2013 12th International Conference on Document Analysis and Recognition

Moment-based Character-normalization Methods using a Contour Image Combined with an Original Image

Toshinori Miyoshi, Takeshi Nagasaki, and Hiroshi Shinjo
Central Research Laboratory, Hitachi, Ltd.
1-280 Higashi-Koigakubo, Kokubunji-shi, Tokyo 185-8601, Japan
E-mail: {toshinori.miyoshi.pd, takeshi.nagasaki.sk, hiroshi.shinjo.tz}@hitachi.com

***Abstract*—Moment-based character-normalization methods are known to improve character-recognition accuracy. These methods use the moments of an input image, which has two dimensions because of the thickness of its stroke lines, to estimate transformation parameters, whereas character is essentially composed of one dimensional stroke lines. This implies that these methods overestimate the moments of the thick parts of character strokes. To solve this problem, moment-based normalization methods, which use the moments of a contour image of a character combined with the input image of that character, are proposed. To extract the contours of character strokes, two methods, chaincode contour (CC) and gradient contour (GC), are used. Character-recognition experiments on two printed-character databases and on two handwritten-character databases show that the character-recognition accuracies of the proposed methods are comparable to or significantly higher than those of conventional methods. In particular, the proposed methods are more effective for printed-character recognition.**

I. Introduction

In regard to character-recognition systems, normalization is one of the most important preprocessing operations. One purpose of normalization is to generate a normalized image of pre-defined size from an input image. Another is to reduce within-class variations of shape and position of characters due to diverse styles of input images; that is, shape and position of characters in an input image vary in accordance with factors such as scanner maker, paper quality, font, handwriting style, and writing tool. Nonlinear normalization (NN) [1] [2] and moment-based normalization [3] [4] are among the most popular methods because they significantly improve recognition accuracy [5] [6].

NN methods reduce character-shape variations by equalizing local line density of character strokes. To calculate local line density, Tsukumo's method [2] and Yamada's method [1] are widely applied. NN methods have been proven to significantly improve recognition accuracy [5]. They, however, often generate excessively deformed normalized character images owing to local transformation, namely, equalization of local line densities [4]. Moreover, the calculations of local line density make the computational cost of NN methods high.

In contrast to NN methods, moment-based normalization methods transform an input image by using global parameters estimated from the moments of an input image, and the resulting transformations are smooth. Moment-based methods include moment normalization (MN) [3], bi-moment normalization (BN) [4], centroid-boundary alignment (CBA) [4], and modified centroid-boundary alignment (MCBA) [7].

Conventional moment-based methods, however, use the moments of an input image, which has two dimensions because of the thickness of its stroke lines, to estimate the transformation parameters of a normalization function, whereas a character is composed of one-dimensional stroke lines. This implies that conventional methods overestimate the moments of thick parts of character strokes; however, in many cases, thickness of stroke lines is not essential for identifying or discriminating characters.

To solve this problem concerning overestimation of the moments, in the present study, moment-based normalization methods, which use the moments of contours of character strokes instead of an input image itself to estimate transformation parameters, are proposed. To extract the contours of the character strokes from an input image, two methods are proposed: chaincode contour (CC) and gradient contour (GC). Other methods that use the moments of contours extracted by gradients have already been reported by the authors in [8] [9]. One of the most important improvements to these reported methods is the use of the moments of a contour image combined with an input image instead of the moments of the contour image itself. This improvement makes the proposed methods more effective for both printed character recognition and handwritten character recognition.

Normalization methods and feature-extraction methods are reviewed and compared in [10] [11] [5]. Overall descriptions and problems concerning character-recognition systems are given, for example, in [12] [13] [14] [15] [16].

The general normalization procedure and conventional normalization methods are reviewed in Section II, and the proposed methods are described in Section III. The results of character-recognition experiments with both kinds of methods are presented in Section IV.

II. Character normalization

The procedure for character normalization is overviewed first. In Section II-A, the general normalization procedure is described. In Section II-B, MN, which is one of the most fundamental moment-based methods, is described as an example of a normalization method. Descriptions of other moment-based methods, namely, BN, CBA and MCBA, are found in [4] [7].

