

Keyword Spotting in Offline Chinese Handwritten Documents Using a Statistical Model

Liang Huang^{1,2}, Fei Yin², Qing-Hu Chen¹, Cheng-Lin Liu²

¹ School of Electronic Information, Wuhan University

39 Luoyu Road, Wuhan 430079, P.R.China

² National Laboratory of Pattern Recognition (NLPR)

Institute of Automation of Chinese Academy of Sciences

95 Zhongguancun East Road, Beijing 100190, P.R. China

Email: hala1579@yahoo.com.cn, cqh@eis.whu.edu.cn, {fyin,liucl}@nlpr.ia.ac.cn

Abstract—This paper proposes a method for keyword spotting in offline Chinese handwritten documents using a statistical model. On a text query word, the method measures the similarity between the query word and every candidate word in the document by combining a character classifier and four classifiers characterizing the geometric contexts. By over-segmenting text lines into primitive segments, candidate characters and words are generated by concatenating consecutive segments, and the beam search strategy is used to search all the candidate words. The character classifier and the model combining weights are trained by optimizing a one-vs-all discrimination objective so as to maximize the similarity of true words and minimize the similarity of imposters. In experiments on a test dataset containing 1,015 pages of 180 writers, the proposed methods yields promising performance. For retrieving three-character words, the recall, precision and F-measure are 86.43%, 69.87% and 77.27%, respectively.

Keywords—Keyword spotting; Chinese handwritten documents; statistical model;

I. INTRODUCTION

Along with the rise of Internet and multimedia communication, the development of technology for efficient document retrieval is increasingly demanded. Despite the many works of document retrieval with OCR, the retrieval of handwritten documents still needs more research efforts to handle the difficulty of layout analysis, character/word segmentation, the large vocabulary and variability of writing styles [1], especially in degraded historical manuscripts [2]. Under these conditions, keyword spotting has been proposed instead of a complete recognition for the restricted task of retrieving words from document images [3].

According to the word similarity scoring technique, two different approaches for word retrieval can be distinguished. Template-based methods match a query image with labeled template images using, e.g., holistic gradient-based binary features (GSC) [4] or local information by means of dynamic time warping (DTW) [5]. Typically, no training is needed for template-based methods. Learning-based methods, on the other hand, train word models for scoring query images or query text. This can also be categorized into two groups:

holistic word model [6] and subword model (also called character-based models) [7][8]. The holistic word model based method is limited in three aspects. Firstly, the words are required to appear in the training set. Secondly, for each word a considerable amount of data is required for training the word model. And thirdly, the method is dependent on word segmentation. When training subword models, e.g., letters rather than words, only a small number of models need to be learned. They can be used to retrieve arbitrary words that are not required to be present in the training set.

Most previous works of Chinese document retrieval have focused on printed documents. We aim to retrieve words from multi-writer handwritten documents using string query. The method can apply to single writer's documents as well. To overcome the difficulties of character and word segmentation, we search the query word from character candidates lattice generated by over-segmentation of text lines. The word similarity is obtained by combining character classifier with geometric context models.

Text search using character segmentation and recognition candidates has been proven effective in Japanese document retrieval [9][10]. These methods are similar to ours, but we have paid more attention to the training of the character classifier for a better similarity measure and geometric context models used to reduce character segmentation and recognition errors.

In this paper, we designed a fused model of geometric context combined with character classifier for word retrieval in Chinese unconstrained handwritten document images. We use four statistical models to evaluate the single-character geometric features and between-character geometric relationships [11]. The geometric models are combined with the character classifier to evaluate the word-to-image matching score and the best match searched by beam search gives the retrieval result. We construct two fusion models, one for character detection, the other for word matching, so the combining weights are optimized by character-level and string-level training respectively. Our experimental results on unconstrained handwritten Chinese text lines show that the

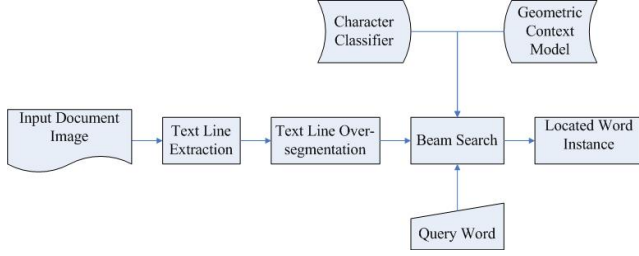


Figure 1. Block diagram of the word retrieval system.

