Transcript Mapping for Handwritten Text Lines Using Conditional Random Fields

Xiang-Dong Zhou [1], Fei Yin [2], Da-Han Wang [2], Qiu-Feng Wang [2], Masaki Nakagawa [1], Cheng-Lin Liu [2]
[1] Tokyo University of Agriculture and Technology, Tokyo 184-8588, Japan
[2] National Laboratory of Pattern Recognition (NLPR),
Institute of Automation, Chinese Academy of Sciences, Beijing 100190, P.R. China
[1] {xdzhou, nakagawa}@cc.tuat.ac.jp, [2] {fyin, dhwang, wangqf, liucl}@nlpr.ia.ac.cn

Abstract—**This paper presents a conditional random field (CRF) model for aligning online handwritten Chinese/Japanese text lines (character strings) with the corresponding transcripts. The CRF model is defined on a lattice which contains all possible segmentation hypotheses. The feature functions characterize the shape and context dependences of characters, including the scores of character recognition and the geometric compatibilities between characters. The combining parameters are optimized by energy minimization. Experimental results on two online databases: CASIA-OLHWDB and TUAT Kondate demonstrate the effectiveness of the proposed method.**

Keywords-transcript mapping; text alignment; conditional random fields

I. INTRODUCTION

To ground-truth handwritten documents in word/character level is of great importance for designing and evaluating document analysis algorithms. However, to annotate words or characters manually is tedious, time consuming and error-prone, which necessitates the automatic ground-truthing tools. Transcript mapping is a technique for such purpose, which is to align the manuscript with the corresponding transcripts.

Depending on how the handwritten text lines are segmented before alignment, previous work can be roughly classified into three categories: frame-level segmentation [1], [2], component-level segmentation [3], [4], [5], [6] and word-level segmentation [7], [8], [9], [10], [11]. Frame-level segmentation blindly slices the text line image into frames of equal width, on which feature vectors are extracted and fed to a pre-trained hidden Markov model (HMM) based recognition system. The optimal word sequence is generated by a Viterbi decoder in a forced alignment mode. Component-level segmentation over-segments the handwritten text line into components, with each component being whole or part of a word/character, and consecutive components are combined as segments (word/character hypotheses). The best match is found by dynamic time warping (DTW) [3], [4], exhaustive search [5] or greedy search [6]. In this paper, we also refer to this approach as segment-based method. Word-level segmentation tries to segment the handwritten text line into a word image sequence, whose length is not necessary to be equal to that of the transcription. The word image sequence is further aligned with the transcripts using DTW [7], [8], [9] or HMM [10]. The work of [11] also considers multiple line segmentation hypotheses, and the dynamic programming (DP) algorithm is used to find the best image-text mapping. Without the guidance of transcripts, the word segmentation step is a major source of errors [2]. Usually a post-processing step is adopted to correct the alignment errors.

The work for Chinese/Japanese documents is relatively fewer. Yin et al. [12] propose a segment-based method incorporating character recognizer for offline handwritten Chinese text line alignment, which views transcript mapping as a special case of lexicon-driven string recognition. They further enhance this model by integrating geometric context in their following work [13].

Our work focuses on aligning online handwritten Chinese/Japanese text lines with the associated transcripts, whose objective is to facilitate the development of string recognition applications. Existing Chinese/Japanese string recognition algorithms usually adopt segmented text lines for parameter learning [14], [15]. Transcript mapping enables training with un-segmented text line data and alleviates the pains taken in human segmentation. We propose a principled framework based on conditional random field (CRF) [16] for transcript mapping. After over-segmenting the input text line into a component sequence with the hope that each component forms a character or part of a character, a lattice including all possible segmentation paths is generated. On this lattice, a segment-based CRF that models the a posteriori probability of a segmentation path given the character string and the transcripts is defined. The CRF model effectively combines the character recognition scores and geometric compatibilities between characters and thus yield high accuracy of alignment.

