DLQDF). The DLQDF is isomorphic to the MQDF2 of Kimura et al. [15] but the parameters are adjusted with aim to minimize the classification error [25]. The PC uses 70 principal components of the pattern space. The DLQDF uses 40 principal eigenvectors for each class and in discriminative training, the parameters are regularized by a penalty of quadratic distances of within-class samples. The DLQDF was shown to perform comparably well to PC.

On training the classifiers, the error rates on the test dataset are listed in Table 3. The table has four blocks corresponding to the combination of two classifiers and two resolutions of direction feature. The average error rates of normalization functions and feature vectors are calculated and the relative performance measures (RPMs) are given. Comparing

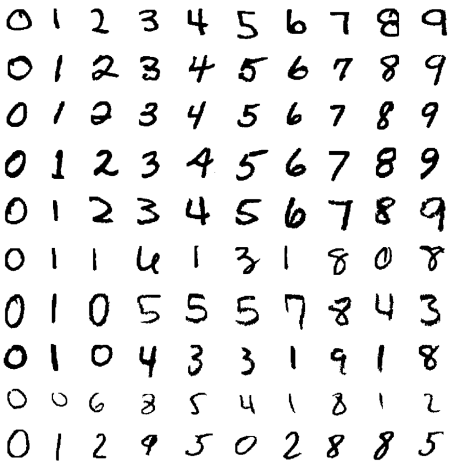

Fig. 10. Sample images of NIST database.

the normalization functions, we can see that the moment normalization F9 performs best in all the four cases in respect of the RPM and the performance of F8 is competitive. For each feature vector, the highest classification accuracy is given by either F8 or F9. Except F0 and F1, the other normalization functions perform fairly well in respect of either the case-specific RPM or the overall RPM.

Comparing the feature vectors, the deslant chaincode feature (*des*) and the gray-scale gradient feature (*grd-g*) perform best in two of the four cases, respectively. They also compete with the best in all the four cases of four blocks. The superiority of gray-scale gradient feature is due to the reliable computation and decomposition of gradient while that of deslant feature is due to the correction of image slant. Among other feature vectors, the advantage of improved NCFE features (*ncf-p* and *ncf-c*) over their baseline counterpart (*ncf*) is evident. The continuous NCFE feature (*ncf-c*) competes with the best in three of four cases in the sense that the RPM is less than 111.1. In the case of 4-orientation feature with DLQDF classification, the enhanced chaincode feature (*mul*) and the enhanced NCFE feature (*ncf-p*) also compete with the best.

To clarify the improvements of normalization functions and features, let us take the chaincode feature (*blr*) with normalization function F5 as a baseline. This baseline feature vector was used in our previous evaluation of various classifiers [1]. The accuracies of 4-orientation baseline feature given by PC and DLQDF are 99.11% and 99.14%, respectively, and those of 8-direction baseline feature are 99.27% and 99.28%, respectively. From Table 3, we can see that by varying the normalization function only, the accuracies of 4-orientation feature and 8-direction feature are promoted to 99.29% and 99.34%, respectively. On the other hand, the globally highest accuracies of 4-orientation feature and 8-direction feature are 99.36% and 99.47%, respectively.

