

# A Model of On-line Handwritten Japanese Text Recognition Free from Line Direction and Writing Format Constraints

Masaki NAKAGAWA<sup>†</sup>, Member, Bilan ZHU<sup>†a)</sup>, and Motoki ONUMA<sup>†</sup>, Student Members

**SUMMARY** This paper presents a model and its effect for on-line handwritten Japanese text recognition free from line-direction constraint and writing format constraint such as character writing boxes or ruled lines. The model evaluates the likelihood composed of character segmentation, character recognition, character pattern structure and context. The likelihood of character pattern structure considers the plausible height, width and inner gaps within a character pattern that appear in Chinese characters composed of multiple radicals (subpatterns). The recognition system incorporating this model separates freely written text into text line elements, estimates the average character size of each element, hypothetically segments it into characters using geometric features, applies character recognition to segmented patterns and employs the model to search the text interpretation that maximizes likelihood as Japanese text. We show the effectiveness of the model through recognition experiments and clarify how the newly modeled factors in the likelihood affect the overall recognition rate.

**key words:** on-line recognition, character recognition, segmentation, probabilistic model, writing constraint

## 1. Introduction

Demand to remove writing constraint from on-line handwriting recognition is getting higher and higher since people can write more freely on enlarged surfaces of tablet PCs, electronic whiteboards and on new paper-based handwriting environments such as the Anoto pen, e-pen and so on. On such surfaces, Asian people whose languages are Chinese origin often write text horizontally, vertically or even slantingly in a mixed way.

Japanese text includes various sizes of character patterns ranging from so-called “half-width” characters like numbers and symbols, Kana characters (two sets of phonetic characters), Kanji characters of Chinese origin consisting of only one radical to those consisting of multiple radicals. Moreover, handwriting even magnifies the size variations as shown in Fig. 1. Some characters may be several times longer and/or wider than others.

The research to remove writing format constraint started from horizontal text recognition without character writing boxes, which is nowadays common on PDAs to avoid segmentation problem. Murase et al. made an initial attempt by applying DP-matching to find the best interpretation of a character pattern sequence [1]. However, the likelihood of segmentation was not considered and the likelihood

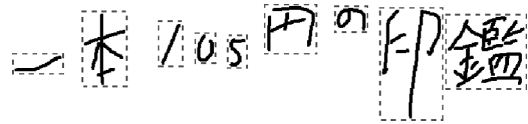


Fig. 1 An example of Japanese handwritten text.

of context (as Japanese text) was used only for verification. Okamoto et al. proposed geometrical rules and features to improve segmentation reliability while decreasing segmentation hypotheses [2]. Aizawa et al. applied a neural network using geometric features to realize real-time segmentation of characters [3]. Fukushima et al. took a probabilistic approach and considered the likelihood based on character segmentation, shape (recognition), context, and character size [4], [5]. By incorporating the factor of character size in determining the likelihood, they showed better performance than without including it. Senda et al. published a similar approach to the above method and formulated the problem as a search for the most probable interpretation of character segmentation, recognition and context, but they did not deal with the likelihood of character size [6], [7].

In off-line handwriting recognition, the same problem of character segmentation and recognition occurs. Due to the difficulty of the problem, however, research and development have been almost limited to the postal address recognition where only smallest and largest character size constraints have been considered as in [8]. For postal address recognition, lexical and contextual constraint is strong enough to find the best address interpretation. For ordinary text, however, the size likelihood is more effective than the size constraint.

Most of the previous publications and systems have been assuming only horizontal or vertical lines of text, while we have been trying to relinquish any writing constraint from on-line text input. We proposed a method to recognize mixtures of horizontal, vertical and slanted lines of text with assuming normal character orientation [9], and a revised method with arbitrary character orientation [10]. This paper presents a formalization of on-line unconstrained, line-direction free handwritten Japanese text recognition with normal character orientation and the effect of newly introduced factors in the likelihood evaluation. Due to the formalization, we have improved the overall recognition rate.

In this paper, Sect. 2 presents the problem that we must solve. Section 3 describes hypothetical segmentation of a text line element into character patterns for which the model of on-line handwritten Japanese text recognition is applied.

Manuscript received November 21, 2004.

Manuscript revised April 11, 2005.

<sup>†</sup>The authors are with the Graduate School of Engineering, Tokyo University of Agriculture and Technology, Koganei-shi, 184-8588 Japan.

a) E-mail: zhubilan@hands.ei.tuat.ac.jp

DOI: 10.1093/ietisy/e88-d.8.1815

Section 4 presents the model in a probabilistic framework and Sect. 5 describes implementation briefly and presents evaluation. Section 6 concludes this paper.

## 2. Problem

Here, we first define some terminologies. A stroke denotes a series of pen-tip coordinates sampled from pen down to pen up. An off-stroke is a vector indicating pen movement between successive strokes. Character orientation is used to specify the direction of a character from its top to bottom while line direction is used to designate the writing direction of a sequence of characters until it changes (Fig. 2). A text line is a piece of text separated by new-line or large space and it is further divided into text line elements at the changing points of line direction. Each text line element has its line direction (Fig. 3). The line direction and the character orientation are independent.