II. CRFS FOR TRANSCRIPT MAPPING

With the maximum a posteriori (MAP) criterion, given a string pattern X and the associated label string Y (referred to as (X,Y)), transcript mapping is to find the optimal segmentation S^* to maximize the a posterior probability $P(S|X,Y)$:

$$S^* = \underset{S:Y}{\arg\max}\, P(S|X,Y), \tag{1}$$

where $S{:}Y$ indicates S is a segmentation of X given Y. Similar to the integrated segmentation and recognition approach for character string recognition [14], before adjusting the character boundaries, the input string X is over-segmented into a sequence of components (Figure. 1 (a)). Between each pair of consecutive components, there is a candidate

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.21

segmentation point. The start and end of the string pattern are viewed as two additional segmentation points. Consecutive components with plausible width to line height ratio are combined to generate candidate character patterns, which constitute the segmentation candidate lattice (Figure. 1 (b) and (c)). From the segmentation candidate lattice, we construct the $S|Y$ lattice, with all possible segmentation paths of (X,Y) included. Compared to our former work on character string recognition [14], which models the joint distribution of $P(Y,S|X)$, the proposed model is also a segment-based CRF, while modeling $P(S|X,Y)$. Compared to the linear-chain model, such as HMM, which considers the characteristics of frames, segment-based model directly manipulates the attributes of characters.

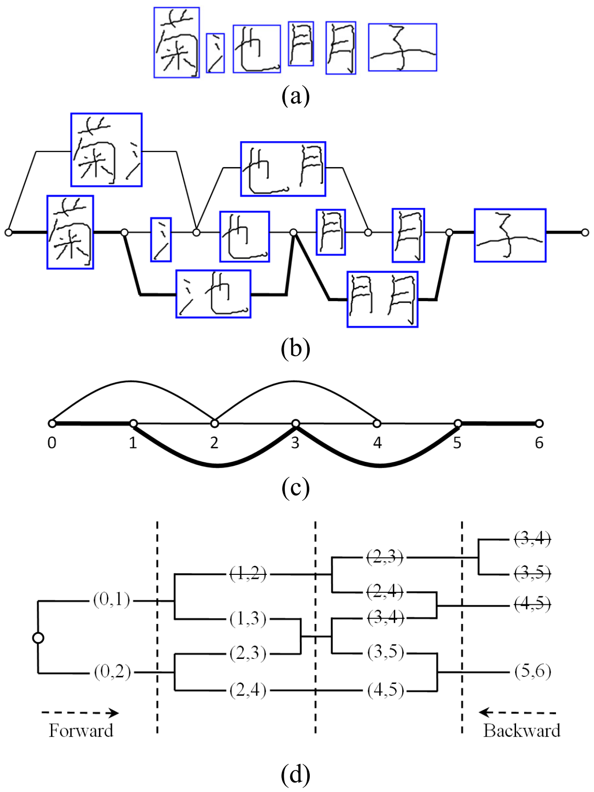

Figure 1. Generation of a $S|Y$ lattice from a segmentation candidate lattice (transcripts: 菊池朋子): (a) component sequence, (b) candidate characters, (c) segmentation candidate lattice (the bold lines indicate the desired segmentation), and (d) forward and backward procedures. The struck out edges will be removed in the backward procedure.

A. Generation of $S|Y$ Lattice

Figure.1 illustrates the construction process of a $S|Y$ lattice, including a forward procedure and a backward procedure, where (i,j) denotes the candidate character between segmentation points i and j. The forward procedure expands the partial paths in a breadth-first character synchronous manner [17], with which the depths of partial paths are equal at each step. The expansion continues until the path depth is equal to the string length $|Y|$ or the terminal segmentation point is reached. After the forward procedure, some paths will be still open. The backward procedure then removes all the paths with depth smaller than $|Y|$ or ending node preceding the terminal segmentation point. The retained paths are the possible alignments of (X,Y), which will be evaluated using the CRF model.