Table 3
Error rates (%) of NIST test dataset

Method	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	Aver	Relative
4-orientation, PC												
blr	0.97	0.94	0.90	0.90	0.91	0.89	0.87	0.90	0.76	**0.71**	0.875	118.7
des	0.77	0.86	0.70	0.69	0.74	0.73	0.74	0.82	**0.64**	0.68	0.737	100
mul	0.91	0.87	0.82	0.83	0.89	0.82	0.79	0.83	0.74	**0.71**	0.821	111.4
ncf	1.01	0.99	0.99	0.98	0.97	0.90	0.93	0.85	0.83	**0.78**	0.923	125.2
ncf-p	0.91	0.89	0.88	0.89	0.87	0.87	0.84	0.81	0.75	**0.70**	0.841	114.1
ncf-c	0.92	0.91	0.87	0.87	0.91	0.88	0.88	0.76	0.70	**0.68**	0.838	113.7
grd	0.99	0.92	0.89	0.92	0.91	0.87	0.88	0.83	**0.75**	0.76	0.872	118.3
grd-g	0.85	0.83	0.79	0.81	0.82	0.80	0.79	0.74	**0.67**	0.68	0.778	105.6
Aver	0.916	0.901	0.855	0.861	0.878	0.845	0.840	0.818	0.730	0.712		
Relative	128.7	126.5	120.1	121	123.3	118.7	118.0	114.9	102.5	100		
4-orientation, DLQDF												
blr	0.91	1.03	0.89	0.87	0.85	0.86	0.94	0.97	**0.78**	**0.78**	0.888	116.7
des	0.81	0.99	0.76	0.71	0.72	0.74	0.76	1.00	**0.68**	**0.68**	0.785	103.1
mul	0.86	0.92	0.85	0.84	0.80	0.80	0.85	0.91	0.75	**0.73**	0.831	109.2
ncf	0.96	1.13	0.94	0.95	0.92	0.93	0.92	0.96	0.81	**0.80**	0.932	122.5
ncf-p	0.85	0.97	0.84	0.87	0.85	0.85	0.82	0.84	**0.73**	0.74	0.836	109.9
ncf-c	0.85	0.96	0.85	0.83	0.85	0.83	0.82	0.83	0.73	**0.68**	0.823	108.1
grd	0.90	0.97	0.87	0.85	0.82	0.81	0.92	0.92	0.76	**0.75**	0.857	112.6
grd-g	0.76	0.90	0.76	0.74	0.76	0.72	0.83	0.76	0.70	**0.68**	0.761	100
Aver	0.862	0.984	0.845	0.833	0.821	0.818	0.858	0.899	0.743	0.730		
Relative	118.1	134.8	115.8	114.1	112.5	112.1	117.5	123.2	101.8	100		
8-direction, PC												
blr	0.84	0.81	0.76	0.73	0.76	0.73	0.79	0.75	0.68	**0.66**	0.751	117.2
des	0.67	0.74	0.61	0.60	0.63	0.61	0.61	0.71	**0.59**	0.64	0.641	100
mul	0.83	0.76	0.78	0.73	0.72	0.70	0.78	0.72	**0.66**	**0.66**	0.734	114.5
ncf	0.81	0.82	0.77	0.77	0.79	0.76	0.73	0.75	0.71	**0.67**	0.758	118.3
ncf-p	0.74	0.77	0.75	0.74	0.74	0.71	0.72	0.69	0.66	**0.62**	0.714	111.4
ncf-c	0.78	0.76	0.72	0.71	0.77	0.72	0.70	0.66	0.64	**0.63**	0.709	110.6
grd	0.78	0.77	0.74	0.70	0.73	0.72	0.78	0.72	**0.64**	0.66	0.724	112.9
grd-g	0.70	0.74	0.67	0.69	0.67	0.69	0.67	0.62	**0.55**	**0.55**	0.655	102.2
Aver	0.769	0.771	0.725	0.709	0.726	0.705	0.723	0.702	0.641	0.636		
Relative	120.9	121.2	114.0	111.5	114.2	110.8	113.7	110.4	100.8	100		
8-direction, DLQDF												
blr	0.79	0.92	0.76	0.69	0.71	0.72	0.78	0.89	**0.67**	0.68	0.761	119.7
des	0.66	0.89	0.64	**0.60**	0.62	0.63	0.65	0.87	0.63	0.61	0.680	106.9
mul	0.78	0.85	0.72	0.69	0.75	0.71	0.74	0.82	**0.65**	0.68	0.739	116.2
ncf	0.74	0.92	0.76	0.72	0.72	0.73	0.77	0.79	0.69	**0.65**	0.749	117.8
ncf-p	0.72	0.87	0.74	0.70	0.73	0.67	0.70	0.74	0.66	**0.64**	0.717	112.7
ncf-c	0.70	0.79	0.70	0.67	0.71	0.68	0.69	0.72	0.66	**0.64**	0.696	109.4
grd	0.75	0.87	0.74	0.68	0.69	0.70	0.77	0.81	0.64	**0.62**	0.727	114.3
grd-g	0.69	0.74	0.64	0.62	0.65	0.64	0.69	0.62	**0.53**	0.54	0.636	100
Aver	0.729	0.856	0.713	0.671	0.698	0.685	0.724	0.783	0.641	0.633		
Relative	115.2	135.2	112.6	106.0	110.3	108.2	114.4	123.7	101.3	100		
Overall	120.7	129.4	115.6	113.1	115.1	112.5	115.9	118.1	101.6	100		

