

A robust model for on-line handwritten Japanese text recognition

Bilan Zhu^a, Xiang-Dong Zhou, Cheng-Lin Liu^b, and Masaki Nakagawa^a

^a Tokyo University of Agriculture and Technology, Tokyo 184-8588, Japan; ^b Institute of Automation, Chinese Academy of Sciences, Beijing 100190

ABSTRACT

This paper describes a robust model for on-line handwritten Japanese text recognition. The method evaluates the likelihood of candidate segmentation paths by combining scores of character pattern size, inner gap, character recognition, single-character position, pair-character position, likelihood of candidate segmentation point and linguistic context. The path score is insensitive to the number of candidate patterns and the optimal path can be found by the Viterbi search. In experiments of handwritten Japanese sentence recognition, the proposed method yielded superior performance.

Keywords: On-line recognition, Character recognition, Recognition model, Segmentation, SVM, Writing constraint.

1. INTRODUCTION

Due to the development of pen-based systems such as tablet PCs, electronic whiteboards, PDAs, pen and paper devices like Anoto pen and so on, handwritten text recognition rather than character recognition is being sought with less constraints since larger writing surfaces allow people to write more freely. This poses new challenges on text line segmentation and character segmentation.

Character segmentation and recognition are usually integrated in a character string (text line or sentence) recognition process because characters cannot be reliably segmented before they are recognized due to the irregularity of character size and spacing. To improve the segmentation and recognition accuracy, the string recognition process should consider character recognition, geometric features and linguistic context for segmentation path evaluation [1][2].

In on-line Japanese text recognition, a stochastic model was presented to evaluate the likelihood of segmentation paths [1]. The likelihood, depending on the number of segmented characters, tends to recognize two or more characters as one character because a longer character sequence tends to have smaller evaluation score than a shorter one [2].

Zhou et al. gave a path evaluation method overcoming the effect of string length by normalizing the evaluation score with respect to the string length [2]. However, this normalized score bias to longer strings, and so, a character with multiple components tends to be split into multiple characters. In hidden Markov model (HMM)-based text recognition, the path score depends on the fixed length of observation sequence, but the character shape cannot be grasped well. Chen et al. proposed a variable duration HMM method for handwritten word recognition, where the probability of a hypothesized character is weighted by the number of primitives composing the character [3]. Yu et al. similarly use the number of primitives composing a character for weighting the character recognition score in segmentation path evaluation [4]. However they did not explain the logical ground for the path evaluation, did not weight other factors such as geometric features and the degree for weighting can not be controlled.

In this paper, we propose a robust recognition model for on-line handwritten Japanese text recognition. We evaluate the likelihood of candidate segmentation paths by combining scores of character pattern size, inner gap, character recognition, single-character position, pair-character position, likelihood of candidate segmentation point and linguistic context. Our method trains and decides the degree for weighting each factor with the number of composing primitives automatically by a genetic algorithm, and presents the ground for weighting them. The path score is insensitive to the number of candidate patterns. Since the path score remains cumulative with respect to the character string, the optimal path can be found by the Viterbi search. In experiments on Japanese sentence recognition, the proposed method outperforms previous methods.

Representing a character string pattern as a sequence of primitive segments: $\mathbf{X}=s_1\dots s_m$, which is partitioned into character patterns $\mathbf{Z}=z_1\dots z_n$, where each candidate pattern contains k_i primitive segments: $z_i = s_{j_i} \cdots s_{j_i+k_i-1}$. The segmented character patterns are assigned classes $\mathbf{C}=C_1\dots C_n$. To evaluate the score of string \mathbf{X} in respect of string class \mathbf{C} , we extract features for scoring the primitive segments (or candidate patterns) and between-segment (or between-character) compatibilities:

- Bounding box feature b_i
- Inner gap feature q_i
- Shape feature s_i or z_i
- Unary position feature p_i^u of single segment (or character)
- Binary position feature p_i^b between adjacent segments (or characters)
- Between-segment gap feature g_i , which is to be classified as SP or NSP