B. CRFs on $S|Y$ Lattice

For a hypothesized segmentation S in the $S|Y$ lattice of (X,Y), from the definition of CRFs, $P(S|X,Y)$ can be written as the normalized product of potential functions:

$$\begin{aligned} P(S \mid X,Y) &= \frac{1}{Z(X,Y)} \prod_{c \in S} \Psi_c(y_c) \\ &= \frac{1}{Z(X,Y)} \exp\{-E(\Lambda, X, Y, S)\} \end{aligned} . \quad (2)$$

$\Psi_c(y_c)$ denotes the potential function on maximal clique c (consecutive characters) with transcripts y_c:

$$\Psi_c(y_c) = \exp\left\{ \sum_{k=1}^{K} \lambda_k f_c^k(y_c) \right\}. \quad (3)$$

$E(\Lambda,X,Y,S)$ is the energy function:

$$E(\Lambda, X, Y, S) = -\sum_{c \in S} \sum_{k=1}^{K} \lambda_k f_c^k(y_c), \quad (4)$$

where $f_c^k(y_c)$ is the kth feature function on clique c, and $\Lambda=\{\lambda_k\}$ are weighting parameters. The feature functions characterize the character shape and geometric compatibilities between characters, as will be specified in Section III. $Z(X,Y)$ is the partition function defined as the summation over all segmentation paths:

$$Z(X,Y) = \sum_{S':Y} \prod_{c \in S'} \Psi_c(y_c) = \sum_{S':Y} \exp\{-E(\Lambda, X, Y, S')\}. \quad (5)$$

C. Parameter Learning

Lattice generation cannot guarantee the desired segmentation is included, so before parameter learning, we first insert the ground-truthed segmentation into the $S|Y$ lattice. Denoting a training sample by (X^i,Y^i,S^i) (string with transcripts and desired segmentation), training is to find the optimal parameters by minimizing a loss function. With stochastic gradient decent, the parameters are updated by:

$$\Lambda(t+1) = \Lambda(t) - \varepsilon(t) \nabla_\Lambda L(X^i, Y^i, S^i)|_{\Lambda=\Lambda(t)}, \quad (6)$$

where $L(X^i,Y^i,S^i)$ denotes the per-sample loss, and $\varepsilon(t)$ is the learning step. The loss function is often the negative log-likelihood (NLL) loss [18], which is a convex function and guarantees global optimization by gradient descent. From Eq. (1), to maximize $P(S^i|X^i,Y^i)$ is equivalent to minimizing $-\log P(S^i|X^i,Y^i)$. The NLL loss is thus defined as

$$\begin{aligned} L_{NLL}(X^i, Y^i, S^i) &= -\log P(S^i \mid X^i, Y^i) \\ &= E(\Lambda, X^i, Y^i, S^i) + \log Z(X^i, Y^i) \end{aligned}, \quad (7)$$

whose partial derivatives with respect to the weighting parameters are computed by

$$\frac{\partial L_{NLL}(X^i,Y^i,S^i)}{\partial \lambda_k} = -\sum_{c \in S^i} f_c^k(y_c) + \sum_{S:Y^i} \sum_{c \in S} f_c^k(y_c) P(S|X^i,Y^i). \quad (8)$$

In the second term, the summation is over all the segmentation paths in the $S|Y$ lattice and all the maximal cliques on each path, which is equivalent to the summation over each position and each maximal clique ending at this position:

$$\begin{aligned}&\sum_{S:Y^i}\sum_{c\in S} f_c^k(y_c)P(S\mid X^i,Y^i)\\&=\sum_{n=1}^{|Y^i|}\sum_{c:D(c)=n} f_c^k(y_c)\sum_{S:Y^i\wedge c\in S\wedge D(c)=n} P(S\mid X^i,Y^i),\\&=\sum_{n=1}^{|Y^i|}\sum_{c:D(c)=n} f_c^k(y_c)P(c,n\mid X^i,Y^i)\end{aligned} \tag{9}$$

where $D(c)=n$ means c ends at position n of Y^i (the position of the last character of c is n), and $P(c,n|X^i,Y^i)$ is the marginal probability that c is on the segmentation path and ends at position n. Similar to the linear-chain CRF [16], $P(c,n|Y^i,X^i)$ can be calculated from the forward-backward recursions.