4.3. Results on Hitachi database

To test the recognition performance on Japanese writing styles, we use a database collected by Hitachi, Ltd. The samples were collected from sampling sheets and real form (insurance application, bank transaction, etc.) images and were scanned at 200DPI. The training dataset contains 164, 158 samples, which we use to train a PC with 80 principal components. From the test dataset, some mis-tagged samples were removed and some ambiguous samples are singled out for separate testing. The rest clean samples form a dataset Test-1, while the ambiguous samples form another dataset

Fig. 11. Sample images of Hitachi dataset Test-2.

Test-2. The numbers of samples of Test-1 and Test-2 are 163,223 and 2,801, respectively. Some samples of Test-2 are shown in Fig. 11.

For NCFE on Hitachi database, smoothing was not performed. Since most character images have thin strokes, smoothing may damage the character structure. The error rates of 4-orientation features and 8-direction features on two test datasets are listed in Table 4. In the results of dataset Test-1, the globally lowest error rate is highlighted, whereas for Test-2, the lowest error rate of each feature vector is highlighted. As for CENPARMI and NIST databases, the average error rates of normalization functions and feature vectors are calculate and the RPMs are given.

On either Test-1 or Test-2, the comparison of normalization functions manifests the superiority of moment normalization functions F8 and F9. On Test-1, the linear normalization functions F2, F3, and F5 also give competitive performance to the best. In comparing the feature vectors, it is shown that the gray-scale gradient feature (*grd-g*) performs best in three of four cases and competes with the best in another case. The improved NCFE features *ncf-p* and *ncf-c* compete with the best in three of four cases. It is noteworthy that on Hitachi database, the performance of deslant chaincode feature (*des*) is inferior while for CENPARMI and NIST databases, its performance is the best. This is because in Japanese writing style, the slant of digit images is not considerable. In this case, the deformation by slant normalization enlarges the within-class shape variation since the slant is estimated individually rather than from field context.

A noteworthy observation is that on Test-2 of Hitachi database, the accuracies of 4-orientation features and 8-direction features are comparable, while for other datasets, the accuracies of 8-direction features are significantly higher. Overall, 8-direction features outperform 4-orientation features in most cases and the performance is comparable in rare case.

To get an overview of performance, the overall RPMs of normalization functions and feature vectors on three databases are shown in Tables 5 and 6, respectively. We can see that the comparison of methods shows different tendencies for different styles of data. Regarding the normalization functions, the moment normalization functions F8 and F9 perform best for all the three databases, while for CENPARMI data, the linear normalization functions F0, F2, F3, F4, and F5 also compete with the best. F2, F3, F4, and F5 also perform fairly well for NIST and Hitachi databases. The aspect ratio preserving normalization functions F1 and F7 give inferior performance for all the three databases.

Comparing the feature vectors, the deslant chaincode feature (*des*) perform best for CENPARMI and NIST databases while the gray-scale gradient feature (*grd-g*) perform best for NIST and Hitachi databases. *grd-g* performs fairly well on CEPARMI data while the performance of *des* on Hitachi data is poor. The improved NCFE features *ncf-p* and *ncf-c* compete with the best on Hitachi data and perform fairly well on CENPARMI and NIST data. Considering that slant normalization is a general strategy for shape recovery and is applicable to NCFE and gradient feature as well, we can conclude that the gray-scale gradient feature is the best for all the three databases. The slant normalization is optional to use, and the slant is estimated either from individual character or from field context.

4.4. Speed of feature extraction

The computation time of feature extraction is a concern in practical recognition systems. Particularly, in integrated segmentation and recognition of character strings, not only the correctly segmented character patterns are recognized, but also many noncharacter patterns generated in trial segmentation are processed through the recognizer. For real-time application, the computation time of both feature extraction and classification should be considered.