A. Normalization function

Normalization is performed by a normalization function, which maps a plane of an input image onto a normalized plane of pre-determined size. The normalized plane is usually given as a square, i.e., $[0, L] \times [0, L]$. Hereafter, the intensity

1520-5363/13 $26.00 © 2013 IEEE
DOI 10.1109/ICDAR.2013.213

function of an input image is denoted by $f(x, y)$, and that of the normalized image generated from the input image is denoted by $f'(x', y')$.

The normalization function is written as

$$h(x, y): [0, W] \times [0, H] \to [0, L] \times [0, L]. \quad (1)$$

This function maps input plane $[0, W] \times [0, H]$ (defined as the minimum bounding box that contains all the strokes (black pixels) of an input image) onto normalized plane $[0, L] \times [0, L]$. It is also written with functions $u(x, y)$ and $v(x, y)$ as

$$h(x, y) = (u(x, y),\ v(x, y)). \quad (2)$$

Since character images should not be distorted by normalization, functions $u(x, y)$ and $v(x, y)$ should satisfy $\partial_x u(x, y) > 0$ and $\partial_y v(x, y) > 0$, respectively. In case of one-dimensional normalization methods, normalization function h is given as

$$h(x, y) = (u(x), v(y)). \quad (3)$$

Functions $u(x, y)$ and $v(x, y)$ are used to calculate $f'(x', y')$ from the intensity function of an input image, $f(x, y)$, according to the following equations:

$$f'(x', y') = f(x, y), \quad (4)$$
$$x' = u(x, y), \quad (5)$$
$$y' = v(x, y). \quad (6)$$

To generate a normalized image using equations (4) to (6), pixel values need to be calculated by interpolation. Pixel values can be interpolated by two methods: forward mapping or backward mapping [12] [5]. Forward mapping is used in the character-recognition experiments described in Section IV.

B. Moment normalization (MN)

MN is a simplification of Casey's method [3]; that is, it excludes slant correction. It estimates transformation parameters from centroid (x_c, y_c) and the second-order moments of the intensity of an input image; centroid (x_c, y_c) is given as

$$x_c = m_{10}/m_{00}, \quad y_c = m_{01}/m_{00}, \quad (7)$$

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

The center of a valid stroke area is given as centroid (x_c, y_c). The width and height of the valid stroke area are given as

$$\delta_x = \alpha \sqrt{\mu_{20}/m_{00}}, \quad \delta_y = \alpha \sqrt{\mu_{02}/m_{00}}, \quad (8)$$

where α is a positive number, and the second-order moments μ_{20} and μ_{02} are given as

$$\mu_{20} = \sum_x \sum_y (x - x_c)^2 f(x, y), \quad (9)$$
$$\mu_{02} = \sum_x \sum_y (y - y_c)^2 f(x, y). \quad (10)$$

The valid stroke area

$$[x_c - \delta_x/2,\ x_c + \delta_x/2] \times \left[y_c - \delta_y/2,\ y_c + \delta_y/2\right] \quad (11)$$

is linearly scaled onto a normalized plane, $[0, L] \times [0, L]$. That is, the normalization function of MN is given as

$$u(x) = L(x - x_c)/\delta_x + L/2, \quad (12)$$
$$v(y) = L(y - y_c)/\delta_y + L/2. \quad (13)$$

III. Proposed normalization methods

Conventional moment-based normalization methods estimate transformation parameters from the moments of an input image, which has two dimensions because of the thickness of the character's stroke lines. However, a character is, in essence, composed of one-dimensional stroke lines, implying that conventional moment-based methods overestimate the moments of the thick parts of character strokes.

To solve this problem concerning overestimation of moments, twelve moment-based normalization methods (which use the moments of the contours of character strokes) are proposed. The following two methods are applied for extracting the contours from an input image.

A. Contour-extraction methods

Two methods for extracting contour image, the intensity function of which is denoted by $f_c(x, y)$, from an input image are proposed here: one uses chaincodes of the input image, and the other uses gradients of the input image.