Denoting by $\mathbf{b}, \mathbf{q}, \mathbf{X}, \mathbf{p}^u, \mathbf{p}^b, \mathbf{g}$ for the sequences of features of primitive segments, the posterior probability of string class is given by:

$$\begin{aligned} P(\mathbf{C} | \mathbf{X}) &= P(\mathbf{C} | \mathbf{b}, \mathbf{q}, \mathbf{X}, \mathbf{p}^u, \mathbf{p}^b, \mathbf{g}) \\ &= \frac{P(\mathbf{b}, \mathbf{q}, \mathbf{X}, \mathbf{p}^u, \mathbf{p}^b, \mathbf{g} | \mathbf{C})P(\mathbf{C})}{P(\mathbf{b}, \mathbf{q}, \mathbf{X}, \mathbf{p}^u, \mathbf{p}^b, \mathbf{g})} \end{aligned} \quad (1)$$

In the above formula, the denominator is independent of the string class. Reasonably assuming independence between the different features, the string class can be equivalently evaluated by a score:

$$\begin{aligned} f(\mathbf{X}, \mathbf{C}) &= \log\left(P(\mathbf{b}, \mathbf{q}, \mathbf{X}, \mathbf{p}^u, \mathbf{p}^b, \mathbf{g} | \mathbf{C})P(\mathbf{C})\right) \\ &= \log P(\mathbf{C}) + \sum_{i=1}^m \left[\log P(b_i | c_i) + \log P(q_i | c_i) + \log P(s_i | c_i) \right. \\ &\quad \left. + \log P(p_i^u | c_i) + \log P(p_i^b | c_{i-1}, c_i) + \log P(g_i | t_i) \right] \end{aligned} \quad (2)$$

where c_i denotes a character category or a hypothetical category for partial character pattern that a primitive segment represents (we call it hyper-category), and t_i denotes SP or NSP.

$P(\mathbf{C})$ is represented by the tri-gram of hyper-categories for primitive segments. Since the tri-gram of hyper-categories is difficult to obtain, we approximate it by the tri-gram of character categories:

$$\begin{aligned}
\log P(\mathbf{C}) &= \sum_{i=1}^m \log P(c_i | c_{i-2}c_{i-1}) \\
&= \sum_{i=1}^n \left[\log P(c_{j_i} | c_{j_i-2}c_{j_i-1}) + \sum_{j=j_i+1}^{j_i+k_i-1} \log P(c_j | c_{j-2}c_{j-1}) \right] \\
&\approx \sum_{i=1}^n \left[\lambda_{11} \log P(C_i | C_{i-2}C_{i-1}) + \lambda_{12} \sum_{j=j_i+1}^{j_i+k_i-1} P(C_i | C_{i-2}C_{i-1}) + \lambda_1 \right] \\
&= \sum_{i=1}^n \{ [\lambda_{11} + \lambda_{12}(k_i - 1)] \log P(C_i | C_{i-2}C_{i-1}) + \lambda_1 \}
\end{aligned} \tag{3}$$

where λ_{11} and λ_{12} are weighting parameters, and λ_1 is a bias for balancing the number of characters. We approximate the transition probability of start segment of a character pattern and that of non-start segment using different weights for their varying effects.

Similarly, we use different weights for the start segment and non-start segment of the other features, obtaining the path score:

$$f(\mathbf{X}, \mathbf{C}) = \sum_{i=1}^n \left\{ \begin{aligned} &\sum_{h=1}^6 [\lambda_{h1} + \lambda_{h2}(k_i - 1)] \log P_h \\ &\lambda_{71} \log P(g_{j_i} | SP) + \lambda_{72} \sum_{j=j_i+1}^{j_i+k_i-1} \log P(g_{j_i} | NSP) \end{aligned} \right\} + \lambda \tag{4}$$

where P_h , $h=1, \dots, 6$, stand for $P(C_i | C_{i-2}C_{i-1})$, $P(b_i | C_i)$, $P(q_i | C_i)$, $P(z_i | C_i)$, $P(p^u_i | C_i)$ and $P(p^b_i | C_{i-1}C_i)$, respectively.

The weighting parameters λ_{h1} , λ_{h2} ($h=1 \sim 7$) and λ are selected using a genetic algorithm to optimize the string recognition performance on a training dataset.

The path score in Eq. (4) is accumulated over the primitive segments, and hence, is insensitive to the number of segmented character patterns. The optimal path can be found by the Viterbi search (dynamic programming).

