IEICE TRANS. INF. & SYST., VOL.E88–D, NO.8 AUGUST 2005

PAPER Special Section on Document Image Understanding and Digital Documents

An On-line Handwritten Japanese Text Recognition System Free from Line Direction and Character Orientation Constraints

Motoki ONUMA[†a)], ***Student Member***, **Akihito KITADAI**[†], ***Member***, **Bilan ZHU**[†], ***Student Member***, ***and*** **Masaki NAKAGAWA**[†], ***Member***

SUMMARY This paper describes an on-line handwritten Japanese text recognition system that is liberated from constraints on line direction and character orientation. The recognition system first separates freely written text into text line elements, second estimates the line direction and character orientation using the time sequence information of pen-tip coordinates, third hypothetically segment it into characters using geometric features and apply character recognition. The final step is to select the most plausible interpretation by evaluating the likelihood composed of character segmentation, character recognition, character pattern structure and context. The method can cope with a mixture of vertical, horizontal and skewed text lines with arbitrary character orientations. It is expected useful for tablet PC's, interactive electronic whiteboards and so on.

key words: *on-line recognition, character recognition, unconstrained handwriting, character recognition system*

1. Introduction

Due to the increasing size of writing surface on a PDA, and the advent of a tablet PC or an electronic whiteboard, people can write text more freely as on a piece of paper. Thus, the demand to remove writing constraints from on-line handwriting recognition is getting higher. A new type of pen interfaces like E-pen [1] and paper interface by Anoto pen and paper [2], [3] are also raising this demand even higher.

Text input interface by a stylus pen started form character writing boxes and they are still in use in most of PDA, but research has been made to remove them. Murase et al. made an initial attempt to remove character writing boxes and applied DP-matching to find the best interpretation of a character pattern sequence [4]. Okamoto et al. proposed geometrical rules and features to improve the segmentation hypothesis [5]. Aizawa et al. presented a real-time segmentation of characters by a neural network using geometric features [6]. Fukushima et al. proposed a probabilistic approach and considered the likelihood based on character segmentation, shape (recognition), context, and character size [7], [8]. 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 [9], [10]. Recently, Microsoft Tablet PC provides on-line text recognition without character writing boxes.

Although many attempts have been made to remove writing constraints, most of them have been assuming only horizontal lines of text. However, people may also write Japanese text vertically or even slantingly. Moreover, they write text with characters rotated when they write text on a whiteboard or write a memo on a piece of paper.

Therefore, 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 [11] and even attempted to avoid character orientation constraint [12].

This paper is the updated version of the last paper and presents a system to recognize mixed on-line handwriting of arbitrary line directions and character orientations.

Section 2 of this paper defines line direction and character orientation. Section 3 presents the detailed flow of processing. Section 4 presents evaluation and shows examples of recognition and errors. Section 5 concludes this paper.

2. Line Direction and Character Orientation

Here, we define some terminologies. A stroke means a series of pen-tip coordinates sampled from pen down to pen up. 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. 1). Although the line direction is the same as common sense, the character orientation might be the opposite from it. We define them in this way since they are consistent with pen-tip movement directions to write Japanese characters.

A text line is a piece of text separated by new-line and

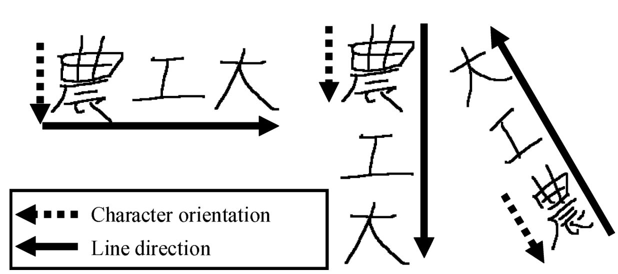


Fig. 1 Line direction and character orientation.

Manuscript received November 24, 2004.
Manuscript revised April 8, 2005.

[†]The authors are with the Graduate School of Engineering, Tokyo University of Agriculture and Technology, Koganei-shi, 184–8588 Japan.

