Online Character Segmentation Method for Unconstrained Handwriting Strings Using Off-stroke Features

Naohiro Furukawa, Junko Tokuno, Hisashi Ikeda

Online Character Segmentation Method for Unconstrained Handwriting Strings Using Off-stroke Features

Naohiro Furukawa† *Junko Tokuno*‡ *Hisashi Ikeda*†

†Central Research Laboratory, Hitachi, Ltd.
1-280 Higashi-Koigakubo, Kokubunji-shi,
Tokyo, 185-8601 Japan
E-mail: {naohiro.furukawa.qv,
hisashi.ikeda.vz}@hitachi.com

‡Center for Innovation and Intellectual Property,
Tokyo University of Agriculture and Technology
2-24-16 Naka-cho, Koganei-shi,
Tokyo, 184-8559 Japan
E-mail: j-tokuno@hands.ei.tuat.ac.jp

Abstract

In this paper, an online character segmentation method for unconstrained strings is proposed. To recognize unconstrained-expression strings such as those in phrases and informal notations, we designed physical features of segmented patterns, in particular, off-stroke features. Segmented-pattern likelihood was also defined from these features using a probabilistic model. Evaluations using a digital pen system showed that the character segmentation rates were 97.8%, 91.7%, and 75.6% of numerals, Japanese characters, and all characters (numerals, alphabets, symbols, and Japanese characters), respectively.

Keywords: character segmentation, online recognition, box-free recognition, off-stroke feature, digital pen.

1. Introduction

Paper is one of the most familiar media to people and has many advantages, such as being easy to read and write on. It is thus still widely used, even in today's information society. For example, when an end user wants to use services at government offices, banks, or other such offices, he/she hands an application form to an employee in the appropriate section, and that employee then initiates the application processes according to the information on the forms. We define the sequences of all these processes between filling in forms and starting the task as "data entry." Many data entry systems using pen and paper have been applied to business tasks, such as notifications to government offices, remittances at banks, insurance applications, patient-record updates in hospitals, maintenance records for utilities services, and inventory management in warehouses.

Online data entry systems such as a digital pen system[1][2] have been introduced, and with them, temporal information can be used. To improve user convenience, the rate of forms that are free of boxes that have to be filled in ("writing-box-free") is increasing more and more. However, a system capable of recognizing not only constrained strings such as addresses and names, but also unconstrained strings such as phrases and informal notations is needed.

Most writing-box-free recognition systems need a character segmentation process that can determine correct segmented patterns from an input string. In general, the character segmentation process is as follows.

(1) Pre-segmentation: each character candidate pattern (segmented pattern) from an input string is segmented; this hypothesis is output as a segmentation graph.
(2) Feature extraction: feature vectors are extracted from each segmented pattern.
(3) Calculating segmented-pattern likelihood: segmented-pattern likelihood is calculated from segmented-pattern features.
(4) Searching a segmentation path: a path is found on the segmentation graph in which the sum of segmented-pattern likelihood has the best value.

In the pre-segmentation step, the segmentation position may not be uniquely decided using physical information. Thus, multiple segmented-pattern candidates are created in the pre-segmentation step and obtained as a directed acyclic graph (DAG) which is called a "segmentation graph." Figure 1 shows an example of such a segmentation graph for a text line. Every segmented pattern in the segmentation graph is tagged with a segmented-pattern likelihood, which indicates whether that pattern seems to have been correctly segmented.

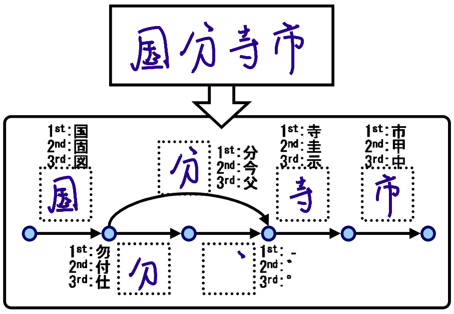

Figure 1. Example of segmentation graph.

Here, there are mainly three types of information for using character segmentation, as follows.

(1) Physical information: geometric and temporal features such as width/height of segmented patterns and writing time of segmented pattern.
(2) Statistical information: co-occurrence frequency of concatenated characters, n-gram probability, and so on.
(3) Linguistic information: using a language model or a morphological analysis, etc.

For an example of using physical information, one method [3] uses eight shape features. This method can evaluate the character pattern candidates by linear transformation of their feature vectors.

An example of using statistical information is a method [4] that uses the transition probability of two characters (bi-gram probability). This method also uses physical information.