1) Chaincode contour (CC): CC can only be applied to a binary input image; $f(p) = 1$ is set for black pixels, and $f(p) = 0$ is set for white pixels. To calculate $f_c(x, y)$, chaincodes of an input image are used. All pixels of the input image are scanned, and, for any pixel p that satisfies $f(p) = 1$,

$$g_{k+1}(p) = \begin{cases} 1 & \text{if } f(d_k) = 0 \text{ and } f(d_{k+1}) = 1 \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

and

$$g_{(k+2)\%8}(p) = \begin{cases} 1 & \text{if } f(d_k) = f(d_{k+1}) = 0 \\ & \quad \text{and } f(d_{(k+2)\%8}) = 1 \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

are set for $k = 0, 2, 4,$ and 6, where $d_k (k = 0, 1, \cdots, 7)$ gives the neighboring pixels of pixel p (Fig. 1). After that, $f_c(x, y)$ is given as

$$f_c(x, y) = \begin{cases} \sum_{k=0}^{7} g_k(p) & \text{if } f(p) = 1 \\ 0 & \text{otherwise} \end{cases}. \quad (16)$$

Examples of input images and their contours generated by CC are shown in Fig. 2. The images on the upper row are original input images taken from ETL9B, and the images on the lower row are their contour images extracted by CC.

2) Gradient contour (GC): Contrary to CC, GC can also be applied to a gray-scale image. To calculate $f_c(x, y)$, first, gradient vector $\vec{g} = (g_x, g_y)$ is calculated by using Sobel masks [17] [14]. Elements g_x and g_y are given as

$$g_x(p) = \left[f(d_1) + 2f(d_0) + f(d_7) - f(d_3) - 2f(d_4) - f(d_5)\right]/8,$$
$$g_y(p) = \left[f(d_1) + 2f(d_2) + f(d_3) - f(d_5) - 2f(d_6) - f(d_7)\right]/8,$$

where d_k is a neighboring pixel of pixel p (Fig. 1). The intensity function of the contours, $f_c(x, y)$, is then given as

$$f_c(x, y) = \sqrt{g_x^2(x, y) + g_y^2(x, y)}. \quad (17)$$

d_3	d_2	d_1
d_4	p	d_0
d_5	d_6	d_7

Fig. 1. **Neighboring pixels of pixel p.**

Fig. 2. **Input images taken from ETL9B (the upper row) and their contour images extracted by CC (the lower row).**

Fig. 3. **Images taken from ETL9B (the upper row) and their contour images (the lower row).**

B. Moment-based normalization using contour

Since a contour image does not have mass in the interior of stroke lines, overestimation of the moments due to thickness of stroke lines is reduced by using a contour image for calculating the moments. However, especially in the case of characters with complex structures, some parts of the contours of stroke lines may be lost. This leads to underestimation of the moments due to the lost parts. An example of this contour loss is given in Figure 3. The images on the upper row are original input images taken from ETL9B, and the images on the lower row are their contour images extracted by CC. For the bottom character image on the right hand side of Fig. 3, some parts of the contours of the stroke lines are lost.

To solve this problem, namely, partially lost contours, the intensity of a contour image, $f_c(x, y)$, is modified by combining it with that of input image $f(x, y)$ as follows:

$$f_p(x, y) = \gamma f_c(x, y) + (1 - \gamma) f(x, y), \tag{18}$$

where $0 \leq \gamma \leq 1$. The proposed versions of moment-based normalization use the moments of $f_p(x, y)$ instead of $f(x, y)$ for estimating transformation parameters. For example, the proposed version of MN is given simply by replacing $f(x, y)$ with $f_p(x, y)$ in equations (7), (9), and (10) in Section II-B.

The proposed version of MN using chaincodes for contour extraction is called "chaincode contour-moment normalization" (CCMN) hereafter, and that using gradients for gradient extraction is called "gradient contour-moment normalization" (GCMN) hereafter. Similar abbreviations are used for the proposed versions of the other moment-based methods with prefix "CC" or "GC," namely, "CCBN,", "CCCBA," "CCMCBA," "GCBN," "GCCBA," and "GCMCBA."

C. Comparison with conventional methods

CCMN is compared with MN in the following simple example. The images shown in Fig. 4 are artificially generated letter "T"s; the upper row shows "T"s whose horizontal line gradually becomes thicker from left to right, and the lower row shows "T"s whose horizontal line gradually becomes longer from left to right. Normalized images generated by MN and CCMN, respectively, from the input images shown in Fig. 4 are shown in Fig. 5 and Fig. 6.