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

DOI: 10.1093/ietisy/e88–d.8.1823

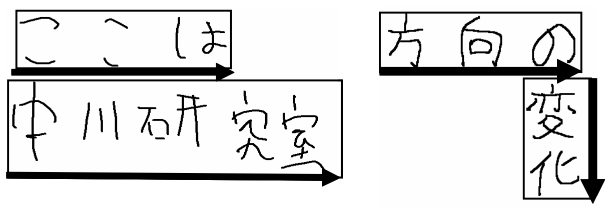


Fig. 2 Text line element and line direction.

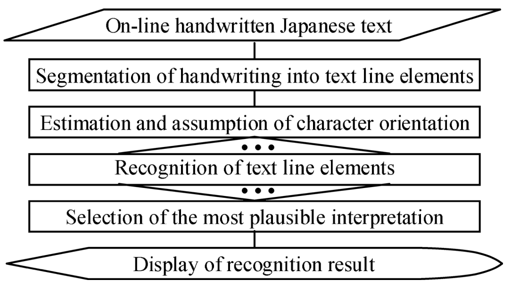


Fig. 3 Flow of processing.

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. 2). The Line direction and the character orientation are independent.

3. Structure of Recognition Process

Recognition of handwritten Japanese text liberated from constraints on line direction and character orientation is composed of the steps shown in Fig. 3. The following sections describe them in more detail.

3.1 Segmentation into Text Line Elements

This step is composed of sub-steps as shown in Fig. 4. Description on each follows.

(1) Estimation of character size

This step estimates the average character size from all the strokes written on a tablet. We assume that most of Japanese characters have the square shape so that the length of one side represents the character size. This size is used to segment handwriting into text line elements, to segment a text line element into characters, to recognize characters and so on.

For each stroke, we take its bounding box and measure its longer side. We sort them, abandon the smaller half and take the average of the remaining larger half. We remove the former since they are short strokes appearing among longer strokes and make the character size estimation too small. This threshold is designed empirically, but it is simple and it produces pretty reliable estimation on the character size. Figure 5 depicts the method.

(2) Detection of new-line and space

Pen movement between consecutive strokes is represented

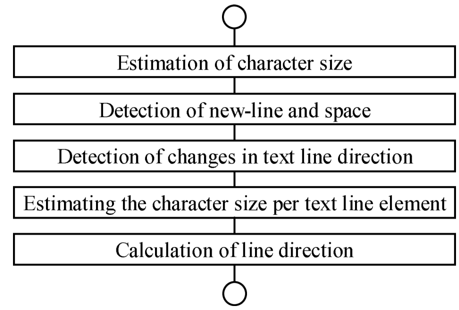


Fig. 4 Flow of segmentation into text line elements.

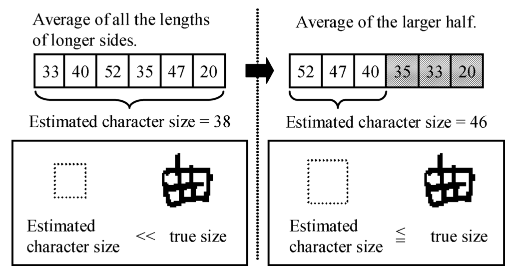


Fig. 5 Estimation of character size.

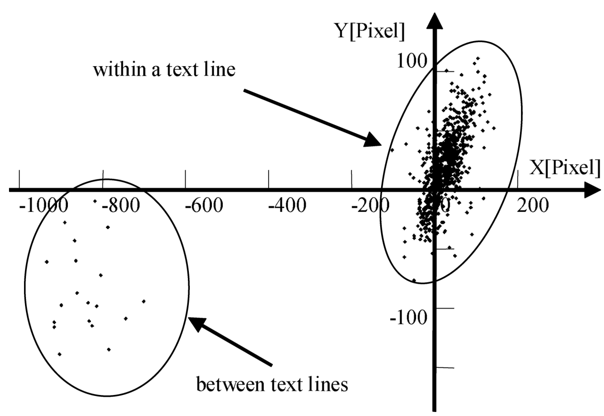


Fig. 6 Clustering of off-strokes.

by a vector from the ending point of the preceding stroke to the starting point of the succeeding stroke and often called an off-stroke or a dark stroke. Off-strokes within a text line are short while those between text lines are considerably long.