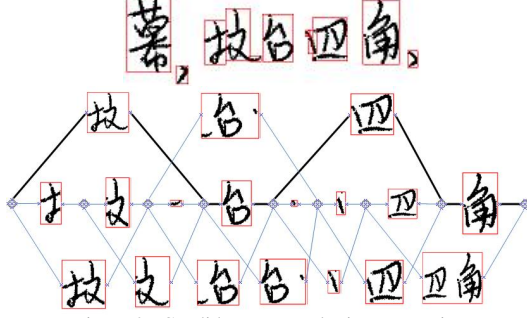


Figure 2. Candidate pattern lattice generation.

geometric models can significantly reduce word matching errors.

II. SYSTEM OVERVIEW

Fig. 1 shows the block diagram of our system for word retrieval in handwritten document image. After being pre-processed, the input document image is segmented into several text lines [12]. Each line is over-segmented into primitive segments. And these consecutive segments are combined to generate candidate character patterns, forming a segmentation candidate lattice [13]. Examples of candidate pattern lattice generation is shown in Fig. 2.

In word retrieval, the query word is matched with sequences of candidate patterns (paths in the candidate lattice) with every primitive segments as the start. The word matching score is obtained by combining the character similarity scores output by character classifier and geometric models. When the word similarity score is greater than a certain threshold, a word instance is located in the document. We use a character-synchronous beam search algorithm [13] for efficient search of query word from sequences of primitive segments. An illustrative example of this search algorithm for retrieval of a word “坟台四角” from sequence of primitive segments is shown as Fig. 3. In Fig. 3, the validated pattern(s) which is matched with the first character “坟” of word “坟台四角” is(are) located at first. And then we search the other characters of the word “坟台四角” using the validated pattern(s) corresponding to the first character as the seed node through character-synchronous beam search [13]. This is the so-called Two-Stage Matching Procedure. Fig. 4 shows an example of retrieval of word “坟台四角” from a document image.

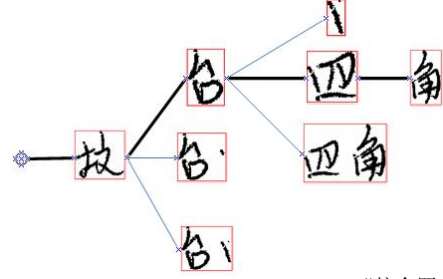


Figure 3. Beam search for locating a word “坟台四角”.

III. WORD SIMILARITY MEASURE

In word retrieval, the query word $W = c_1c_2 \cdots c_n$ is matched with each sequence of candidate patterns $X = x_1x_2 \cdots x_n$ (each pattern represented as multiple feature vectors for different purpose) in the document. The similarity measure is designed such that true words have high scores and the opposite have low scores. According to the Bayesian Decision Theory, the posterior probability $p(W|X)$ is preferred to be used to measure the word similarity. The logarithm arithmetic expression is as below:

$$\log p(W|X) = \log p(X|W) + \log p(W) - \log p(X).$$

On the right side of the formula given above, $\log p(X|W)$ is modeled by character classifier, $\log p(W)$ is the abstraction of language model, but when the query word is given, the output of language model is a constant, so it can be ignored in the procedure of retrieval, $\log p(X)$ can be seen as the abstraction of geometric contextual model. The details of implementation of these two abstract concepts are described as followed.

