Learning-Based Candidate Segmentation Scoring for Real-Time Recognition of Online Overlaid Chinese Handwriting

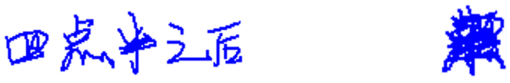
Yan-Fei Lv[1], Lin-Lin Huang[2], Da-Han Wang[1], Cheng-Lin Liu[1]
[1]National Laboratory of Pattern Recognition,
Institute of Automation of Chinese Academy of Sciences, Beijing 100190, China
[2]School of Electronic and Information Engineering,
Beijing Jiaotong University, Beijing 100044, China
Email: liucl@nlpr.ia.ac.cn

***Abstract*—In overlaid handwriting, multiple characters are written sequentially in the same area. This needs special consideration for segmenting the stroke sequence into characters. We propose a learning-based model for scoring the candidate stroke cuts and segments for online overlaid Chinese handwriting recognition. Based on stroke cut classification using support vector machine (SVM), strokes are grouped into segments, and consecutive segments are concatenated into candidate characters. The likeliness of candidate characters (unary geometry) and the compatibility between adjacent characters (binary geometry) are measured by combining the stroke cut score and the between-segment geometric score, and are integrated with the character classification score and linguistic context for character string recognition. Experiments on a large database of online Chinese handwriting demonstrate the effectiveness of the proposed method.**

Keywords*—Online overlaid Chinese handwriting, over-segmentation, stroke cut, geometric scores.

I. Introduction

With the popular use of pen-based and touch-based devices such as tablet PC, PDA, and smart phones, online handwriting recognition-based text input meets more application opportunities, and sentence (character string) recognition has many advantages over isolated character recognition [1]. Due to the small surface of hand-held devices, writing the characters of a sentence in the same area continuously (overlaid handwriting) has been proposed [2]–[5]. Fig. 1 shows an example of handwritten sentence and its overlaid writing. Due to the loss of horizontal character shift, it is difficult to segment the characters in overlaid handwriting. We propose a method of geometric scoring for improving the character segmentation performance in overlaid Chinese handwriting.



(a) (b)

Fig. 1. A handwritten sentence (a) and the corresponding overlaid writing (b).

In previous attempts of overlaid handwriting recognition, Shimodaira et al. [2] used a hierarchical dictionary of character models with substrokes modeled as hidden Markov models (HMMs), and used a network search algorithm to segment and recognize the characters in a string. Tonouchi et al. [3] used discrete Markov models consisting of stroke and up-stroke states for characters, and an inter-character state for segmentation. Both the HMM-based methods are stroke-order dependent and need multiple models for the characters with stroke-order variation. For overlaid Chinese handwriting, Zou et al. [4] use a neural network to classify stroke cuts (whether a stroke starts a new character or not), and perform character segmentation and recognition in two separate steps. Based on stroke cuts classification using support vector machine (SVM), Wan et al. [5] evaluate candidate character pairs by combining stroke information, recognition confidence and language model, and search for best character pairs recursively. Both the methods of [4], [5] consider the strokes of the whole character string for segmentation and recognition, and thus do not enable real-time recognition (segmenting and recognizing during writing, utilizing the past strokes only).

Recently, a real-time handwriting recognition approach under the principled sequence classification framework was proposed [1]. To apply this approach to overlaid recognition, we herein propose a method for candidate segmentation scoring, which is combined with the ordinary character classification and linguistic scores for character string recognition. Based on stroke cut classification using SVM, we over-segment the stroke sequence into primitive segments. Then, on forming candidate characters by concatenating consecutive segments, we measure the likeliness of candidate characters (unary geometry) and the between-character compatibility (binary geometry) by combining the stroke cut scores and between-segment geometric scores. Both segmentation and recognition consider the past strokes only. In our experiments on the online Chinese handwriting database CASIA-OLHWDB [6], the proposed segmentation score is shown to improve the string recognition performance significantly.