Compared with the estimated character size, if there exist some off-strokes that are much longer than the size, we apply clustering to all the off-stokes and divide them into the two groups as shown in Fig. 6.

(3) Detection of the change in line direction

When we write a list of items, we sometimes write a header horizontally and write items vertically as shown in Fig. 7. When we write some text around a figure or a picture, we sometimes write text horizontally and then the remaining vertically. For these cases, we must find changing points of line direction, segment the text there into text line ele-

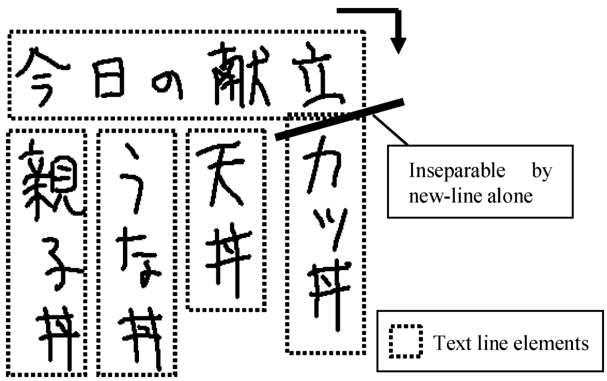


Fig. 7 Mixture of horizontal and vertical writing.

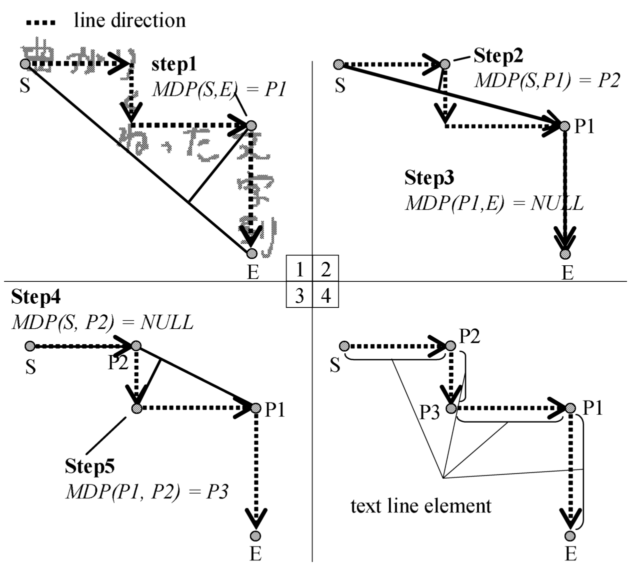


Fig. 8 Detection of directional changing points.

ments and assign the correct line direction to each text line element. Otherwise, text is recognized assuming wrong line direction.

In order to detect changing points of line direction, we employ a recursive procedure similar to that to detect corner points [9]. Among a series of pen-tip coordinates forming a handwritten text line, it finds the most distant point (MDP) from the straight line connecting the starting point and ending point of the series of coordinates, and if the distance is larger than the threshold then apply the same procedure to the straight line from the starting point to the MDP and that from the MDP to the ending point with the result of detecting multiple points of directional change as shown in Fig. 8. Thus, a text line segmented by new-line or large space is further segmented into text line elements having different line directions.

Here, it is worth noting that points thus detected may not be the best segmentation points, that is, they might be within character patterns rather than between characters. The subsequent steps treat them indecisively and the recognition step described in Sect. 3.3 determines the best segmentation points while recognizing handwritten text.

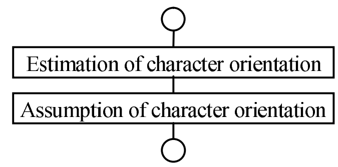


Fig. 9 Flow of processing (3.2).

