A search method for on-line handwritten text employing writing-box-free handwriting recognition

Hideto Oda, Akihito Kitadai, Motoki Onuma and Masaki Nakagawa
Graduate School of Technology, Tokyo University of Agriculture and Technology
2-24-16 Naka-cho, Koganei-shi Tokyo, 184-8588 Japan
oda@hands.ei.tuat.ac.jp

Abstract

This paper presents a method for writing-box-free on-line handwritten text search. It searches for a target keyword in the lattice composed of candidate segmentations and candidate characters. By considering the accuracy of the recognition method and the length of the keyword, the method decreases noises to be output from the lattice effectively. When the keyword consists of three characters, we have achieved the recall rate 89.4%, the precision rate 93.2% and F measure 0.912.

Keywords: search method, on-line handwritten text, character recognition, text recognition

1. Introduction

Handwriting input interfaces has been employed in environments where a keyboard is not suitable. Portable devices like PDA, interactive electronic whiteboards and tablet PC's are examples of such environments where a keyboard is too large for mobile systems or it is not suitable for annotations.

A considerable number of studies have been made on handwriting input interfaces with pen devices from both sides of hardware and software. High-specification portable devices are equipped with character input functions incorporating recognition systems. An interactive electric whiteboard, which is composed of a large display integrated with a digitizer, have been used in classrooms. It allows teachers to introduce the power of IT into their classrooms while exploiting their skills to teach students using a blackboard. A tablet PC can be employed for dictation or annotation while reading electronic documents. On-line handwritten patterns, handwritings sampled in the form of a sequence of pen-tip coordinates, produced on these devices will be accumulated. Without a search method, however, accumulated on-line handwritten data will not be utilized effectively. Search of off-line handwritten documents have been studied for many years but research on on-line handwritten text search is not fully considered yet.

Although on-line patterns and off-line patterns are different, it would be worth mentioning previous work in the field of off-line document search. Maruyama et al. proposed a search method, which reduced search loss from incorrect recognition results by using two or more character recognition candidates and a confusion matrix [1]. Ota et al. further extended the above idea by producing search terms considering mis-segmentation as well as mis-recognition with the confidence from them and from bi-gram [2]. Imagawa et al. investigated reliability of recognition results using a neural network and they showed that both the recall rate and the precision rate were improved by their method [3]. Lopresti et al. examine how OCR noises effect the performance of common information search models [4].

For on-line handwritten text search, a pattern matching method without character recognition was reported in [5]. Lopresti et al. also proposed stroke search method "ScriptSearch Algorithm", which searches through a long handwritten text pattern and find approximate patterns of a pattern given as a keyword [6].

This paper is composed of the following sections. Section 2 describes our on-line handwritten text recognition method on which a proposed search method is based. Section 3 presents the on-line handwritten text search method. Section 4 describes experiments and evaluations of our method. Section 5 concludes this paper.

2. An on-line handwritten text recognition method

2.1 Outline of the method

Our on-line handwritten text recognition method is composed of the following four processes [7].

i) Segmentation of handwriting into text line elements
ii) Segmentation of text line elements into character pattern elements
iii) Generation of candidate lattice
iv) Determination of the optimum text recognition candidate

We describe them in the following sections.

2.2. Segmentation of handwriting into text line elements

First, we define some terminology as follows. Character orientation designates 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. A text line is a piece of text separated by new-line and large space and it is further divided into text line elements at the changing points of writing direction. Each text line element has its line direction. The line direction and the character orientation are independent. Our recognition system hypothetically segments a handwriting pattern into text line elements by examining geometric features of strokes and movements between pen-up and pen-down.

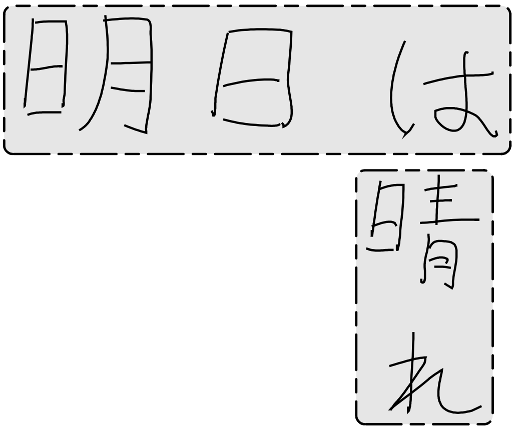

Figure 1. Text line elements.