An example of using linguistic information is a method [5] using a TRIE-structure to obtain the expression knowledge. Another example is using a recursive transition network (RTN) [6]. These methods can permit character-classification failure and character-segmentation failure.

In general, statistical and linguistic information are useful for constrained string segmentation. However, these types of information cannot be applied or are incorrect with unconstrained string segmentation. Thus, for unconstrained string segmentation, it is important to use physical information. Even if for constrained string segmentation, physical information is used as a basic factor for statistical and linguistic approaches.

Some previously proposed methods [3] and [4] use physical information such as the width/height of segmented patterns, the aspect rate of segmented patterns, gaps between segmented patterns, etc. The gap between segmented patterns is an especially useful feature. If all the pitches between characters in a string are wide enough, such strings can be segmented using only the gaps. However, when pitches are narrow, or likewise, some characters are touching, a string cannot be segmented correctly using only traditional features.

Therefore, we focused on the features of the off-stroke. Even if some characters are touching, the distance and the time period of the off-stroke between the last stroke of the previous character and the first stroke of the next character should both be bigger than those of the off-strokes that occur within characters.

Consequently, we propose new off-stroke features. Segmented-pattern likelihood combined with these new features and traditional features by using a probabilistic model is also proposed. This paper reports our character segmentation method and its evaluation tests.

2. Proposed Method

2.1. Pre-segmentation

Strings are usually filled in from left to right. In this case, if each segmented pattern is segmented according to the stroke order, the segmented pattern can be correctly determined. With touching characters, it is only necessary to segment according to stroke order. However, a user sometimes inserts additional characters between previously written characters. In this case, using the stroke order only would not work because any added characters would be regarded as the characters at the end of the string. Thus, it is important to utilize not only temporal information, but also geometrical information.

The digital pen can obtain three variables (x_i, y_i, t_i) of sampling point p_i : (1) x-axis; (2) y-axis; and (3) writing time. A criterion for merging strokes is calculated from these variables, and the segmented pattern is created according to the criterion. We focused on linearly located characters, and a linear sum function of the sampling point was defined as:

$$f(x_i, y_i, t_i) = w_0 + w_1 x_i + w_2 y_i + w_3 t_i . \qquad (1)$$

This function converts three variables to the criterion.

Here, variables of a whole stroke (x, y, t) were defined as variables of a sampling point that had the smallest f of all sampling points. Thus, the linear sum function of a stroke was defined as follows.

$$f(x, y, t) = \min(f(x_i, y_i, t_i)) . \qquad (2)$$

2.2. Segmented-pattern feature extraction

We designed 25 features for segmented patterns. These include the geometrical and temporal features such as the width/height of the segmented pattern, the aspect rate of the segmented pattern, and the writing time of the segmented pattern (Table 1).

In this section, we explain in detail our proposed features, the off-stroke features within segmented patterns and between segmented patterns.

Table 1. List of segmented-pattern features.

#	Segmented-pattern features	No.
A	Shape features of segmented pattern	6
B	Position features of segmented pattern	4
C	Gap features within segmented pattern and between segmented patterns	3
D	Length of strokes	1
R	Character classification results	1
F	Off-stroke features within segmented pattern	2
G	Off-stroke features between segmented patterns	8

2.2.1. Off-stroke features within segmented pattern

The digital pen we used can capture the time period used to write strokes and the writing order of strokes. Therefore, off-stroke information can be obtained from the last sampling point of the previous stroke and the first sampling point of the next stroke.

Both the distance and time of off-strokes within a correct segmented pattern have a relatively small value. Using this property, we defined two segmented-pattern features (Table 2). Here, W_e is the estimation of the width of a correct segmented pattern and is defined as $W_e = 0.75 \cdot \{$the maximum width of segmented pattern in a string$\}$. N is the number of strokes in a segmented pattern. The other parameters are described in Figure 2.

2.2.2. Off-stroke features between segmented-patterns

Both the distance and time of off-strokes between segmented patterns have a relatively large value if the patterns are correct. Here, when a focused segmented pattern is correct, two off-strokes, from the preceding stroke to the current stroke, and from the current stroke to the succeeding stroke, are both correct. In most correct off-strokes between segmented patterns, their start points may be located in the bottom/right position, and their end points may be located in the top/left position because the start points of characters are mainly in the top/left side of the pattern, and the end points in the bottom/right position. Thus, we think that the angle of the off-stroke between segmented patterns is one of the most important factors. Using this property, we defined eight segmented-pattern features (Table 3).

Table 2. List of off-stroke features within segmented-pattern.

