Integrating Geometric Context for Text Alignment of Handwritten Chinese Documents

Fei Yin, Qiu-Feng Wang, Cheng-Lin Liu

National Laboratory of Pattern Recognition (NLPR),

Institute of Automation, Chinese Academy of Sciences,

P.O. Box 2728, Beijing 100190, P.R. China

E-Mail: {fyin, wangqf, liucl}@nlpr.ia.ac.cn

Abstract

The alignment of text line images with text transcript is a crucial step of handwritten document annotation. Handwritten text alignment is prone to errors due to the difficulty of character segmentation and the variability of character shape, size and position. In this paper, we propose to incorporate the geometric context of character strings to improve the alignment accuracy for offline handwritten Chinese documents. We use four statistical models to evaluate the geometric features of single characters and between-character relationships. By combining the geometric models with a character recognizer, we have achieved a large improvement of alignment accuracy in our experiments on unconstrained handwritten Chinese text lines.

1. Introduction

Annotating document images by segmenting and labeling the text lines and characters is an important but tedious and expensive task for building databases for research and development. Automated annotating tools can greatly facilitate this work, but for handwritten documents, it is hard to annotate correctly due to the difficulty of character segmentation and the variability of character shape, size and position. Recently, we presented an annotation tool GTLC (Ground-truthing Text Lines and Characters) for text line segmentation and text alignment of handwritten Chinese documents [1]. This tool has greatly assisted our data collection and annotation work, but still needs appreciable human intervention to correct the alignment errors.

A crucial step of handwriting annotation is to align text line images with their text transcripts. The previous methods can be roughly divided into two groups depending on whether word/character recognition models are used or not. In the first group without recognition models, outline features are extracted from text line image for transcript alignment by dynamic time warping (DTW) [2][3] or hidden Markov model (HMM) [4]. Recognition models better measure the similarity between image segment and word/character and thus can improve the alignment accuracy. The method in [5] uses HMM-based word recognizer, the ones in [6][7] use character prototype-based word models, while our previous method [1] uses a modified quadratic discriminant function (MQDF)-based character classifier [8].

The alignment accuracy is still not sufficient despite the promise of recognition models. In Chinese documents, the mixed alphanumeric characters and punctuation marks are prone to segmentation and labeling errors because they have distinct geometric features such as size, aspect ratio and position in text line. The misalignment of Chinese characters is mainly due to character touching and the gaps within characters composed of multiple radicals. Fig.1 shows typical annotation errors caused by a punctuation mark and a radical of Chinese character. Geometric context features would be helpful to reduce such errors.

Fig.1 Annotation errors caused by punctuation mark (red box) and radical of Chinese character (green box).

Geometric context has been used in handwriting recognition to reduce character segmentation and recognition errors, by using various geometric features (such as character size, inter-character and between-character gaps) and statistical models (geometric class means, Gaussian density models, discriminative classifiers, etc.) for handwritten word and Japanese character string recognition [9-13]. These methods cannot be used straightforwardly for Chinese handwriting due to its greater challenge.

In this paper, we design geometric models for text alignment in Chinese handwriting annotation. We use four statistical models to evaluate the single-character geometric features and between-character relationships. The geometric models are combined with the character recognizer to evaluate the text-to-handwriting matching cost and the best match searched by DTW gives the alignment result (character segmentation and labeling). The combining weights are optimized by string-level training. Our experimental results on unconstrained handwritten Chinese text lines show that the geometric models can significantly reduce alignment errors.

2. Outline of Annotation System

The block diagram of our Chinese handwriting annotation system is shown in Fig. 2. The input handwritten document image is first segmented into text lines. Each line image is then aligned with its transcript for character segmentation and labeling. Line segmentation and alignment are automatic, with human intervention to correct remaining errors.

