Handwritten Chinese Text Recognition by Integrating Multiple Contexts

Qiu-Feng Wang, Fei Yin, and Cheng-Lin Liu, *Senior Member*, *IEEE*

Abstract—This paper presents an effective approach for the offline recognition of unconstrained handwritten Chinese texts. Under the general integrated segmentation-and-recognition framework with character oversegmentation, we investigate three important issues: candidate path evaluation, path search, and parameter estimation. For path evaluation, we combine multiple contexts (character recognition scores, geometric and linguistic contexts) from the Bayesian decision view, and convert the classifier outputs to posterior probabilities via confidence transformation. In path search, we use a refined beam search algorithm to improve the search efficiency and, meanwhile, use a candidate character augmentation strategy to improve the recognition accuracy. The combining weights of the path evaluation function are optimized by supervised learning using a Maximum Character Accuracy criterion. We evaluated the recognition performance on a Chinese handwriting database CASIA-HWDB, which contains nearly four million character samples of 7,356 classes and 5,091 pages of unconstrained handwritten texts. The experimental results show that confidence transformation and combining multiple contexts improve the text line recognition performance significantly. On a test set of 1,015 handwritten pages, the proposed approach achieved character-level accurate rate of 90.75 percent and correct rate of 91.39 percent, which are superior by far to the best results reported in the literature.

Index Terms—Handwritten Chinese text recognition, confidence transformation, geometric models, language models, refined beam search, candidate character augmentation, maximum character accuracy training.

1 Introduction

Handwritten Chinese character recognition has long been considered a challenging problem. It has attracted much attention since the 1970s and has achieved tremendous advances [1], [2]. Both isolated character recognition and character string recognition have been studied intensively but are not solved yet. In isolated Chinese character recognition, most methods were evaluated on data sets of constrained writing styles though very high accuracies (say, over 99 percent on Japanese Kanji characters and over 98 percent on Chinese characters) have been reported [1]. The accuracy on unconstrained handwritten samples, however, is much lower [3]. In Chinese character string recognition, most works aimed at the recognition of text lines or phrases in rather constrained application domains, such as legal amount recognition in bank checks [4] and address phrase recognition for postal mails [5], [6], [7], [8], where the number of character classes is very small or there are very strong lexical constraints. Works on Chinese handwriting recognition of general texts have been reported only in recent years, and the reported accuracies are quite low. For example, Su et al. reported character-level correct rate (CR) of 39.37 percent on a Chinese handwriting data set HIT-MW with 853 pages containing 186,444 characters [9]. Two later works on the same data set, using character classifiers and statistical language models (SLM) based on oversegmentation, reported a character-level correct rate of 78.44 [10] and 73.97 percent [11], respectively. On the other hand, many works on online Japanese/Chinese handwritten text recognition have reported higher accuracies [12], [13], [14], [15]. Online handwriting recognition has the advantage over offline recognition in that the sequences of strokes are available for better segmenting and discriminating characters.

Handwritten Chinese text recognition (HCTR) is a challenging problem due to the large character set, the diversity of writing styles, the character segmentation difficulty, and the unconstrained language domain. Fig. 1 shows an example of a Chinese handwritten page. The large set of Chinese characters (tens of thousands of classes) brings difficulties to efficient and effective recognition. The divergence of writing styles among different writers and in different geographic areas aggravates the confusion between different classes. Handwritten text recognition is particularly difficult because the characters cannot be reliably segmented prior to character recognition. The difficulties of character segmentation originate from the variability of character size and position, character touching and overlapping. A text line of Chinese handwriting must be recognized as a whole because it cannot be trivially segmented into words (there is no more extra space between words than between characters). Last, handwritten text recognition is more difficult than bank check recognition and mail address reading because the lexical constraint is very weak: Under grammatical and semantic constraints, the number of sentence classes is infinite.

Due to the large number of character classes and the infinite sentence classes of Chinese texts, HCTR can only be

- *The authors are with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, 95 Zhongguancun East Road, Beijing 100190, P.R. China.*
E-mail: {wangqf, fyin, liucl}@nlpr.ia.ac.cn.

Manuscript received 18 July 2011; revised 15 Nov. 2011; accepted 22 Nov. 2011; published online 20 Dec. 2011.
Recommended for acceptance by R. Manmatha.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2011-07-0476.
Digital Object Identifier no. 10.1109/TPAMI.2011.264.

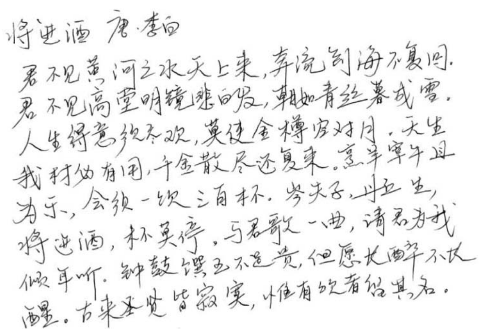

将进酒 唐·李白
君不见黄河之水天上来，奔流到海不复回。
君不见高堂明镜悲白发，朝如青丝暮成雪。
人生得意须尽欢，莫使金樽空对月。天生
我材必有用，千金散尽还复来。烹羊宰牛且
为乐，会须一饮三百杯。岑夫子，丹丘生，
将进酒，杯莫停。与君歌一曲，请君为我
倾耳听。钟鼓馔玉不足贵，但愿长醉不复
醒。古来圣贤皆寂寞，惟有饮者留其名。

Fig. 1. A page of handwritten Chinese text.

solved by segmentation-based approaches using character models [16], preferably by explicit segmentation, also called oversegmentation, which can take advantage of the character shape and overlapping and touching characteristics to better separate the characters at their boundaries. The result of oversegmentation is a sequence of primitive segments, each corresponding to a character or a part of a character, such that candidate characters can be generated by concatenating consecutive segments [5]. The candidate character sequences can be represented in a network called a candidate lattice [17], and each candidate segmentation path in the lattice can be split into many segmentation-recognition paths by assigning character classes to the candidate characters. The result of character segmentation and recognition is obtained by evaluating the paths in the lattice and searching for the optimal path.

In integrated segmentation-and-recognition, the candidate segmentation-recognition paths are usually evaluated by combining the character recognition scores (classifier outputs), geometric context, and linguistic context [16]. Many efforts have been made this direction, but there has not been a satisfactory solution. The existing methods either integrated incomplete contexts [9], [10], [18] or combined the contexts heuristically without optimizing the combining weights [12], [13], [19], [20]. Zhou et al. optimize the combining weights using the conditional random field (CRF) model [14], which is hard to incorporate into language models of higher order than the bi-gram. Zhu et al. optimize the combining weights using the genetic algorithm (GA) [15], which is computationally expensive and is sensitive to some artificial parameters. The previous works have addressed handwritten text (character string) recognition from different viewpoints and have contributed various techniques. However, none has investigated these techniques comprehensively and integrated them in a high-performance system for Chinese/Japanese handwritten text recognition.

In this study, we investigate three key issues of integrated segmentation-and-recognition for HCTR: candidate path evaluation, path search, and parameter estimation. By elaborating the techniques for these issues, we achieved significant improvements on unconstrained handwritten Chinese texts. In path evaluation, we integrate character recognition scores, geometric context, and linguistic context from the Bayesian decision view, and convert the classifier outputs to posterior probabilities via confidence transformation (CT). In path search, a refined beam search algorithm is used to improve the search efficiency and, meanwhile, a candidate character augmentation (CCA) strategy is applied to benefit the recognition accuracy. To balance the multiple contexts in path evaluation function, we optimize the combining weights on a data set of training text lines using a Maximum Character Accuracy (MCA) criterion. We evaluated the recognition performance on a large database CASIA-HWDB [21] of unconstrained Chinese handwritten characters and texts, and demonstrated superior performance by the proposed methods.

The rest of this paper is organized as follows: Section 2 reviews some related works, Section 3 gives an overview of our HCTR system, Section 4 provides a statistical foundation of the path evaluation issue from the Bayesian decision view, Section 5 describes the confidence transformation, geometric context, and linguistic context in details, Section 6 introduces the refined beam search algorithm and candidate character augmentation strategy, Section 7 presents the weights learning method, Section 8 reports the experimental results, and Section 9 draws concluding remarks.

2 Related Works

In the context of handwritten text (character string[1]) recognition, many works have contributed to the related issues of oversegmentation, character classification, confidence transformation, language model, geometric model, path evaluation and search, and parameter estimation.

For oversegmentation, connected component analysis has been widely adopted, but the splitting of connected (touching) characters has been a concern [5], [22], [23]. After generating candidate character patterns by combining consecutive primitive segments, each candidate pattern is classified using a classifier to assign similarity/dissimilarity scores to some character classes. Character classification involves character normalization, feature extraction, and classifier design. The state-of-the-art methods have been reviewed in [24], [25]. For classification of Chinese characters with large number of classes, the most popularly used classifiers are the modified quadratic discriminant function (MQDF) [26] and the nearest prototype classifier (NPC) [27]. The MQDF provides higher accuracy than the NPC but suffers from high expenses of storage and computation.

Transforming the similarity/dissimilarity measures output by classifiers to probabilistic confidence measures can benefit from fusing multiple classifiers or fusing multiple patterns, as has been demonstrated in previous works (e.g., [28], [29]). In character string recognition, Jiang et al. [18] transformed classifier outputs to confidence values under the soft-max framework. Li et al. [30] used the logistic regression model for confidence transformation. Our recent work [31] compared various confidence transformation methods in HCTR and found a better solution.

Language models are widely used in speech recognition, machine translation, handwriting recognition, and so on [32]. The most popular language model is the n-gram, which characterizes the statistical dependency between characters or words. Character-level n-gram models have been popularly used in character string recognition (e.g., [12],

1. We will use the terms text line recognition and string recognition interchangeably because, in this study, a text line is treated as a character string.