D. Inference on $S|Y$ Lattice

In this work, we consider only binary context, thus the maximal clique size is 2 (at the beginning of the paths, the clique size is 1, on which the binary feature functions are set to 0). We denote a maximal clique by t_0^2, where t_i, i=0, 1, 2 are three ordered segmentation points. With $Y=y_1^{|Y|}$ being the transcripts of string X, the forward variables $\alpha_{t_1^2}^n(y_n)$ at each position n of Y can be deduced recursively by

$$\alpha_{t_1^2}^n(y_n)=\begin{cases}\Psi_{t_1^2}(y_1), & t_1=0, n=1\\ \sum_{t_0}\Psi_{t_0^2}(y_{n-1}^n)\alpha_{t_0^1}^{n-1}(y_{n-1}), & t_1\neq 0, 2\leq n\leq |Y|\end{cases}, \tag{10}$$

Similarly, the backward variables $\beta_{t_0^1}^n(y_n)$ can be deduced by

$$\beta_{t_0^1}^n(y_n)=\begin{cases}1, & t_1=T, n=|Y|\\ \sum_{t_2}\Psi_{t_0^2}(y_n^{n+1})\beta_{t_1^2}^{n+1}(y_{n+1}), & t_1\neq T, 1\leq n\leq |Y|-1\end{cases}, \tag{11}$$

where T is the index of the terminal segmentation point. The partition function can be calculated from both forward and backward variables:

$$\begin{aligned}Z(X,Y)&=\sum_{t_1}\alpha_{t_1^2}^{|Y|}(y_{|Y|}), & t_2=T,\\ &=\sum_{t_1}\beta_{t_0^1}^1(y_1)\Psi_{t_0^1}(y_1), & t_0=0.\end{aligned} \tag{12}$$

With the forward and backward variables, we can calculate the marginal probabilities:

$$P(t_0^1,n\mid X,Y)=\frac{1}{Z(X,Y)}\alpha_{t_0^1}^n(y_n)\beta_{t_0^1}^n(y_n), \tag{13}$$

$$P(t_0^2,n|X,Y)=\frac{1}{Z(X,Y)}\alpha_{t_0^1}^{n-1}(y_{n-1})\beta_{t_1^2}^n(y_n)\Psi_{t_0^2}(y_{n-1}^n). \tag{14}$$

Replacing the summation by maximization in Eq. (10) yields the Viterbi-like (max-product) recursion:

$$\hat{\alpha}_{t_1^2}^n(y_n)=\begin{cases}\Psi_{t_1^2}(y_1), & t_1=0, n=1\\ \max_{t_0}\Psi_{t_0^2}(y_{n-1}^n)\hat{\alpha}_{t_0^1}^{n-1}(y_{n-1}), & t_1\neq 0, 2\leq n\leq |Y|\end{cases}, \tag{15}$$

The optimal preceding segmentation point at each step is recorded by

$$\xi_{t_1^2}^n(y_n)=\operatorname*{argmax}_{t_0}\Psi_{t_0^2}(y_{n-1}^n)\hat{\alpha}_{t_0^1}^{n-1}(y_{n-1}),\ t_1\neq 0, 2\leq n\leq |Y|. \tag{16}$$

The globally most probable segmentation defined in Eq. (1) can be achieved by backtracking.

III. Experiments

We evaluated the proposed method on online handwritten text lines of Chinese CASIA-OLHWDB database [19] and Japanese TUAT Kondate database [15]. The training and test set of CASIA-OLHWDB is obtained by merging the corresponding string samples of OLHWDB 2.0~2.2, respectively. Horizontal text lines extracted from the TUAT Kondate database are used in our experiments. The details of the two databases are listed in TABLE I.

TABLE I. Details of training and testing sets. First row: training; second row: testing.

Databases	# line	# class	# char/line
CASIA-OLHWDB	41,711	2,651	25.95
	10,510	2,631	25.66
TUAT Kondate	10,174	1,106	10.23
	3,511	790	16.89

Four feature functions are employed in our experiments, including character recognition score, unary and binary class-dependent geometric context, as well as binary class-independent geometric context [14]. In our experiments, they are represented by f_1, f_2, f_3 and f_4, respectively. For CASIA-OLHWDB database, the modified quadratic discriminant function (MQDF) [20] classifier for character recognition was trained on the merged training set of OLHWDB1.0~1.2 (7,356 classes) [19] together with the isolated characters extracted from the training string set falling into the 7,356 classes. For TUAT Kondate database, the MQDF classifier for character recognition was trained on TUAT Nakayosi database (4,438 classes) [21] together with the isolated characters extracted from the Kondate training set falling into the 4,438 classes. For both the two databases, the quadratic discriminant function (QDF) classifiers and the support vector machine (SVM) classifier for geometric feature functions were trained on the samples extracted from the respective training strings.