The rest of this paper is organized as follows. Section 2 gives an overview of the recognition system; Section 3 and Section 4 presents the methods for character over-segmentation and geometric context scoring, respectively; Section 5 presents the experiment results and Section 6 offers concluding remarks.

1520-5363/13 $26.00 © 2013 IEEE
DOI 10.1109/ICDAR.2013.23

II. Recognition System Overview

The real-time overlaid handwriting recognition scheme is similar to that of [1] with dynamically maintained candidate segmentation-recognition lattice, except that the geometric models are specially designed for scoring candidate segmentations. Fig. 2 shows the overall processing flow of overlaid handwriting recognition. On each pen lift, the system checks whether the last stroke starts a new character or not, groups the last strokes into primitive segments, and concatenates segments into candidate characters. The candidate characters are assigned candidate classes by a character classifier and linguistic scores by a statistical language model (character bi-gram in our case). When the pen lift exceeds a time threshold (0.2 second, say), the optimal path in the candidate lattice is searched for by dynamic programming to give the character string segmentation and recognition result.

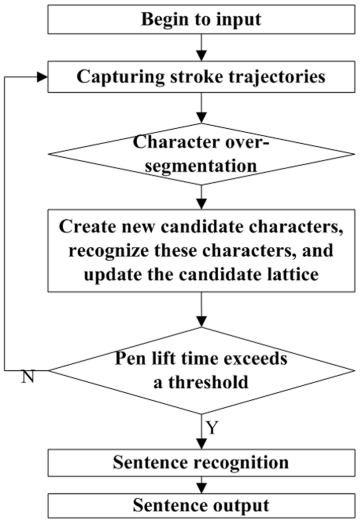


Fig. 2. Flow chart of real-time overlaid handwriting recognition.

Over-segmentation is to group the strokes into primitive segments. In overlaid writing, the characters do not shift from left to right, such that the characters cannot be segmented according to the horizontal overlapping of strokes [1]. We thus use a two-class classifier, support vector machine (SVM), to judge whether a pen lift is a between-character cut (i.e. its next stroke starts a new character or not. This stroke cut classifier detects excessive candidate cuts by setting a low threshold such that between-character cuts are detected with a high recall rate. Then, the strokes between two adjacent candidate cuts are grouped into a primitive segment. Fig. 3(a) shows the primitive segments of an overlaid string and the candidate characters by concatenating consecutive segments, where each candidate character is assigned two candidate classes by character classification. The character classification scores are combined with the language model and geometric scores for evaluating the candidate segmentation-recognition paths in candidate lattice. When a new stroke is produced, only the related partial lattice is updated (Fig. 3(b)). When the pen lift exceeds a time threshold, the written sentence is assumed complete, and the optimal path in the updated candidate lattice is searched for by frame-synchronous dynamic programming search [1]. Since the generation of segmentation-recognition lattice (including candidate character classification and geometric scoring) consumes the majority of computing and is performed in real time during writing, sentence recognition result is obtained immediately with a fast search algorithm after a pen lift of long time.

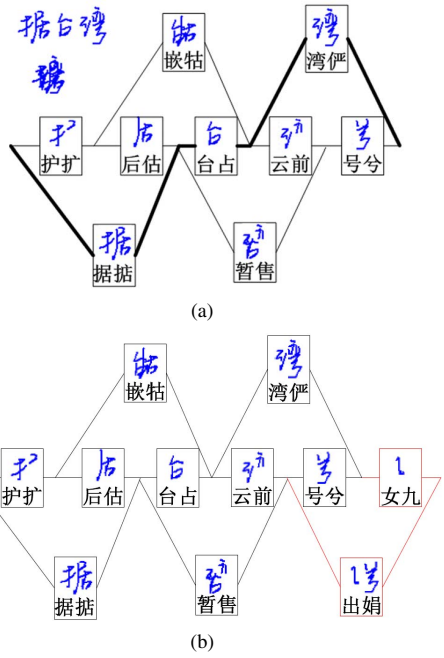


Fig. 3. (a) A character string grouped into five segments (boxes in the middle line), and the candidate segmentation-recognition lattice; (b) the updated lattice due to a new stroke.