Our on-line recognition system of handwritten Japanese text, which is free from character orientation, line direction and any writing format constraint, takes the following steps:

- Step 1:** separates freely written text into text line elements and estimates the average character size of each text line element.
- Step 2:** estimates line direction and character orientation of each text line element and rotates it so that character orientation is normalized.
- Step 3:** hypothetically segments a character-orientation-normalized text line element into character patterns using geometric features.
- Step 4:** recognizes hypothetically segmented patterns as characters.
- Step 5:** applies the model to search the text interpretation that maximizes the likelihood composed of character segmentation, character recognition, character pattern structure and context.

We focus our attention to the model for the step 4

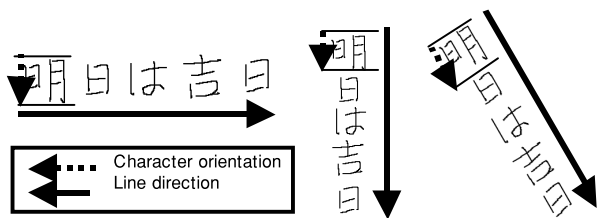


Fig. 2 Line direction and character orientation.

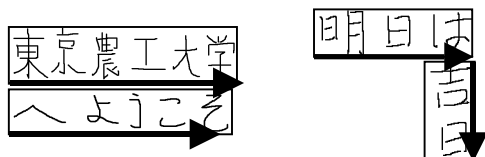


Fig. 3 Text line element and line direction.

and step 5, i.e., the problem of recognizing a character-orientation-normalized handwritten Japanese text line element of arbitrary line direction. Hereafter, we assume a text line element as a character-orientation-normalized text line element unless it is explicitly mentioned.

## 3. Segmentation of a Text Line Element

Before proceeding to the model construction, however, we must consider the step 3 of hypothetically segmenting a text line element into character patterns. It is the step that outputs candidates of segmentation by which a handwritten text line element is split into a sequence of character patterns.

There are many characters in the Japanese character set that can be divided into multiple character patterns. For example, the patterns shown in Fig. 4 (a) can be read as either  $C_1$ , a character in itself, or as the two consecutive characters  $C_2C_3$ . Which of the two is correct is determined by the characters (or strings) proceeding and/or following it. In the example of Fig. 4 (b), the character  $C_4$  follows, which causes the pattern of Fig. 4 (a) to be read as  $C_1$ . In Fig. 4 (c), on the other hand, the characters  $C_5C_6$  follow, which causes the pattern to be read as two characters  $C_2C_3$ . This example shows that the position of character segmentation can be different even for the same handwritten pattern depending on the context and it is therefore difficult to segment characters deterministically on the basis of geometrical features alone.

Hypothetical segmentation depends also on character orientation and line direction. After character orientation is normalized, it depends on line direction of a text line element as shown in Fig. 5. Note that segmentation hypothesis is often made within character pattern and it is different even for the same character pattern depending on the line direction. The quantization can be finer but the 4-directional quantization shown in Fig. 5 is adequate and effective to prevent a text line element from being segmented excessively. When the line direction is classified into, say, downward or upward (rightward or leftward), a considerable gap projected on the vertical axis (the horizontal axis) or a long off-stroke to the quantized line direction is employed as a candidate for segmentation but strokes or off-strokes to the opposite direction are used to merge its crossing strokes with the result that hypotheses on segmentation can be decreased, which is then effective to speed up the text recognition and to increase the recognition rate.

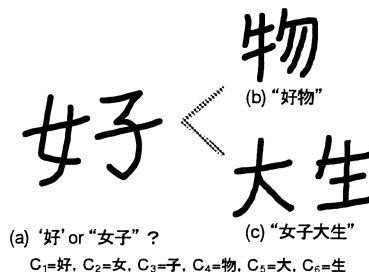


Fig. 4 An example of segmentation ambiguity.

Consequently, a text line element is hypothetically segmented according to its quantized direction. For rightward or leftward (horizontal) direction, a text line element is segmented vertically while producing bounding boxes  $b_i$  of character candidates and gaps  $g_i$  between them along the text line. For downward or upward (vertical) direction, it is segmented horizontally. Even if the direction is not perfectly along one of the four directions, it is quantized into one of them and segmented as shown in Fig. 6.

Here, we do not go into the detail of the hypothetical segmentation. Only what we need is the assumption that a text line element is segmented with all the true segmentation positions included as well as some false positions into a segmentation  $S$ , i.e., a sequence of character pattern structures and gaps. A character pattern structure is defined as a bounding box with an arbitrary number of inner gaps within it.