The path evaluation scorer of [1] and that of [4] can be viewed as a special case of the proposed one in Eq. (4) by setting $\lambda_{h1}=1, \lambda_{h2}=0$ ($h=1 \sim 7$) and $\lambda=0$ for [1], and by setting $\lambda_{41}=\lambda_{42}$, $\lambda_{h2}=0$ ($h=1 \sim 3, 5 \sim 7$), and $\lambda=0$ for [4], respectively.

3.2 Evaluation of terms

$P(C_i | C_{i-2}, C_{i-1})$ is the transition probability from character C_{i-2} and C_{i-1} to C_i (tri-gram probability). It is reduced to unigram or bi-gram when C_i is the first or second character of a sentence. The tri-gram is smoothed to overcome the imprecision of training with insufficient text [5]:

$$P(C_i | C_{i-2}, C_{i-1}) = \beta_1 P(C_i | C_{i-2}, C_{i-1}) + \beta_2 P(C_i | C_{i-1}) + \beta_3 P(C_i) + \beta_4 \tag{5}$$

where the weights (subject to $\beta_1 + \beta_2 + \beta_3 + \beta_4 = 1$) are obtained by using text different from that for training tri-grams.

The values of geometric features b_i , q_i , p_{ui} and p_{bi} are normalized with respect to the average character size acs for scaling invariance. Several geometric features are shown in Fig. 2.

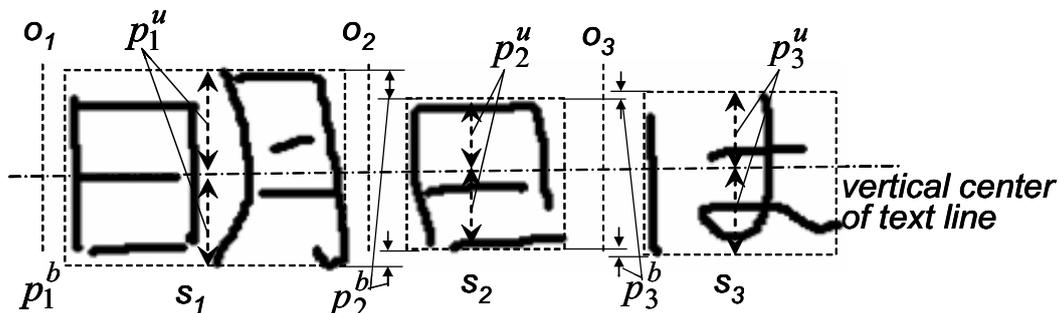


Fig. 2. Some geometric features.

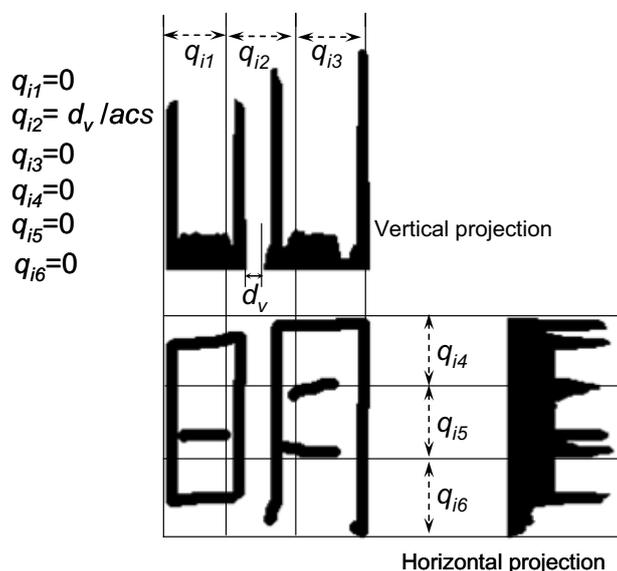


Fig. 3. Feature values of character pattern inner gap.

The feature vector b_i comprises the height and width of character pattern bounding box.

The feature vector q_i comprises six values as shown in Fig. 3. The first three values represent the horizontal gaps of three vertical slits in vertical projection, and the last three ones represent the vertical gaps of three horizontal in the horizontal projection. In each slit, if there are gaps more than 1, the sum of their lengths is used.