Figure 5 shows normalized images generated by MN. It is clear that the position of the horizontal line of the "T" becomes lower as that of original image becomes thicker, and the vertical line of the "T" becomes thicker as the horizontal line of the original "T" becomes shorter. In the case of MN, it is concluded that the positions of the normalized "T" are unstable in the presence of width or thickness variations.

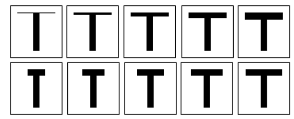
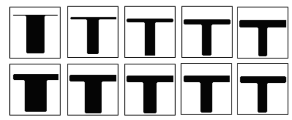


Fig. 4. **Input character images.**



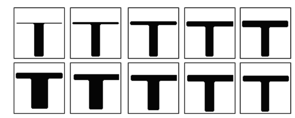


Fig. 5. **Images generated by MN.** Fig. 6. **Images generated by CCMN.**

The images in Fig. 6 are normalized images generated by CCMN by substituting $\gamma = 1.0$ into equation (18). It is clear that the positions of the "T" in the upper row of Fig. 6 are stable compared with those in the upper row of Fig. 5, and the thickness of "T" in the lower row of Fig. 6 are stable compared with those in the lower row of Fig. 5.

IV. Character-recognition experiments

Aiming to evaluate the proposed methods, character-recognition experiments were conducted. Two printed-character databases, ETL2 and CEDAR Japanese-character image databases, and two handwritten-character databases, ETL8B and ETL9B, were used. ETL2, ETL8B, and ETL9B were collected by the Electro Technical Laboratory in Japan[1], which was reorganized as the National Institute of Advanced Industrial Science and Technology (AIST). The CEDAR Japanese-character image database was created by the Center of Excellence for Document Analysis and Recognition[2]. ETL8B, ETL9B, and CEDAR databases are composed of binary character images. The ETL2 database is composed of gray-scale images; that is, each pixel of an image is represented by a 64-level gray scale. In the experiments, the images of ETL2 were binarized by unifying 0 to 31 levels and unifying 32 to 63 levels. CEDAR and ETL9B contain many characters whose contours are partially lost.

ETL2 and CEDAR are composed of Japanese *kanji*, *hiragana*, and *katakana* characters, alphanumeric characters, and symbolic characters. ETL2 is composed of five subsets, namely, ETL2-{1,2,3,4, and 5}. ETL2-5 was used as a test data set. As a training data set, character images, which belong to classes contained in ETL2-5, were extracted from ETL2-{1,2,3, and 4}. Consequently, the training data set is composed of 571 classes, and each class contains 20 samples. Similarly, the test data set is composed of 571 classes, and each class contains 20 samples (except two classes, whose SJIS codes are 96C5 and 97D6, that contain 40 samples). The CEDAR database is divided into a training data set and a testing data set. The training data set is composed of 118,414 samples of

[1] https://projects.itri.aist.go.jp/etlcdb/wordpress/?lang=en

[2] http://www.cedar.buffalo.edu/Databases/JOCR/

3,354 character classes, and the testing data set is composed of 57,571 samples with 3,319 character classes. Since some characters in CEDAR DB have identical shapes, for example, "o", "O", and "0", 3,035 categories contained in ETL9B were selected; consequently, 96,911 samples were used for training, and 46,985 samples were used for testing.

ETL8B is composed of binary images with 956 character classes: 75 *hiragana* character classes and 881 *kanji* characters classes. ETL8B contains 160 or 161 character images in each class, and the total number of samples is 153,600. ETL8B is composed of three subsets, namely, ETL8B2.{001,002,003}. The $(2k+1)$-th samples of each subset were used for training, and the $2k$-th samples of each subset were used for testing; the total numbers of test samples and training samples were 76,800 and 76,800, respectively. ETL9B is composed of binary images with 3,036 character classes: 71 *hiragana* character classes and 2,965 *kanji* characters classes. ETL9B contains 200 character images in each class, and the total number of samples is 607,200. The first 20 and the last 20 samples of each class were used for testing, and the remaining 160 samples of each class for training; the total numbers of test samples and training samples were 121,440 and 485,760 respectively. This dataset setting has been tested on [18] [19] [20].