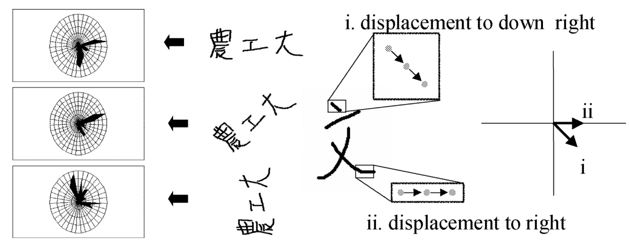


Fig. 10 Two peaks in pen movement direction.

(4) Estimating the character size per text line element

Since the character size may vary among text line elements, we estimate the character size again for every element. Although the method is the same as described in 3.1 (1), it is applied to each element, so that the character size is more accurately estimated for each.

(5) Calculation of line direction

The direction from the starting point to the ending point of a text line element is assigned as the line direction of the text line element.

3.2 Estimation and Assumption of Character Orientation

This is made by the two steps as shown in Fig. 9. It produces multiple hypotheses and the succeeding recognition stages determine the best estimation.

(1) Estimation of character orientation

When Japanese characters are written, principal pen movement within real strokes is the same as the character orientation or $\pi/2$ counter clockwise to it. This is because Japanese characters, especially Kanji characters, are composed of downward and rightward strokes. Because of this, if we take the histogram of displacement direction of pen-tip coordinates, we will see two peaks as shown in Fig. 10. These peaks are not so stable if characters are few in a text line element, but they become more stable as the number of characters increases.

Therefore, we can estimate the character orientation from the histogram of displacement direction for a text line element. Once, the character orientation is estimated, the text line element can be recognized by rotating characters until their orientation become downward.

Let us assume the intensity of the histogram at the angle θ as $f(\theta)$. Then, compute $f(\theta) * f(\theta+\pi/2)$. This is to find

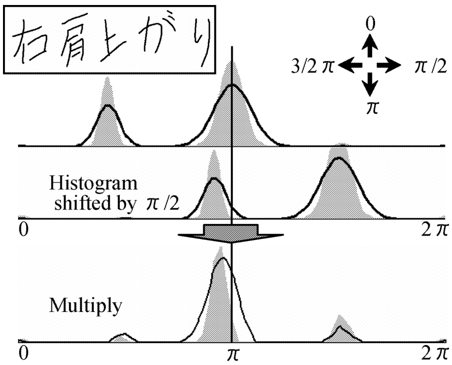


Fig. 11 Estimation method of character orientation.

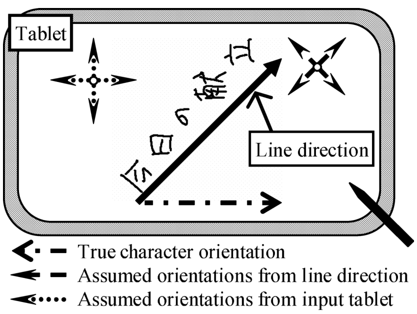


Fig. 12 Direction assumption of character orientation.

the overlap between $f(\theta)$ and $f(\theta+\pi/2)$. If we can find a single and strong peak, this implies that the peak at θ and that of $\theta + \pi/2$ are notable and θ is the character orientation. In order to make the peak detection more robust, we take convolution of $f(\theta)$ and the Gauss function $g(\delta) = \exp(-\delta^2/\sigma^2)$ to blur the peak as shown in Fig. 11 so that it works for slanted characters that have rightward strokes with slightly upward inclination.

The system estimates the character orientation as 0, 30, 60 and so on, namely being quantized by 30 degrees, since is a tolerance of our character recognizer.

(2) Assumption of character orientation

This step assumes character orientation commonly appearing in relation to the line direction of a text line element. Assumed orientations are the same and opposite of the line direction, two perpendicular orientations to it and the four orientations (upward, downward, rightward and leftward) to the orientation of the input tablet as shown in Fig. 12. We can expand the assumption more than these orientations or restrict them when the character orientation has the strong sign or it is confined from applications.