2.3. Segmentation of a text line element into character pattern elements

A text line element consists of one or more character patterns. However, It is difficult to segment a text line element into character patterns only by geometric features and recognition results. Our on-line handwritten text recognition method over-splits a text line element into two or more partial patterns called character pattern elements. A character pattern element or a combination of two or more adjacent character pattern elements forms a character pattern. Over-split character pattern elements are merged in the process of the Viterbi search for the candidate lattice that will be described later.

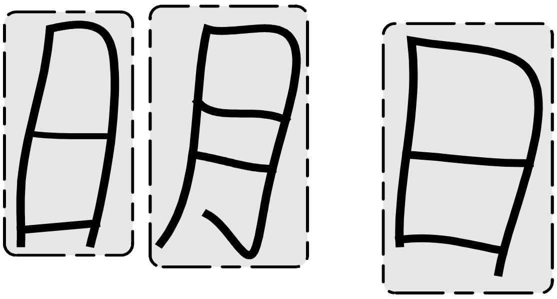

Figure 2. Character pattern elements.

2.4. Candidate lattices

Our on-line handwritten text recognition method hypothetically segments a text line element into character pattern elements and attempt to recognize one or more adjacent character pattern elements as a character and assign a set of candidate classes with recognition scores as shown in Fig. 3 [8]. We define a character pattern candidate as one or more adjacent character pattern elements to be recognized as a character. We assume a text line element to be a sequence of character pattern candidates. Since the segmentation is not deterministic at this stage, there are multiple ways of segmentations (defined as segmentation candidates) and multiple candidate classes (defined as character recognition candidates) assigned to each character pattern candidate so that a text line element is represented as a lattice where each node represents a character pattern candidate with possibly multiple character recognition candidates and each path represents a sequence of candidate segmentations between a previous character pattern candidate and a succeeding character pattern candidate. We call it a candidate lattice as shown in Fig. 4.

Rank	Code	Score
1st	明	891
2nd	朋	873
3rd	門	860

Figure 3. Recognition candidate classes.

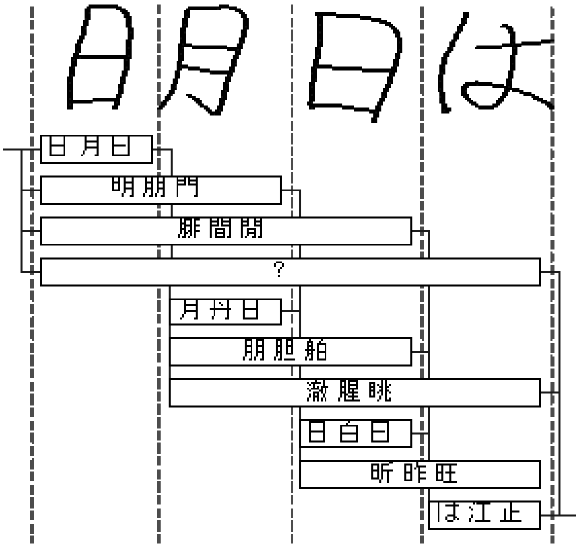

Figure 4. Candidate lattice.

2.5. Determination of the optimum text recognition candidate

Our on-line handwritten text recognition method calculates an evaluation score for each path in the candidate lattice for a text line element from the likelihood composed of character segmentation, character recognition, character pattern structure and context.

A sequence of character recognition candidates in the candidate lattice with the highest evaluation score is the recognition result. We search the candidate lattice using the Viterbi search.

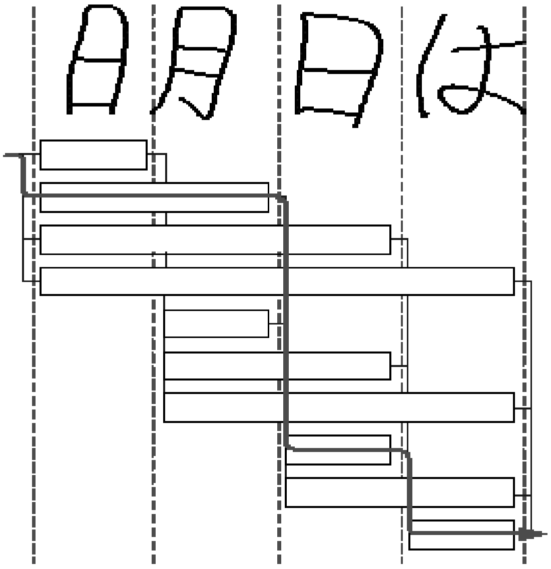

Figure 5. Path search in the candidate lattice.