In the character-recognition experiments, conventional normalization methods and the proposed normalization methods were compared. As for the conventional methods, linear normalization (LN) [12], nonlinear normalization (NN) [2], two-dimensional nonlinear normalization (2D-NN) [21], MN, BN, MCBA, and their P2D extensions [19] were compared. As for NN and 2D-NN, Tsukumo's method for local-line-density calculations [2] was adopted, and the improvements for adjusting the density functions of marginal and stroke areas [11] [19] were applied. As for the proposed methods, CCMN, CCBN, CCMCBA, GCMN, GCBN, GCMCBA, and their P2D extensions were compared. The pseudo two-dimensional extensions are denoted with prefix "P2D." For MN, the parameter of equation (8) was set as $\alpha = 4$, and for BN [4], $\beta = 2$ was set. For P2D extensions, parameter w_0, which controls the rigidity of shape deformations, was set to 0.75 [19]. These parameter settings were used in [4] [19]. In equation (18), $\gamma = 0.5$ was set.

For feature extraction, CNCGF (continuous normalization cooperated gradient feature) was used [22]. The size of the normalized plane were set to 64×64 pixels, i.e., $L = 64$. When feature vectors were generated, the normalized plane was partitioned into 64 sub-squares with size of 8×8; consequently, the number of dimensions of a feature vector is $8 \times 64 = 512$. For normalization, ARAN was applied to partially maintain the aspect ratio of an input character image [23]. All the measurements of features were transformed (by the so-called Box-Cox transformation [24]) by $x^{0.5}$ to improve classification performance.

For classification of ETL2, ETL8B, and ETL9B, the dimensionality of a feature vector was first reduced from 512 to 160 by Fisher discriminant analysis on the training data set. An MQDF2 classifier was used, and its principal eigenvectors for each class were 10 for ETL2 and 40 for ETL8B and ETL9B[25]. The minor eigenvalues were replaced by a class-independent constant, which was optimized by holdout cross-validation on the training data set.

For classification of characters in CEDAR Japanese-character image database, the dimensionality of feature vectors was first reduced from 512 to 160 by Fisher linear discriminant analysis using the training data set. Since some classes contain only one or a few characters, an LVQ classifier was used [15]. For LVQ training, one representative vector was used for each class.

The databases used in the experiments are summarized in Table I. The table lists type of database, number of classes, number of samples for training, and number of samples for testing. In the table, "HW" stands for handwritten, and "P" for printed.

TABLE I
SUMMARY OF DATABASES

	type	class	train	test
ETL2	P	571	11,460	11,420
CEDAR	P	3,035	96,911	46,985
ETL8B	HW	956	76,800	76,800
ETL9B	HW	3,036	485,760	121,440

The experimental results are listed in Table II. The table indicates that GC versions of the proposed moment-based normalization methods yield higher recognition accuracies than their corresponding conventional moment-based normalization methods in almost all the cases. As for the proposed methods, the recognition accuracy of each GC version is slightly higher than that of the corresponding CC version.

For printed-character recognition, the recognition accuracies of the proposed methods are significantly higher than those of the conventional methods. The table also indicates that the moment-based methods (including the conventional methods) are superior to nonlinear methods for printed character recognition. Moreover, the recognition accuracies of the nonlinear methods, NN and 2D-NN, are lower than those of LN. This result indicates that the nonlinear methods deteriorate recognition accuracies for printed character recognition.

For handwritten recognition, 2D-NN achieved the highest recognition accuracies for both ETL8B and ETL9B. However, proposed methods P2D-GCBN and P2D-GCMCBA are comparable to 2D-NN. From the viewpoint of computational complexity, the proposed methods are more advantageous than 2D-NN because moment calculations and pseudo 2D operation in the moment-based methods cost much less than local-line-density calculations and blurring operation in 2D-NN.

Recognition accuracies for different settings of parameter γ for P2D-GCBN are shown in Fig. 7. The proposed method matches the corresponding conventional method when $\gamma = 0$. This result indicates that the combination of a contour image with an input image is effective for classification of character images in CEDAR and ETL9B, which contain many characters with complex structures whose contours are partially lost.