#	Definitions
F1	Average distance of off-strokes within segmented pattern: $\dfrac{\frac{1}{N}\sum_{i=1}^{N}\sqrt{(EndX_{i-1}-StartX_i)^2+(EndY_{i-1}-StartY_i)^2}}{W_e}$
F2	Average time of off-strokes within segmented pattern: $\frac{1}{N}\sum_{i=1}^{N}(StartTime_i - EndTime_{i-1})$

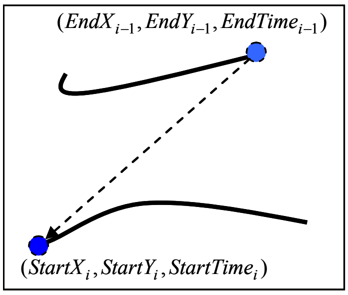
$(EndX_{i-1}, EndY_{i-1}, EndTime_{i-1})$
$(StartX_i, StartY_i, StartTime_i)$

Figure 2. Parameters for calculation of pattern features.

Table 3. List of off-stroke features between segmented patterns.

#	Definitions
G1	Distance of off-stroke from preceding stroke to current stroke: $\dfrac{\sqrt{(EndX_{pre}-StartX\ )^2+(EndY_{pre}-StartY\ )^2}}{W_e}$
G2	Time of off-stroke from preceding stroke to current stroke: $StartTime - EndTime_{pre}$
G3	Sine of the angle of off-stroke from preceding stroke to current stroke: $\dfrac{StartY - EndY_{pre}}{\sqrt{(StartX-EndX_{pre})^2+(StartY-EndY_{pre})^2}}$
G4	Cosine of the angle of off-stroke from preceding stroke to current stroke: $\dfrac{StartX - EndX_{pre}}{\sqrt{(StartX-EndX_{pre})^2+(StartY-EndY_{pre})^2}}$
G5	Distance of the off-stroke from current stroke to succeeding stroke: $\dfrac{\sqrt{(StartX_{suc}-EndX\ )^2+(StartY_{suc}-EndY\ )^2}}{W_e}$
G6	Time of the off-stroke from current stroke to succeeding stroke: $StartTime_{suc} - EndTime$
G7	Sine of the angle of off-stroke from current stroke to succeeding stroke: $\dfrac{StartY_{suc} - EndY}{\sqrt{(StartX_{suc}-EndX)^2+(StartY_{suc}-EndY)^2}}$
G8	Cosine of the angle of off-stroke from current stroke to succeeding stroke: $\dfrac{StartX_{suc} - EndX}{\sqrt{(StartX_{suc}-EndX)^2+(StartY_{suc}-EndY)^2}}$

2.3. Segmented-pattern likelihood calculation

In this paper, we suppose that each feature distribution fits a normal distribution. Segmented-pattern likelihood can be calculated from means and variances of features using a probabilistic model.

Let μ be a mean of a feature and σ^2 be a variance. Then the probability of x is $P(x \mid \mu, \sigma^2)$. Let x_i be the i-th feature of a segmented-pattern. Here, the i-th feature's segmented-pattern likelihood L_i is as follows,

$$L_i = \log P(x_i \mid \mu_i, {\sigma_i}^2)$$

$$= -\frac{1}{2}\log(2\pi {\sigma_i}^2) - \frac{(x_i - \mu_i)^2}{2{\sigma_i}^2}. \qquad (3)$$

We defined segmented-pattern likelihood as the sum of the likelihood of all features:

$$L = \sum L_i. \qquad (4)$$

2.4. Searching for a segmentation path

This step finds a path from the starting node to the ending node on the segmentation graph in which the sum of segmented-pattern likelihood is the best value. We applied dynamic programming (DP) to this step, as reported elsewhere[7][8][9]. Here, the i-th node's score is defined as:

$$S_i = \max_{\{k \mid L_{k,i}\}} \{S_k + L_{k,i}\}. \qquad (5)$$

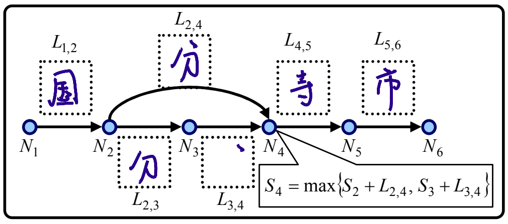

Figure 3. Example of node score.

3. Experimental results

3.1. Evaluation of the pre-segmentation

First, we collected 2,662 samples using digital pens for the evaluation of our pre-segmentation method. These samples consisted of several kinds of characters including numerals, letters, symbols, and Japanese characters. There were 98 participants.