A. Character Classifier Model

The character classifier is desired to give high similarity to the genuine class of the input pattern and low scores to all the others. On an input non-character pattern, the scores of all classes should be low. We recently proposed an algorithm for training prototype classifier with one-vs-all objective such that the prototypes of each class perform as a binary classifier for separating the class from the others. This property is desirable for retrieval. For comparison, we also use a one-vs-all SVM classifier.

1) *Prototype Classifier*: Recently we proposed a new prototype learning method which decomposes the multi-class classification problem into multiple binary classification problems and train the prototypes using a one-vs-all (OVA) criterion, the cross-entropy (CE) [14].

In the OVA formulation, each prototype m_{ij} (the j -th prototype of class i) functions as a binary discriminant function, and the discriminant function of a class is the maximum over the prototypes:

$$f_i(x) = \max_j f_{ij}(x) = -\min_j (\|x - m_{ij}\|_2 - \tau_{ij}), \quad (1)$$

where τ_{ij} is the threshold for prototype m_{ij} . The training objective is to minimize the cross-entropy on a training

dataset for good convergence. After training the OVA prototype classifier, we only use the class-specific discriminant function as the similarity:

$$\text{sim}(c_i, x) = f_i(x). \quad (2)$$

2) *SVM Classifier*: Considering the large number of Chinese characters, we use a one-vs-all trained SVM classifier with linear kernel. The similarity of each class is a linear discriminant function:

$$f_i(x) = w_i^T x - \tau_i. \quad (3)$$

The weight vector and the threshold of each class are calculated from support vectors selected by large margin learning. For the large number of classes and huge sample set, we use the successive overrelaxation(SOR) algorithm of [15] for SVM training.

3) *Character Feature Extraction*: For character extraction of candidate character patterns, we use the gradient direction histogram feature, which has been popularly used in offline character recognition. Particularly, we adopt the method of [18] for gradient direction feature extraction from offline characters. The gradient elements are decomposed into 8 directions and each direction is extracted 8×8 values by Gaussian blurring. To reduce the complexity of classifier, the 512D feature vector is projected onto a 160D subspace learned by Fisher linear discriminant analysis (FLDA).

B. Geometric Context Model

By modeling the geometric context, the distinct outline features of alphanumeric characters, punctuation marks and radicals can be exploited to improve the retrieval accuracy. We design four statistical models for evaluating the geometric context, each of them contains two categories, one is the class-dependent feature, the other is class-independent feature, more technique details about feature extraction could be found in our earlier work [11].

For modeling the class-dependent geometry, we cluster the character classes into six super-classes similar with our earlier work [11]. After clustering, each single character is assigned to one of six super-classes and a pair of successive characters thus belongs to one of 36 binary super-classes. For estimating the statistical geometric models, training character samples are related to 6 super-classes. We use a quadratic distance function for both the unary and binary class-dependent geometric models. For unary geometry, 43D feature vector is reduce to 5-dimensional subspace by Fisher linear discriminant analysis (FLDA), also for binary geometry, the 36-class QDF is estimated using samples of 35D geometric features.

The unary class-independent geometry indicates whether a candidate pattern is a valid character or not. The binary class-independent geometry indicates whether a segmentation point is a between-character gap or not. Both of them are describing a binary classification task. To sum up, we use two SVM classifiers trained for these two class-independent geometry.

C. Fused Word Model

In word retrieval by beam search, we measure the character matching by combining the outputs of a character classifier and the geometric models. The combining weights and thresholds are optimized by training with annotated text line samples(string-level training).

1) *Two-Stage Matching Model*: After over-segmentation, a text line image is represented as a sequence of primitive segments ordered from left to right: $I = I_1 \cdots I_m$, the text of query word is represented as $W = c_1 c_2 \cdots c_k \cdots c_{n-1} c_n$, usually $m \geq n$. We assume that there is a subsequence of primitive segments: $I' = I_p \cdots I_q$ corresponding to the query word. And also there is a smaller subsequence $I_{j-k_i+1} \cdots I_j$ corresponding to character c_i . We allow a candidate character pattern to be formed by at most 4 primitive segments ($1 \leq k_i \leq 4$).