Consider a candidate segmentation path $X = \mathbf{x}_1 \cdots \mathbf{x}_n$ paired with string class $C = c_1 \cdots c_n$ (candidate segmentation-recognition path), the path score is:

$$f(X,C) = \sum_{i=1}^{n} \{k_i \log P(c_i|\mathbf{x}_i) + \lambda_1 \log P(c_i|c_{i-1}) + \lambda_2 \log P(z_i^p = 1|\mathbf{g}_i^{ui}) + \lambda_3 \log P(z_i^g = 1|\mathbf{g}_i^{bi})\} \quad (1)$$

where k_i is the number of segments forming the candidate character $\mathbf{x}_i$; the four probabilities are the character classification score, bi-gram linguistic score, unary class-independent geometric score, and binary class-independent geometric score, respectively; λ_j $(j = 1, 2, 3)$ are combining weights, which can be trained on a character string dataset [1], [7]. We do not use class-dependent geometric models, which were observed to not benefit overlaid handwriting recognition in our experiments. Taking into account the memory limitation of mobile devices, the character classifier is a nearest prototype classifier (NPC) [8], whose output (distance) is transformed to probabilistic confidence as done in [1], [7].

III. Character Over-Segmentation

Whenever a new stroke is captured, the pen lift preceding it is judged by a two-class classifier (nonlinear SVM with 2nd-order polynomial kernel) whether it is a candidate between-character cut or not. If the confidence is over a threshold, the stroke is treated as a new segment; otherwise, it is grouped

with its predecessors to form a segment. Thus, the detection of candidate cuts (pen lifts between strokes) over-segments the character string into primitive segments.

For SVM classification, we extract 42 geometric features from a pair of strokes $Strk_1$ and $Strk_2$, which are listed in Table I, where P_1 is the end point of $Strk_1$, P_{21} and P_{22} are the start point and end point of $Strk_2$, $Strk_{12}$ is the union of $Strk_1$ and $Strk_2$. Except the aspect ratios, all the features are normalized by the estimated character height.

TABLE I. GEOMETRIC FEATURES EXTRACTED FROM A PAIR OF STROKES.

No.	Features
1-4	Geometric centers of $Strk_1$ and $Strk_2$
5-6	Distance between horizontal geometric centers and between the vertical geometric centers of $Strk_1$ and $Strk_2$
7-10	Distance between the left bounds, right bounds, upper bounds, and lower bounds of $Strk_1$ and $Strk_2$
11-14	Distance between the upper-lower bounds, lower-upper bounds, left-right bounds, and right-left bounds of $Strk_1$ and $Strk_2$
15-18	Height and width of the bounding boxes of $Strk_1$ and $Strk_2$
19-20	Logarithm of the aspect ratios of $Strk_1$ and $Strk_2$
21-22	Diagonal length of $Strk_1$ and $Strk_2$
23-24	Square root of the bounding boxes of $Strk_1$ and $Strk_2$
25-28	x-y coordinates of P_1 and P_{21}
29-30	Difference of x-y coordinates between P_1 and P_{21}
31	Euclidean distance between P_1 and P_{21}
32-33	Distance between P_1 and the left bound and lower bound of $Strk_1$
34-35	Distance between P_{21} and the right bound and lower bound of $Strk_1$
36-37	Distance between P_{22} and the left bound and lower bound of $Strk_2$
38-39	Height and width of the bounding box of $Strk_{12}$
40	Logarithm of the aspect ratio of $Strk_{12}$
41-42	x-y coordinates of the geometric center of $Strk_{12}$

The SVM classifier is trained with a dataset of stroke pairs labeled as genuine character cut or within-character pen lift. If the SVM output score is smaller than a threhold, the pen lift is judged to be a within-character lift, and so, the adjacent strokes are forced to be merged into a primitive segment. If a between-character cut is misclassified as within-character list, the adjacent two characters cannot be segmented due to the forced stroke merging. Decreasing the threshold of SVM output, we can reduce the error of merging between-character strokes, i.e. detect between-character cuts at high recall rate.