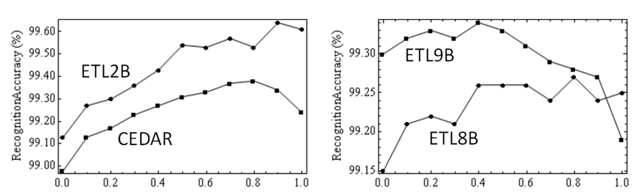


Fig. 7. Parameter γ and recognition accuracies on printed (left) and handwritten (right) databases.

TABLE II
RECOGNITION ACCURACY (%)

		ETL2	CEDAR	ETL8B	ETL9B
size normalization	LN	99.20	99.26	98.42	97.77
nonlinear methods	NN	98.94	99.16	99.20	99.21
	2D-NN	98.68	99.14	**99.29**	**99.36**
conventional moment-based methods	MN	99.18	99.12	99.08	99.14
	BN	98.96	99.21	99.04	99.16
	MCBA	99.02	99.24	99.04	99.15
	P2D-MN	99.20	98.94	99.11	99.30
	P2D-BN	99.13	98.98	99.15	99.30
	P2D-MCBA	99.52	99.16	99.18	99.30
proposed methods (CC)	CCMN	99.40	99.34	99.12	99.13
	CCBN	99.42	99.38	99.14	99.19
	CCMCBA	99.30	99.36	99.10	99.12
	P2D-CCMN	99.53	99.18	99.15	99.27
	P2D-CCBN	99.41	99.24	99.21	99.30
	P2D-CCMCBA	99.57	99.34	99.24	99.30
proposed methods (GC)	GCMN	99.41	99.36	99.03	99.15
	GCBN	99.33	**99.39**	99.14	99.17
	GCMCBA	99.35	**99.39**	99.11	99.14
	P2D-GCMN	**99.58**	99.27	99.19	99.32
	P2D-GCBN	99.54	99.31	**99.26**	**99.33**
	P2D-GCMCBA	**99.61**	99.35	**99.26**	99.29

V. SUMMARY

Twelve moment-based normalization methods, which use the moments of a contour image of a character combined with the input image of that character, are proposed. To extract the contours of character strokes, two methods, chaincode contour (CC) and gradient contour (GC), are used. Character-recognition experiments on two printed-character databases and on two handwritten-character databases show that the character-recognition accuracies of the proposed methods are comparable to or significantly higher than those of the conventional methods. In particular, the proposed methods are more effective for printed-character recognition. The combination of a contour image with the input image is effective for classification of characters in database which contains characters whose contours are partially lost.

VI. ACKNOWLEDGMENTS

The authors are grateful to the National Institute of Advanced Industrial Science and Technology (AIST) and the Center of Excellence for Document Analysis and Recognition (CEDAR) for providing the image databases.

REFERENCES

[1] H. Yamada, K. Yamamoto, and T. Saito, "A nonlinear normalization method for handprinted Kanji character recognition—line density equalization," *Pattern Recognition*, vol. 23, no. 9, pp. 1023–1029, 1990.

[2] J. Tsukumo and H. Tanaka, "Classification of handprinted Chinese characters using non-linear normalization and correlation methods," in *ICPR88*, 1988, pp. I: 168–171.

[3] R. Casey, "Moment normalization of handprinted characters," *IBM Journal of Research and Development*, vol. 14, no. 5, pp. 548–557, September 1970.

[4] C. Liu, H. Sako, and H. Fujisawa, "Handwritten Chinese character recognition: Alternatives to nonlinear normalization," in *Proceedings of the Seventh International Conference on Document Analysis and Recognition*, 2003, pp. 524–528.

[5] C. Liu, K. Nakashima, H. Sako, and H. Fujisawa, "Handwritten digit recognition: investigation of normalization and feature extraction techniques," *Pattern Recognition*, vol. 37, no. 2, pp. 265–279, February 2004.

[6] M. UMEDA, "Advances in recognition methods for handwritten kanji characters (special issue on character recognition and document understanding)," *IEICE transactions on information and systems*, vol. 79, no. 5, pp. 401–410, 1996-05-25.

[7] C.-L. Liu and K. Marukawa, "Global shape normalization for handwritten Chinese character recognition: A new method," in *Proceedings of the Ninth International Workshop on Frontiers in Handwriting Recognition*, ser. IWFHR '04. Washington, DC, USA: IEEE Computer Society, 2004, pp. 300–305.