Using our pre-segmentation method, we obtained the segmentation results and checked visually to determine if there was a correct segmented pattern in the results. Table 4 shows this evaluation result.

The accuracy of the pre-segmentation was 100%. Samples with touching characters were segmented correctly.

However, more than 95 samples (3.4%) had mistakes, for example, a wrong character with a double line as a correction mark. In that case, our pre-segmentation method was not able to segment characters correctly because of extra strokes. These samples were excluded from the evaluation in this paper. From a practical point of view, a correction mark recognition function is necessary in the future.

Table 4. Evaluation result of pre-segmentation.

Correct	Error
100% (2,662)	0% (0)

3.2. Evaluation of the character segmentation

The samples for character segmentation were collected using digital pens from 387 people. We divided the collected samples into two data equally, for learning and evaluation. Table **5** gives details of the sample sets for evaluation. Using the learning sample sets, we pursued each mean and variance of segmented pattern feature.

The evaluation results are listed in Table **6**. For string-based evaluation, if even one segmented pattern in the path was not a correct pattern, the string was counted as a failure.

Evaluation results showed that the character-based character segmentation rates were 97.8%, 91.7%, and 75.6% of numerals, Japanese characters, and all characters, respectively.

Figure **4** shows examples of correct segmentation samples. Samples such as (a) and (b) have little character spacing; other samples such as (c), (d), and (e) have touching characters. Samples (f), (g), and (h) have different font sizes and/or big gaps within characters. Our proposed method was able to segment these samples correctly. For example, sample (e) has two pairs of touching characters, '道' and '石', and '新' and '篠'. In this case, our proposed method segmented them correctly. Moreover, the character '川' which has wide internal gaps was also segmented correctly.

However, some samples were segmented incorrectly; these are shown in Figure **5**. There were mainly five types of failure: (a) an over-segmented pattern; (b) over-merged pattern; (c) character classification failure; (d) small character mismerge; and (e) a mixture of character sizes.

For example, in sample (a1), '5' is over-segmented because of the wide internal gap in the '5.' This is especially likely to occur when an internal gap in the character widens, because internal gaps in numerals rarely exist.

In addition, the failures in the over-segmented and over-merged patterns, such as in (a2), (b1), and (b2), sometimes occur in strings consisting of different character sizes, like a mixture of Kanji characters and numerals. This is because the average pattern size in the string is different from the size of the Kanji characters and the numerals. For instance, (b2) contains many Kanji characters; thus the average pattern size is bigger than the Hiragana characters (e.g., enclosed in boxes). In this case, the occurrence of over-merged patterns is slightly higher.

In the case of (c), when a correct pattern is rejected by character classification, character segmentation tends to result in a mistake due to the use of the character classification results as the pattern feature.

The case of (d) is a special type of over-merged failures. Small characters such as commas and hyphens are sometimes mismerged with neighboring characters.

The case of (e) is a failure according to mixture of character sizes. In general, the width of Kanji characters is several times as large as the width of numerals and letters. Thus occurrence of numerals/letters over-merged or Kanji characters over-segmented is higher, in this case. To improve such failures, we should use for the feature normalization not only global information but also local information such a mean within a short range.

Table **7** lists the comparative experiments of the character segmentation methods using physical information. The top line in shows the result of the method using physical features described by Senda, et al. [3]. The result of the second line is from the method using physical features described by Fukushima and Nakagawa [4].

From these results, we confirmed that our designed off-stroke features are effective information for character segmentation.

Table **5**. Sample sets for character segmentation evaluation.

Set	No. of char.	No. of str.	Char. /str.	String properties
Set_N	20,588	2,077	9.9	Numerals, hyphens and commas only.
Set_K	1,789	189	9.5	Japanese characters only.
Set_M	50,371	3,580	14.0	All characters: numerals, alphabets, symbols, and Japanese characters.

Table **6**. Evaluation results of character segmentation.

Set	Character based	String based
Set_N	97.8% (20,143)	88.0% (1,827)
Set_K	92.3% (1,652)	71.4% (135)
Set_M	75.6% (38,065)	20.6% (736)
All sets	82.3% (59,860)	46.2% (2,698)

Table **7**. Comparative experiments using all sets.

Method	Character-based	String-based
Using features from [3]	69.1%	23.0%
Using features from [4]	53.4%	15.4%
Proposed method	82.3%	46.2%

4. Conclusion

We developed an online character segmentation method for box-free strings. To recognize strings of unconstrained expressions such as phrases and informal notations, we designed physical features of segmented patterns, notably off-stroke features. Segmented-pattern likelihood was also defined from these features using a probabilistic model. Evaluations using a digital pen system showed that the respective character segmentation rates were 97.8%, 91.7%, and 75.6% for numerals, Japanese characters, and all characters (numerals, alphabets, symbols, and Japanese characters). Moreover, from the comparative experiments, we confirmed that our designed off-stroke features are effective information for character segmentation.