A. Performance Metrics

Given the test set $\aleph=\{(X^i,S^i,Y^i)\mid i=1,\ldots,N\}$, where S^i is the ground-truthed segmentation and Y^i is the label sequence of string X^i. The following metrics are calculated to measure the algorithm performance.

String error rate (SER) is used to measure the string level performance, which is as the percentage of mis-aligned strings (those with mis-aligned characters).

Character error rate (CER) is used to measure the character level performance, which is defined as the percentage of mis-aligned characters in all the test strings.

Lattice error rate (LER) is used to evaluate the quality of the $S|Y$ lattices. If a character-label pair in a test string (X^i,S^i,Y^i) is not included in the $S|Y$ lattice of (X^i,Y^i), we say that there is a lattice error. LER is as the percentage of number of lattice errors among all characters in the test strings.

Alignment error rate (AER) measures the errors caused by the alignment model, which is defined as the difference of CER and LER.

B. Experimental Results

TABLE II lists the experimental results on test string sets, as well as the effects of each feature function. The character recognizer can help to improve the tolerance to character shape and size variations, therefore the character recognition score (f_1) is the most important feature function. The incorporation of geometric context can further decrease the error rates. When segmenting the text lines, the features characterizing between-character gaps and overlaps are discriminative, and consequently the binary class-independent geometric feature function (f_4) plays an important role. In Chinese/Japanese text lines, the alphanumeric characters and the punctuation marks usually have distinct size, aspect ratio and position from the Chinese characters, so the class-dependent feature functions, especially the unary one (f_2), take effect. The combination of the four feature functions gives the best alignment performance, which means that they are complementary.

TABLE II. EXPERIMENTAL RESULTS (%) ON TEST STRING SETS. UPPER PART: CASIA-ONHWDB; LOWER PART: TUAT KONDATE.

f_1	f_2	f_3	f_4	SER	CER	AER
	○	○	○	46.39	9.26	8.20
○				38.68	4.69	3.63
○	○			24.37	2.72	1.65
○		○		36.14	4.27	3.21
○			○	21.89	2.45	1.39
○	○	○	○	20.46	2.26	1.20
	○	○	○	22.19	9.82	9.37
○				11.09	2.75	2.31
○	○			8.57	2.06	1.62
○		○		11.09	2.70	2.26
○			○	4.91	1.14	0.69
○	○	○	○	4.76	1.07	0.63

TABLE III. COMPARISON WITH OTHER METHODS (%).

Methods	CASIA-OLHWDB			TUAT Kondate		
	SER	CER	AER	SER	CER	AER
Proposed	20.46	2.26	1.20	4.76	1.07	0.63
Ref. [5]	65.16	17.70	16.64	46.37	26.15	25.70
Ref. [6]	93.60	68.35	67.29	55.68	46.13	45.69

In CASIA-OLHWDB, 339 test strings contain outlier characters (those out of the classes modeled by the character classifier). In the proposed method, if a clique contains outlier, all the feature functions on this clique are set to 0, which leaves the segmentation to be determined by other cliques. By calculating the proportion of incorrectly aligned outlier characters (5.21%), we can see that most of the outlier characters are correctly aligned, which means that the proposed method is robust to outliers.

Finally, we draw a comparison with two other segment-based methods. The work presented by Zinger et al. [5] exploits the relationship between the image-text word lengths and the best results are given by exhaustive search. In our method, character width is a unary class-dependent geometric feature. However, to perform exhaustive search is impracticable and unnecessary in our work, because the global optimal segmentation path can be found by DP, which is more efficient. Stamatopoulos et al. [6] calculate the distances between components and adopt greedy search to segment the text lines. According to their method, we sort the distances between adjacent components in descending order and recursively select the largest one to separate the text line until the character number is equal to that of the transcripts. From TABLE III we can see that the propose method outperforms the two methods significantly. The superior performance of the proposed method is attributed to the principled combination of character recognition scores and multiple geometric features. While the recognition score plays critical role in alignment, it is not used in the two compared methods.