[8] T. Miyoshi, T. Nagasaki, and H. Shinjo, "Character normalization methods using moments of gradient features and normalization cooperated feature extraction," *Proceedings of the 2009 Chinese Conference on Pattern Recognition (CCPR2009) and the First CJK Joint Workshop on Pattern Recognition (CJKPR)*, vol. 2, pp. 934–938, 2009.

[9] ——, "Character normalization methods using moments of gradient features (in Japanese)," *IEICE Tech. Rep.*, vol. 108, no. 432, pp. 187–192, 2009-02-12.

[10] C.-L. Liu, Y.-J. Liu, and 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., 1997, pp. 161–168.

[11] C.-L. Liu, "Handwritten Chinese character recognition: effects of shape normalization and feature extraction," in *Proceedings of the 2006 conference on Arabic and Chinese handwriting recognition*, ser. SACH'06. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 104–128.

[12] M. Cheriet, N. Kharma, C. lin Liu, and C. Suen, *Character Recognition Systems: A Guide for Students and Practitioners*. Wiley-Interscience, 2007.

[13] R. Dai, C. Liu, and B. Xiao, "Chinese character recognition: history, status and prospects," *Frontiers of Computer Science in China*, vol. 1, no. 2, pp. 126–136, 2007.

[14] C.-L. Liu, K. Nakashima, H. Sako, and H. Fujisawa, "Handwritten digit recognition: benchmarking of state-of-the-art techniques," *Pattern Recognition*, vol. 36, no. 10, pp. 2271–2285, 2003.

[15] C.-L. Liu, H. Sako, and H. Fujisawa, "Performance evaluation of pattern classifiers for handwritten character recognition," *Int. J. Document Anal. Recognition*, vol. 4, no. 3, pp. 191–204, 2002.

[16] C. Liu and H. Fujisawa, "Classification and learning methods for character recognition: Advances and remaining problems," *Machine learning in document analysis and recognition*, pp. 139–161, 2008.

[17] A. Kawamura, K. Yura, T. Hayama, Y. Hidai, T. Minamikawa, A. Tanaka, and S. Masuda, "Online recognition of freely handwritten Japanese characters using directional feature densities," in *In Proceedings of the 11th International Conference on Pattern Recognition*, 1992, pp. II:183–186.

[18] C. Liu, H. Sako, and H. Fujisawa, "Handwritten Chinese character recognition: Alternatives to nonlinear normalization," 2003.

[19] C.-L. Liu and K. Marukawa, "Pseudo two-dimensional shape normalization methods for handwritten Chinese character recognition." *Pattern Recognition*, vol. 38, no. 12, pp. 2242–2255, 2005.

[20] T. NAKAJIMA, T. WAKABAYASHI, F. KIMURA, and Y. MIYAKE, "Accuracy Improvement by Compound Discriminant Functions for Resembling Character Recognition." *IEICE Transactions on Information and Systems*, vol. 83, no. 2, pp. 623–633, 2000.

[21] T. Horiuchi, R. Haruki, H. Yamada, and K. Yamamoto, "Two-dimensional extension of nonlinear normalization method using line density for character recognition," *Proceedings of the Fourth International Conference on Document Analysis and Recognition*, pp. 511–514, 1997.

[22] C.-L. Liu, "Normalization-cooperated gradient feature extraction for handwritten character recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 8, pp. 1465–1469, 2007.

[23] C.-L. Liu, M. Koga, H. Sako, and H. Fujisawa, "Aspect ratio adaptive normalization for handwritten character recognition," in *ICMI '00: Proceedings of the Third International Conference on Advances in Multimodal Interfaces*. London, UK: Springer-Verlag, 2000, pp. 418–425.

[24] R. Vanderheiden and F. Groen, "The Box-Cox metric for nearest-neighbor classification improvement," *Pattern Recognition*, vol. 30, no. 2, pp. 273–279, February 1997.

[25] F. Kimura, K. Takashina, S. Tsuruoka, and Y. Miyake, "Modified quadratic discriminant functions and the application to Chinese character recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 9, no. 1, pp. 149–153, 1987.