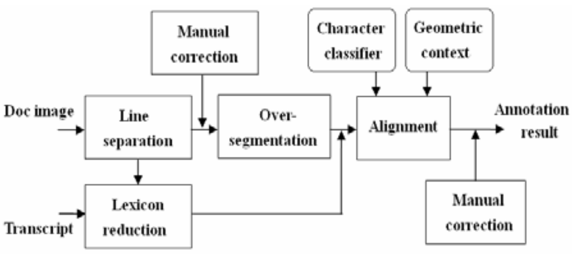

Fig. 2 Block diagram of the annotation system.

Text lines are separated by clustering connected components (CCs) using minimal spanning tree (MST) with distance metric learning [14].

For segmenting and labeling the characters in a text line image, the line image is over-segmented into primitive segments such that one or more consecutive segments form a character. Touching characters are split in over-segmentation by contour analysis to detect touching points.

In text alignment, the sequence of primitive segments is matched with the character string of transcript. This is a dynamic programming (DP) problem, which minimizes an edit distance, and the alignment result largely depends on the cost defined for segment-to-character match. The character recognizer measures the shape similarity (or distance) between a candidate pattern (composed of one or more consecutive primitive segments) and a character, and geometric models measure the similarity of character outline and between-character compatibility. The best alignment, corresponding to a path in a grid space, is searched by DP. Fig.3 shows an example, where a string of seven characters is aligned with 10 primitive segments, which are correctly segmented and labeled by the path of red line.

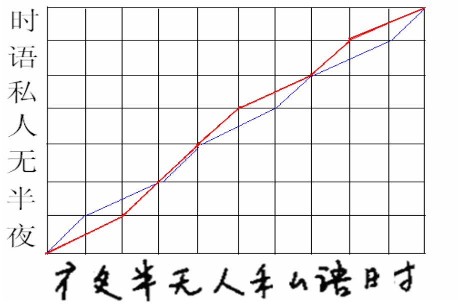

Fig. 3 An example of text line alignment.

3. Text Line Alignment

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

3.1 Problem Formulation

After over-segmentation, a text line image is represented as a sequence of primitive segments ordered from left to right: $I=\{I_1 \cdots I_m\}$, to align with a character string $T=\{C_1 \cdots C_n\}$, $(n \leqslant m)$. The alignment A is a correspondence between candidate patterns and characters:

$$A=\{(C_1, I_1 \cdots I_{k_1-1}), \cdots ; (C_i, I_{j-k_i+1} \cdots I_j), \cdots ; (C_n, I_p \cdots I_m)\}, \quad (1)$$

where the subsequence $I_{j-k_i+1} \cdots I_j$ forms a candidate pattern to match with character C_i. We allow a candidate pattern to be formed by at most 4 primitive segments $(1 \leqslant k_i \leqslant 4)$.

Each possible alignment corresponds to a path from left-bottom to upper-right in the grid of Fig. 3. To search for the best alignment, each path is evaluated by a cost function. Inspired by [11][12], we define the alignment cost as a weighted sum:

$$g(A) = \sum_{h=0}^{4} \lambda_h \cdot F_h \,, \quad (2)$$

where λ_h (h=0,...,4) are the weight coefficients, F_0 is the character recognition cost, F_1, F_2, F_3 and F_4 are four geometric model costs, for the class-dependent and class-independent single character geometry, class-dependent and class-independent between-character relationships, respectively. Each term is the sum of costs over the path:

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

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

$$F_3 = \sum_{i=2}^{n} f_3[(C_i, I_{j-k_i+1} \cdots I_j) \,|\, (C_{i-1}, I_p \cdots I_{j-k_i})],$$

$$F_4 = \sum_{i=2}^{n} f_4(I_j \,|\, I_{j-1}),$$

To search for the best alignment with minimum cost by DP, we define $D(i, j)$ as the accumulated cost of optimal alignment between a partial string $\{C_1 \cdots C_i\}$ and partial image $\{I_1 \cdots I_j\}$. $D(i, j)$ can be updated from the preceding partial alignments by

$$D(i,j) = \min_k \begin{cases} D(i, j-1) + penalty(I_j) + \lambda_4 * f_4(I_j \,|\, I_{j-1}) \\ D(i-1, j) + penalty(C_i) \\ D(i-1, j-k) + \theta \end{cases}, \quad (3)$$