[13], [14], [15], [18], [19], [20]. Word-level and hybrid language models were used in postprocessing for correcting recognition errors after character segmentation [30], [33], but have been rarely used in integrated segmentation-and-recognition [10].

In addition to the character recognition scores and linguistic context, the geometric context also plays an important role in character string recognition, particularly for disambiguating character segmentation [12], [13], [14], [15], [19], [20], [34]. Zhou et al. elaborated the geometric context models into unary and binary, character class-dependent and class-independent models in online handwriting recognition [13], [14]. Yin et al. elaborated the geometric context models for offline handwriting and applied to transcript mapping of handwritten Chinese documents [35].

A key issue in character string recognition is to design an objective function evaluating each candidate segmentation-recognition path. The path evaluation function is hoped to be insensitive to the path length (number of characters on the path). The summation of classifier output similarity/dissimilarity scores or product of class probabilities is not appropriate since this is biased to short paths. Normalizing the summation or product by the path length overcomes the bias problem [36], [37], but this normalized form does not enable optimal path search by dynamic programming (DP). Beam search can be used instead, but does not guarantee optimality [37]. Another way to overcome the path length bias is to add a compensative constant in the summated path evaluation function [20], but the constant needs to be estimated empirically. Wuthrich et al. [38] called this constant a word insertion penalty, and Quiniou et al. [39] also used this constant to control the deletion and insertion of words. Another effective way is to weight the character classification score with the number of primitive segments forming the character pattern [10], [15], motivated by the variable duration HMM of Chen et al. [40]. This not only makes the number of summated terms in the path evaluation function equal the number of primitive segments (and thus independent of the path length), but also preserves the summation form and enables optimal path search by DP.

In weighted combination of context models for path evaluation, the weights were sometimes determined by trial and error. Some works have applied the supervised learning approach to estimate the weights by optimizing a string recognition criterion. Recently, Zhou et al. [14] proposed learning the weights by minimizing the negative log-likelihood (NLL) loss under the framework of conditional random field, and compared its performance with the minimum classification error (MCE) criterion [41]. Yin et al. [35] optimized the weights by MCE learning for transcript mapping. Zhu et al. [15] optimized the combining weights for handwriting recognition using the genetic algorithm. More discriminative learning criteria have been proposed by the speech recognition community, such as minimum phone error (MPE) and its variant, minimum word error (MWE) [42], [43].

The search of optimal path in Chinese character string recognition is not trivial because of the large number of candidate segmentation-recognition paths. The search is further complicated when using word-level language models because the word segmentation is again a combinatorial problem [10]. The speech recognition community has contributed many efficient search algorithms based on dynamic programming and some variants (e.g., beam search) [44]. The beam search strategy provides a good tradeoff between efficiency and accuracy. The character-synchronous beam search strategy is appropriate for lexicon-driven string recognition [5], while the frame-synchronous (also called as time-synchronous in speech recognition) strategy is appropriate for lexicon-free string recognition [37].

In character string recognition, the pruning or augmentation of character classes affects the search efficiency and accuracy. Ideally, a candidate character pattern is assigned as few classes as possible by the classifier, including the true class. For Chinese handwriting, it often entails a large number (e.g., several hundred) of candidate classes to guarantee a high probability of including the true class, however. This complicates the search space on one hand and, on the other hand, may deteriorate the recognition accuracy because there are too many wrong classes competing with the true class. Therefore, some works have attempted to reduce the candidate classes output by the classifier by confidence evaluation [45], [46], and some other works attempted to supplement candidate classes for reducing the probability of missing the true class, according to the linguistic context [47] or the classification confusion matrix [33], [48]. These techniques, however, have not been evaluated in integrated segmentation-and-recognition.

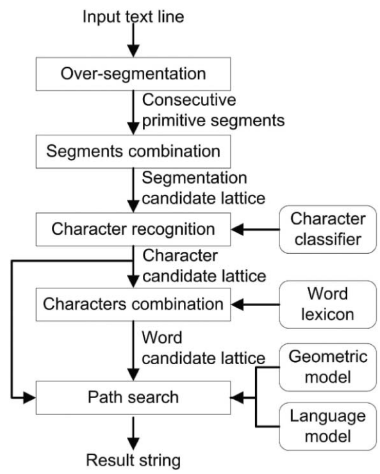


Fig. 2. System diagram of handwritten Chinese text line recognition.

3 System Overview

This study focuses on the recognition of text lines, which are assumed to have been segmented externally. For the convenience of academic research and benchmarking, the text lines in our database have been segmented and annotated at character level [49].

Fig. 2 shows the block diagram of our system for text line recognition. First, the input text line image is oversegmented into a sequence of primitive segments (Fig. 3a) using the connected component-based method [5]. Consecutive primitive segments are combined to generate candidate character patterns, forming a segmentation candidate lattice

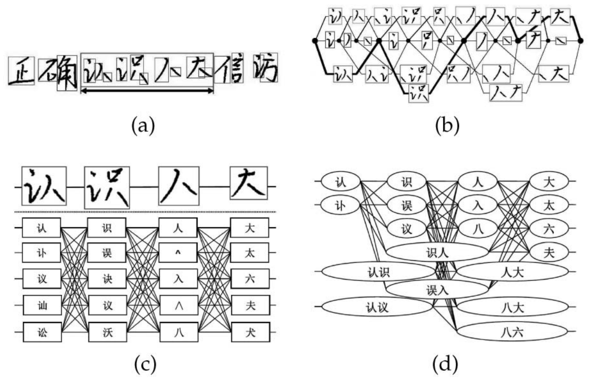


Fig. 3. (a) Oversegmentation to a sequence of primitive segments (each is bounded by a small box), (b) segmentation candidate lattice of the arrowed part of (a), (c) character candidate lattice of the thick path in (b), (d) word candidate lattice of (c).

(Fig. 3b). After that, each candidate pattern is classified to assign a number of candidate character classes, and all the candidate patterns in a candidate segmentation path generate a character candidate lattice (Fig. 3c). If a word-level language model is used, each sequence of candidate characters is matched with a word lexicon to segment into candidate words, forming a word candidate lattice (Fig. 3d). All of these character (or word) candidate lattices are merged to construct the segmentation-recognition lattice of text line image. Each path in this lattice is constructed by a character sequence paired with a candidate pattern sequence, and this path is called a candidate segmentation-recognition path. Finally, the task of string recognition is to find the optimal path in this segmentation-recognition lattice. Considering that the text lines are segmented from text pages, we utilize the linguistic dependency between consecutive lines to improve the recognition accuracy by concatenating multiple top-rank recognition results of the previous line to the current line for recognition.

4 Statistical Foundation

We formulate the problem of handwritten Chinese text recognition from Bayesian decision view. According to Bayesian decision under the 0/1 loss, maximizing a posterior probability of character sequence (string class) $C = <c_1 \dots c_m>$ given a text line image X is the optimal criterion for recognition. This posterior probability is formulated by

$$\begin{aligned} P(C|X) &= \sum_s P(C,s|X) = \sum_s P(s|X)P(C|s,X) \\ &= \sum_s P(s|X)P(C|X^s), \end{aligned} \tag{1}$$

where s is the segmentation path index, $P(s|X)$ denotes the posterior probability of the sth segmentation path given the text line image, and $P(C|X^s)$ represents the posterior probability of string class given the sth segmentation path. $P(s|X)$ is formulated by

$$P(s|X) = \prod_{i=1}^{m} p\big(z_i^{\mathrm{p}} = 1|g_i^{\mathrm{ui}}\big)p\big(z_i^{\mathrm{g}} = 1|g_i^{\mathrm{bi}}\big), \tag{2}$$

where m is the number of segmented candidate patterns (i.e., character number) of the segmentation path, $z_i^{\mathrm{p}} = 1$ means that the ith candidate pattern is a valid character, and $z_i^{\mathrm{g}} = 1$ means that the gap between the $(i-1)$th and ith candidate patterns is a valid between-character gap; the terms g_i^{ui} and g_i^{bi} are the class-independent geometric features extracted from the ith candidate pattern and from the pair of the $(i-1)$th and ith candidate patterns, respectively. The two probabilistic terms in (2) correspond to the unary and binary class-independent geometric model (see Section 5.2), respectively. To avoid summing over a huge number of segmentation paths in (1), the optimal string class can be decided approximately by

$$C^* = \arg\max_{s,C} P(s|X)P(C|X^s). \tag{3}$$

This is to search for the optimal segmentation and string class simultaneously in the lattice.