In a special case, overlaid writing of two characters makes a third character (Fig. 4). This makes the between-character cut mostly misclassified as within-character lift. To overcome this, we use the character classifier to verify the candidate character composed of two strokes. If it is recognized as the third character of Fig. 4, a candidate cut is added. The final decision of this cut is delayed to the character string recognition stage incorporating linguistic context.

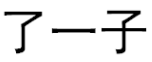


Fig. 4. Overlaid writing of two characters (left) makes a third character (right).

IV. GEOMETRIC CONTEXT MODELING

For modeling the class-independent unary and binary geometry, we use two linear SVM classifiers on the features extracted from a candidate character and on the features from a pair of adjacent candidate characters, respectively. As will be shown below, both the unary and binary geometric models utilize the compatibility between adjacent primitive segments.

From a pair of adjacent segments, we extract 23 geometric features, most of which are similar to the between-character geometric features used in [9], and one being the score of stroke cut classification between the last stroke of the preceding segment and the first stroke of the following one (SVM classifier as in Section 3). On these features, an SVM classifier is used to give the between-segment score. The compatibility score between the adjacent segments of two candidate characters is used as the class-independent binary geometry score.

For scoring single-character (unary) geometry, we extract 10 features from a candidate character, two of which are the mean and the maximum of between-segment scores for the between-segment gaps within the character. The other eight features are similar to the class-independent unary geometric features in [9]. The unary geometric score is given by a linear SVM on these features.

We further propose a hybrid geometric score by combining the between-segment scores of two characters. As shown in Fig. 4, a sequence of six segments is partitioned into two characters by a cut between segments #1 and #2. Denote the between-segment score of S_i and S_j as $P_{i,j}$, the probability that the two segments belong to the same character is then $1 - P_{i,j}$. The hybrid score is thus

$$P(z_i^g = 1|\mathbf{g}_i^{bi}) = \min_{i \neq k} (P_{k,k+1}, 1 - P_{i,i+1}), \qquad (2)$$

where k is the index of the last segment of the character before the hypothesized cut. This hybrid score combines the binary geometric scores of all the adjacent pairs of segments, both between-character ones and within-character ones.

The unary and binary geometric scores used here are peculiar that they both incorporate the compatibility between primitive segments, which in turn incorporates the score of stroke cut classification.

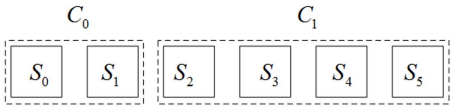


Fig. 5. A segment sequence hypothesized into two characters.

V. EXPERIMENTS

A. Dataset

We evaluated the performance of the proposed method on a database of online Chinese handwriting CASIA-OLHWDB [6], which is divided into six datasets: three for isolated characters (DB1.0-1.2) and three for handwritten texts (DB2.0-2.2). We use the text dataset DB2.1 to evaluate the performance, which is divided into disjoint training set of 240 writers and test set of 60 writers. For simulating overlaid writing process, we re-place the characters in DB2.1 to generate overlaid handwriting data. We consider only Chinese, numeral and English letters, and remove the non-alphanumeric symbols in the text data, since the non-alphanumeric symbols (such as punctuation marks) are hard to be segmented from overlaid

handwriting, but instead, they can be easily typed from soft keyboard. We imagine an overlaid writing area with center (x_c, y_c) and size as character height h_a. The generation process has four steps:

1) Mark the end of text lines as the end of sentences.
2) Remove the non-alphanumeric symbols, mark their positions as the end of sentences, and delete the sentences that contain only one character.
3) For each sentence, estimate h_a as the average height of the characters. Assume that the distribution of characters' centers as two-dimensional Gaussian with mean (x_c, y_c) and diagonal covariance $diag(\sigma_x^2, \sigma_y^2)$. We set $\sigma_x = \sigma_y = 0.3 * h_a/2.58$ to ensure that 99% of characters' centers are within a circle of radius $0.3 * h_a$. Random numbers are generated according to this distribution (subject to the constraint that the generated center is within $0.3 * h_a$ from (x_c, y_c)) for each character to place.
4) Characters with exceptionally long strokes (Fig. 5) are removed from the generated dataset because they are unlikely to appear in overlaid writing.

Fig. 6. Characters with exceptionally long strokes that are unlikely to appear in overlaid writing.

The generated overlaid text dataset is called GDB2.1. We further removed the data of 13 training writers and one test writers because they have appreciable numbers of characters of exceptionally long strokes. Finally, the GDB2.1 has 51,893 sentences or called strings, including 41,194 training strings of 227 writers and 10,699 test strings of 59 writers. There are totally 347,567 characters in the strings. The number of characters in strings varies from 2 to 44 and the average number is 6.70.

B. Experimental Setting and Results

In CASIA-OLHWDB, there are 3,912,017 isolated character samples and 52,220 handwritten pages (consisting of 1,348,904 character samples). Both the isolated data and handwritten text data have been divided into standard training and test subsets. The character classifier (discriminative NPC [8]) was trained with all the isolated character samples and the characters in the text data of 816 training writers in CASIA-OLHWDB, 4,207,801 samples in total. From the training samples, 4/5 were used for training classifier and the remaining 1/5 used for confidence parameter estimation. The training samples fall in 7,356 classes, including 7,185 Chinese characters and 171 alphanumeric characters and symbols. The classifier inputs direction histogram features extracted from online character samples by the normalization-cooperated feature extraction (NCFE) method proposed in [10]. The extracted 512D feature vector is reduced to 160D by Fisher linear discriminant analysis (FLDA). The character bi-gram language model used in string recognition is the one of [7], trained on a text corpus containing about 50 million characters.

From the training strings of GDB2.1, 218,468 stroke pairs, about 1/5 of all, were randomly sampled for training the stroke cut classifier (nonlinear SVM). The unary and binary geometric models (linear SVMs) and their confidence parameters were trained with 3/4 of training strings, while the remaining 1/4 training strings were used for tuning the combining weights λ_j of Eq. (1). There are 281,685 segment pairs for training the binary geometric model. For training the unary geometric model, there are in total 1,444,630 candidate characters, from which 152,579 were sampled, including 48,177 positive samples and 104,402 negative samples. In estimating the combining weights, to avoid combinatorial search, we selected weight values sequentially: tune one weight at a time and fix the former ones when tuning a later one.

We evaluate the performance of over-segmentation from the viewpoint of between-character cut detection, and measure the rates of recall (R), precision (P) and harmonic mean (F):

$$R = \frac{\text{\#correctly detected between-character cuts}}{\text{\#true between-character cuts}} \tag{3}$$

$$P = \frac{\text{\#correctly detected between-character cuts}}{\text{\#detected between-character cuts}} \tag{4}$$

$$F = \frac{2}{1/R + 1/P} \ . \tag{5}$$

The performance of overlaid string recognition is measured in character-level correct rate (CR) and accurate rate (AR) [1], [7]:

$$CR = (N_t - D_e - S_e)/N_t \ , \tag{6}$$

$$AR = (N_t - D_e - S_e - I_e)/N_t \ , \tag{7}$$

where N_t is the total number of characters in the test strings, S_e, D_e and I_e are the numbers of substitution errors, deletion errors and insertion errors, respectively.