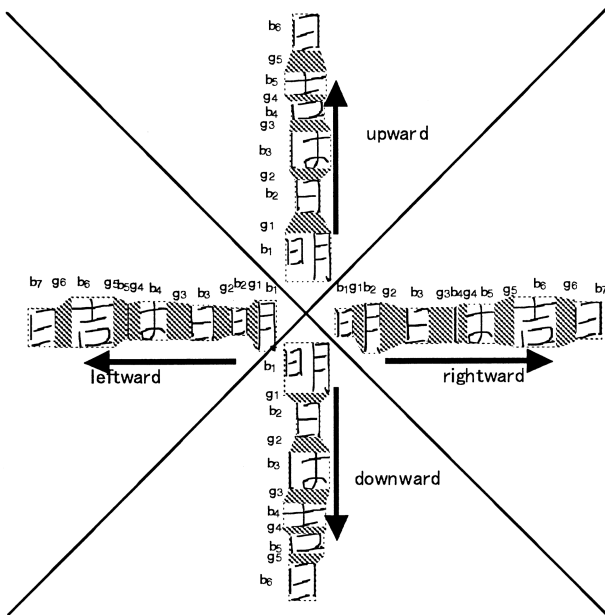


Fig. 5 Quantization of line direction.

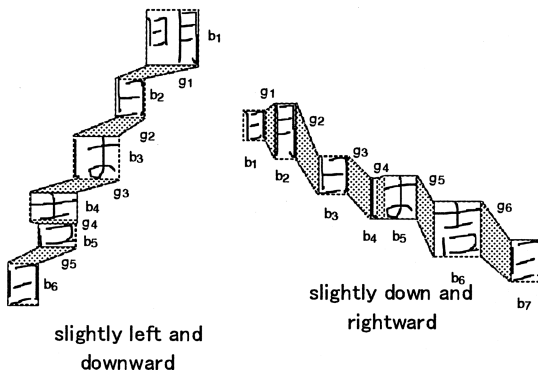


Fig. 6 Segmentation of various text line elements.

#### 4. Model of Recognition

The probability that a given pattern  $X$  is segmented as a segmentation  $S = s_1g_1s_2g_2 \cdots s_i g_i \cdots s_m g_m$ , where  $s_i$  is an  $i$ -th character pattern structure that is bounded by a box  $b_i$  of the height  $h_i$  and the width  $w_i$  and includes inner gaps  $q_{ik}$  ( $k = 0, 1, 2, \dots$ ) while  $g_i$  is an  $i$ -th outer gap, and then recognized as a character sequence  $C = C_1C_2 \cdots C_i \cdots C_m$  is defined as the conditional probability  $P(C,S|X)$  and it is transformed as follows:

$$P(C,S|X) = \frac{P(C).P(X,S|C)}{P(X)} \tag{1}$$

The goal is to find the segmentation  $S$  and the character sequence  $C$  that maximize  $P(C,S|X)$  among candidate segmentations as shown in Fig. 7 and among candidate character sequences as shown in Fig. 8. Since  $P(X)$  is the probability that a pattern  $X$  occurs regardless of  $S$  and  $C$ , we ignore it. Hereafter, we will consider  $P(C)$  and  $P(X,S|C)$ .

##### 4.1 Probability $P(C)$

In Eq. (1),  $P(C)$  is the probability that a character sequence  $C$  occurs. Assuming the 1st order Markov chain,  $P(C)$  is transformed with  $C_i$  denoting the  $i$ -th character in  $C$  as follows:

$$P(C) = \prod_{i=1}^m P(C_i|C_{i-1}) \tag{2}$$

$m$ : the number of characters in  $C$ .

$P(C_i|C_{i-1})$ : the probability that a character  $C_{i-1}$  is

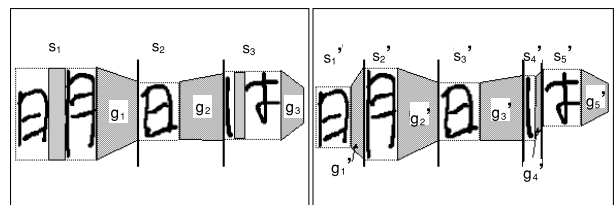


Fig. 7 Different segmentations for the same pattern.

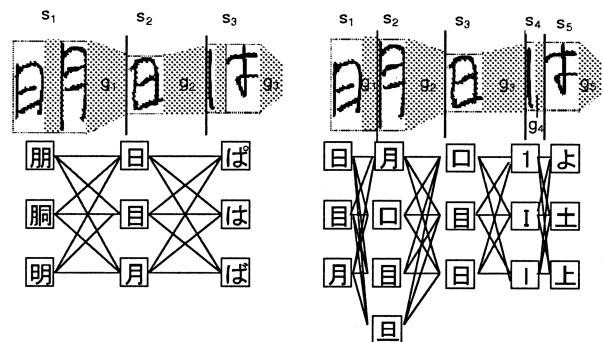


Fig. 8 Different candidate character sequences for the same segmentation.

succeeded by  $C_i$  (bi-gram probability).

$C_0$ : the state before the first character occurs.

$P(C_1|C_0)$ : the probability that a character  $C_1$  occurs at the top of text.

The term  $P(C_i|C_{i-1})$  is computed from a large volume of corpus. A more elaborate model using tri-gram or generally n-gram is possible, but its actual implementation takes larger computational cost.