4.1 Posterior Probability of String Class

The posterior probability of string class can be obtained by the Bayes' formula:

$$P(C|X^s) = \frac{p(X^s|C)p(C)}{p(X^s)}, \tag{4}$$

where the prior probability $p(C) = p(c_1 \dots c_m)$ is given by a statistical language model (Section 5.3). The likelihood function $p(X^s|C)$ can be decomposed by

$$p(X^s|C) = \prod_{i=1}^{m} p(\mathbf{x}_i|c_i)p\big(g_i^{\mathrm{uc}}|c_i\big)p\big(g_i^{\mathrm{bc}}|c_{i-1}c_i\big), \tag{5}$$

where we assume that the feature vectors $(\mathbf{x}_i, g_i^{\mathrm{uc}}, g_i^{\mathrm{bc}})$ are only dependent on the corresponding character patterns, and they represent the intrinsic shape feature for character recognition, unary and binary outline geometric features for geometric models (Section 5.2), respectively. Similarly, we can get $p(X^s)$ as

$$p(X^s) = \prod_{i=1}^{m} p(\mathbf{x}_i)p\big(g_i^{\mathrm{uc}}\big)p\big(g_i^{\mathrm{bc}}\big). \tag{6}$$

Consequently, the posterior probability $P(C|X^s)$ is obtained by

$$P(C|X^s) = p(C)\prod_{i=1}^{m} \frac{p(\mathbf{x}_i|c_i)}{p(\mathbf{x}_i)} \frac{p\big(g_i^{\mathrm{uc}}|c_i\big)}{p\big(g_i^{\mathrm{uc}}\big)} \frac{p\big(g_i^{\mathrm{bc}}|c_{i-1}c_i\big)}{p\big(g_i^{\mathrm{bc}}\big)}, \tag{7}$$

and transformed to

$$P(C|X^s) = p(C)\prod_{i=1}^{m} \frac{p(c_i|\mathbf{x}_i)}{p_1(c_i)} \frac{p\big(c_i|g_i^{\mathrm{uc}}\big)}{p_2(c_i)} \frac{p\big(c_{i-1}c_i|g_i^{\mathrm{bc}}\big)}{p_3(c_{i-1}c_i)}, \tag{8}$$

where the three posterior probabilities can be approximated by confidence transformation (see Section 5.1), and three corresponding prior probabilities $p_1(c_i), p_2(c_i), p_3(c_{i-1}c_i)$ are viewed as constants in classifier design (the uneven distribution $p(c_i)$ is considered elsewhere in the language model $p(C)$), denoted by p_1, p_2, p_3, respectively. Substituting $p(C) = \prod_{i=1}^{m} p(c_i|h_i)$ (see Section 5.3) in (8) gives

$$P(C|X^s) = \prod_{i=1}^{m} \frac{p(c_i|\mathbf{x}_i)p(c_i|g_i^{\mathrm{uc}})p(c_{i-1}c_i|g_i^{\mathrm{bc}})p(c_i|h_i)}{P}, \quad (9)$$

where $P = p_1p_2p_3$. The four probabilistic terms in (9) correspond to the character recognition model, unary and binary class-dependent geometric model, and language model, respectively.

4.2 Path Evaluation Function

Combining the posterior probabilities of segmentation path (2) and string class (9), the optimal string class of (3) can be obtained by

$$\begin{aligned} C^* = \arg\max_{s,C} \frac{1}{P^m} \prod_{i=1}^{m} \big[&p(c_i|\mathbf{x}_i)p(c_i|g_i^{\mathrm{uc}})p(c_{i-1}c_i|g_i^{\mathrm{bc}}) \\ &p(z_i^{\mathrm{p}} = 1|g_i^{\mathrm{ui}})p(z_i^{\mathrm{g}} = 1|g_i^{\mathrm{bi}})p(c_i|h_i)\big]. \end{aligned} \quad (10)$$

Note that all the terms m, c_i, $\mathbf{x}_i$, g_i^{uc}, g_i^{bc}, g_i^{ui}, g_i^{bi}, z_i^{p}, z_i^{g}, h_i are related to the sth segmentation path, and the index s is dropped for simplification. However, the probability formulation (10) is still insufficient, because it does not consider the different contribution and reliability of different models (character recognition, geometric, and language models). In the following, we take the logarithm of probability (denoted by $lp_i^0 = \log p(c_i|\mathbf{x}_i)$, $lp_i^1 = \log p(c_i|g_i^{\mathrm{uc}})$, $lp_i^2 = \log p(c_{i-1}c_i|g_i^{\mathrm{bc}})$, $lp_i^3 = \log p(z_i^{\mathrm{p}} = 1|g_i^{\mathrm{ui}})$, $lp_i^4 = \log p(z_i^{\mathrm{g}} = 1|g_i^{\mathrm{bi}})$, $lp_i^5 = \log p(c_i|h_i)$, $lp^6 = \log \frac{1}{P}$) and incorporate the weights of different models to get a generalized likelihood function $f(X^s, C)$ for the segmentation-recognition path evaluation:

$$f(X^s, C) = \sum_{i=1}^{m} \left(lp_i^0 + \sum_{j=1}^{5} \lambda_j \cdot lp_i^j \right) + \lambda_6 \cdot m \cdot lp^6, \quad (11)$$

and $C^* = \arg\max_{s,C} f(X^s, C)$, where λ_j, $j = 1, \ldots, 6$, are the weights to balance the effects of different models.

In the above, the positive constant lp^6 is also called the word insertion penalty in [38], and used to overcome the bias to short strings (without this term, the path evaluation score decreases as the path length m increases). Besides this formulation, there are some heuristic methods to deal with the bias problem. One straightforward strategy used in the previous works [13], [36], [37] is to normalize the evaluation function with the path length:

$$f(X^s, C) = \frac{1}{m} \sum_{i=1}^{m} \left(lp_i^0 + \sum_{j=1}^{5} \lambda_j \cdot lp_i^j \right). \quad (12)$$

In our previous works [10], [31], we weighted the lp_i^0 with the number of constituent primitive segments k_i (similar to the variable length HMM of [40]), and got the evaluation function

$$f(X^s, C) = \sum_{i=1}^{m} \left(k_i \cdot lp_i^0 + \sum_{j=1}^{5} \lambda_j \cdot lp_i^j \right). \quad (13)$$

This function is not sensitive to the path length because the sum of k_i is a constant (equals the total number of primitive segments in the text line). Consider that the width of each primitive segment is variable; in this study, we also try another form of evaluation function by replacing the segments number with the width of candidate pattern

$$f(X^s, C) = \sum_{i=1}^{m} \left(w_i \cdot lp_i^0 + \sum_{j=1}^{5} \lambda_j \cdot lp_i^j \right), \quad (14)$$

where w_i is the width of the ith character pattern after normalizing by the estimated height of the text line. The four path evaluation functions in the above (adding Word Insertion Penalty (11), Normalization with Path Length (12), Weighting with primitive Segments Number (13), and Weighting with Character pattern Width (14)) are abbreviated as **WIP**, **NPL**, **WSN**, and **WCW**, respectively.

5 Probabilistic Models of Contexts

The path evaluation functions (11)-(14) entail the estimation of the context models and the combining weights. We describe the context models in the following, while the estimation of weights is addressed in Section 7.

5.1 Confidence Transformation

The character recognition score, ideally, the posterior probability $p(\omega|\mathbf{x})$ (ω refers to a class and $\mathbf{x}$ is the feature vector), is an important context for string recognition. Most classifiers, however, do not output class posterior probabilities. We hence resort to confidence transformation methods for converting classifier outputs to posterior probabilities [31].

Two commonly used functions for probabilistic confidence transformation are the **sigmoidal** function (15) and the **soft-max** function (16):

$$P^{sg}(\omega_j|\mathbf{x}) = \frac{\exp[-\alpha d_j(\mathbf{x}) + \beta]}{1 + \exp[-\alpha d_j(\mathbf{x}) + \beta]}, j = 1, \ldots, M, \quad (15)$$

$$P^{sf}(\omega_j|\mathbf{x}) = \frac{\exp[-\alpha d_j(\mathbf{x})]}{\sum_{i=1}^{M} \exp[-\alpha d_i(\mathbf{x})]}, \quad j = 1, \ldots, M. \quad (16)$$

In the above, M is the total number of defined classes, $d_j(\mathbf{x})$ is the dissimilarity score of class ω_j output by the classifier, α and β are the confidence parameters. Both forms have insufficiencies: The sigmoidal form gives multiple one-versus-all two-class probabilities instead of multiclass probabilities, while the soft-max form forces the sum of posterior probabilities to one even on noncharacter (outlier) patterns.

For the sigmoidal form, we combine such two-class probabilities into multiclass probabilities according to the **Dempster-Shafer (D-S) theory of evidence** [50], and the probabilities can be formulated by [31]

$$p^{ds}(\omega_j|\mathbf{x}) = \frac{\exp[-\alpha d_j(\mathbf{x}) + \beta]}{1 + \sum_{i=1}^{M} \exp[-\alpha d_i(\mathbf{x}) + \beta]}, j = 1, \ldots, M. \quad (17)$$

We also introduce an outlier class dissimilarity score (assuming $d_o(\mathbf{x}) = \beta/\alpha$) in soft-max confidence, and the result is extended to the same form of (17) [31]. After getting multiclass probabilities, the probability of outlier class is

$$p^{ds}(\omega_{outlier}|\mathbf{x}) = \frac{1}{1 + \sum_{i=1}^{M} \exp[-\alpha d_i(\mathbf{x}) + \beta]}, \quad (18)$$

which is the complement probability to the M defined classes.

The confidence parameters are optimized by minimizing the cross entropy (CE) loss function on a validation data set (preferably different from the data set for training classifiers) [31].

5.2 Geometric Models

Considering that Chinese texts mix with alphanumeric characters and punctuation marks and different characters show distinct outline features (e.g., size, position, aspect ratio, and within-character gap), we design two class-dependent geometric models, namely, single-character geometry (unary geometric model) and between-character geometry (binary geometric model), respectively. In addition, two class-independent geometric models are designed to indicate whether a candidate pattern is a valid character or not, and whether a gap is a between-character gap or not, respectively. The four geometric models (unary and binary class-dependent, unary and binary class-independent) are abbreviated as "**ucg**," "**bcg**," "**uig**," and "**big**," respectively, and have been used successfully in transcript mapping of handwritten Chinese documents [35].

To build geometric models, we extract features for unary and binary geometry from the bounding boxes and profiles of a candidate character pattern, and of two consecutive character patterns, respectively [35]. Since the number of Chinese characters is very large and many different characters have similar geometric features, we cluster the character classes empirically into six superclasses using the EM algorithm. After clustering, we use a 6-class quadratic discriminant function (QDF) for the "ucg" model, and a 36-class QDF for the "bcg" model. In addition, we use a linear support vector machine (SVM) trained with character and noncharacter samples for the "uig" model, and similarly, a linear SVM for the "big" model. In path evaluation function, we convert both QDF and SVM outputs to posterior probabilities via sigmoidal confidence transformation.