For obtaining high recall rate of between-character cut detection, we empirically set the threshold of SVM output as low as 0.006. As result, we achieved a high recall rate of 99.59% with precision 62.33%. This ensures that most characters in the overlaid strings can be separated.

In string recognition after over-segmentation, we set the maximum number of concatenated segments as $SN = 6$ and the number of candidate classes output by the character classifier as $CN = 10$ as done in [1]. This guarantees that the characters in the strings can be mostly hypothesized by concatenating segments and the correct class is mostly included in the candidate classes.

Table II shows the performance of string recognition integrating character classification, geometric and linguistic contexts (Eq. (1)). In the table, R, P and F indicate the performance of finalized character segmentation. f_0, f_1, f_2, f_3 and f_4 stand for the scores of character classification, bi-gram language model, unary geometry, binary geometry and hybrid binary geometry (Eq. (2)), respectively. We can see that when only character classifier is used, the character segmentation and recognition performance is fair. In this recognition process, the stroke cut classification plays a promising role for over-segmentation. The language model f_1 gives significant improvement. The unary geometry score f_2 further improves the performance evidently, particularly, the CR is improved from

89.53% to 90.62%. The binary geometry f_3 is less effective than the unary geometry. The hybrid geometry score f_4 yields the best improvement, particularly, the CR is improved to 91.55%. Based on $f_0 + f_1 + f_4$, further combining f_2 or f_3 yields little improvement. This is because the hybrid geometry score f_4 already reflects both unary and binary geometry scores very well.

TABLE II. PERFORMANCE OF OVERLAID STRING RECOGNITION ON THE TEST SET.

	CR (%)	AR (%)	R (%)	P (%)	F (%)
f_0	81.23	68.70	98.15	85.86	91.59
$f_0 + f_1$	89.53	87.39	95.41	95.43	95.42
$f_0 + f_1 + f_2$	90.62	88.48	97.47	96.37	96.92
$f_0 + f_1 + f_3$	90.13	89.63	95.96	**98.50**	97.21
$f_0 + f_1 + f_4$	**91.55**	90.35	98.18	97.82	98.00
$f_0 + f_1 + f_4 + f_2$	91.52	90.27	**98.23**	97.80	98.01
$f_0 + f_1 + f_4 + f_3$	91.50	**90.48**	98.05	98.05	**98.05**
$f_0 + f_1 + f_4 + f_2 + f_3$	91.50	**90.48**	98.07	98.04	**98.05**

In comparison, the recognition accuracy on the segmented isolated characters from the test strings using the character classifier (NPC) is 88.10%. The correct rate of overlaid string recognition, 91.55%, is even higher than that of isolated character recognition despite the difficulty of character segmentation. This is because string recognition explores the linguistic and geometric contexts.

We also compare the performance of overlaid string recognition with that of ordinary (horizontally shifted) string recognition. Using the recognizer of [1] on the naturally written sentences corresponding to the test overlaid strings in GDB1.1, the character correct rate is 91.68%, which is comparable to the correct rate 91.55% of overlaid string recognition. This justifies that the proposed overlaid string recognition method is effective, though the overlaying of characters brings new challenge of character segmentation.

We looked into some mis-recognized overlaid strings, and observed that the recognition errors can be categorized into three types: over-segmentation error, character classification error and path search failure. Over-segmentation error (shown in Fig. 6(a)) happens when the between-character stroke cut was not detected such that two characters are merged. Character classification error (Fig. 6(b)) is the case that a character is correctly segmented but the true class is not included in the top CN candidate classes output by the classifier. Path search failure is mainly due to the imperfection of path evaluation criterion or search algorithm, as shown in Fig. 6(c).

VI. CONCLUSION

This paper presented an effective method for scoring the candidate stroke cuts and segments for real-time online overlaid Chinese handwriting recognition. Due to the high recall rate of stroke cut classification and the effectiveness of the hybrid binary geometric score, we achieved recognition accuracy of overlaid string recognition comparable to that of natural writing recognition despite the challenge brought by overlaying. Further improvements can be made by using character classifier of higher accuracy and linguistic model of higher order.