To accelerate word retrieval, we use two thresholds to prune candidate matches: a character threshold T_c to prune those characters with $\text{sim}(c_i, x) \leq T_c$, and a word threshold T_w to prune those words with $\text{Sim}(W, X) \leq T_w$.

As inspired by [14], the equation by which the character matching score is calculated is represented as below:

$$g_c(c_i | I_{j-k_i+1} \cdots I_j) = \sum_{h=0}^2 \alpha_h \cdot f_h - \tau_c, \quad (4)$$

where α_h ($h = 0 \cdots 2$) are the weight coefficients, f_0 is the output of character classifier, f_1, f_2 are two geometric cost, for the class-dependent and class-independent single character geometry respectively. The parameter τ_c could be either class-dependent or class-independent.

Also inspired by [11], the equation by which the word matching score is calculated is represented as below:

$$g_w(A) = \frac{1}{n} \sum_{h=0}^4 \beta_h \cdot F_h - \tau_w, \quad (5)$$

where $A = \{(c_1; I_p, \cdots, I_{p+k_1-1}), \cdots, (c_n; I_{q-k_n+1}, \cdots, I_q)\}$, β_h ($h = 0, \cdots, 4$) are the weight coefficients, F_0 is the character classifier output, F_h ($h = 1, \cdots, 4$) are outputs of four geometric models, for the class-dependent and class-independent single character geometry, class-dependent and class-independent between-character relationships, respectively. Each item is the sum of costs over the candidate subpath. They can be represented as below:

$$F_h = \sum_{i=1}^n f_h(c_i, I_{j-k_i+1} \cdots I_j) (h = 0, 1), \quad (6)$$

$$F_2 = \sum_{i=1}^n f_2(I_{j-k_i+1} \cdots I_j), \quad (7)$$

$$F_3 = \sum_{i=2}^n f_3((c_i, I_{j-k_i+1} \cdots I_j) | (c_{i-1}, I_{j-m} \cdots I_{j-k_i})), \quad (8)$$

$$F_4 = \sum_{j=p}^q f_4(I_j | I_{j-1}). \quad (9)$$

2) *Parameter Estimation of Models*: The objective of training is to tune the combining weights and threshold so as to promote correct word matching and depress incorrect word matching. The character-level training using the minimum cross-entropy error criterion has shown superior performance in pattern retrieval task [18]. As the implementation in [17], we add a regularization item to the error function for better convergence when optimization is done. For a training data set $S = \{(X^n, c^n) | n = 1, \dots, N\}$, we define M binary classification problems: ω_i versus non- ω_i , $i = 1, \dots, M$. The character-level training objective is rewritten as below:

$$\min(CE) = - \sum_{n=1}^N \left\{ \sum_{i=1}^M [y_i^n \log(p_i) + (1 - y_i^n) \log(1 - p_i)] \right\} + \gamma \sum_h \alpha_h^2. \quad (10)$$

The parameter's iterated update formulas are shown below:

$$\alpha_h^{t+1} = \alpha_h^t + \eta_1(t) [\xi(1 - p_i) f_h + 2\gamma\alpha_h] (c^n = \omega_i), \quad (11)$$

$$\alpha_h^{t+1} = \alpha_h^t - \eta_1(t) [\xi p_i f_h + 2\gamma\alpha_h] (c^n \neq \omega_i), \quad (12)$$

$$\tau_c^{t+1} = \tau_c^t + \eta_2(t) \xi [y_i^n (1 - p_i) - (1 - y_i^n) p_i], \quad (13)$$

where $p_i = p_i(x^n) = \text{sigmoid}(\xi g_{\omega_i}(x^n))$, $h = 0, 1, 2$.