3.3 Recognition of Text Line Elements

Assuming each of the character orientations derived above, recognition of a text line element is made for each orientation according to the flow shown in Fig. 13.

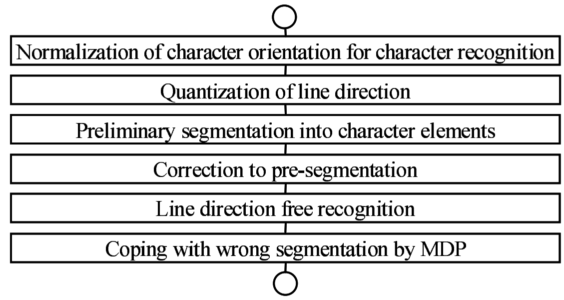


Fig. 13 Flow of recognition for text line elements.

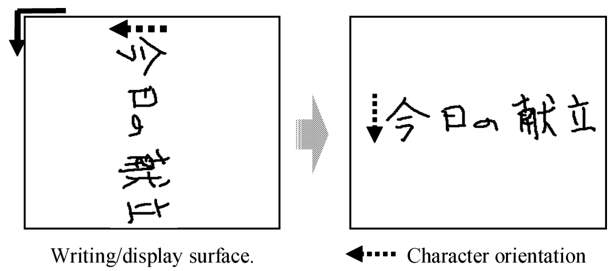


Fig. 14 Normalization of character orientation.

(1) Normalization of character orientation

A text line element is rotated so that the estimated or assumed character orientation is turned downward as shown in Fig. 14. Since it is a sequence of pen-tip coordinates, its rotation to an arbitrary angle is easy and quick. We call the rotated pattern as a character orientation normalized (CO-normalized in short) text line element. For the display of formatted recognition result, the center and amount of rotation is kept for each CO-normalized text line element.

(2) Quantization of line direction

We apply the same amount of rotation as the character orientation normalization to the line direction calculated in 3.1 and call the revised line direction as CO-normalized line direction. This direction is quantized into 4 directions (downward, upward, rightward or leftward). The Quantization can be finer but the 4 directional quantization is adequate and effective to prevent the CO-normalized text line element from being segmented excessively. For example, if you are writing horizontally to the right direction, then a leftward stroke or off-stroke is used to merge its preceding strokes crossed by the (off-) stroke. In general, if we know the line direction, pen movement of the opposite direction is used to merge its preceding strokes with the result that hypotheses on segmentation are decreased, which is then effective to speed up the text recognition and to increase the recognition rate.

(3) Preliminary segmentation into character elements

Each text line element is hypothetically segmented into character elements using its estimated character size, the projected distance between the gravity centers of strokes to x-axis when the revised line direction is horizontal or y-axis

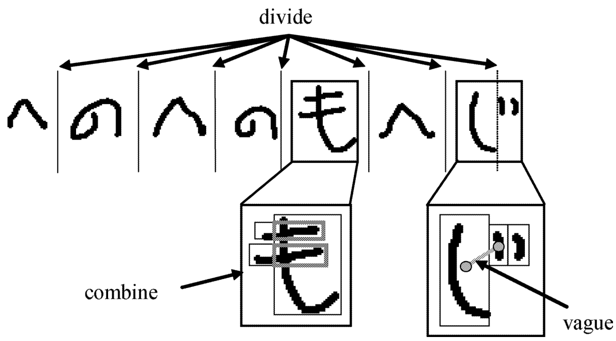


Fig. 15 Segmentation into characters.

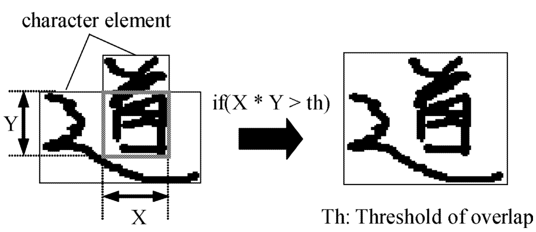


Fig. 16 Correction to preliminary segmentation.