where $penality(I_j)$ is the cost of deleting primitive segment I_j, and $penality(C_i)$ is the cost of skipping character C_i. Moreover, the term θ is defined as:

$$\begin{aligned}\theta = &\sum_{h=0}^{1} \lambda_h * f_h(C_i, I_{j-k_i+1} \cdots I_j) + \lambda_2 * f_2(I_{j-k_i+1} \cdots I_j) \\ &+ \lambda_3 * f_3[(C_i, I_{j-k_i+1} \cdots I_j) \,|\, (C_i^{pre}, I_p \cdots I_{j-k_i})] \\ &+ \lambda_4 * f_4(I_{j-k_i+1} \,|\, I_{j-k_i})\end{aligned},$$

where C_i^{pre} represents the preceding non-skipped character of C_i. That is, $C_i^{pre}=C_l$, $l=i-1$ if C_{i-1} is not skipped, otherwise recursively set $l=l-1$ until C_l is not skipped).

The DP search starts with $D(0,0)=0$, then for $i=1,\ldots,n$ and $j=1,\ldots,m$, $D(i, j)$ are iteratively updated according to Eq. (3). Finally, $D(n, m)$ gives the total cost of optimal alignment, and the path backtracked to the start gives the character segmentation and labeling.

3.2 String-Level MCE training

The objective of training is to tune the combining weights so as to promote correct alignment and depress incorrect alignment. String-level training using the minimum classification error (MCE) criterion has shown superior performance in speech recognition and handwriting recognition [15][16].

By string-level training, the weights are estimated on a dataset of string samples (X^n, A_c^n), $n=1,\ldots,N$, where A_c^n denotes the correct alignment of the sample $X^n=(I^n,T^n)$. The weights are initialized as equal values and then iteratively updated on each string sample based on alignment using the current weights. Denoting the cost between a sample X^n and its alignment A^n as $g(X^n, A^n, \Lambda)$, where Λ is the set of weights. By the MCE method, the misclassification measure for correct alignment A_c^n is approximated by:

$$d(X^n, \Lambda) = -g(X^n, A_c^n, \Lambda) + g(X^n, A_r^n, \Lambda), \quad (4)$$

where $g(X^n, A_r^n, \Lambda)$ is the cost of the closest rival (the minimum cost alignment excluding the correct one).

The misclassification measure is transformed to loss by the sigmoidal function:

$$l(X^n, \Lambda) = \delta[\xi d(X^n, \Lambda)], \quad (5)$$

Where ξ is a parameter to control the hardness of sigmoidal nonlinearity. On the training sample set, the empirical loss is computed by

$$L(\Lambda) = \frac{1}{N} \sum_{n=1}^{N} l(X^n, \Lambda), \quad (6)$$

By stochastic gradient descent, the parameters are updated on each training sample by

$$\Lambda(t+1) = \Lambda(t) - \varepsilon(t) \frac{\partial l(X^t, \Lambda)}{\partial \Lambda} \Big|_{\Lambda=\Lambda(t)}, \quad (7)$$

Where $\varepsilon(t)$ is the learning step. Accordingly, the updating formula of each weight parameter can be derived as:

$$\lambda_h^{t+1} = \lambda_h^t - \xi \varepsilon(t) l(1-l) \sum_{i=1}^{n} [f_h(C_i, A_r^t) - f_h(C_i, A_c^t)], \quad h=0,1,\ldots,4, \quad (8)$$

where n is the string length of sample X^t, (C_i, A^t) denotes the alignment of character C_i.

4. Geometric Context Modeling

Since the distinct outline features of alphanumeric characters and punctuation marks can be exploited to improve the alignment accuracy, we design four statistical models for the class-dependent and class-independent single-character geometry (unary geometric context), and the class-dependent and class-independent between-character relationships (binary geometric context), respectively. In the following, we first describe the geometric features, and then describe the statistical models scoring these features.