Like what is done in character-level training, by string-level training, the weights and threshold are estimated on data set $S = \{(X^n, A_c^n) | n = 1, \dots, N\}$ where A_c^n denotes the correct matching of the sample $X^n = (I^n, W^n)$, the iterated parameter update formulas are shown as below:

$$\beta_h^{t+1} = \beta_h^t + \eta_1(t) [\xi(1 - p_i) F_h + 2\gamma\beta_h] (W^n = w), \quad (14)$$

$$\beta_h^{t+1} = \beta_h^t - \eta_1(t) [\xi p_i F_h + 2\gamma\beta_h] (W^n \neq w), \quad (15)$$

$$\tau_w^{t+1} = \tau_w^t + \eta_2(t) \xi [y_i^n (1 - p_i) - (1 - y_i^n) p_i], \quad (16)$$

where $p_i = p_i(X^n) = \text{sigmoid}(\xi g_w(X^n))$, $h = 0, 1, 2, 3, 4$. The parameters of fused word model are independent of word-class. For acceleration, on each training sample, we only update the correct class and a few selected rival class.

IV. EXPERIMENT & RESULTS

To evaluate the retrieval performance of the presented method, we tested with a dataset containing 1015 handwritten pages of 180 writers. We trained two classifiers for testing, one is one-vs-all prototype classifier, the other is linear SVM classifier.

The dataset for character classifier training contains 4,198,494 samples of 7356 classes. This is a union datasets of database CASIA-HWDB1.0~1.2 and CASIA-HWDB2.0~2.2 [18]. The 1015 pages for retrieval test were written by 180 different person from test set. We selected 60,000 frequently used words from the corpus collected by Sougou Lab for retrieval, including 39,057 two-character words, 9975 three-character words and 9451 four-character words, and 1517 words containing more than 4 characters. For better evaluating the performance of our method, we use the segmented text lines as our test samples to avoid being affected by the error of text line segmentation.

Table I
CHARACTER DETECTION RATE (%) OF FUSED MODELS

		Before CT			After CT		
		Recall	Precis	F-Score	Recall	Precis	F-Score
LVQ (OVA)	f_0	59.51	50.37	54.56	60.79	50.89	55.40
	$f_0 + f_1$	68.22	76.29	72.03	69.53	77.97	73.51
	$f_0 + f_2$	64.97	72.77	68.65	65.67	73.83	69.51
	$f_0 + f_1 + f_2$	72.29	79.97	75.94	69.53	77.97	73.51
LSVM (OVA)	f_0	68.98	59.45	63.86	70.21	60.11	64.77
	$f_0 + f_1$	79.38	84.77	81.98	80.26	85.93	82.29
	$f_0 + f_2$	74.93	79.52	77.16	75.27	80.19	77.65
	$f_0 + f_1 + f_2$	84.51	89.23	86.81	85.73	90.48	88.04

Table II
TWO-CHARACTER WORD RETRIEVAL RATE (%) OF FUSED MODELS

		Before CT			After CT		
		Recall	Precis	F-Score	Recall	Precis	F-Score
LVQ (OVA)	f_0	68.31	56.56	61.88	69.23	57.29	62.69
	a	72.53	57.39	64.08	73.95	59.23	65.78
	b	77.69	59.87	67.63	78.41	60.39	68.23
	c	75.36	58.63	65.95	76.29	59.76	67.02
	d	79.68	60.73	68.69	81.31	61.27	69.88
LSVM (OVA)	f_0	70.64	60.48	65.17	71.58	60.97	65.85
	a	75.39	61.53	67.76	76.29	62.39	68.64
	b	80.93	63.05	70.88	81.23	64.01	71.60
	c	78.69	62.17	69.46	79.64	63.54	70.68
	d	82.37	63.79	71.90	83.17	64.27	72.5

A. Character Detection

We first evaluated the character detection performance. If a character pattern of class i is accepted by the discriminant function of class j ($j \neq i$), it is a false positive of class j . If it is rejected by class i , it is a false negative of class i . We can obtain the rates of recall, precision and the F-measure using the same evaluation criterion as in [19].