## 4.2 Probability $P(X,S|C)$

In Eq.(1),  $P(X,S|C)$  is the probability that a character sequence  $C$  is written as a segmentation  $S = s_1g_1s_2g_2 \cdots s_i g_i \cdots s_m g_m$  and a character pattern sequence  $X = X_1X_2 \cdots X_i \cdots X_m$ , where  $X_i$  denotes a stroke sequence  $X_i = x_{i1}x_{i2} \cdots x_{ik}$  within  $s_i$ . Using the Basian law:

$$P(X,S|C) = P(X|S,C).P(S|C) \quad (3)$$

The terms  $P(S|C)$  and  $P(X|S,C)$  are the probability that a character sequence  $C$  is written so as to be segmented as  $S$  and the probability that a character sequence  $C$  segmented as  $S$  produces a character pattern sequence  $X$ , respectively.

## 4.3 Probability $P(S|C)$

This is the probability that a character sequence  $C$  is written so as to be segmented as  $S$ . We assume the probability that a character  $C_i$  is written in a structure  $s_i$  depends only on the character  $C_i$  and the average size  $\bar{C}$  of the character sequence  $C$ . We also assume the probability that an outer gap  $g_i$  occurs between the characters  $C_i$  and  $C_{i+1}$  depends only on the characters  $C_i$  and  $C_{i+1}$ , and the average size  $\bar{C}$  of the character sequence  $C$ :

$$P(S|C) \approx \prod_{i=1}^m P(s_i|C_i, \bar{C}).P(g_i|C_i, C_{i+1}, \bar{C}) \quad (4)$$

where the term with  $C_{m+1}$  is ignored.

If we assume that the scale of  $s_i$  and  $g_i$  is proportional to the average size  $\bar{C}$ , we can scale them by  $\bar{C}$ . Then,  $P(s_i|C_i, \bar{C})$  and  $P(g_i|C_i, C_{i+1}, \bar{C})$  are replaced by  $P(s_i/\bar{C}|C_i)$  and  $P(g_i/\bar{C}|C_i, C_{i+1})$ , respectively (cf. Appendix).

$P(s_i/\bar{C}|C_i)$  is the probability that a character  $C_i$  is written in a structure  $s_i$  whose height is  $h_i/\bar{C}$ , width is  $w_i/\bar{C}$  and includes inner gaps  $q_{ik}/\bar{C}$  ( $k = 0, 1, 2, \dots$ ). The character pattern structure is an extension of character size in [4], [5]. The simplest approximation to  $P(s_i/\bar{C}|C_i)$  is to assume a constant probability regardless of  $C_i$ . The second simplest way is to classify characters into several groups  $G_i$  and apply distinct probabilities  $P(s_i/\bar{C}|G_i)$ . Grouping can be made for numerals, alphabets, simple Kanji characters composed of single radicals, those composed of left and right radicals, those composed of top and bottom radicals and so on.

On the other hand,  $P(g_i/\bar{C}|C_i, C_{i+1})$  is the probability that an outer gap  $g_i$  occurs between the characters  $C_i$  and  $C_{i+1}$  and it can be assumed as a constant regardless of  $C_i$  and  $C_{i+1}$  or can be approximated by distinct probabilities depending on  $G_i$  including  $C_i$  and  $G_{i+1}$  including  $C_{i+1}$ .

## 4.4 Probability $P(X|S,C)$

This is the probability that a character sequence  $C$  segmented as  $S$  produces a pattern  $X$  and approximated as:

$$P(X|S,C) \approx \prod_{i=1}^m P(X_i|s_i, C_i) \quad (5)$$

The probability  $P(X_i|s_i, C_i)$  is that each character  $C_i$  is written in a structure  $s_i$  and represented by the stroke sequence  $X_i$ .

## 4.5 Total Evaluation Function

If we summarize the above transformations and approximations:

$$P(C).P(X,S|C) = \left( \prod_{i=1}^m P(C_i|C_{i-1}) \right) \times \left( \prod_{i=1}^m P(s_i/\bar{C}|C_i).P(g_i/\bar{C}|C_i, C_{i+1}).P(X_i|s_i, C_i) \right) \quad (6)$$

Then, by taking log of the both sides:

$$\log(P(C).P(X,S|C)) = \sum_{i=1}^m \log P(C_i|C_{i-1}) + \sum_{i=1}^m (\log P(s_i/\bar{C}|C_i) + \log P(g_i/\bar{C}|C_i, C_{i+1}) + \log P(X_i|s_i, C_i)) \quad (7)$$

In the right-hand side of Eq. (7), the first term considers context likelihood in terms of bi-gram, the second term is related to character recognition likelihood, the third term and forth term evaluates character pattern structure likelihood and outer gap likelihood, respectively.

## 5. Implementation and Evaluation

In this section, we describe implementation of the model briefly and present evaluation of the model.

### 5.1 Implementation

We prepared the bi-gram table for the context likelihood from 55,000,000 characters of text in the year 1993 volume of the ASAHI newspaper (one of the major Japanese newspaper publishers). By suppressing not occurring bi-grams, we reduced its size to 3.49 MB for 4,799 character categories. It is possible to reduce the bi-gram table even more by neglecting a small number of occurrences, but we used the above table.