5.3 Statistical Language Models

In character string recognition, the statistical language model is used to give the prior probability of a certain character sequence [51]. If the sequence C contains m characters, $p(C)$ can be decomposed by

$$p(C) = \prod_{i=1}^{m} p\left(c_i|c_1^{i-1}\right) = \prod_{i=1}^{m} p(c_i|h_i), \tag{19}$$

where $h_i = c_1^{i-1} = <c_1 \cdots c_{i-1}>$ denotes the history of character c_i (h_1 is null). An n-gram model only considers the $n-1$ history characters in (19):

$$p(C) = \prod_{i=1}^{m} p\left(c_i|c_{i-n+1}^{i-1}\right), \tag{20}$$

where n is called the order of the model. For high complexity, the **character bigram** and **trigram** are usually used:

$$p_{cbi}(C) = \prod_{i=1}^{m} p(c_i|c_{i-1}), \tag{21}$$

$$p_{cti}(C) = \prod_{i=1}^{m} p(c_i|c_{i-2}c_{i-1}). \tag{22}$$

Compared to the character-level, word-level models can better explore the syntactic and semantic meaning. Segmenting the character sequence C into word sequence $C = w_1 w_2 \cdots w_L$, the **word bigram** model is

$$p_{wbi}(C) = \prod_{i=1}^{L} p(w_i|w_{i-1}). \tag{23}$$

Due to the large size of the word lexicon (about 0.3 million words), we only use the word bigram. Further, we cluster the words into a number of word classes by the exchange algorithm [52], and the **word class bigram** is calculated by

$$p_{wcb}(C) = \prod_{i=1}^{L} p(w_i|W_i)p(W_i|W_{i-1}), \tag{24}$$

where the term W_i is the class of word w_i, and the cluster number is set empirically to 1,000 [10]. In addition, the word class bigram is often used by interpolating with the word bigram [32]:

$$\log p_{iwc}(C) = \log p_{wbi}(C) + \lambda \cdot \log p_{wcb}(C), \tag{25}$$

where the logarithm is used for more general purposes, and this model is called **interpolating word and class bigram**.

We use the SRI Language Model (SRILM) toolkit [53] to give the parameters of n-gram models. By the toolkit, the default smoothing technique (Katz smoothing) and the entropy-based pruning are used. The thresholds of the pruning for character bigram, character trigram and word bigram are set empirically as 5×10^{-8}, 10^{-7}, and 10^{-7}, respectively [10]. Since the word class number (1,000) leads to a moderate model size, the parameters are not pruned.

6 Path Search

On defining a score for each path in the segmentation-recognition lattice, the next issue is how to efficiently find the path of maximum score. In addition, to alleviate the loss that the candidate classes assigned by character classifier do not contain the true class, we propose an augmentation technique to supplement candidate classes in the lattice.

6.1 Search Algorithm

If the segmentation-recognition path is evaluated by the accumulated score (WIP, WSN, and WCW), it satisfies the principle of optimality, and the optimal path with maximum score can be found by dynamic programming. Nevertheless, when binary or higher order contexts are used, the complexity of DP search is high. For the NPL function, which does not satisfy the principle of optimality, DP search does not guarantee finding the optimal path, and the beam search strategy can better find an approximately optimal solution. In beam search, it is critical to retain the correct partial path in fewer survived paths. A simple strategy of beam search is to retain the multiple top-rank partial paths ending at each primitive segment [16]. This simple strategy, though it works efficiently, is too rough, particularly when high-order context models are used. A

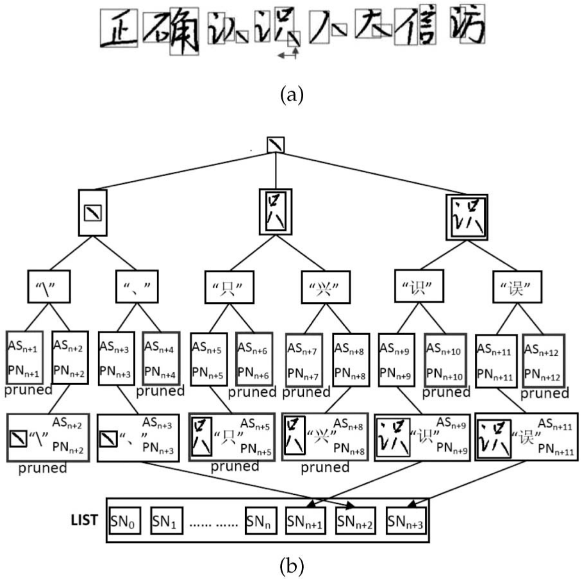


Fig. 4. An illustrative example of refined beam search ($K = 3$, $CN = 2$, $BW = 3$) at a primitive segment. (a) A sequence of consecutive primitive segments (the upward arrow points to current primitive segment and the leftward arrow points to the direction of segments combination to generate candidate patterns), (b) search space expansion at the pointed primitive segment of (a) (the pruned nodes are labeled).

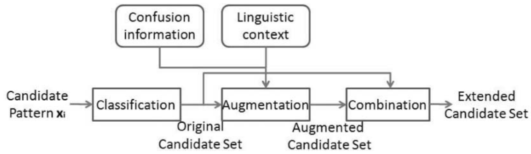


Fig. 5. Diagram of candidate character augmentation.

refined beam search algorithm was presented in our previous work (called pruned DP there) [10], which is suitable for using high-order context models.

After oversegmentation, the text line image is represented as a sequence of primitive segments. A candidate pattern composed of k consecutive segments and ending at the ith segment is denoted by (i, k). A node in the search space is represented as a quadruple $SN = \{CP, CC, AS, PN\}$, where SN denotes a search node, CP is a candidate pattern, CC is a candidate character of CP, and AS is the accumulated score from the root node (calculated by (11)-(14), where m is the length of the current partial path), and PN is a pointer to the parent node of SN. All nodes are stored in a list named **LIST** to backtrack the final path. The refined beam search process is described in detail as follows, and Fig. 4 gives an illustrative example.

Refined Beam Search in frame-synchronous fashion:

1. Initialize the first search node (i.e., the root) of **LIST**, $SN_0 = \{null, null, 0, null\}$, set $i = 1$.
2. Generate nodes of $CP = (i, k)$ over k (the second level nodes in Fig. 4b, $i - k \geq 0$, $k \leq K$, K is the maximum number of segments to be concatenated). For each CP, the top CN (Candidate Number) candidate characters are assigned by the character classifier (the third level nodes in Fig. 4b). In total, at most $K \times CN$ nodes are generated.
3. Link to parent nodes for current nodes ($CP = (i, k)$, $CC = c_{i,k}$). For multiple such parent nodes ($CP' = (i - k, k')$, $CC' = c_{i-k,k'}$), the current node generates multiple copies, each linked to a respective parent node (PN) and associated to an accumulated score (AS) (the fourth level nodes in Fig. 4b). In these copies, only the node with maximum AS over $(k', c_{i-k,k'})$ is retained (the fifth level nodes in Fig. 4b).
4. Sort the retained nodes in above in decreasing order according to AS over $(k, c_{i,k})$, and the leading BW (Beam Width) nodes are retained and added to **LIST**, while the others are pruned to accelerate search.
5. Set $i = i + 1$, back to Step 2 and iterate until the last primitive segment is reached (such nodes called terminal nodes).
6. Backtrack the terminal node in **LIST** of maximum score along the element PN, and obtain the result character string.

We can see that if $BW = K \times CN$, the above algorithm guarantees finding the optimal path for context models up to order 2 when the principle of optimality is satisfied, i.e., it is equivalent to DP. For context models of order 3 (e.g., character trigram) or higher, it does not guarantee finding the optimal path but significantly accelerates search compared to DP. Further, if $BW < K \times CN$, the search procedure is further accelerated. Compared to simple beam search, the two-step pruning strategy in the refined beam search algorithm has at least two advantages: 1) The first step pruning (in Step 3) observes the principle of optimality; 2) sorting the nodes has lower complexity.

If we use word-level n-grams, the search process works on a word candidate lattice, which is constructed from character lattice by combining several consecutive characters according to the word lexicon. So, search in the word candidate lattice is very complex [10]. To accelerate this search process, we first prune the original character lattice using the above character search process (many nodes are pruned in Steps 3 and 4), then use it to construct a succinct word lattice.

6.2 Candidate Character Augmentation

The character classifier assigns a number of candidate classes to each candidate pattern with the risk of missing the true class. In Chinese handwriting recognition, even assigning hundreds of classes cannot guarantee 100 percent inclusion of the true class. Therefore, we propose a Candidate Character Augmentation (**CCA**) method, as diagramed in Fig. 5, to supplement candidate classes during search.

The CCA method exploits both confusion information of the classifier and linguistic context. First, a candidate pattern $\mathbf{x}_i$ is classified to assign a number of candidate classes, called the Original Candidate Set (OCS). Then, the confusion information and the linguistic context are used to supplement two types of candidate classes, forming the Augmented Candidate Set (ACS). Last, the Extended Candidate Set (ECS), as the union of the OCS and the ACS, is used to generate candidate nodes at Step 2 of the search process.

To predict the true class from the OCS by confusion information, we calculate the probability of the hypothesized truth class ω_t given an output class ω_o:

$$p(\omega_t|\omega_o) = \frac{p(\omega_o|\omega_t)p(\omega_t)}{\sum_{\omega_t} p(\omega_o|\omega_t)p(\omega_t)}, \tag{26}$$

where $p(\omega_o|\omega_t) = n_{t,o}/\sum_{\omega_o} n_{t,o}$, $n_{t,o}$ is the number of times that characters of class ω_t are classified as ω_o, counted on a validation data set, and the prior probability $p(\omega_t)$ is usually regarded as equal for all classes. According to (26), we select several top most likely truth classes for each output class ω_o in OCS, and all these likely truth classes form the first ACS.