We measured the CPU time of feature extraction (including normalization) on the NIST test dataset. The experiments were implemented on personal computer with CPU Pentium-4-1.7 GHz. The average CPU times of eight feature vectors combined with ten normalization functions are plotted in Fig. 12. The two figures show similar tendency except that 8-direction features cost more time than 4-orientation features because for 8-direction feature extraction, blurring is performed eight times other than four times.

To extract the same feature vector, the CPU time varies according to the normalization function. This is because the actual size (occupied area in the standard plane) of normalized image depends on normalization. For example, F0 fills both the two dimensions of standard plane while moment normalization rarely fills a dimension. The computation time of blurring depends on the number of nonzero

Table 4
Error rates (%) of Hitachi datasets (classifier: PC)

Method	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	Aver	Relative
Test1, 4-orientation												
blr	0.16	0.19	0.13	0.13	0.15	0.13	0.18	0.18	0.12	0.13	0.150	119.0
des	0.18	0.22	0.17	0.16	0.18	0.17	0.19	0.21	0.15	0.16	0.179	142.1
mul	0.14	0.18	0.13	0.13	0.14	0.13	0.16	0.16	0.12	0.12	0.141	112.0
ncf	0.14	0.16	0.13	0.13	0.15	0.14	0.17	0.14	0.14	0.13	0.143	113.6
ncf-p	0.13	0.15	0.13	0.12	0.13	0.13	0.14	0.13	0.13	0.12	0.131	104.0
ncf-c	0.13	0.14	0.12	0.12	0.13	0.12	0.15	0.14	0.12	0.12	0.129	102.4
grd	0.16	0.18	0.13	0.14	0.15	0.13	0.18	0.17	0.13	0.13	0.150	119.0
grd-g	0.13	0.15	0.12	0.11	0.12	0.12	0.16	0.14	**0.10**	0.11	0.126	100
Aver	0.146	0.171	0.133	0.130	0.144	0.134	0.166	0.159	0.126	0.128		
Relative	115.9	135.7	105.6	103.2	114.3	106.3	131.7	126.2	100	101.6		
Test1, 8-direction												
blr	0.12	0.15	0.11	0.10	0.11	0.10	0.14	0.15	0.10	0.11	0.119	132.2
des	0.15	0.18	0.14	0.13	0.14	0.13	0.15	0.17	0.13	0.14	0.146	162.2
mul	0.11	0.14	0.11	0.10	0.10	0.09	0.14	0.14	0.10	0.10	0.113	125.6
ncf	0.12	0.13	0.11	0.11	0.12	0.11	0.12	0.12	0.10	0.10	0.114	126.7
ncf-p	0.11	0.13	0.10	0.10	0.10	0.10	0.12	0.12	0.09	0.09	0.106	117.8
ncf-c	0.10	0.12	0.10	0.10	0.10	0.10	0.11	0.11	0.09	0.09	0.102	113.3
grd	0.11	0.13	0.10	0.09	0.10	0.10	0.13	0.15	0.10	0.10	0.111	123.3
grd-g	0.09	0.11	0.09	**0.08**	**0.08**	**0.08**	0.11	0.10	**0.08**	**0.08**	0.090	100
Aver	0.114	0.136	0.107	0.101	0.106	0.101	0.128	0.133	0.099	0.101		
Relative	115.2	137.4	108.1	102.0	107.1	102.0	129.3	134.3	100	102.0		
Test2, 4-orientation												
blr	26.17	27.20	25.53	25.28	25.28	25.28	24.13	24.13	22.38	**22.13**	24.751	107.0
des	25.74	28.63	26.13	25.81	26.67	26.31	25.60	26.81	23.13	**23.06**	25.789	111.5
mul	24.78	26.31	25.06	24.85	24.35	24.74	23.35	23.53	22.17	**21.64**	24.078	104.1
ncf	24.63	25.21	24.88	24.42	24.35	25.03	22.67	23.35	21.10	**20.96**	23.660	102.3
ncf-p	24.38	24.31	23.78	23.99	24.24	24.67	21.96	22.78	20.92	**20.39**	23.142	100.0
ncf-c	23.92	24.81	24.46	23.53	23.92	23.92	22.56	22.56	21.03	**20.71**	23.147	100.0
grd	25.38	26.70	24.96	24.81	25.10	24.92	24.13	24.38	21.60	**21.03**	24.301	105.0
grd-g	25.24	25.74	23.71	23.60	24.21	23.46	22.85	22.35	**19.99**	20.24	23.139	100
Aver	25.030	26.114	24.814	24.536	24.636	24.791	23.406	23.736	21.540	21.270		
Relative	117.7	122.8	116.7	115.4	115.8	116.6	110.0	111.6	101.3	100		
Test2, 8-direction												
blr	27.13	26.78	25.63	24.88	25.96	25.35	23.56	23.10	22.38	**21.96**	24.673	106.6
des	26.31	27.70	25.71	26.70	26.88	25.78	24.92	24.88	**23.24**	23.35	25.547	110.3
mul	25.85	25.60	24.85	24.60	24.67	24.78	22.85	22.78	21.78	**21.53**	23.929	103.3
ncf	24.92	25.28	24.71	24.49	25.31	24.63	22.13	23.21	20.89	**20.60**	23.617	102.0
ncf-p	24.88	24.63	24.21	24.42	24.53	24.17	21.78	22.81	20.14	**19.99**	23.156	100
ncf-c	24.63	24.85	23.99	23.88	24.46	24.31	21.67	22.31	20.89	**20.64**	23.163	100.0
grd	26.10	25.85	25.03	24.56	25.28	24.81	22.85	22.88	21.71	**21.60**	24.067	103.9
grd-g	25.53	25.63	24.06	23.85	24.92	24.42	22.63	21.71	**19.96**	20.17	23.288	100.6
Aver	25.669	25.790	24.774	24.672	25.251	24.781	22.799	22.960	21.374	21.230		
Relative	120.9	121.5	116.7	116.2	118.9	116.7	107.4	108.1	100.7	100		
Overall	117.6	128.9	111.2	109.5	115.8	111.5	120.2	118.0	100.5	100.8		