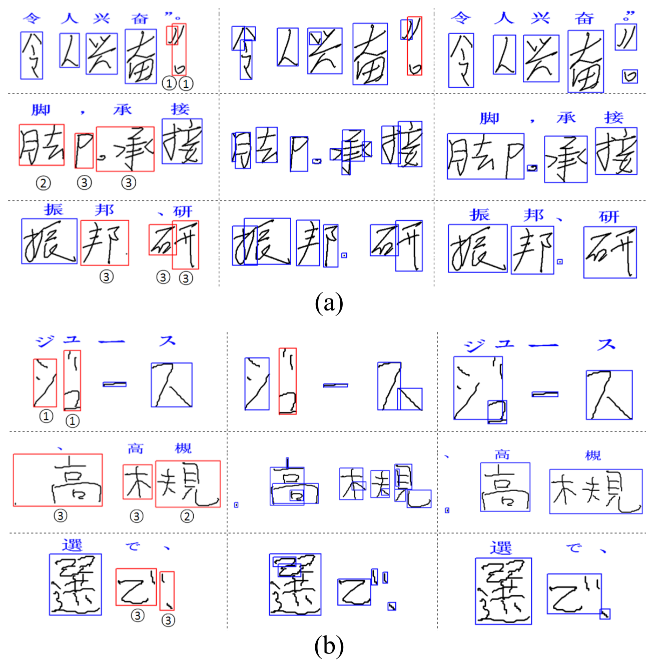

Figure 2. Examples of errors after alignment: (a) CASIA-OLHWDB, and (b) TUAT Kondate. Blue bounding boxes: correct segmentation, and red bounding boxes: incorrect segmentation. Left part: errors, middle part: components, and right part: ground-truths. ①: merging error, ②: splitting error, and ③: alignment error.

C. Error Analysis

TABLE IV lists the error rates for each error type, and Figure. 2 illustrates some errors. In TABLE IV, the lattice errors include two parts. The merging errors are caused by incorrectly merging strokes of adjacent characters into components in over-segmentation. The splitting errors are introduced by excluding candidate characters whose width exceeds a threshold (1.6 times the text line height in our experiments) in lattice generation. The text lines in CASIA-OLHWDB are more cursive, such that strokes of consecutive characters usually overlap with each other, which cause considerable merging errors. In Figure 2, an incorrectly merged component will definitely cause merging errors for each character with strokes in it. Splitting errors usually cause alignment errors, and most of the alignment errors are accompanied.

TABLE IV. ERROR RATES (%) FOR EACH ERROR TYPE.

Databases	LER		AER
	Merging	Splitting	
CASIA-OLHWDB	1.05	0.01	1.20
TUAT Kondate	0.14	0.31	0.63

IV. CONCLUSION

This paper presents a CRF-based transcript mapping method for online handwritten Chinese/Japanese text lines. By incorporating character recognizer and geometric context, the proposed method is robust to character shape/size variability. Experimental results on CASIA-OLHWDB database and TUAT Kondate database show that the proposed method can achieve quite low alignment error rates (1.20% and 0.63%, respectively). Moreover, it is robust to outliers. Our future work will investigate into the lattice pruning method to reduce the time cost in training and decoding, and apply transcript mapping in character string recognizer training.

ACKNOWLEDGEMENTS

This work is supported by the R&D fund for "Development of Pen & Paper-Based User Interaction" of Japan Science and Technology Agency and the National Natural Science Foundation of China (NSFC) grant no.60933010.

REFERENCES

[1] M. Zimmermann, H. Bunke, Automatic segmentation of the IAM off-line database for handwritten English text, Proceedings of the 16th International Conference on Pattern Recognition, Quebec, Canada, 2002, vol.4, pp.35-39.

[2] E. Indermühle, M. Liwicki, H. Bunke, Combining alignment results for historical handwritten document analysis, Proceedings of the 10th International Conference on Document Analysis and Recognition, Barcelona, Spain, 2009, pp. 1186–1190.

[3] A. Kumar, A. Balasubramanian, A. Namboodiri, C.V. Jawahar, Model-based annotation of online handwritten datasets, Proceedings of the 10th International Workshop on Frontiers in Handwriting Recognition, La Baule, France, 2006, pp.9-14.