4.1 Class-Dependent Geometric Features

For modeling single-character geometry, we extract 42 geometric features from a candidate character pattern, which are divided into three categories:

(1) 10 scalar features relative to the bounding box of the character, as the No.1-10 in Table 1.

(2) 4 scalar features relative to the vertical center of text line, as No.11-14 in Table 1.

(3) 28 profile-based features inspired by [17][18], as No.15-42 in Table 1.

Table 1. Class-dependent single-character geometric features (The last column denotes whether normalized w.r.t. the text line height or not)

No.	Feature	Norm
1-2	Height and width of bounding box	Y
3	Sum of inner gap	Y
4-5	Distances of horizontal/vertical gravity center to left/upper bound	Y
6	Logarithm of aspect ratio	N
7	Square root of bounding box area	Y
8	Diagonal length of bounding box	Y
9-10	Distances of horizontal/vertical gravity center to horizontal/vertical geometric center	Y
11-12	Distances of vertical gravity/geometric center to text line vertical gravity/geometric center	Y
13-14	Distances of upper/lower bound to text line vertical geometric center	Y
15-16	Means of horizontal/vertical projection profiles	Y
17-22	Normalized amplitude deviations, coefficients of skewness and kurtosis of the horizontal and vertical projection profiles	N
23-30	Means and deviations of the upper, lower, left and right outline profiles	Y
31-42	Normalized amplitude deviations, coefficients of skewness and kurtosis of the upper, lower, left and right outline profiles	N

We also extract 24 features for binary class-dependent geometric context, which are divided into two categories:

(1) 16 scalar features between the bounding boxes of two consecutive character patterns, as No.1-16 in Table 2.

(2) 8 features between the profiles of two consecutive character patterns, as No.17-24 in Table 2.

Table 2. Class-dependent between-character geometric features

No.	Features	Norm
1-6	Distances between the upper bounds, lower bounds, upper-lower bounds, lower-upper bounds, left bounds and right bounds	Y
7-8	Distances between the horizontal gravity centers and the vertical gravity centers	Y
9-10	Distances between the horizontal geometric centers and the vertical geometric centers	Y
11-12	Height and width of the box enclosing two consecutive characters	Y
13	Gap between the bounding boxes	Y
14	Ratio of heights of the bounding boxes	N
15	Ratio of widths of the bounding boxes	N
16	Square root of the common area of the bounding boxes	Y
17-20	Differences between the mean of upper, lower, left and right outline profiles	Y
21-24	Differences between the deviation of upper, lower, left and right outline profiles	Y

4.2 Class-Independent Geometric Features

For measure whether a candidate pattern is a valid character or not, we extract 12 class-independent geometric features. Ten of them are the same as the No.1-10 in Table. 1 and the remaining two are the No. 15-16 in Table. 1.

The class-independent geometry between two consecutive candidate patterns describes whether an over-segment gap is a valid between-character gap or not. For this we extract 14 class-independent features, 13 of them are the same as No.1-13 in Table 2, and the last is the convex hull-based distance as calculated in [19].

4.3 Statistical Models

For modeling the class-dependent geometry, we should first reduce the number of character geometry classes, since the number of Chinese characters is very large and many different characters have similar geometric features. Particularly, the binary class-dependent geometric model considers pairs of characters, and it is formidable to store $M \times M$ models (M is the number of character classes) and get enough training samples for so many models. Hence, we cluster the character classes into five super-classes using the EM algorithm based on our prior knowledge. After clustering, each single character is assigned to one of five super-classes and a pair of successive characters thus belongs to one of 25 binary super-classes. For estimating the statistical geometric models, training character samples are relabeled to 5 super-classes.

We use a quadratic distance function (QDF) for both the unary and binary class-dependent geometric models. For unary geometry, the 42D feature vector is reduced to 4-dimensional subspace by Fisher linear discriminant analysis (FLDA), and the projected samples are used to estimate the parameters of 5-class QDF. For binary geometry, the 25-class QDF is estimated using samples of 24D geometric features.