As for the character structure likelihood  $P(s_i/\bar{C}|C_i)$ , we approximated it as  $P(h_i/\bar{C}, w_i/\bar{C}|C_i).P(q_{ik}/\bar{C}|C_i)$ . We employed the most faithful implementation. For each character category, we obtained the distribution of its height

( $h$ ), width ( $w$ ) and inner gaps  $q_k$  ( $k = 0, 1, \dots$ ) from our on-line handwritten character pattern databases, kuchibue\_d and nakayosi\_t [11].

In order to obtain  $P(h_i/\bar{C}, w_i/\bar{C}|C_i)$ , we counted the frequencies of the height and the width over the average character size  $\bar{C}$  within every 0.1 step from 0.0 to 1.9 for each character category  $C_i$ , as shown in Table 1. We blurred the matrix by another matrix of the Gaussian values as follows:

$$h(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad (8)$$

We set its size as  $15 \times 15$  and its variance  $\sigma$  as 0.18 by testing several values in experiments, and took  $x$  and  $y$  at every 0.1 step from  $-0.7$  to  $0.7$ .

Dividing all the values by the total number of occurrences  $N_{C_i}$  of the character category ( $C_i$ ) produces the probability that each quantized height and width over  $\bar{C}$  occurs. For zero components, we approximated them by a very small value close to 0. This naive implementation requires the large memory space. i.e., 3.39 MB for 4,443 categories. For practical systems, however, this size could be reduced.

As for the inner gap, we considered the vertical gap and the horizontal gap independently. The former has some width ( $w_g$ ) and the same height as the character while the latter has some height ( $h_g$ ) and the same width as the character. In order to obtain  $P(q_{ik}/\bar{C}|C_i)$ , we counted the frequencies of the gap width over  $\bar{C}$  within every 0.1 step from 0.0 to 0.9 for each character category  $C_i$ . We represented the frequencies by a one-dimensional array. Similarly, we counted the frequencies of the gap height over  $\bar{C}$  and represented them by another one-dimensional array. We blurred the one-dimensional arrays by another one-dimensional array of the Gaussian values as follows:

$$h(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right) \quad (9)$$

We set its size as 5 and its variance  $\sigma$  as 0.04 by testing several values in experiments, and took  $x$  at every 0.1 step from  $-0.2$  to  $0.2$ .

By replacing zero frequency by a very small value close to 0 and dividing all the frequencies by their sum, we have produced the probability that each quantized gap height or gap width over  $\bar{C}$  occurs. Memory requirement is 173 KB for 4,443 character categories.

**Table 1** Frequencies of height and width of the punctuation character “.”.

$h_i/\bar{C} \backslash w_i/\bar{C}$	0.0	0.1	0.2	0.3	...	1.9
0.0	9	8	1	0	0	0
0.1	17	30	7	0	0	0
0.2	4	11	5	0	0	0
0.3	0	0	0	0	0	0
0.4	0	0	0	0	0	0
...	0	0	0	0	0	0
1.9	0	0	0	0	0	0

On the other hand, we chose the simplest implementation for  $P(g_i/\bar{C}|C_i, C_{i+1})$ , i.e., we determined a constant distribution regardless of  $C_i$  and  $C_{i+1}$  from a small amount of unconstrained handwritten text since we did not have a large database of unconstrained handwritten text. The table size to store this distribution is only 56 Byte.

As for  $P(X_i|s_i, C_i)$ , we approximate it from the score of character recognition.

For character recognition, we normalize its shape, namely we rather neglect its structure so that:

$$P(X_i|s_i, C_i) \approx P(X_i|C_i) \quad (10)$$

Hereafter, we omit the suffix  $i$ . Given  $C$ , the recognition system outputs score  $\delta$  to  $X$ . Therefore:

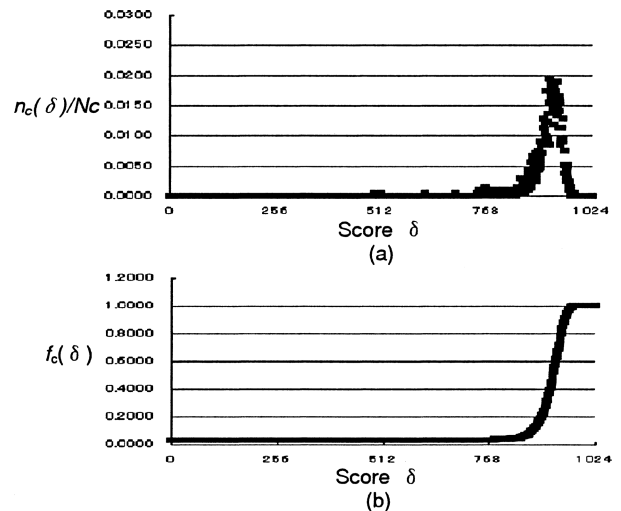
$$\begin{aligned} P(X|C) &= P(X, \delta|C) \\ P(X, \delta|C) &= P(\delta|C).P(X|\delta, C) \end{aligned} \quad (11)$$

Assuming that  $(\delta, C)$  and  $X$  have the strongest correlation, we remove the second term of the right hand side with the result that:

$$P(X, \delta|C) = P(\delta|C) \quad (12)$$

Assuming that the term  $n_c(\delta)$  denotes the number of learning patterns of the category  $C$  that are scored as  $\delta$  by the recognizer. We expect that  $P(\delta_a|C) \ll P(\delta_b|C)$  if  $\delta_a \ll \delta_b$ .