[4] L.M. Lorigo, V. Govindaraju, Transcript mapping for handwritten Arabic documents, Proceeding of the IS&T/SPIE Symposium on Electronic Imaging, Document Recognition and Retrieval XIV, X. Lin and B. Yanikoglu (Eds.), San Jose, CA, 2007, pp. 65000W.1-65000W.8.

[5] S. Zinger, J. Nerbonne, L. Schomaker, Text-image alignment for historical handwritten documents, Proceeding of the IS&T/SPIE Symposium on Electronic Imaging, Document Recognition and Retrieval XIV, K. Berkner and L. Likforman-Sulem (Eds.), San Jose, CA, 2009, pp. 724703.1-724703.8.

[6] N. Stamatopoulos, G. Louloudis, B. Gatos, Efficient transcript mapping to ease the creation of document image segmentation ground truth with text-image alignment, Proceedings of the 12th International Conference on Frontiers in Handwriting Recognition, Kolkata, India, 2010, pp. 226-231.

[7] E.M. Kornfield, R. Manmatha, J. Allan, Text alignment with handwritten document, Proceedings of the First International Workshop on Document Image Analysis for Libraries, Palo Alto, CA, 2004, pp.195-209.

[8] E.M. Kornfield, R. Manmatha, J. Allan, Further explorations in text alignment with handwritten documents, Int. J. Document Analysis and Recognition, 10 (1)(2007) 39-52.

[9] C. Huang, S.N. Srihari, Mapping transcripts to handwritten text, Proceedings of the 10th International Workshop on Frontiers in Handwriting Recognition, La Baule, France, 2006, pp.15-20.

[10] J. Rothfeder, T.M. Rath, R. Manmatha, Aligning transcripts to automatically segmented handwritten manuscripts, Proceedings of the 7th International Workshop on Document Analysis Systems, Nelson, New Zealand, 2006, pp. 84-95.

[11] C.I. Tomai, B. Zhang, V. Govindaraju, Transcript mapping for historic handwritten document images, Proceedings of the 8th International Workshop on Frontiers in Handwriting Recognition, Ontario, Canada, 2002, pp. 413-418.

[12] F. Yin, Q.F. Wang, C.L. Liu, A Tool for ground-truthing text lines and characters in off-line handwritten Chinese documents, Proceedings of the 10th International Conference on Document Analysis and Recognition, Barcelona, Spain, 2009, pp. 951-955.

[13] F. Yin, Q.F. Wang, C.L. Liu, Integrating geometric context for text alignment of handwritten Chinese documents, Proceedings of the 12th International Conference on Frontiers in Handwriting Recognition, Kolkata, India, 2010, pp. 7-12.

[14] X.D. Zhou, C.L. Liu, M. Nakagawa, Online handwritten Japanese character string recognition using conditional random fields, Proceedings of the 10th International Conference on Document Analysis and Recognition, Barcelona, Spain, 2009, pp.521-525.

[15] B. Zhu, X.D. Zhou, C.L. Liu, M. Nakagawa, A robust model for on-line handwritten Japanese text recognition, Int. J. Document Analysis and Recognition, 13 (2)(2010) 121-131.

[16] J. Lafferty, A. McCallum, F. Pereira, Conditional random fields: probabilistic models for segmenting and labeling sequence data, Proceedings of the 18th International Conference on Machine Learning, Morgan Kaufmann, San Francisco, CA, 2001, pp. 282-289.

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

[18] Y. LeCun, S. Chopra, R. Hadsell, R. Marc'Aurelio, F. Huang, A tutorial on energy-based learning. In: G. Bakir, et al. (Eds.), Predicting Structured Data, MIT Press, 2007.

[19] C.L. Liu, F. Yin, D.H. Wang, Q.F. Wang, CASIA online and offline Chinese handwriting databases, submited to ICDAR2011.

[20] F. Kimura, K. Takashina, S. Tsuruoka, Y. Miyake, Modified quadratic discriminant functions and its application to Chinese character recognition, IEEE Trans. Pattern Anal. Mach. Intell. 9 (1) (1987) 149-153.

[21] K. Matsumoto, T. Fukushima, M. Nakagawa, Collection and analysis of on-line handwritten Japanese character patterns, Proceedings of the 6th International Conference on Document Analysis and Recognition, Seattle, WA, 2001, pp. 496-500.