The feature vector p_{ui} comprises two elements: the vertical length from the center line to the top and bottom of the bounding box. The feature vector p_{bi} has two elements measured from the bounding boxes of two adjacent character patterns: the vertical distance between the upper bounds and the vertical distance between the lower bounds. $P(p_{b1}|C_1, C_0)$ is set as 1. To reduce the cardinality of $P(p_{bi}|C_{i-1}, C_i)$, we cluster the character classes into six super-classes according to the mean vector of the unary position features of each class on training samples using the k-means algorithm. $P(p_{bi}|C_{i-1}, C_i)$ is then replaced by $P(p_{bi}|C'_{i-1}, C'_i)$, where C'_{i-1}, C'_i are the super-classes of C_{i-1} and C_i , respectively.

The feature vector g_j comprises multiple features measuring the relationship between two primitive segments adjacent to a candidate segmentation point [8]. We approximate $P(g_j|SP)$ and $P(g_j|NSP)$ using a SVM classifier: training a binary classifier on the feature data of sampled segmentation points labeled as SP or NSP. The SVM output is warped to obtain

probabilities $P(o_i|SP)$ and $P(o_i|NSP)$, where o_j is the output of the SVM for g_j . The warping function is obtained from the distribution of SVM outputs on a validation dataset.

4. EXPERIMENTS

For evaluating the proposed character string recognition model, we trained the character recognizer and geometric scoring functions using a Japanese online handwriting database Nakayosi [6][7]. The character recognizer combined off-line and on-line recognition method by normalizing the recognition scores of off-line and on-line to conditional probabilities $P(s_i|C_i)$ [6]. For the geometric scores, four quadratic discriminant function (QDF) classifiers are trained for $P(b_i|C_i)$, $P(q_i|C_i)$, $P(p^u_i|C_i)$ and $P(p^b_i|C_{i-1}, C_i)$, respectively.

For scoring linguistic context, we prepared an initial tri-gram table from the year 1993 volume of the ASAHI newspaper and the year 2002 volume of the NIKKEI newspaper. We estimated the smoothing parameters $\beta_1, \beta_2, \beta_3, \beta_4$ using the Nakayosi database. The data size of tri-gram was reduced to 6MB by suppressing non-occurring terms, neglecting a small number of occurrences, and quantizing the logarithm values of tri-gram probabilities.

Table 1. Statistics of training/test text lines.

	#Text lines	#Character patterns	#Character classes	#Characters per line
Training	10,174	10,4093	1,106	10.23
Testing	3,511	35,686	790	16.89

For training weighting parameters and evaluating the performance of character string recognition, we extracted horizontally written text lines from the database *HANDS-Kondate_t_bf-2001-11* collected from 100 people. We used 75 persons' text lines (training dataset) to train the SVM model for the candidate segmentation point probability and to train the weighting parameters of path evaluation score using the genetic algorithm. After training the weighting parameters, we used the text lines of the remaining 25 persons for testing. The statistics of the training and test text lines are listed in Table 1.

We compare the performance of the proposed method and the ones presented in [1] and [2]. For fair comparison, all the three methods use the same tri-gram for language context and same classifiers for character recognition and geometric context. The weighting parameters are optimized using the genetic algorithm for each method. The three methods combines the same seven terms in Eq. (4) for path evaluation, but the method of [1] (Method 1) does not use the term related to k_i (number of primitive segments composing a character pattern), the method of [2] (Method 2) normalizes the path score of Method 1 using the number of segmented characters. The experiments were implemented on a Pentium (R) 4 2.80 GHz CPU with 512 MB memory.

For Method 2, we use beam search for finding the optimal paths in the candidate lattice, because the path score is not cumulative with the character sequence. For the proposed method and the Method 1, the optimal paths are found by the Viterbi search. For all the three methods, the candidate lattice retains 10 candidate classes for each character pattern.