From a set of learning patterns, we can obtain the right hand side as  $n_c(\delta)/N_c$  where  $N_c$  is the number of learning patterns of the category  $C$ . Figure 9 (a) shows a histogram for a Chinese character “三” by our on-line recognizer where we neglect the number of rejection. The problem here is that the number of occurrences drops sharply as the score approaches to the best score. This is in some sense reasonable since the number of learning patterns decreases as they come close to the prototype of each category. This is strange, however, if we take  $n_c(\delta)/N_c$  as  $p(\delta/C)$ . Therefore, we take the following cumulative function instead of  $n_c(\delta)/N_c$ :



**Fig. 9** Histogram for a Chinese character “三”.

$$f_c(\delta) = \frac{\sum_{\delta=0}^{\delta} n_c(\delta)}{N_c} \quad (13)$$

This has the similar shape and prove monotonic increase as shown in Fig. 9 (b).

We may consider this function for each character category or take the summation over the entire character set and take the following single function where  $N_{cc}$  denotes the number of character categories:

$$f_s(\delta) = \frac{\sum_c f_c(\delta)}{N_{cc}} \quad (14)$$

or the weighted version:

$$f_w(\delta) = \frac{\sum_c N_c \cdot f_c(\delta)}{\sum_c N_c} \quad (15)$$

In practice, we took the Eq.(15) and obtained  $f_w(\delta)$  from the database nakayosi\_t with its size as 4.88 KB. Space complexity of the recognition system is summarized in Table 2.

We have fixed the number of candidate categories as 10 for each hypothetically segmented pattern when the character recognizer produces candidates for each segmented pattern.

Time complexity of the recognition system on a Pentium (R) 4 2.26 GHz CPU with 1.0 GB memory is shown in Table 3. Since the step 1 and the step 2 in Sect. 2 are made in negligibly small time, we only show the time complexity from the step 3 to the step 5.

### 5.2 Evaluation

We collected 136 test patterns consisting of 1,385 characters from 17 people.

We define character recognition rate (*crr*) as the number of correctly recognized characters over the number of total character patterns. On the other hand, we employ the *F* measure composed of *recall*(*R*) and *precision*(*P*) to show the segmentation performance since there are two kinds of

error in segmentation that a true segmentation is not detected and a false segmentation is detected:

$$F = \frac{2}{1/R + 1/P}$$

$R$  = number of correctly detected segmentation positions / number of true segmentation positions

$P$  = number of correctly detected segmentation positions / number of detected segmentation positions (including false) (16)

Table 4 shows how each term of the likelihood damages the performance when not employed in comparison with the last row where all the terms are employed.

If we do not consider all the terms of the context likelihood, the character recognition likelihood, the character pattern structure likelihood and the outer gap likelihood, i.e., if we output the sequence of the top candidate for every hypothetically segmented pattern, the performance is low as shown in the first row. The worst *Fmeasure* for segmentation, 0.7989 is observed since it is the result of hard decision on segmentation.

The second row shows the result of neglecting the character recognition likelihood among candidate categories selected as the result of character recognition. Although 65.59% character recognition rate looks too good without the character recognition likelihood, it is the result after candidates are reduced into 10 by the character recognition process.

The third row shows large importance of the context likelihood since its unemployment damages the character recognition performance as the same as the above two.

The fourth row shows still large effect of the character pattern structure likelihood. An example of its effect seen in Fig. 10. Without this term, two characters written narrowly were recognized incorrectly into a single character. With this term, however, they are recognized correctly.

**Table 2** Memory requirement for tables and dictionaries.

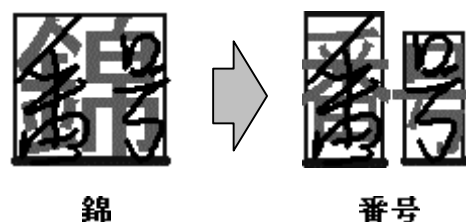
Tables/Dictionaries		memory size
Bi-gram		3.49 MB
Character structure likelihood	Height and Width	3.39 MB
	Inner gap	173 KB
Outer gap likelihood		56B
Conversion from score to likelihood		4.88 KB
Character pattern prototypes		838 KB

**Table 3** Time complexity per character.

process	processing time / character
Step 3: hypothetical segmentation	1.99 m sec
Step 4: recognition of hypothetically segmented patterns	14.17 m sec
Step 5: application of the model to search the best interpretation	0.68 m sec
Total recognition time	16.84 m sec

**Table 4** Effect of the term for the likelihood (%).

measure	<i>crr</i>	<i>Fmeasure</i>
Evaluation function		
without any term (top candidate for every segmentation)	65.13	0.7989
without character recognition likelihood	65.59	0.9616
without context likelihood in terms of bi-gram	69.72	0.9416
without character pattern structure likelihood	76.17	0.9168
without outer gap likelihood	82.50	0.9776
with all the terms	82.55	0.9778



**Fig. 10** Effect of evaluating the character pattern structure likelihood.

On the other hand, the effect of the outer gap likelihood alone is not so notable as others as shown in Table 4. This might have been due to the simplest implementation of this term.