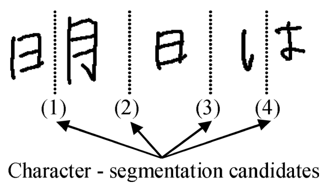


Fig. 17 An example of hypothetical character segmentation.

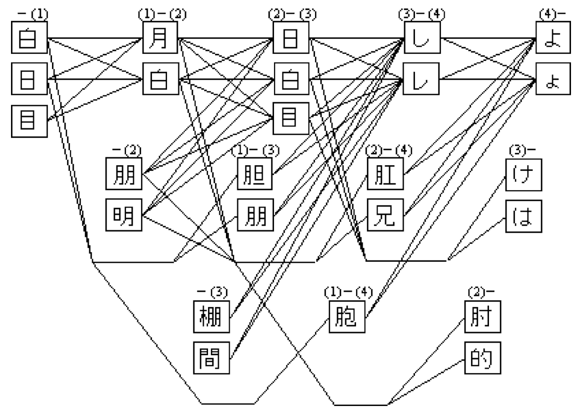
when vertical, and the overlap between the bounding box of a stroke with that of another stroke. Clear segmentation points are marked “divide”, Points that should not be segmented are “combine” and points undetermined are marked “vague” as shown in Fig. 15.

(4) Correction to preliminary segmentation

By classifying points judged “vague” into either of “divide” or “combine” as much as possible, the search space of the succeeding text recognition is reduced. Here, we consider the overlap of the bounding box of a character element and that of another element rather than a stroke. The process is depicted in Fig. 16.

(5) Line direction free recognition

Every CO-normalized text line element has the downward character orientation, so that it is recognized by the line-direction-free recognizer already developed [11]. The recognizer applies single character recognition for each hypothetically segmented pattern and outputs candidates for each pattern as well as links between those for a preceding segmented pattern and those for a succeeding segmented pattern, which is called a candidate lattice. Figure 17 shows a hypothetical segmentation and Fig. 18 shows its candidate lattice.



Each box shows a candidate of character recognition and (n)-(m) on top of them shows segmentation points.

Fig. 18 A candidate lattice.

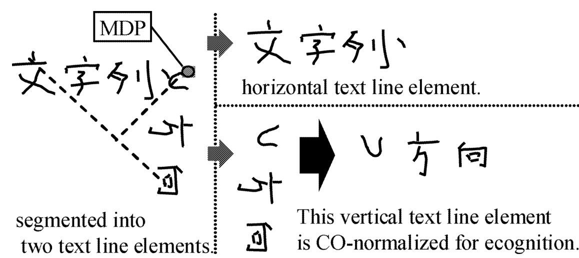


Fig. 19 Segmentation by MDP.

(6) Coping with wrong segmentation by MDP

Segmentation of a text line into text line elements by MDP should not be decisive. Wrong segmentation within a character pattern into two text line elements and rotation of the segmented text line elements so as to normalize character orientation may damage their recognition as shown in Fig. 19.

In order to avoid this problem, we produce multiple alternatives of segmentation of a text line by changing the segmentation point around MDP. The range that the segmentation point is perturbed can be confined within the average character size before or after MDP. These multiple alternatives are CO-normalized and recognized by the above line direction free recognizer.