Table 5
Overall relative performance measures of normalization functions

	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9
CENPAR	105.7	119.5	103.5	104.1	106.4	105.9	125.9	137.2	107.1	103.1
NIST	120.7	129.4	115.6	113.1	115.1	112.5	115.9	118.1	101.6	100
Hitachi	117.6	128.9	111.2	109.5	115.8	111.5	120.2	118.0	100.5	100.8

elements on the feature plane. Generally, the feature planes of larger image have more nonzero elements. In nonlinear normalization (F6), the computation of line density histograms costs some time.

In comparing the CPU times of feature vectors, we can see that the gradient feature is the most time-consuming while the NCFE feature is the most time efficient. In NCFE, the time of generating normalized image is saved. The CPU time of gradient feature is mainly consumed in gradient computation and decomposition. It is reasonable that extracting gradient feature from gray-scale image is more expensive than from binary image due to the generation of pseudo-gray normalized image. Similarly, the enhanced NCFE feature (*ncf-p*) and the continuous NCFE feature (*ncf-c*) are more computationally expensive than the discrete NCFE feature. The CPU time of chaincode feature (*blr*) is intermediate, and the enhanced feature (*mul*) and deslant feature (*des*) are more expensive than it.

Table 6
Overall relative performance measures of feature vectors

	CENPAR	NIST	Hitachi
blr	124.2	118.1	116.2
des	100	102.5	131.5
mul	115.6	112.8	111.3
ncf	127.1	121.0	111.1
ncf-p	114.5	112.0	105.5
ncf-c	119.5	110.5	103.9
grd	126.9	114.5	112.8
grd-g	116.4	102.0	100.2

5. Conclusion

We compared the performances of ten normalization functions and eight feature vectors in handwritten digit recognition on large databases of distinct sources. The normalization functions implement dimension-based linear/nonlinear normalization and moment-based normalization with varying aspect ratio mapping. The recognition results show that the moment normalization function F8 and F9 mostly yield highest classification accuracies. The aspect ratio mapping is influential to the recognition performance for both dimension-based and moment-based normalization. Either preserving the aspect ratio or forcing the aspect ratio to one is not a good choice. Comparing the feature vectors, the deslant chaincode feature performs best for CENPARMI and NIST data while the gradient feature from (pseudo) gray-scale normalized image performs best for NIST and Hitachi data. The improved NCFE features (enhanced NCFE and continuous NCFE) mostly perform well. The NCFE is very time efficient, while the superior performance of gradient feature is traded off by the intensive computation. However, the gradient feature is directly applicable to gray-scale images whereas other features are restricted to binary images. The reported results provide useful insights for selecting right implementation options in developing recognition systems.