### 5.3 Consideration on Recognition Errors

We investigated the reasons of misrecognitions. Although it is not simple to determine the single term in the total evaluation function for each misrecognition, we can roughly enumerate the reasons.

#### 1. Problem of context likelihood

Figure 11 (a) and (b) show recognition errors mainly due to this reason. Correct character answers are within the top 5 candidates so that the context likelihood should be able to save these cases.

Solution to this problem can be two ways. The bi-gram table for the context likelihood is obtained from the 1993 year's volume of the ASAHI newspaper, but it should be obtained from a large amount of text of the

target domain. Tri-gram should also work better. The other is to employ a word dictionary.

#### 2. Problem of character recognition

Figure 11 (c) and (d) show recognition errors mainly due to character recognition. The context likelihood selects meaningful answers. Character recognition should score incorrectly segmented patterns very low so that they are not within recognition candidates.

#### 3. Problem of outer gap likelihood

In many cases, outer gaps were very small even compared with inner gaps with the result that they were merged into misrecognized character patterns as shown in Fig. 11 (e). In fact, the effect of this likelihood was smallest in the above evaluation. We might have oversimplified this likelihood as the same for all the character pairs.

#### 4. Problem of character pattern structure likelihood

Although it is very hard to pick up recognition errors solely due to this reason, when two characters are incorrectly merged, they are often misrecognized as a character pattern having two radicals. This type of misrecognitions is even supported by the character pattern structure likelihood.

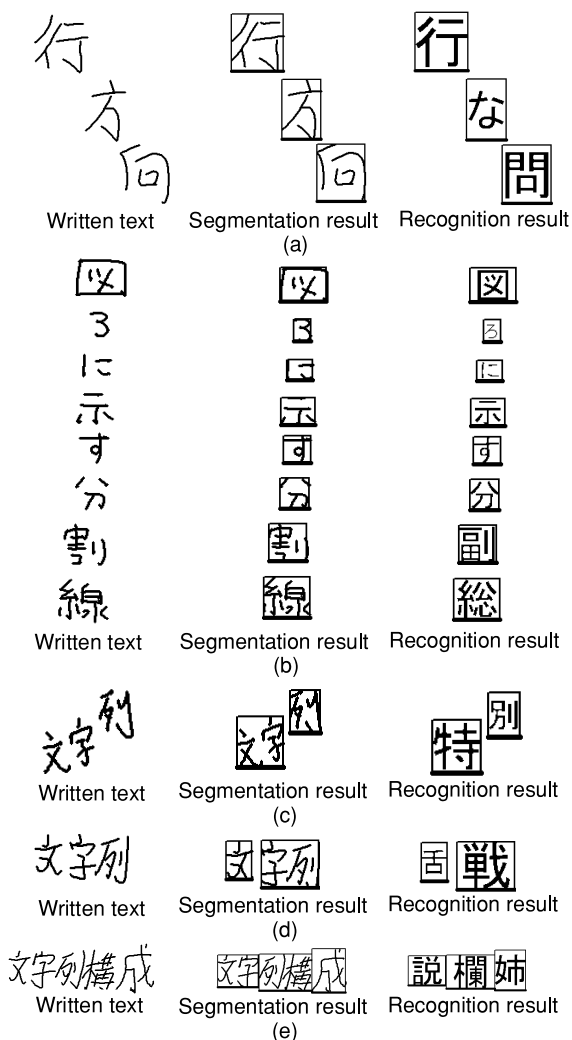


Fig. 11 Recognition errors.

### 6. Conclusion

This paper presented a probabilistic model of unconstrained, line-direction free handwritten Japanese text recognition and its effect. The model can be applied to text of arbitrary character orientation after character orientation is estimated and normalized. We showed the effectiveness of the model through recognition experiments and clarified how each factor in the likelihood contributes to the overall character recognition rate and segmentation rate.

On the other hand, the effect of the outer gap likelihood was small probably due to its simplest implementation. Therefore, we must try more elaborate implementation and make its intrinsic contribution clear in the likelihood.

Character size and inner gap representation by tables is memory consuming but their effects are not high. Alternatively, they should be modeled by distribution functions and trained by learning patterns. It should not require large memory space while improving their robustness.

Another work to be made is to enlarge the database for evaluating the proposed model and avail it for open use as well as train and evaluate the model using a larger amount of sample patterns.

### Acknowledgments

This research was supported by the Strategic Information and Communication R&D Promotion Scheme under the Ministry of Public Management, Home Affairs, Post and Telecommunications.