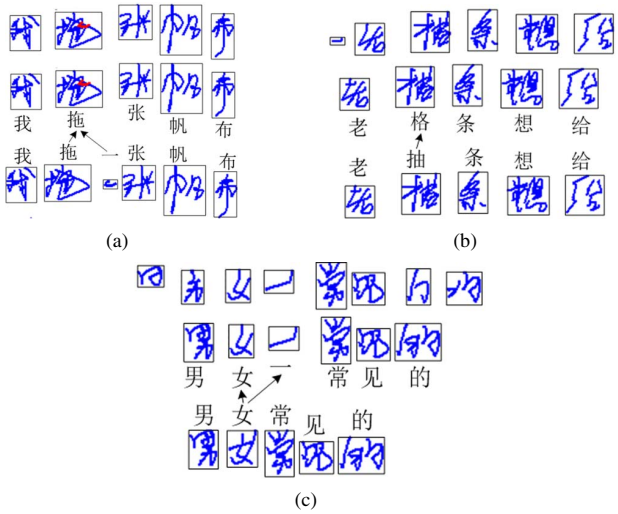


Fig. 7. Examples of errors in overlaid string recognition (displayed with horizontal shift): (a) Over-segmentation error; (b) Character classification error; (c) Path search failure. Upper: over-segmentation; middle: segmentation-recognition result; bottom: ground truth.

ACKNOWLEDGEMENTS

This work has been supported by the National Natural Science Foundation (NSFC) Grant 60933010.

REFERENCES

[1] D.-H. Wang, C.-L. Liu, and X.-D. Zhou, An approach for real-time recognition of online Chinese handwritten sentences, *Pattern Recognition*, 45 (10): 3661–3675, 2012.

[2] H. Shimodaira, T. Sudo, M. Nakai, and S. Sagayama, On-line overlaid-handwriting recognition based on substroke HMMs, *Proc. 7th ICDAR*, Edinburgh, Scotland, 2003, pp.1043–1047.

[3] Y. Tonouchi and A. Kawamura, Text input system using online overlapped handwriting recognition for mobile devices, *Proc. 9th ICDAR*, 2007, Curitiba, Brazil, pp.754–758.

[4] Y. Zou, Y. Liu, Y. Liu, and K. Wang, Overlapped handwriting input on mobile phones, *Proc. 11th ICDAR*, 2011, Beijing, China, pp.369–373.

[5] X. Wan, C. Liu, and Y. Zou, On-line Chinese character recognition system for overlapping samples, *Proc. 11th ICDAR*, 2011, Beijing, China, pp.799–803.

[6] C.-L. Liu, F. Yin, D.-H. Wang, and Q.-F. Wang, CASIA online and offline Chinese handwriting database, *Proc. 11th ICDAR*, 2011, Beijing, China, pp.37–41.

[7] Q.-F. Wang, F. Yin, and C.-L. Liu, Handwritten Chinese text recognition by integrating multiple contexts, *IEEE Trans. Pattern Anal. Mach. Intell.*, 34 (8): 1468–1481, 2012.

[8] X.-B. Jin, C.-L. Liu, and X. Hou, Regularized margin-based conditional log-likelihood loss for prototype learning, *Pattern Recognition*, 2010, 43 (7): 2428–2438.

[9] F. Yin, Q.-F. Wang, C.-L. Liu, and Q.-F. Wang, Integrating geometric context for text alignment of handwritten Chinese documents, *Proc. 12th ICFHR*, 2010, Kolkata, India, pp.16–18.

[10] C.-L. Liu and X.-D. Zhou, Online Japanese character recognition using trajectory-based normalization and direction feature extraction, *Proc. 10th IWFHR*, 2006, La Baul, France, pp.217–222.