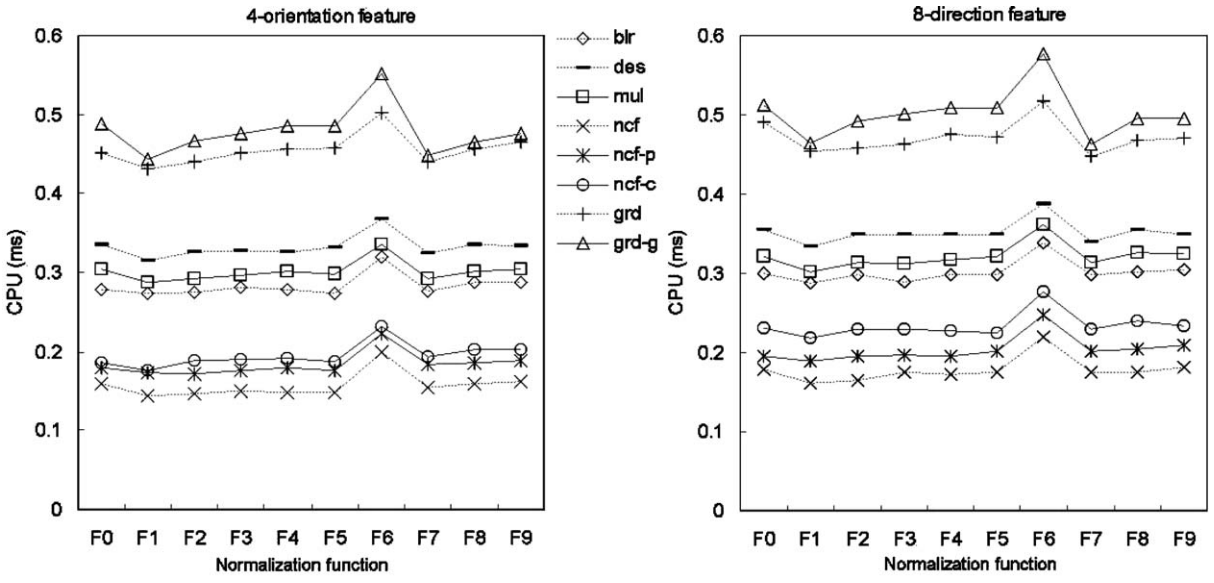

Fig. 12. CPU time (ms) of feature extraction on NIST test dataset.

References

[1] C.-L. Liu, H. Sako, H. Fujisawa, Performance evaluation of pattern classifiers for handwritten character recognition, Int. J. Document Anal. Recognition 4 (3) (2002) 191–204.

[2] C.-L. Liu, K. Nakashima, H. Sako, H. Fujisawa, Handwritten digit recognition: benchmarking of state-of-the-art techniques, Pattern Recognition, 36 (10) (2003) 2271–2285.

[3] C.-L. Liu, M. Koga, H. Sako, H. Fujisawa, Aspect ratio adaptive normalization for handwritten character recognition, in: T. Tan, Y. Shi, W. Gao (Eds.), Advances in Multimodal Interfaces—ICMI 2000, Lecture Notes in Computer Science, Vol. 1948, Springer, Berlin, 2000, pp. 418–425.

[4] M. Hamanaka, K. Yamada, J. Tsukumo, Normalization-cooperated feature extraction method for handprinted Kanji character recognition, Proceedings of the Third International Workshop on Frontiers of Handwriting Recognition, Buffalo, NY, 1993, pp. 343–348.

[5] A. Gudessen, Quantitative analysis of preprocessing techniques for the recognition of handprinted characters, Pattern Recognition 8 (1976) 219–227.