3. Keyword search method

3.1. An overview of the search method

We propose a full-text search method for locating a keyword in on-line handwritten text patterns produced in free format without imposing any constraint of writing boxes, grids or baselines. Here the problem is that the keyword may be presented from a keyboard as character codes while handwritten text patterns are on-line patterns rather than a sequence of character codes. Senda et al. proposed a search method for on-line handwritten text patterns by a keyword in the form of on-line patterns. In this case, pattern matching is the method for search [5]. This method does not need a character recognition engine and it is language independent but the search reliability may not be high enough. Search efficiency may also be degraded due to the pattern matching employed. Moreover, if the search keyword is input form a keyboard, its on-line handwritten pattern must be produced virtually and compared with on-line handwritten text patterns.

On the other hand, we can consider a search method to on-line handwritten text recognition results, i.e., a sequence of character codes, but incorrect recognition results would cause search losses. We propose a search method by a keyword with taking mis-recognition into account, which searches into the candidate lattice generated from the recognition process. The candidate lattice can be generated when the on-line handwritten patterns are saved in a database so that it does not incur its recognition time when the keyword is searched. Since we search into the candidate lattice, we could reduce search losses by considering 2nd, 3rd and n-th character recognition candidates.

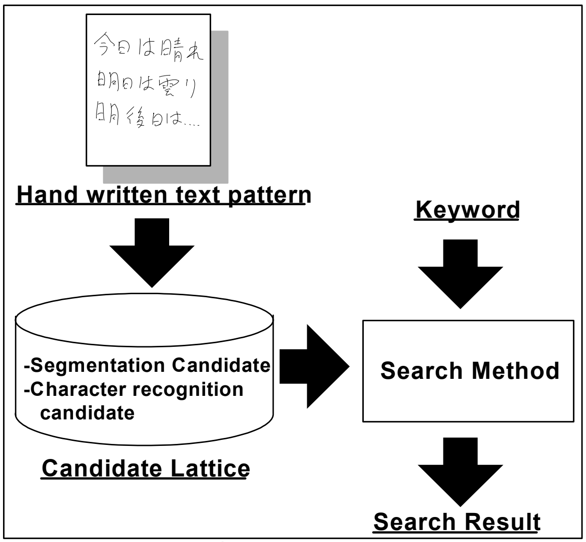

Figure 6. Searching process.

3.2. Reduction of search noises and an amount of calculation

Among possible paths in a candidate lattice, only a single path is the correct answer and others are noises. Therefore, it is important for the search method into the candidate lattice to reduce the amount of computation and search noises.

We could use the Viterbi search for searching into the candidate lattice. In addition, we could reduce the amount of computation and search noises by pruning segmentation and character recognition candidates having low evaluation scores. We will describe this in more detail in section 3.3.

3.3. Criterions for reduction of recognition results

We propose two ways of pruning character recognition candidates in every node in order to reduce the amount of computation and search noises.

i) Pruning of character recognition candidates whose ranks in character recognition are below the threshold "Tr".
ii) Pruning of character recognition candidates whose recognition scores are below the threshold score "Ts".

We will use the term "candidate pruning threshold" to refer to Tr and Ts.

In addition to the above, we could employ the evaluation of character pattern sizes in the process of generating the candidate lattice without having less promising branches or pruning unlikely nodes by the context process, but they remain to be implemented in the next step.

3.4. Criterions for evaluation

We evaluate the search method by counting the F measure. The F measure is defined by the formula (1).

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

In the formula (1), R is recall, P is precision and they are expressed by the formula (2)-(3).

The recall rate represents tolerance to search losses, while the precision rate represents tolerance to search noises. We evaluate the overall performance of the search method in terms of the F measure.

$$R = \frac{\text{The number of the information which conformed}}{\text{The number of conformable information in the search target}} \qquad (2)$$

$$P = \frac{\text{The number of the information which conformed}}{\text{The number of the searched information}} \qquad (3)$$

3.5. Length of the search keyword and candidate pruning threshold

We can calculate the optimal candidate pruning threshold Tr or Ts by employing a sufficient number of training sets.

We assume that the optimal candidate pruning threshold may vary depending on the length of a search keyword. This is because the probability of search loss by incorrect recognition is high when a search keyword is long; on the other hand, the probability of search noise by incorrect recognition is high when a search keyword is short.