References

[1] http://www.hitachi.co.jp/tegaki

[2] N. Furukawa, H. Ikeda, Y., Kato, and H. Sako, "D-Pen: a digital pen system for public and business enterprises," *Proc. of 9th IWFHR*, pp. 269-274, 2004.

[3] S. Senda, M. Hamanaka, and K. Yamada, "An Online Handwritten Character Segmentation Method of which Parameters can be Decided by Learning," *Technical Report of IEICE*, PRMU97-219, pp.17-24, 1998 (in Japanese).

[4] T. Fukushima, and M. Nakagawa, "On-line Writing-box-free Recognition of Handwritten Japanese Text Considering Character Size Variations," *Proc. of ICPR'00*, 2000.

[5] M. Koga, R. Mine, H. Sako, and H. Fujisawa, "Lexical Search Approach for Character-String Recognition," *Proc. of DAS'98*, Nov. 19, pp. 237-251, 1998.

[6] H. Ikeda, N. Furukawa, M. Koga, H. Sako, and H. Fujisawa, "Context-Free Grammar-Based Language Model for String Recognition," *International Journal of Computer Processing of Oriental Languages*, Vol. 15, No. 2, pp. 149-163, 2002.

[7] H. Bunke, "A Fast Algorithm for Finding the Nearest Neighbor of a Word in a Dictionary," Report of Institut fur Informatik und Angewandte Mathematik, Universitat Bern, 1993.

[8] F. Kimura, M. Shridhar, and Z. Chen, "Improvements of a lexicon directed algorithm for recognition of unconstrained handwritten words," *Proc. of 2nd ICDAR*, pp. 18-22, 1993.

[9] N. Furukawa, A. Imaizumi, M. Fujio, and H. Sako, "Document Form Identification Using Constellation Matching," *Technical Report of IEICE*, PRMU 2001-125, Vol. 101, No. 421, pp. 85-92, 2001 (in Japanese).

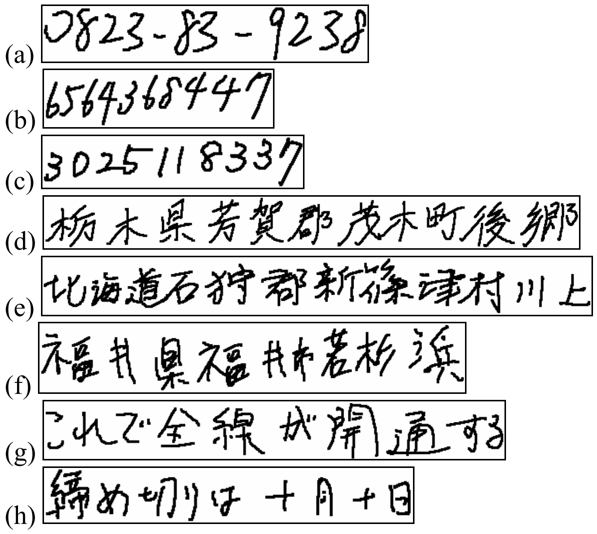

Figure **4**. Examples of correct segmentation samples.

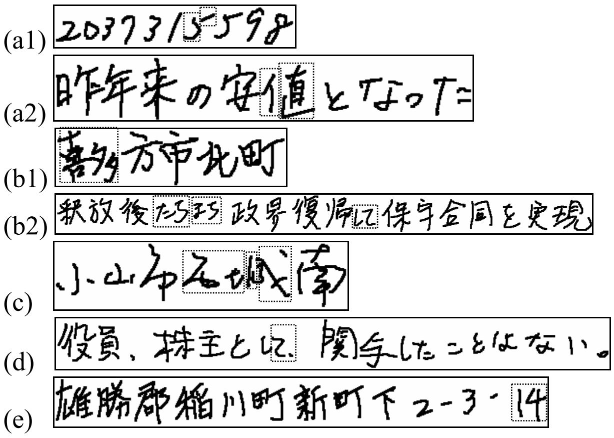
(a1) 2037315598
(a2) 昨年来の安値となった
(b1) 喜多方市北町
(b2) 釈放後たちまち政界復帰に保守合同を実現
(c) 小山市石城南
(d) 役員、株主として関与したことはない。
(e) 雄勝郡稲川町新町下2-3-14

Figure **5**. Examples of segmentation samples with errors.