[6] G. Srikantan, D.-S. Lee, J.T. Favata, Comparison of normalization methods for character recognition, Proceedings of the Third International Conference Document Analysis and Recognition, Montreal, Canada, 1995, pp. 719–722.

[7] J.J. de Oliveira Jr., L.R. Veloso, J.M. de Carvalho, Interpolation/decimation scheme applied to size normalization of characters images, Proceedings of the 15th International Conference Pattern Recognition, Barcelona, Spain, Vol. 2, 2000, pp. 577–580.

[8] G. Nagy, N. Tuong, Normalization techniques for handprinted numerals, Commun. ACM 13 (8) (1970) 475–481.

[9] R.G. Casey, Moment normalization of handprinted character, IBM J. Res. Dev. 14 (1970) 548–557.

[10] J. Tsukumo, H. Tanaka, Classification of handprinted Chinese characters using nonlinear normalization and correlation methods, Proceedings of the Ninth International Conference on Pattern Recognition, Rome, Italy, 1988, pp. 168–171.

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

[12] A. de S. Britto Jr., et al., Improvement in handwritten numeral string recognition by slant normalization and contextual information, Proceedings of the Seventh International Workshop on Frontiers of Handwriting Recognition, Amsterdam, The Netherlands, 2000, pp. 323–332.

[13] O.D. Trier, A.K. Jain, T. Taxt, Feature extraction methods for character recognition—a survey, Pattern Recognition 29 (4) (1996) 641–662.

[14] M. Yasuda, H. Fujisawa, An improvement of correlation method for character recognition, Trans. IEICE Japan J62-D (3) (1979) 217–224.

[15] 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) (1987) 149–153.

[16] S. Tsuruoka, et al., Handwritten Kanji and hiragana character recognition using weighted direction index histogram method, Trans. IEICE Japan J70-D (7) (1987) 1390–1397.

[17] J.T. Favata, G. Srikantan, S.N. Srihari, Handprinted character/digit recognition using a multiple feature/resolution philosophy, Proceedings of the Fourth International Workshop on Frontiers of Handwriting Recognition, Taipei, 1994, pp. 57–66.

[18] G. Srikantan, S.W. Lam, S.N. Srihari, Gradient-based contour encoder for character recognition, Pattern Recognition 29 (7) (1996) 1147–1160.

[19] M. Shi, Y. Fujisawa, T. Wakabayashi, F. Kimura, Handwritten numeral recognition using gradient and curvature of gray scale image, Pattern Recognition 35 (10) (2002) 2051–2059.

[20] C.-L. Liu, Y-J. Liu, R-W. Dai, Preprocessing and statistical/structural feature extraction for handwritten numeral recognition, in: A.C. Downton, S. Impedovo (Eds.), Progress of Handwriting Recognition, World Scientific, Singapore, 1997, pp. 161–168.

[21] V. Vapnik, The Nature of Statistical Learning Theory, Springer, New Work, 1995.

[22] C.J.C. Burges, A tutorial on support vector machines for pattern recognition, Knowledge Discovery Data Mining 2 (2) (1998) 1–43.

[23] J. Schürmann, Pattern Classification: A Unified View of Statistical and Neural Approaches, Wiley Interscience, New York, 1996.

[24] U. Kreßel, J. Schürmann, Pattern classification techniques based on function approximation, in: H. Bunke, P.S.P. Wang (Eds.), Handbook of Character Recognition and Document Image Analysis, World Scientific, Singapore, 1997, pp. 49–78.

[25] C.-L. Liu, H. Sako, H. Fujisawa, Learning quadratic discriminant function for handwritten character recognition, Proceedings of the 16th International Conference Pattern Recognition, Quebec, Canada, Vol.4, 2002, pp. 44–47.

[26] C.Y. Suen, et al., Computer recognition of unconstrained handwritten numerals, Proc. IEEE 80 (7) (1992) 1162–1180.

[27] P.J. Grother, NIST special database 19: handprinted forms and characters database, Technical report and CDROM, 1995.

[28] S.-W. Lee, Multilayer cluster neural network for totally unconstrained handwritten numeral recognition, Neural Networks 8 (5) (1995) 783–792.