We did the detection test on data set according to different combinatorial model generated by character classifier integrated with geometric context. And we have also do the confidence transformation (CT) [17] using logistic regression to the outputs of character classifier and geometric models. The parameter τ_c of discriminant function of (4) is set to class-independent for lack of character categories in textual document images for training. Before and After confidence transformation, the recall, precision and F-score of character detection task are listed in Table I.

B. word retrieval

In word retrieval from the test documents, we set variable combined model generated by character classifier integrated with geometric models. According to the character detection rate, there are totally five combinatorial mode for word retrieval. They are listed below: (a) $f_0 + f_1 + f_2$; (b) $f_0 + f_1 + f_2 + f_3$; (c) $f_0 + f_1 + f_2 + f_4$; (d) $f_0 + f_1 + f_2 + f_3 + f_4$ (only inner-word geometric model). The notions f_0, f_1, f_2, f_3, f_4 are represented the same statistical models described in section 3 part C. The retrieval result are listed in Table II~IV.

From retrieval rates of our fused model for words of different lengths as shown in Table II, Table III, Table IV, our model have a rather good performance for recall rates, but a little bad for precision rate, this is because most of the

Table III
THREE-CHARACTER WORD RETRIEVAL RATE (%) OF FUSED MODELS

		Before CT			After CT		
		Recall	Precis	F-Score	Recall	Precis	F-Score
LVQ (OVA)	f_0	69.57	60.56	64.75	70.29	61.39	65.54
	a	74.23	62.37	67.79	74.97	63.57	67.50
	b	79.23	63.89	70.74	81.97	65.72	72.95
	c	77.39	63.09	69.51	78.35	64.98	71.04
	d	81.74	64.37	72.02	83.37	66.43	73.94
LSVM (OVA)	f_0	72.64	63.48	67.75	73.49	64.37	68.63
	a	77.58	66.27	71.48	78.56	67.18	72.43
	b	82.47	68.32	74.73	85.96	69.18	76.66
	c	80.39	67.49	73.38	82.67	68.03	74.64
	d	84.37	68.53	75.63	86.43	69.87	77.27

Table IV
FOUR-CHARACTER WORD RETRIEVAL RATE (%) OF FUSED MODELS

		Before CT			After CT		
		Recall	Precis	F-Score	Recall	Precis	F-Score
LVQ (OVA)	f_0	71.35	61.78	66.22	72.19	62.49	66.99
	a	75.69	64.97	69.92	77.07	65.97	71.09
	b	82.47	67.04	73.96	83.65	68.03	75.04
	c	78.85	65.83	71.75	81.96	66.53	73.44
	d	84.27	67.59	75.01	85.76	68.97	76.45
LSVM (OVA)	f_0	74.96	65.48	69.90	75.86	66.59	70.92
	a	78.53	70.23	74.15	79.83	70.96	75.13
	b	83.94	72.94	78.05	84.23	73.54	78.52
	c	81.67	71.46	76.22	82.96	72.39	77.32
	d	85.47	73.28	78.91	86.31	74.56	80.01

words used for test does not exist in the document images for retrieval at all. This Research result indicates that our model is not good enough for outlier sample detection. We should integrates more contextual model with our model, such as linguistic context etc.

V. CONCLUSION

We proposed a word retrieval method for offline Chinese handwritten documents employing the one-vs-all (OVA) trained character classifier combined with four geometric contextual models. The query text word is matched with sequences of candidate character patterns to locate handwritten words with similarity greater than a threshold. The word similarity measure is obtained by combining character classifier model with single-character's and between-character's geometric models. Due to the merit of the geometric contextual models that it reduces the error of character segmentation and false accept rates of invalid characters to be valid one, the fused model combined it with OVA character classifier gives high similarity to target class and low scores to the other classes. The proposed method has yielded high performance of word retrieval in our experiments. Our future works aim to further improve the word matching similarity measure to and reject imposters.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China (NSFC) under grants no. 60825301 and no. 60933010.