3.4 Selection of the Most Plausible Interpretation

All the candidate lattices for each CO-normalized text line element are merged into a single lattice for a unit of a text line. Then, the Viterbi search is made into the lattice so as to maximize the likelihood $L(C|X)$ that a handwritten text line pattern $\mathbf{X}$ is recognized as the character string $\mathbf{C} = C_1C_2 \ldots C_i \ldots C_m$, which is defined as follows [13]:

$$L(C|X) = \sum_{i=1}^{m} \log P(C_i|C_{i-1})$$
$$+ \sum_{i=1}^{m} \Big(\log P(X_i|s_i, C_i)$$

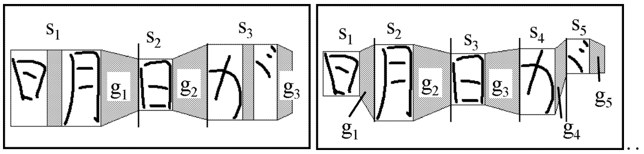


Fig. 20 Character pattern structures and outer gaps.

$$+ \log P(s_i/\overline{C}|C_i) + \log P(g_i/\overline{C}|C_i, C_{i+1})\Big)$$

- m: the number of characters in C.
- $P(C_i|C_{i-1})$: the probability that a character C_i follows C_{i-1} (bi-gram probability).
- C_0: the state before the first character occurs.
- $P(C_1|C_{i-0})$: the probability that a character C_1 occurs at the top of text.
- s_i: character pattern structure
- g_i: outer gap
- $P(X_i|s_i, C_i)$: the probability that a character C_i is written in a structure si and represented by the stroke sequence $X_i = x_{i1}x_{i2}\dots x_{ik}$.
- $\overline{C}$: the average size of the character sequence C.
- $P(s_i/\overline{C}|C_i)$: the probability that a character C_i is written in a structure s_i.
- $P(g_i/\overline{C}|C_i, C_{i+1})$: the probability that an outer gap g_i appears between C_i and C_{i+1}.

Here, the character pattern structure s_i is a bounding box of the character pattern X_i with including some inner gaps, while the outer gap g_i denotes the gap between two character patterns X_i and X_{i+1}. We need to consider multiple interpretations to a single text line pattern as shown in Fig. 20.

In the right-hand side of the above equation, 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.

This step outputs the path to produce the highest likelihood that shows the best hypothesis among multiple combinations of hypotheses on segmentation into text line element by MDPs, character orientation/assumption, segmentation into characters and character recognition.

We consider the likelihood in the unit of text line rather than text line element, because we can get the benefit of bigger context and easier structure to accommodate the above mentioned hypotheses.

3.5 Display of Recognition Result

This step displays the recognition result close to the original handwriting. It places fonts according to the original line direction, character orientation, average character size, and average character interval for every text line element.

4. Result

An example of recognition is shown in Fig. 21. The method works for a mixture of text lines that have various line directions and character orientations.

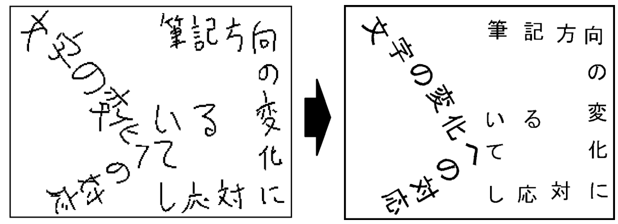


Fig. 21 An example of recognition.

Table 1 Database HANDS-Kondate_t_bf-2001-11 (100 people).

Page	Text	Format
1-11	Fixed	line direction: horizontal character orientation: downward
12-22	Fixed	line direction: vertical character orientation: downward
23-25	Free	line direction: mixture of vertical, horizontal and skewed character orientation: unspecified
26	Fixed	line direction: mixture of vertical and horizontal character orientation: downward
27	Fixed	line direction: mixture of horizontal and skewed character orientation: slanting
28	Fixed	line direction: mixture of vertical and horizontal character orientation: down, up, left and right

Table 2 Performance on rotated horizontal/vertical handwritings.

Rotation	Estimation on character orientation (%)	Estimation online direction (%)	Segmentation measure	Character recog. rate (%)
original (0)	99.03	99.71	83.86	71.52
30 degree	98.49	97.85	82.12	70.25
90 degree	99.30	99.14	83.75	71.57
130 degree	96.40	98.83	82.32	69.86
240 degree	99.08	99.08	82.29	70.16

We preliminarily evaluated the method using the database HANDS-Kondate_t_bf-2001-11 (Kondate_t in short) collected from 100 people as shown in Table 1. Some samples are shown in Fig. 22. Pages 1 through 11 are just horizontal handwritings and Pages 12 through 22 are just vertical lines so that we added different line directions and character orientations by rotating the original 22 pages × 100 people's handwritings by the amount of 30, 90, 130 and 240 degrees as shown in Table 2.