The unary class-independent geometry indicates whether a candidate pattern is a valid character or not. For this two-class problem, we use a linear support vector machine (SVM) trained with character and non-character samples. The binary geometry indicates whether a segmentation point is a between-character gap or not. We similarly use a linear SVM for this two-class problem, trained with two-class labeled samples.

5. Experimental Results

We evaluated the text alignment performance on a database of unconstrained offline handwritten Chinese documents, collected by Institute of Automation of Chinese Academy of Sciences (CASIA). We extracted 4,600 text lines with correct annotation of character boundaries, in which 3,600 lines were used for estimating the geometric model parameters and the remaining 1,000 were used for evaluation.

In the annotation system, the character recognition is a MQDF classifier on character shape features extracted from candidate patterns: 8-direction contour direction histogram feature using continuous NCFE (normalization-cooperated feature extraction) method combined with the MCBA (modified centriod boundary alignment) normalization method [19]. The 512D feature vector is reduced to 160D by FLDA. On the 160D feature vector, the MQDF uses 40 principal eigenvectors for each of 7,356 classes (7,185 Chinese characters, 10 Arabic numerals, 52 English letters and 109 other symbols). The output of the MQDF is a quadratic distance (cost).

Alignment accuracy is defined as the number of correctly aligned characters divided by the total number of characters in the transcript [6]. In our experiments, a match between a transcript character and primitive segments, $(C_i, I_{j-k+1} \cdots I_j)$, is judged as correct if the bounding box of the primitive segments and the bounding box of the true character image overlap sufficiently (the difference of top, bottom, left and right bounds does not exceed a threshold).

The five classifiers utilized in our system are summarized in Table 3. Except the character classifier that was trained with isolate character samples, the other four classifiers were trained with the text line data.

Table.3 Summary of classifiers in our system.

	Type	Dimension	Classifier
f_0	Character recognizer	512→160	MQDF
f_1	Unary class-dependent	42→4	QDF
f_2	Unary class-independent	12	SVM
f_3	Binary class-dependent	24	QDF
f_4	Binary class-independent	14	SVM

Table 4. Effects of geometric contexts.

Character recognizer + Geometric context					Alignment accuracy
f_0	f_1	f_2	f_3	f_4	
○					96.85
○	○				97.39
○		○			97.66
○			○		98.03
○				○	98.01
○	○	○			97.90
○	○		○		98.00
○		○		○	98.04
○			○	○	98.18
○	○	○	○	○	98.20

Table 4 shows the effects of geometric context models (f_1-f_4) on text line alignment. We can see that the incorporation of geometric contexts improve the alignment accuracies remarkably when using any geometric model or the combination of them. The binary geometric contexts perform better than the unary geometric contexts. This justifies the importance of between-character relationship. Comparing the class-dependent models and the class-independent models, the results show little difference between them. However, the combination of them gives the best alignment performance. This verifies that class-dependent and class-independent geometric models are complementary.

6. Conclusions and future works

We presented a method for modeling the geometric context in handwritten Chinese text lines and integrating to text alignment by combining with the character recognizer. Specifically, we built four statistical geometric models for class-dependent and class-independent single-character (unary) geometry and between-character (binary) relationships. Our experimental results demonstrate that the geometric models can improve the alignment accuracy significantly. Based on the text line alignment method, we have used the annotation tool to annotate a large number of handwritten Chinese document images. Further improvements using better character classifiers and applying the geometric models to character string recognition are undergoing.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (NSFC) under grant no.60775004.

References

[1] F. Yin, Q.-F Wang, C.-L Liu, A Tool for ground-truthing text lines and characters in off-line handwritten Chinese documents, *Proc. 10th Int'l Conf. on Document Analysis and Recognition*, Barcelona, Spain, pp. 951-955, 2009.

[2] E.M. Kornfield, R. Manmatha, J. Allan, Text alignment with handwritten document, *Proc. Int. Workshop on Document Image Analysis for Libraries (DIAL)*, pp.195-209, 2004.

[3] A. Kumar, A. Balasubramanian, A. Namboodiri, C.V. Jawahar, Model-based annotation of online handwritten datasets, *Proc. 10th Int. Workshop on Frontiers in Handwriting Recognition (IWFHR)*, pp.9-14, 2006.