For augmenting candidate characters from the linguistic context, we use three strategies: forward character bigram, backward character bigram, and bigram cache model. In forward character bigram, we predict the character with the maximum probability:

$$c_i^* = \arg\max_{c_i} \ p(c_i|c_{i-1}), \tag{27}$$

where c_{i-1} is an immediately preceding character in the search space, while Kigo [47] used all characters whose preceding character is c_{i-1} in his smaller bigram table. Similarly, we can predict characters by the backward character bigram:

$$c_i^* = \arg\max_{c_i} \ p(c_i|c_{i+1}), \tag{28}$$

where $p(c_i|c_{i+1}) = p(c_{i+1}|c_i)p(c_i)/p(c_{i+1})$ and the character c_{i+1} is from the immediately succeeding OCS. For prediction from the cache model, we assume that a document covers a single topic, such that character sequence is likely to repeat in the document. Accordingly, the candidate character is predicted by both forward and backward bigram: $\{c_i : (c_{i-1}, c_i) \in \text{cache } \mathit{or} \ (c_i, c_{i+1}) \in \text{cache}\}$, where the cache is the history text (the best result string until the current character pattern of the document).

7 Maximum Character Accuracy Training

Since the parameters of multicontext models in path evaluation function are estimated in advance, the object of training is to tune the combining weights to optimize the recognition performance. To do this, we optimize a Maximum Character Accuracy (**MCA**) criterion similar to the Minimum Word Error (**MWE**) [42] in speech recognition. MCA is a smoothed approximation to the accuracy of the R string samples (text line images) in the training data set:

$$\max \Psi(\Lambda) = \frac{1}{R}\sum_{r=1}^{R}\sum_{j=1}^{N_r} P_\Lambda\big(C_r^j|X_r\big)A\big(C_r^j, T_r\big), \tag{29}$$

where N_r is the number of all segmentation-recognition paths in the lattice of the rth text line image X_r, and C_r^j is the character sequence of the jth path. The term $A(C_r^j, T_r)$ is the character accuracy score, which equals the number of characters in the ground-truth transcript T_r minus the number of errors in C_r^j (including substitution, insertion, and deletion errors, see Section 8.2). Note that the posterior probability $P_\Lambda(C_r^j|X_r)$ can be computed by

$$P_\Lambda\big(C_r^j|X_r\big) = \frac{\exp\left[\xi f_\Lambda\big(X_r^j, C_r^j\big)\right]}{\sum_{i=1}^{N_r}\exp\left[\xi f_\Lambda\big(X_r^i, C_r^i\big)\right]}, \tag{30}$$

where ξ is a scaled constant value, and $f_\Lambda(X_r^j, C_r^j)$ can be any path evaluation function in (11)-(14) under the weights set Λ. MCA is degenerated to MCE [41] if the character accuracy score is calculated by $A(C_r^j, T_r) = \delta(C_r^j, T_r) \in \{0, 1\}$ [43].

We optimize the MCA object (29) by stochastic gradient ascent method. However, the gradients are difficult to calculate precisely due to the huge number N_r; moreover, the precise calculation of $A(C_r^j, T_r)$ needs a completed path. Therefore, we only consider the top N paths of maximum evaluation score while viewing the probabilities of the remaining paths as zero.

8 Experimental Results

We evaluated the performance of our approach on a large database of unconstrained Chinese handwriting, CASIA-HWDB [21], and on a small data set, HIT-MW [54].

8.1 Database and Experimental Setting

The CASIA-HWDB database contains both isolated characters and unconstrained handwritten texts, and is divided into a training set of 816 writers and a test set of 204 writers. The training set contains 3,118,477 isolated character samples of 7,356 classes (7,185 Chinese characters, 109 frequently used symbols, 10 digits, and 52 English letters) and 4,076 pages of handwritten texts. The text pages have a few miswritten characters and characters beyond the 7,356 classes, which we call noncharacters and outlier characters, respectively. The characters in the training text pages (except for the noncharacters and outlier characters, 1,080,017 samples) were also segmented and used together with the isolated samples for training the character classifier. We evaluated the text line recognition performance on the 1,015 handwritten pages of 204 test writers, which were segmented into 10,449 text lines containing 268,629 characters (including 723 noncharacters and 368 outlier characters).

To compare our results with those reported in the literature [9], [10], [11], we also tested on the data set HIT-MW [54], from which a test set of 383 text lines contains 8,448 characters (7,405 Chinese characters, 780 symbols, 230 digits, eight English letters, 16 noncharacters, and nine outlier characters).

To build the character classifier, we extract features from gray-scale character images (background eliminated) using the normalization-cooperated gradient feature (NCGF) method [55]. Before feature extraction, the gray levels of foreground pixels in each image are normalized to a standard mean and deviation. The 512D feature vector obtained is reduced to 160D by Fisher linear discriminant analysis (FLDA), and then input into a modified quadratic discriminant function classifier. The classifier parameters were learned on 4/5 samples of the training set, and the remaining 1/5 samples were used for confidence parameter estimation and confusion matrix construction. For parameter estimation of the geometric models, we extracted geometric features from 41,781 text lines of training text pages. The statistical language models were trained on a text corpus containing about 50 million characters (about 32 million words) [10]. On obtaining the context models, the combining weights of path evaluation function were learned on 300 training text pages.

Table 1 shows some statistics of character samples segmented from the test text pages of CASIA-HWDB. The "number" row gives the numbers of different types of characters (including noncharacters and outlier characters). We can see that the majority of segmented characters are

TABLE 1
Statistics of Character Types, Recognition, and Segmentation Correct Rates on the Test Set

	All	Chinese	symbol	digit	letter	non	outlier
number	268,629	233,329	26,583	6,879	747	723	368
rec (%)	83.78	87.28	60.34	69.40	77.24	0	0
rec20 (%)	98.24	98.55	99.36	98.90	97.86	0	0
rec200 (%)	99.18	99.58	99.64	99.40	98.93	0	0
seg (%)	95.54	95.69	96.84	86.97	83.53	94.05	92.66

Chinese characters, and the number of symbols (mostly punctuation marks) is appreciable. Some samples of non-characters and outlier characters are shown in Figs. 6a and 6b, respectively. The "rec" row gives the correct rate of the segmented character recognition by the character classifier, and "rec20" and "rec200" are the cumulative accuracies of top 20 and 200 ranks, respectively. We can see that the correct rate of Chinese character is highest (87.28 percent). The overall correct rate, 83.78 percent, is lower because of the low correct rates of symbols, digits, and letters. The noncharacters and outlier characters cannot be recognized by the classifier, which covers a defined character set of 7,356 classes.

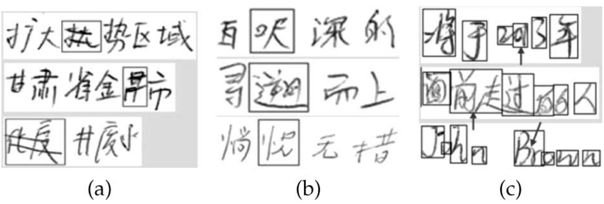
Table 1 (the last row, "seg") also shows the accuracy of oversegmentation (a character is correctly oversegmented when it is separated from other characters despite the within-character splits). We observe that 4.46 percent of characters were not correctly separated (i.e., they are undersegmented and cannot be correctly segmented and recognized by the subsequent character string recognition). This implies that the oversegmentation of characters is still a challenge. Some examples of such errors are shown in Fig. 6c.

8.2 Performance Metrics

We evaluate the recognition performance of text lines using two character-level accuracy metrics following [9]: Correct Rate (**CR**) and Accurate Rate (**AR**):

$$\begin{aligned} CR &= (N_t - D_e - S_e)/N_t, \\ AR &= (N_t - D_e - S_e - I_e)/N_t, \end{aligned} \tag{31}$$

where N_t is the total number of characters in the transcript. The numbers of substitution errors (S_e), deletion errors (D_e), and insertion errors (I_e) are calculated by aligning the recognition result string with the transcript by dynamic programming. The metric CR denotes the percentage of characters that are correctly recognized. Further, the metric AR considers the number of characters that are inserted due to oversegmentation, and is possibly negative. Vinciarelli et al. [51] suggested that the AR (called recognition rate there) is an appropriate measure for document transcription, while CR (called accuracy rate there) is a good metric for tasks of content modeling (e.g., document retrieval). For analyzing the performance on different types of characters, we also give the CR for four types: Chinese characters (**ch**), symbols (**sb**), digits (**dg**), and letters (**lt**).



(a) (b) (c)

Fig. 6. (a) Noncharacters (in small boxes), (b) outlier characters (in small boxes), (c) oversegmentation errors (indicated by the arrows).

TABLE 2
Recognition Results of Different Path Evaluation Functions

	AR (%)	CR (%)	ch (%)	sb (%)	dg (%)	lt (%)	time (h)
w/o	74.63	74.72	77.97	57.45	46.10	46.85	12.46
WIP	88.96	89.68	91.31	81.77	80.69	74.97	11.88
NPL	89.07	90.67	92.29	83.07	81.31	75.64	11.65
WSN	89.54	90.49	92.17	82.27	81.25	76.04	11.85
WCW	90.20	90.80	92.94	79.10	79.63	74.43	11.73

8.3 Text Line Recognition Results

We evaluated the effects of different techniques. First, we compared the effects of different path evaluation functions. Second, the effects of different confidence transformation methods, combinations of geometric models and language models were evaluated. Last, we show the results of different numbers of candidate character classes, beam widths, and candidate character augmentation methods in path search. We report the recognition rates of different techniques on the CASIA-HWDB test set, and give the processing time on all test pages (1,015 pages) consumed on a desktop computer of 2.66 GHz CPU, programming using Microsoft Visual C++. With several selected combinations of techniques, we also report results on the HIT-MW test set.