## References

- [1] H. Murase, T. Wakahara, and M. Umeda, "Online writing-box free character string recognition by candidate character lattice method," IEICE Trans. Inf. & Syst. (Japanese Edition), vol.J68-D, no.4, pp.765-772, April 1985.
- [2] M. Okamoto, H. Yamamoto, T. Yoshikawa, and H. Horii, "On-line character segmentation method by means of physical features," IEICE Technical Report, PRU95-13, May 1995.
- [3] H. Aizawa, T. Wakahara, and K. Odaka, "Real-time handwritten character string segmentation using multiple stroke features," IEICE Trans. Inf. & Syst. (Japanese Edition), vol.J80-D-II, no.5, pp.1178-1185, May 1997.
- [4] T. Fukushima and M. Nakagawa, "On-line writing box free recognition of handwritten text based on stochastic models," IEICE Technical Report, PRMU98-139, Dec. 1998.
- [5] T. Fukushima and M. Nakagawa, "On-line writing-box-free recognition of handwritten Japanese text considering character size variations," Proc. 15th ICDAR, vol.2, pp.359-363, Barcelona, Sept. 2000.
- [6] S. Senda, M. Hamanaka, and K. Yamada, "Box-free online character recognition integrating confidence values of segmentation, recognition and language processing," IEICE Technical Report, PRMU98-138, Dec. 1998.
- [7] S. Senda and K. Yamada, "A maximum-likelihood approach to segmentation-based recognition of unconstrained handwriting text," Proc. 6th ICDAR, pp.184-188, Seattle, Sept. 2001.
- [8] C-L. Liu, M. Koga, and H. Fujisawa, "Lexicon-driven segmentation and recognition of handwritten character strings for Japanese address reading," IEEE Trans. Pattern. Anal. Mach. Intell., vol.24, no.11, pp.1425-1437, Nov. 2002.
- [9] Y. Inamura, T. Fukushima, and M. Nakagawa, "An on-line writing-box-free and writing-direction free recognition system for handwritten Japanese text," IEICE Technical Report, PRMU2000-37, June 2000.
- [10] M. Nakagawa and M. Onuma, "On-line handwritten Japanese text recognition free from constraints on line direction and character orientation," Proc. 7th ICDAR, pp.519-523, Edinburgh, Aug. 2003.
- [11] M. Nakagawa and K. Matsumoto, "Collection of on-line handwritten Japanese character pattern databases and their analysis," Int. J. Document Analysis and Recognition, vol.7, no.1, pp.69-81, 2004.

## Appendix

- $P(s_i|C_i, \bar{C})$  is replaced by  $P(s_i/\bar{C}|C_i)$ , due to the following:

Assuming the size  $s_i$  is proportional to the average size  $\bar{C}$ , the event that  $s_i$  occurs is that the coefficient  $\alpha = s_i/\bar{C}$  occurs. Therefore,

$$P(s_i|C_i, \bar{C}) = P(\alpha|C_i, \bar{C})$$

From the Basian law,

$$P(\alpha|C_i, \bar{C}) = \frac{P(\alpha, \bar{C}|C_i)}{P(\bar{C}|C_i)}$$

As for  $P(\alpha, |C_i)$ ,  $\alpha$  is independent from  $\bar{C}$ , so that,

$$P(\alpha, \bar{C}|C_i) = P(\alpha|C_i) \cdot P(\bar{C}|C_i)$$

Therefore,

$$P(\alpha|C_i, \bar{C}) = \frac{P(\alpha|C_i) \cdot P(\bar{C}|C_i)}{P(\bar{C}|C_i)}$$

$$\begin{aligned} &= P(\alpha|C_i) \\ &= P(s_i/\bar{C}|C_i) \end{aligned}$$

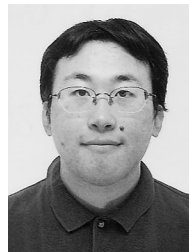
- $P(g_i|C_i, C_{i+1}, \bar{C})$  is replaced by  $P(g_i/\bar{C}|C_i, C_{i+1})$  due to the similar transformation.



**Masaki Nakagawa** was born on 31 October 1954 in Japan. He received B. Sc. and M. Sc. degrees from the University of Tokyo in 1977 and 1979, respectively. During the academic year 1977/78, he followed Computer Science course at Essex University in England, and received M.Sc. with distinction in Computer Studies in July 1979. He received Ph.D. in Information Science from the University of Tokyo in December 1988. From April 1979, he has been working at Tokyo University of Agriculture and Technology. Currently, he is a Professor of Media Interaction in Department of Computer, Information and Communication Sciences.



**Bilan Zhu** received M.Sc. degree from Tokyo University of Agriculture and Technology, Tokyo (TUAT), Japan in 2004. She is currently a Ph.D. degree student in the Department of computer, information and communication sciences at TUAT. She is working on forms processing, on-line, off-line handwriting recognition and context analysis of Japanese, Chinese and English text.



**Motoki Onuma** received M.Sc. degree from Tokyo University of Agriculture and Technology, Tokyo (TUAT), Japan in 2004. He is currently a Ph.D. degree student in the Department of computer, information and communication sciences at TUAT. He is working on on-line handwriting recognition.