[4] J. Rothfeder, T.M. Rath, R. Manmatha, Aligning transcripts to automatically segmented handwritten manuscripts, *Proc. 7th Int. Workshop on Document Analysis Systems (DAS)*, pp. 84-95, 2006.

[5] M. Zimmermann, H. Bunke, Automatic segmentation of the IAM off-line database for handwritten English text, *Proc. 16th Int. Conf. on Pattern Recognition*, vol.4, pp.35-39, 2002.

[6] B. Zhang, C. Tomai, S. Srihari, V. Govindaraju, Construction of handwriting databases using transcript-based mapping, *Proc. First Int. Workshop on Document Image Analysis for Libraries (DIAL)*, pp. 288-298, 2004.

[7] C. Huang, S.N. Srihari, Mapping transcripts to handwritten text, *Proc. 10th Int. Workshop on Frontiers in Handwriting Recognition (IWFHR)*, pp.15-20, 2006.

[8] F. Kimura, K. Takashina, S. Tsuruoka, 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, 1987.

[9] H. Xue, V. Govindaraju, Incorporating contextual character geometry in word recognition, *Proc. 8th Int Workshop on Frontiers in Handwriting Recognition*, Ontario, Canada, pp.123-127, 2002.

[10] M. Koga, T. Kagehiro, H. Sako, H. Fujisawa, Segmentation of Japanese handwritten characters using peripheral feature analysis, *Proc. 14th Int'l Conf. Pattern Recognition*, Brisbane, Australia, Vol.2, pp.1137-1141, 1998

[11] T. Fukushima, M. Nakagawa, On-line writing-box-free recognition of handwritten Japanese text considering character size variations, *Proc. 15th Int'l Conf. Pattern Recognition*, Barcelona, Spain, vol.2, pp.359-363, 2000.

[12] B. Zhu, M. Nakagawa, Online handwritten Japanese text recognition by improving segmentation quality, *Proc. 11th Int'l. Conf. on Frontiers in Handwriting Recognition*, Montreal, Canada, pp.379-384, 2008.

[13] X.-D. Zhou, J.-L. Yu, C.-L. Liu, T. Nagasaki, K. Marukawa, Online handwritten Japanese character string recognition incorporating geometric context", *Proc. 9th Int'l Conf. on Document Analysis and Recognition*, Curitiba, Brazil, Vol.1, pp.48-52, 2007.

[14] F. Yin, C.-L. Liu, Handwritten text line segmentation by clustering with distance metric learning, *Pattern Recognition*, Vol.42, pp. 3146-3157, 2009

[15] W. Chou, Discriminant-function-based minimum recognition error pattern recognition approach to speech recognition, *Proc. IEEE*, vol.88, no.8, pp.1201-1223, 2000.

[16] C.-L. Liu, K. Marukawa, Handwritten numeral string recognition: character-level training vs. string-level training, *Proc. 17th Int'l Conf. Pattern Recognition*, Cambridge, UK, Vol.1, pp.405-408, 2004.

[17] V. Lavrenko, T. M. Rath, R. Manmatha, Holistic word recognition for handwritten historical documents, *Proc. Int'l Workshop on Document Image Analysis for Libraries DIAL'04*, pp.278-287, 2004.

[18] Y.Y. Chung, M.T. Wong, High accuracy handwritten character recognition system using contour sequence moments, *Procs. 4th Int'l Conf. Signal Processing*, Beijing, China, Vol.2, pp. 1249-1252, 1998.

[19] G. Louloudis, B. Gatos, I. Pratikakis, C. Halatsis, Text line and word segmentation of handwritten documents, *Pattern Recognition*, Vol.42, pp.3169-3183, 2009.

[20] C.-L. Liu, K. Marukawa, Global shape normalization for handwritten Chinese character recognition: A new method, *Proc. 9th Int. Workshop on Frontiers in Handwriting Recognition*, Tokyo, Japan, pp. 300-305, 2004.