REFERENCES

- [1] H. Bunke and T. Varga, Off-line Roman cursive handwriting recognition, in *Digital Document Processing: Major Directions and Recent Advances*, B.B. Chaudhuri, Ed. Springer, 2007, pp.165-173.
- [2] A. Antonacopoulos and A. Downton (eds.), Special issue on the analysis of historical documents, *Int.Journal on Document Analysis and Recognition*, vol. 9, no. 2-4, pp. 75-77, 2007.
- [3] R. Manmatha, C. Han, and E. Riseman, Word spotting: a new approach to indexing handwriting, *Int. Conf. on Computer Vision and Pattern Recognition*, pp. 631-637, 1996
- [4] B. Zhang, S.N. Srihari, and C. Huang, Word image retrieval using binary features, in *Proc. Document Recognition and Retrieval XI*, vol. 5296, 2004, pp. 45-53.
- [5] T.M. Rath and R. Manmatha, Word spotting for historical documents, *Int. Journal on Document Analysis and Recognition*, vol. 9, pp. 139-152, 2007.
- [6] J. Rodriguez and F. Perronnin, Handwriting word-spotting using hidden Markov models and universal vocabularies, *Pattern Recognition*, vol. 42, no. 9, pp. 2106-2116, 2009
- [7] J. Chan, C. Ziftci, and D. Forsyth, Searching off-line Arabic documents, *Int. Conf. on Computer Vision and Pattern Recognition*, 2006, pp. 1455-1462.
- [8] J. Edwards, Y.W. Teh, D. Forsyth, R. Bock, M. Marie, and G. Vesom, Making latin manuscripts searchable using gHMM's, in *NIPS, 2004*,
- [9] S. Senda, M. Minoh, K. Ikeda, Document image retrieval system using character candidates generated by character recognition process, *Proc. 2nd ICDAR*, Tsukuba, Japan, 1993, pp.541-546.
- [10] H. Oda, A. Kitadai, M. Onuma, M. Nakagawa, A search method for on-line handwritten text employing writing-box-free handwriting recognition, *Proc. 9th IWFHR*, Tokyo, Japan, 2004.
- [11] F. Yin, Q.F. Wang, C.L. Liu, Integrating geometric context for text alignment of handwritten Chinese documents, *Proc. 12th ICFHR*, Kolkata, India, pp.7-12, 2010.
- [12] F. Yin, C.L. Liu, Handwritten text line segmentation by clustering with distance metric learning, *Pattern Recognition*, vol.42, pp.3146-3157, 2009.
- [13] C.L. Liu, M. Koga, H. Fujisawa, Lexicon-driven segmentation and recognition of handwritten character strings for Japanese address reading, *IEEE Trans.Pattern Analysis and Machine Intelligence*, vol.24, pp.1425-1437, 2002
- [14] C.L. Liu, One-vs-all training of prototype classifier for pattern classification and retrieval, *Proc. 20th ICPR*, Istanbul, Turkey, 2010.
- [15] O.L. Mangasarian, D.R. Musicant, Successive overrelaxation for support vector machines, *IEEE Trans.Neural Networks*, vol.10, pp.1032-1037, 1999.
- [16] C.L. Liu, Normalized-cooperated gradient feature extraction for handwritten character recognition, *IEEE Trans.Pattern Analysis and Machine Intelligence*, vol.29, pp.1465-1469, 2007.
- [17] L. Gillick, Y. Ito, J. Young, A probabilistic approach to confidence estimation and evaluation, *Proc. ICASSP'97*, Munich, Germany, Vol.2, pp.879-882.
- [18] C.L. Liu, F. Yin, D.H. Wang, Q.F. Wang, CASIA online and offline Chinese handwriting databases, to appear in *Proc. 11th ICDAR*, Beijing, China, 2011.
- [19] C. Cheng, B.L. Zhu, M. Nakagawa, A dicriminative model for On-line handwritten japanese text retrieval, *Proc. CJKPR'2010*, pp. 94-97.