We use a character segmentation measure f , the character recognition rate C_r , and average string recognition time $T_{av_rec_tl}$ to evaluate the performance of text line recognition. The measure f is defined below:

$$f = \frac{2}{1/r + 1/p}$$

$$r = \frac{\text{number of correctly classified segmentation points}}{\text{number of true segmentation points}}$$

$$p = \frac{\text{number of correctly classified segmentation points}}{\text{number of classified segmentation points (including false)}}$$

Table 2 shows the string recognition results of the three methods. For reference, the trained weighting parameters of Eq. (4) are as follows:

$(\lambda_{11}, \lambda_{12}, \lambda_{21}, \lambda_{22}, \lambda_{31}, \lambda_{32}, \lambda_{41}, \lambda_{42}, \lambda_{51}, \lambda_{52}, \lambda_{61}, \lambda_{62}, \lambda_{71}, \lambda_{72}, \lambda) = (0.351, 0.000, 0.265, 0.001, 0.199, 0.000, 1.000, 0.641, 0.009, 0.000, 0.100, 0.000, 0.323, 0.120, 0.100)$

Table 2. Results of text line recognition.

Performance \ Method		Method	Method proposed in this paper	Method 1	Method 2
Training dataset	f		0.9941	0.9906	0.9850
	Rc		92.65%	91.68%	91.12%
	$T_{av_rec_tl}$		1.31 (s)	1.32 (s)	1.32 (s)
Testing dataset	f		0.9934	0.9903	0.9827
	Rc		92.80	91.94%	91.07%
	$T_{av_rec_tl}$		1.32(s)	1.33 (s)	1.33 (s)

From the results we can see that our model improved the character recognition and segmentation accuracy. For Method 1 depending on the number of segmented characters, a longer character sequence tends to have smaller evaluation score than a shorter one. For Method 2 this normalized score bias to longer strings, and so, a character with multiple components tends to be split into multiple characters. Our model resolved these problems, and the path score is insensitive to the number of segmented characters. The three methods take almost the same processing time. Method 2 employs beam search while the proposed method and Method 1 apply the Viterbi search for the limited number of candidate paths.

5. CONCLUSION

We presented a robust recognition model for on-line handwritten Japanese text. Our model evaluated the likelihood of candidate segmentation paths by combining multiple features such as geometric features, character recognition and linguistic context, and it nearly independent of the length of segmentation paths. The experiments showed its superior performance. We will improve the accuracy of over-segmentation and examine the effect of each geometric feature.

ACKNOWLEDGMENTS

This research is partially supported by Grant-in-Aid for Scientific Research under the contract number (B)17300031 and the MEXT Research and Education Fund for Promoting Research on Symbiotic Information Technology.

REFERENCES

- [1] Nakagawa M., Zhu B. and Onuma M., "A model of on-line handwritten Japanese text recognition free from line direction and writing format constraints," *IEICE Trans. Inf. & Syst.*, E88-D, 8, 1815-1822 (2005).
- [2] Zhou X.-D., Yu J.-L., Liu C.-L., Nagasaki T. and Marukawa K., "Online handwritten Japanese character string recognition incorporating geometric context," *Proc. 9th ICDAR*, Curitiba, Brazil, 48-52 (2007).
- [3] Chen M.-Y., Kundu A. and Srihari S. N., "Variable duration hidden Markov model and morphological segmentation for handwritten word recognition," *IEEE Trans. on Image Processing*, vol. 4, no. 12, 1675-1687 (1995).
- [4] Yu J.-L., Zhou X.-D. and Liu C.-L., "Search strategies in online handwritten character string recognition (in Chinese)," *Proc. CCPR 2007 (Chinese Conference on Pattern Recognition)*, China, 299-305 (2007).
- [5] Mori H. and Aso H., "Postprocessing for Japanese document recognition based and 2nd order Markov model (in Japanese)," *Technical Report of IPS J, NL*, 94, 63, 89-96 (1994).
- [6] Oda H., Zhu B., Tokuno J., Onuma M., Kitadai A. and Nakagawa M., "A compact on-line and off-line combined recognizer," *Proc. IWFHR-10*, La Baule, France, 133-138 (2006).
- [7] Nakagawa M. and Matsumoto K., "Collection of on-line handwritten Japanese character pattern databases and their analysis," *IJDAR*, vol.7, no.1, 69-81 (2004).
- [8] Zhu B. and Nakagawa M., "Segmentation of on-line freely written Japanese text using SVM for improving text recognition," *IEICE Trans. Inf. & Sys.*, E91-D(1), 105-113 (2008).