[29] A. Kawamura, et al., On-line recognition of freely handwritten Japanese characters using directional feature densities, Proceedings of the 11th International Conference on Pattern Recognition, The Hague, Vol. 2, 1992, pp. 183–186.

[30] T. Wakabayashi, S. Tsuruoka, F. Kimura, Y. Miyake, On the size and variable transformation of feature vector for handwritten character recognition, Trans. IEICE Japan J76-D-II (12) (1993) 2495–2503.

[31] R.V.D. Heiden, F.C.A. Gren, The Box-Cox metric for nearest neighbor classification improvement, Pattern Recognition 30 (2) (1997) 273–279.

About the Author—CHENG-LIN LIU received the B.S. degree in electronic engineering from Wuhan University, the M.E. degree in electronic engineering from Beijing Polytechnic University, the Ph.D. degree in pattern recognition and artificial intelligence from the Institute

of Automation, Chinese Academy of Sciences, in 1989, 1992 and 1995, respectively. He was a postdoctor fellow in Korea Advanced Institute of Science and Technology (KAIST) and later in Tokyo University of Agriculture and Technology from March 1996 to March 1999. Since then, he has been a research staff and later a senior researcher at the Central Research Laboratory, Hitachi, Ltd. His current work is mainly in developing classification algorithms for OCR applications. His research interests include pattern recognition, artificial intelligence, image processing, neural networks, machine learning, and especially the applications to character recognition and document analysis.

About the Author—KAZUKI NAKASHIMA received the B.E. degree from Nagoya Institute of Technology, Japan, in 1989, and the M.E. degree in Information Engineering from Nagoya University, Japan, in 1991. Since 1991, he has been working for Central Research Laboratory, Hitachi, Ltd., Japan. He has been working in research and development on pattern recognition and image processing. He is a member of the Institute of Electronics, Information and Communication Engineers (IEICE) of Japan.

About the Author—HIROSHI SAKO received the B.E. and M.E. degrees in mechanical engineering from Waseda University, Tokyo, Japan, in 1975 and 1977, respectively. In 1992, he received the Dr. Eng. degree in computer science from the University of Tokyo. From 1977 to 1991, he worked in the field of industrial machine vision at the Central Research Laboratory of Hitachi, Ltd., Tokyo, Japan (HCRL). From 1992 to 1995, he was a senior research scientist at Hitachi Dublin Laboratory, Ireland, where he did research in facial and hand gesture recognition. Since 1996, he has been with the HCRL, where he directs a research group of image recognition and character recognition. Currently, he is a chief researcher there. As concurrent posts, he has been a visiting professor at Japan Advanced Institute of Science and Technology, Hokuriku (Postgraduate University) since 1998, and a visiting lecturer at Hosei University since 2003, and from 1998 to 2003, he was a visiting lecturer at Nihon University. Dr. Sako was a recipient of the 1988 Best Paper Award from the Institute of Electronics, Information, and Communication Engineers (IEICE) of Japan , and one of the recipients of the Industrial Paper Awards from the 12th ICPR, Jerusalem, Israel, 1994. He is a senior member of the IEEE, and a member of IEICE, JSAI (the Japanese Society of Artificial Intelligence) and IPSJ (the Information Processing Society of Japan).

About the Author—HIROMICHI FUJISAWA received the B.E., M.E. and Doctor of Engineering degrees in Electrical Engineering from Waseda University, Tokyo, in 1969, 1971, and 1975, respectively. He joined Central Research Laboratory, Hitachi, Ltd. in 1974. He has engaged in research and development work on character recognition, document understanding including mailpiece address recognition and forms processing, and document retrieval. Currently, he is a Senior Chief Researcher at Central Research Laboratory. From 1980 through 1981, he was a visiting scientist at Computer Science Department of Carnegie Mellon University, Pittsburgh. Besides working at Hitachi, he has been a visiting lecturer at Waseda University (1985 to 1997) and at Kogakuin University, Tokyo (1998-present). He is an IAPR Fellow, IEEE Fellow, and a member of ACM, AAAI, Information Processing Society of Japan (IPSJ), and Institute for Electronics, Information and Communication Engineers (IEICE), Japan.