8.3.1 *Comparing Path Evaluation Functions*

In evaluating the effects of path evaluation functions and CT methods, the character trigram language model and all geometric models were used. The search algorithm was the refined beam search with $K = 4, CN = 20$, and $BW = 10$, but CCA methods were not used in the search process. In evaluating the path evaluation functions, the D-S evidence confidence was taken. The recognition results of different path evaluation functions (11)-(14) are shown in Table 2, where "w/o" denotes the path evaluation function without word insertion penalty ((11) removing the last term). We can see that by considering the balance of path length using different heuristics, the string recognition performance is largely improved. Among the four strategies, the one of weighting with character width performs best with respect to both AR and CR. The normalized path function gives a little lower CR but significantly lower AR. This is because NPL tends to generate more oversegmentation. The performance of weighting with primitive segment number is higher than that of NPL, but lower than that of WCW. We hence used the strategy WCW for all the following experiments.

8.3.2 *Comparing CT Methods*

Table 3 shows the results of different CT methods for character classifier introduced in Section 5.1. Compared to the recognition without CT ("w/o" row, it means that lp_i^j, $j = 0, \ldots, 4$, take the classifiers similarity outputs directly in WCW function (14)), the sigmoidal confidence improves the AR from 83.60 to 89.42 percent and CR from 85.52 to 90.19 percent; the D-S evidence improves AR from 83.60 to 90.20 percent and CR from 85.52 to 90.80 percent. The soft-max

TABLE 3
Effects of Different CT Methods

	AR (%)	CR (%)	ch (%)	sb (%)	dg (%)	lt (%)	time (h)
w/o	83.60	85.52	87.61	74.18	75.03	57.70	10.95
sigmoidal	89.42	90.19	92.68	75.32	79.58	73.63	11.87
soft-max	74.67	74.75	79.39	49.87	31.53	17.80	12.20
D-S	90.20	90.80	92.94	79.10	79.63	74.43	11.73

confidence performs inferiorly, however, because it does not consider the outlier probability. The benefit of CT (particularly, sigmoidal and D-S evidence) is attributed to the fact that the converted posterior probabilities (character classification and geometric models) and the statistical language model are more compatible to be combined. We thus used the D-S evidence confidence in the other experiments by default.

8.3.3 Comparing Context Models

The effects of different combinations of context models are shown in Table 4, where "**cls**," "**cg**," "**ig**," "**g**," and "**cti**" denote the character classifier, the class-dependent geometric models ("ucg+bcg"), the class-independent geometric models ("uig+big"), all geometric models ("cg+ig"), and the character trigram language model, respectively. We can see when using the character classifier only ("cls"), the string recognition performance is inferior. Adding geometric models to the classifier, the string recognition performance is remarkably improved. By combining four geometric models, the AR is improved from 47.89 to 77.34 percent and the CR is improved from 68.52 to 79.43 percent. It is observed that the binary geometric models yield larger improvement than the unary models. This justifies the importance of the between-character relationship. Also, the class-dependent geometric models ("cls+cg") perform better than the class-independent geometric models ("cls+ig"). Compared to the geometric models, the statistical language model ("cls+cti") is much more effective to yield a large improvement of AR and CR. Further, the combination of both geometric and language models to the character classifier yields the best recognition result, justifying that geometric context and linguistic context are complementary.

Based on the character classifier and geometric models, we then evaluated different language models: character bigram ("**cbi**"), character trigram ("**cti**"), word bigram ("**wbi**"), word class bigram ("**wcb**"), interpolating word

TABLE 4
Effects of Different Combinations of Contexts

	AR (%)	CR (%)	ch (%)	sb (%)	dg (%)	lt (%)	time (h)
cls	47.89	68.52	70.63	55.17	59.75	63.72	10.49
cls+ucg	69.74	76.93	78.57	71.49	56.32	58.63	10.83
cls+bcg	75.21	78.81	80.50	72.77	59.62	59.44	11.08
cls+uig	72.27	76.23	80.90	48.72	38.81	52.07	10.64
cls+big	74.00	77.08	81.09	54.15	44.19	54.22	10.72
cls+cg	75.70	79.02	80.79	72.59	58.86	55.96	11.36
cls+ig	74.37	76.94	81.62	50.02	37.40	49.80	10.52
cls+g	77.34	79.43	81.95	69.69	46.69	55.69	11.33
cls+cti	89.03	90.24	92.29	78.34	82.63	75.10	11.03
cls+cg+cti	89.98	90.56	92.66	78.49	81.93	74.30	11.99
cls+ig+cti	89.63	90.31	92.70	77.64	74.52	72.96	11.08
cls+g+cti	90.20	90.80	92.94	79.10	79.63	74.43	11.73

TABLE 5
Effects of Different Language Models

	AR (%)	CR (%)	ch (%)	sb (%)	dg (%)	lt (%)	time (h)
w/o	77.34	79.43	81.95	69.69	46.69	55.69	11.33
cbi	89.56	90.27	92.40	78.67	78.47	74.97	11.54
cti	90.20	90.80	92.94	79.10	79.63	74.43	11.73
wbi	90.30	90.94	93.08	79.61	78.75	71.22	12.02
wcb	90.07	90.77	92.73	80.64	79.60	73.76	12.33
iwc	90.53	91.17	93.21	80.45	79.75	72.96	12.70

and class bigram ("**iwc**"). The recognition results are shown in Table 5, where "w/o" denotes recognition without language model. We can see that the character trigram outperforms the character bigram significantly. The advantage of trigram is due to its capturing long-distance text dependency. The extension to character 4-gram is not trial due to the high complexity, however. Even the modeling of third-order word dependency is intractable due to the huge number of words. The use of second-order word dependency models nevertheless shows promise: The "wbi" and "wcb" both perform comparably with the "cti." Further, by interpolating the word-lavel bigram models, the "iwc" yields the best recognition performance.

8.3.4 Comparing Search Strategies

The above experiments used the default number of 20 candidate classes assigned to each candidate pattern, refined beam search with beam width 10 was used, and the CCA techniques were not used. We then evaluated the effects of different candidate class numbers (CN), beam widths (BW), and CCA techniques based on the combinations of geometric models and character trigram language model in the path evaluation function of WCW with the D-S evidence confidence. Figs. 7a and 7c show the effects of different CN of refined beam search algorithm, and Figs. 7b and 7d show the effects of different BW of both refined and simple beam search methods. Compared to the simple beam search, our refined beam search algorithm yields much higher recognition accuracy at comparable speed. We can also see that the number of 20 candidate classes and the beam width 10 perform

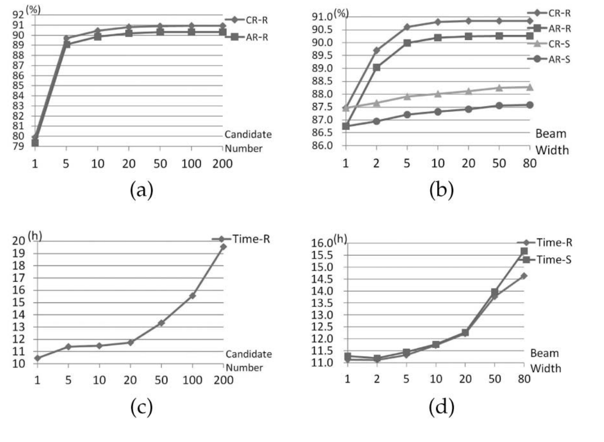


Fig. 7. Accuracies of different CN and BW in Steps 2 and 4 of two beam search methods ("-R" and "-S" denote refined and simple beam search, respectively). (a) Accuracies of different CN with $BW = 10$, (b) accuracies of different BW with $CN = 20$, (c) processing times of (a), (d) processing times of (b).

TABLE 6
Results of CCA Techniques

CCA	AR (%)	CR (%)	ch (%)	sb (%)	dg (%)	lt (%)	time (h)
w/o	90.20	90.80	92.94	79.10	79.63	74.43	11.73
ci	90.19	90.79	92.96	79.05	78.79	74.70	14.17
fcb	90.27	90.87	93.04	79.13	78.72	74.43	13.54
bcb	90.24	90.84	93.01	79.14	78.79	74.03	13.57
bcm	90.27	90.87	93.04	79.11	78.94	74.43	13.01
lc	90.35	90.95	93.13	79.11	78.85	73.76	16.17
cca	90.35	90.95	93.14	79.04	78.83	74.16	17.90
iwc+cca	**90.75**	**91.39**	93.46	80.45	79.76	72.69	18.78

sufficiently well with respect to the recognition accuracy and the speed of the refined beam search. Increasing CN and BW, though it improves the coverage of correct path, does not improve the recognition accuracy. This is because the search algorithm does not guarantee finding the correct path in case of a large number of candidate paths due to the insufficient quantitative evaluation of paths.

Table 6 shows the effects of different candidate character augmentation techniques, namely, confusion information ("**ci**"), forward character bigram ("**fcb**"), backward character bigram ("**bcb**"), bigram cache model ("**bcm**"), combination of three CCA techniques based on linguistic context ("**lc,**" i.e., "fcb+bcb+bcm"), and combination of all CCA techniques ("**cca**", i.e., "ci+lc"). It is shown that compared to recognition without CCA ("w/o"), the CCA techniques yield only slight improvement of recognition performance. Particularly, augmentation by confusion information makes almost no difference. This is because many noisy candidate characters are also added by CCA techniques. To get the best performance by the proposed approach, we evaluated the effects of combining all CCA techniques based on the "iwc" language model, and the results are shown in the last row of Table 6 (others are based on the "cti" language model). Compared to the result of "iwc" without CCA in Table 5, CCA improves the AR from 90.53 to 90.75 percent and the CR from 91.17 to 91.39 percent.

8.3.5 Performance on the HIT-MW Test Set

Finally, we show the recognition results of our approach on the HIT-MW test set. To apply our character classifier trained with gray-scale character images to the binary images of HIT-MW, we converted the binary images to gray-scale images of two levels: 0 for background pixels and G for foreground pixels (G is the mean value in gray-level normalization for gray-scale images). For evaluating the effects of several representative context models on this test set, we used the path evaluation function of WCW with D-S evidence confidence, and the search algorithm was the refined beam search with $K = 4$, $CN = 20$, and $BW = 10$. The recognition results are shown in Table 7. We can see that both geometric models and language models improve the performance largely, and the best performance is achieved by combining all the contexts and the CCA methods ("cls + g + iwc + cca").