From Table 2, we can see that the method is estimating the character orientation and line direction almost correctly. Table 2 also presents segmentation measure and character recognition rate where we employ the following F measure for the segmentation measure.

$$F = \frac{2}{1/R + 1/P} \times 100$$

$$R = \frac{number\ of\ correctly\ detected\ segmentation\ positions}{number\ of\ truesegmentation\ positions}$$

$$P = \frac{number\ of\ correctly\ detected\ segmentation\ positions}{number\ of\ detected\ segmentation\ positions\ (including\ false)}$$

The segmentation measure and character recognition rate are unsatisfactory, so that we need to review the method for segmentation and recognition.

Fig. 22 Examples of Kondate_t database.

Table 3 Performance on mixture of vertical, horizontal and skewed lines.

Page Kondate_t	Segmentation	Character Recognition(%)
23	75.23	63.87
24	73.68	62.66
25	71.18	60.51
26	86.98	72.81
27	72.45	62.10
28	75.83	64.61
Average	75.56	64.18

The Kondate_t database includes mixture of horizontal, vertical and skewed text lines with various character orientations in 23 through 28 pages. Especially, 24 and 25 pages are freely handwritten patterns under certain topics. Since we are aiming to remove the writing constraints on line direction and character orientation as much as possible, we include extreme patterns as shown in Fig. 22.

Table 3 shows performance on them. Average recognition time is about 1.58 sec. per page on a Pentium IV 2 GB CPU.

Although the architecture of the system is working for a mixture of vertical, horizontal and skewed lines with arbitrary character orientations, the performance is premature. We need to improve segmentation and recognition by analyzing errors, reviewing its empirical design and enhancing each component as well as the total system.

5. Conclusion

This paper has presented an on-line handwritten Japanese text recognition system that is liberated not only from writing boxes or rules lines but also from constraints on line direction and character orientation. This system first separates freely written text into text line elements, second estimates the line direction and character orientation, third hypothetically segment it into characters, fourth apply character recognition and finally select the most plausible interpretation by evaluating the likelihood.

As far as a preliminary test is made, the method is working for a mixture of vertical, horizontal and skewed lines with arbitrary character orientations.

Although we have fixed the architecture of the system, we need to train each component and the total system.

References

[1] http://www.e-pen.com
[2] http://www.anoto.com
[3] http://www.otmtech.com
[4] 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.
[5] 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.
[6] 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.
[7] 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.
[8] T. Fukushima and M. Nakagawa, "On-line writing-box-free recognition of handwritten Japanese text considering character size variations," Proc. 15th ICPR, vol.2, pp.359–363, Barcelona, Sept. 2000.
[9] 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.
[10] 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.
[11] 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.
[12] M. Nakagawa and M. Onuma, "On-line handwritten Japanese text recognition free from constrains on line direction and character orientation," Proc. 7th ICDAR, pp.519–523, Edinburgh, Aug. 2003.
[13] M. Nakagawa, B. Zhu, and M. Onuma, "A formalization of on-line handwritten Japanese text recognition free from line direction constraint," Proc. 17th ICPR, 2P. Tu-i, Cambridge, England, Aug. 2004.

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

Akihito Kitadai received B.E. and M.E. degree from Tokyo University of Agriculture and Technology, Tokyo (TUAT), Japan in 1999/2001. In 2004, he received his Ph.D. degree in Engineering from TUAT. He is now working at TUAT. His research interests include pattern recognition, learning algorithm and analysis of historical documents.

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 of forms processing, on-line, off-line handwriting recognition and context analysis of Japanese, Chinese and English text.

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.