Compared to the previous results reported on this test set, 34.64 percent of AR and 39.37 percent of CR in [9], 73.97 percent of CR in [11], and 78.44 percent of RCR (similar to CR, but matched by character boundaries) in [10], the proposed approach achieved 91.86 percent of AR and 92.72 percent of CR, demonstrating significant improvement and advantage.

TABLE 7
Recognition Results on the HIT-MW Test Set

model	AR (%)	CR (%)	ch (%)	sb (%)	dg (%)	lt (%)	time (h)
cls	40.70	66.16	68.25	51.28	56.96	50.00	0.32
cls+g	77.89	80.85	83.66	67.82	44.35	50.00	0.34
cls+cti	89.95	91.52	93.67	76.15	84.78	87.50	0.33
cls+g+cti	91.44	92.22	94.68	76.41	77.39	75.00	0.35
cls+g+iwc	91.68	92.57	94.81	77.69	81.30	75.00	0.39
cls+g+iwc +cca	91.86	92.72	95.02	77.44	81.30	75.00	0.54

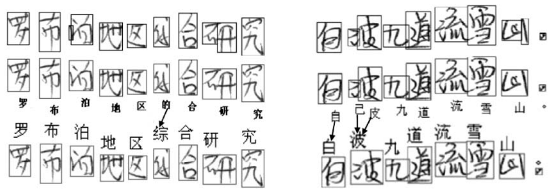
8.3.6 Examples of Recognition Errors

The string recognition errors of our approach can be categorized into three types: 1) oversegmentation failure (undersegmentation), 2) character classification error, including the failure for noncharacters and outlier class, 3) path search failure. In Table 1, we can see that 4.46 percent of characters were not correctly separated by oversegmentation. Character classification error (about 1.53 percent of characters when $CN = 20$) implies that the truth class of candidate pattern is missed in the top CN ranks so that the candidate paths miss the correct one. Path search failure (about 2.62 percent of characters when $K = 4$, $CN = 20$, $BW = 10$) is the case where even though the correct path is included in the candidate paths, it is not the "optimal" path with maximum score due to the imperfect evaluation of paths.

Some examples of noncharacters, outliers, and oversegmentation errors have been shown in Fig. 6. In addition, two examples of character classification error and path search failure are shown in Fig. 8. In Fig. 8a, the misclassified character (indicated by the arrow) was written as a scrawl and the classifier failed to assign the correct class in the top 20 ranks and the CCA techniques failed to pick up the correct class. In Fig. 8b, the first character was misclassified, while the second character was missegmented into two characters; both are due to the effect of language model because the general-purpose language model does not suit the context of this text of ancient poem.

9 CONCLUSION

This paper presented an approach for handwritten Chinese text recognition under the character oversegmentation and candidate path search framework. We evaluate the paths from the Bayesian decision view by combining multiple contexts, including the character classification scores, geometric and linguistic contexts. The combining weights of path evaluation function are optimized by a string



(a) (b)

Fig. 8. Two examples of recognition errors; only the part with recognition error is shown. (a) Character classification error, (b) path search failure. Upper: oversegmentation,; middle: segmentation-recognition result, bottom: ground truth.

recognition objective, namely, the Maximum Character Accuracy criterion. In path search, we use a refined beam search algorithm to improve the accuracy and efficiency. In experiments on the unconstrained Chinese handwriting database CASIA-HWDB, the proposed approach achieved the character-level accurate rate of 90.75 percent and correct rate of 91.39 percent. The experimental results justify the benefits of confidence transformation of classifier outputs, geometric context models, and language models. Nevertheless, the effect of candidate character augmentation is limited. We also evaluated performance on the HIW-MW test set and achieved an accuracy rate of 91.86 percent and correct rate of 92.72 percent, which are significantly higher than those reported in the literature.

The analysis of recognition errors indicates that further research efforts are needed to improve the character oversegmentation, character classification, and path evaluation. The objective of oversegmentation is to improve the tradeoff between the number of splitting points (affecting the complexity of search space) and the accuracy of separating characters at their boundaries. The objective of character classification is to improve the classification accuracy and the tradeoff between the number of candidate classes and the probability of including the true class. For path evaluation, both the geometric model and the language model deserve elaboration. Particularly, our experimental results show that mismatch of language model and text domain leads to inferior recognition performance. Therefore, the domain adaptation of language model will be an important research direction. In addition, the real semantic context and long-distance context will also be considered in the future.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) under Grants 60825301 and 60933010.

References

[1] R.-W. Dai, C.-L. Liu, and B.-H. Xiao, "Chinese Character Recognition: History, Status and Prospects," *Frontiers of Computer Science in China,* vol. 1, no. 2, pp. 126-136, 2007.

[2] H. Fujisawa, "Forty Years of Research in Character and Document Recognition—An Industrial Perspective," *Pattern Recognition,* vol. 41, no. 8, pp. 2435-2446, Aug. 2008.

[3] C.-L. Liu, F. Yin, D.-H. Wang, and Q.-F. Wang, "Online and Offline Handwritten Chinese Character Recognition: Benchmarking on New Databases," *Proc. Second CJK Joint Workshop Pattern Recognition,* Oct. 2010.

[4] H.-S. Tang, E. Augustin, C.Y. Suen, O. Baret, and M. Cheriet, "Spiral Recognition Methodology and Its Application for Recognition of Chinese Bank Checks," *Proc. Ninth Int'l Workshop Frontiers in Handwriting Recognition,* pp. 263-268, Oct. 2004.

[5] C.-L. Liu, M. Koga, and H. Fujisawa, "Lexicon-Driven Segmentation and Recognition of Handwritten Character Strings for Japanese Address Reading," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 11, pp. 1425-1437, Nov. 2002.

[6] C.-H. Wang, Y. Hotta, M. Suwa, and S. Naoi, "Handwritten Chinese Address Recognition," *Proc. Ninth Int'l Workshop Frontiers in Handwriting Recognition,* pp. 539-544, Oct. 2004.

[7] Z. Han, C.-P. Liu, and X.-C. Yin, "A Two-Stage Handwritten Character Segmentation Approach in Mail Address Recognition," *Proc. Eigth Int'l Conf. Document Analysis and Recognition,* pp. 111-115, Aug. 2005.

[8] Q. Fu, X.-Q. Ding, T. Liu, Y. Jiang, and Z. Ren, "A Novel Segmentation and Recognition Algorithm for Chinese Handwritten Address Character Strings," *Proc. 18th Int'l Conf. Pattern Recognition,* pp. 974-977, Aug. 2006.

[9] T.-H. Su, T.-W. Zhang, D.-J. Guan, and H.-J. Huang, "Off-Line Recognition of Realistic Chinese Handwriting Using Segmentation-Free Strategy," *Pattern Recognition,* vol. 42, no. 1, pp. 167-182, 2009.

[10] Q.-F. Wang, F. Yin, and C.-L. Liu, "Integrating Language Model in Handwritten Chinese Text Recognition," *Proc. 10th Int'l Conf. Document Analysis and Recognition,* pp. 1036-1040, July 2009.

[11] N.-X. Li and L.-W. Jin, "A Bayesian-Based Probabilistic Model for Unconstrained Handwritten Offline Chinese Text Line Recognition," *Proc. IEEE Int'l Conf. Systems, Man, and Cybernetics,* pp. 3664-3668, 2010.

[12] M. Nakagawa, B. Zhu, and M. Onuma, "A Model of On-Line Handwritten Japanese Text Recognition Free from Line Direction and Writing Format Constraints," *IEICE Trans. Information and Systems,* vol. 88, no. 8, pp. 1815-1822, Aug. 2005.

[13] X.-D. Zhou, J.-L. Yu, C.-L. Liu, T. Nagasaki, and K. Marukawa, "Online Handwritten Japanese Character String Recognition Incorporating Geometric Context," *Proc. Ninth Int'l Conf. Document Analysis and Recognition,* pp. 48-52, Sept. 2007.

[14] X.-D. Zhou, C.-L. Liu, and M. Nakagawa, "Online Handwritten Japanese Character String Recognition Using Conditional Random Fields," *Proc. 10th Int'l Conf. Document Analysis and Recognition,* pp. 521-525, July 2009.

[15] B. Zhu, X.-D. Zhou, C.-L. Liu, and M. Nakagawa, "A Robust Model for On-Line Handwritten Japanese Text Recognition," *Int'l J. Document Analysis and Recognition,* vol. 13, no. 2, pp. 121-131, 2010.

[16] M. Cheriet, N. Kharma, C.-L. Liu, and C.Y. Suen, *Character Recognition Systems: A Guide for Students and Practitioners.* John Wiley & Sons, Inc., 2007.

[17] H. Murase, "Online Recognition of Free-Format Japanese Handwritings," *Proc. Ninth Int'l Conf. Pattern Recognition,* pp. 1143-1147, 1988.

[18] Y. Jiang, X. Ding, Q. Fu, and Z. Ren, "Context Driven Chinese String Segmentation and Recognition," *Proc. Structural, Syntactic, and Statistical Pattern Recognition: Joint IAPR Int'l Workshops,* pp. 127-135, 2006.

[19] X. Ding and H. Liu, "Segmentation-Driven Offline Handwritten Chinese and Arabic Script Recognition," *Proc. Conf. Arabic and Chinese Handwriting,* pp. 61-73, 2006.

[20] S. Senda and K. Yamada, "A Maximum-Likelihood Approach to Segmentation-Based Recognition of Unconstrained Handwriting Text," *Proc. Sixth Int'l Conf. Document Analysis and Recognition,* pp. 184-188, Sept. 2001.

[21] C.-L. Liu, F. Yin, D.-H. Wang, and Q.-F. Wang, "CASIA Online and Offline Chinese Handwriting Databases," *Proc. 11th Int'l Conf. Document Analysis and Recognition,* pp. 37-41, Sept. 2011.

[22] L.Y. Tseng and R.C. Chen, "Segmenting Handwritten Chinese Characters Based on Heuristic Merging of Stroke Bounding Boxes and Dynamic Programming," *Pattern Recognition Letters,* vol. 19, no. 10, pp. 963-973, Aug. 1998.

[23] Z. Liang and P. Shi, "A Metasynthetic Approach for Segmenting Handwritten Chinese Character Strings," *Pattern Recognition Letters,* vol. 26, no. 10, pp. 1498-1511, July 2005.

[24] C.-L. Liu, "Handwritten Chinese Character Recognition: Effects of Shape Normalization and Feature Extraction," *Proc. Conf. Arabic and Chinese Handwriting Recognition,* S. Jaeger and D. Doermann, eds., pp. 104-128, 2008.

[25] C.-L. Liu and H. Fujisawa, "Classification and Learning in Character Recognition: Advances and Remaining Problems," *Machine Learning in Document Analysis and Recognition,* S. Marinai and H. Fujisawa, eds., pp. 139-161, Springer, 2008.

[26] F. Kimura, K. Takashina, S. Tsuruoka, and Y. Miyake, "Modified Quadratic Discriminant Functions and the Application to Chinese Character Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 9, no. 1, pp. 149-153, Jan. 1987.

[27] C.-L. Liu and M. Nakagawa, "Evaluation of Prototype Learning Algorithms for Nearest Neighbor Classifier in Application to Handwritten Character Recognition," *Pattern Recognition,* vol. 34, no. 3, pp. 601-615, Mar. 2001.

[28] X. Lin, X. Ding, M. Chen, R. Zhang, and Y. Wu, "Adaptive Confidence Transform Based on Classifier Combination for Chinese Character Recognition," *Pattern Recognition Letters,* vol. 19, no. 10, pp. 975-988, Aug. 1998.

[29] C.-L. Liu, "Classifier Combination Based on Confidence Transformation," *Pattern Recognition,* vol. 38, no. 1, pp. 11-28, 2005.

[30] Y.X. Li, C.L. Tan, and X.Q. Ding, "A Hybrid Post-Processing System for Offline Handwritten Chinese Script Recognition," *Pattern Analysis and Applications,* vol. 8, no. 3, pp. 272-286, 2005.

[31] Q.-F. Wang, F. Yin, and C.-L. Liu, "Improving Handwritten Chinese Text Recognition by Confidence Transformation," *Proc. 11th Int'l Conf. Document Analysis and Recognition,* pp. 518-522, Sept. 2011.

[32] R. Rosenfeld, "Two Decades of Statistical Language Modeling: Where Do We Go from Here?" *Proc. IEEE,* vol. 88, no. 8, pp. 1270-1278, Aug. 2000.

[33] R.F. Xu, D.S. Yeung, and D.M. Shi, "A Hybrid Post-Processing System for Offline Handwritten Chinese Character Recognition Based on a Statistical Language Model," *Int'l J. Pattern Recognition and Artificial Intelligence,* vol. 19, no. 3, pp. 415-428, 2005.

[34] M. Koga, T. Kagehiro, H. Sako, and H. Fujisawa, "Segmentation of Japanese Handwritten Characters Using Peripheral Feature Analysis," *Proc. 14th Int'l Conf. Pattern Recognition),* vol. 2, pp. 1137-1141, 1998.

[35] F. Yin, Q.-F Wang, and C.-L. Liu, "Integrating Geometric Context for Text Alignment of Handwritten Chinese Documents," *Proc. 12th Int'l Conf. Frontiers in Handwriting Recognition,* pp. 7-12, Nov. 2010.

[36] S. Tulyakov and V. Govindaraju, "Probabilistic Model for Segmentation Based Word Recognition with Lexicon," *Proc. Sixth Int'l Conf. Document Analysis and Recognition,* pp. 164-167, Sept. 2001.

[37] C.-L. Liu, H. Sako, and H. Fujisawa, "Effects of Classifier Structures and Training Regimes on Integrated Segmentation and Recognition of Handwritten Numeral Strings," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 11, pp. 1395-1407, Nov. 2004.

[38] M. Wuthrich, M. Liwicki, A. Fischer, E. Indermuhle, H. Bunke, G. Viehhauser, and M. Stolz, "Language Model Integration for the Recognition of Handwritten Medieval Documents," *Proc. 10th Int'l Conf. Document Analysis and Recognition,* pp. 211-215, July 2009.

[39] S. Quiniou, M. Cheriet, and E. Anquetil, "Handling Out-of-Vocabulary Words and Recognition Errors Based on Word Linguistic Context for Handwritten Sentence Recognition," *Proc. 10th Int'l Conf. Document Analysis and Recognition,* pp. 466-470, July 2009.

[40] M.-Y. Chen, A. Kundu, and S.N. Srihari, "Variable Duration Hidden Markov Model and Morphological Segmentation for Handwritten Word Recognition," *IEEE Trans. Image Processing,* vol. 4, no. 12, pp. 1675-1688, Dec. 1995.

[41] B.-H. Juang, W. Chou, and C.-H. Lee, "Minimum Classification Error Rate Methods for Speech Recognition," *IEEE Trans. Speech and Audio Processing,* vol. 5, no. 3, pp. 257-265, May 1997.

[42] D. Povey, "Discriminative Training for Large Vocabulary Speech Recognition," PhD dissertation, Cambridge Univ., Cambridge, UK, 2003.

[43] X.-D. He, L. Deng, and W. Chou, "Discriminative Learning in Sequential Pattern Recognition," *IEEE Signal Processing Magazine,* vol. 25, no. 5, pp. 14-36, Sept. 2008.

[44] H. Ney and S. Ortmanns, "Progress in Dynamic Programming Search for LVCSR," *Proc. IEEE,* vol. 88, no. 8, pp. 1224-1240, Aug. 2000.

[45] C.-L. Liu and M. Nakagawa, "Precise Candidate Selection for Large Character Set Recognition by Confidence Evaluation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 6, pp. 636-642, June 2000.

[46] E. Ishidera and A. Sato, "A Candidate Reduction Method for Handwritten Kanji Character Recognition," *Proc. Sixth Int'l Conf. Document Analysis and Recognition,* pp. 8-13, Sept. 2001.

[47] K. Kigo, "Improving Speed of Japanese OCR through Linguistic Preprocessing," *Proc. Second Int'l Conf. Document Analysis and Recognition,* pp. 214-217, 1993.

[48] Y.-X. Li, C.-L. Tan, X.-Q. Ding, and C.-S. Liu, "Contextual Post-Processing Based on the Confusion Matrix in Offline Handwritten Chinese Script Recognition," *Pattern Recognition,* vol. 37, no. 9, pp. 1901-1912, 2004.

[49] F. Yin, Q.-F. Wang, and C.-L. Liu, "A Tool for Ground-Truthing Text Lines and Characters in Off-Line Handwritten Chinese Documents," *Proc. 10th Int'l Conf. Document Analysis and Recognition,* pp. 951-955, July 2009.

[50] J.A. Barnett, "Computational Methods for a Mathematical Theory of Evidence," *Proc. Seventh Int'l Joint Conf. Artificial Intelligence,* pp. 868-875, 1981.

[51] A. Vinciarelli, S. Bengio, and H. Bunke, "Offline Recognition of Unconstrained Handwritten Texts Using HMMs and Statistical Language Models," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 6, pp. 709-720, June 2004.

[52] S. Martin, J. Liermann, and H. Ney, "Algorithms for Bigram and Trigram Word Clustering," *Speech Comm.,* vol. 24, no. 1, pp. 19-37, 1998.

[53] A. Stolcke, "SRILM—An Extensible Language Modeling Toolkit," *Proc. Seventh Int'l Conf. Spoken Language Processing,* pp. 901-904, Sept. 2002.

[54] T.-H. Su, T.-W. Zhang, and D.-J. Guan, "Corpus-Based HIT-MW Database for Offline Recognition of General-Purpose Chinese Handwritten Text," *Int'l J. Document Analysis and Recognition,* vol. 10, no. 1, pp. 27-38, 2007.

[55] C.-L. Liu, "Normalization-Cooperated Gradient Feature Extraction for Handwritten Character Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29, no. 8, pp. 1465-1469, Aug. 2007.

Qiu-Feng Wang received the BS degree in computer science from Nanjing University of Science and Technology, China, in 2006. He is currently working toward the PhD degree in pattern recognition and intelligent systems at the Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests include Handwritten text recognition, language model, and pattern recognition.

Fei Yin received the BS degree in computer science from Xidian University of Posts and Telecommunications, Xi'an, China, the ME degree in pattern recognition and intelligent systems from Huazhong University of Science and Technology, Wuhan, China, the PhD degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 1999, 2002, and 2010, respectively. He is an assistant professor at the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests include document image analysis, handwritten character recognition, and image processing.

Cheng-Lin Liu received the BS degree in electronic engineering from Wuhan University, China, the ME degree in electronic engineering from Beijing Polytechnic University, China, the PhD degree in pattern recognition and intelligent control from the Chinese Academy of Sciences, Beijing, China, in 1989, 1992, and 1995, respectively. He is a professor at the National Laboratory of Pattern Recognition (NLPR), Institute of Automation of Chinese Academy of Sciences, Beijing, China, and is now the deputy director of the laboratory. He was a postdoctoral fellow at the Korea Advanced Institute of Science and Technology (KAIST) and later at Tokyo University of Agriculture and Technology from March 1996 to March 1999. From 1999 to 2004, he was a research staff member and later a senior researcher at the Central Research Laboratory, Hitachi, Ltd., Tokyo, Japan. His research interests include pattern recognition, image processing, neural networks, machine learning, and especially applications to character recognition and document analysis. He has published more than 130 technical papers in prestigious international journals and conferences. He is a senior member of the IEEE.