4. Experiment and evaluation

4.1. A database used for experiments

We employ the database "TUAT Nakagawa Lab. HANDS-kuchibue_d97_06" (hereafter, we call it Kuchibue) in the experiment and evaluation for the search method. Kuchibue is a set of on-line handwritten text patterns written by 120 participants with each composed of 11,962 character patterns written by a single participant (10,152 character patterns in meaningful context, 1,810 character patterns without context) [9]. Participants wrote characters in a sequence of writing boxes one by one in each box on a LCD-integrated tablet. This is the style that we prepare text on manuscript papers. We made 113 virtually writing-box-free on-line handwritten text patterns by throwing away the box information from on-line handwritten text patterns. Writing-box-free on-line handwritten text patterns thus created may have different features from actual on-line writing-box-free handwritten text patterns. However, There is no writing-box-free on-line handwritten text pattern database compatible with the scale of Kuchibue. Therefore, this is the first step to evaluate the search method. We use 57 sets including 578,664 character patterns as training sets and 56 sets of 568,512 patterns as testing sets.

4.2. Generation of a candidate lattice

We generated a candidate lattice for each set of on-line handwritten text patterns. It takes about four minutes to generate a lattice for each set on a Pentium IV 3.06 GHz processor with 512MB RAM. A candidate lattice generated for each set occupies about 10 MB while the size of each set is about 4 MB, both without compression. These sizes can vary depending on implementation but we consider that the size of the candidate lattice is not too large compared with that of on-line handwritten text patterns.

4.3. Setting candidate pruning threshold

We obtained Tr and Ts, which bring the highest F measure from the experiment of searching keywords into the whole 57 training sets. We tested 1,000 kinds of search keywords for every length of two, three and four characters. Figure 7 to Figure 9 show the F measure to each length of the search keyword.

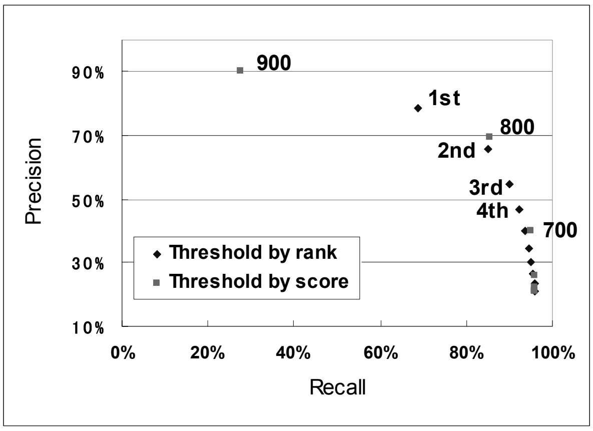

Figure 7. Relationship between recall rate and precision rate (Keyword of 2 characters)

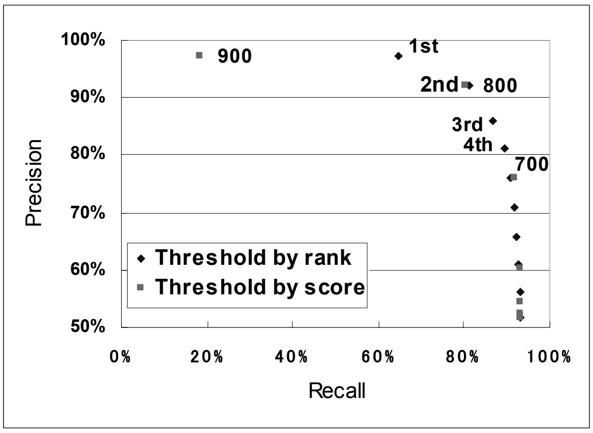

Figure 8. Relationship between recall rate and precision rate (Keyword of 3 characters).

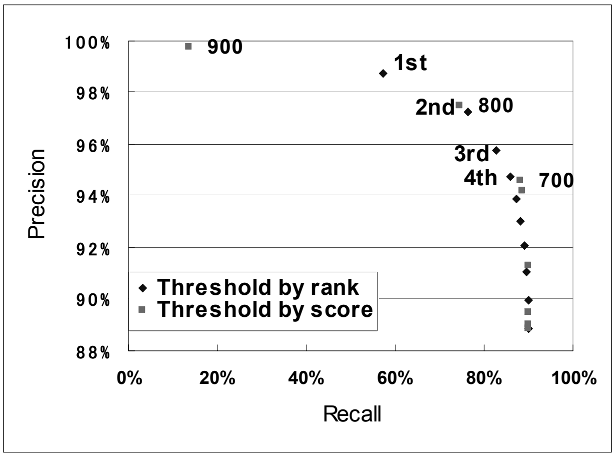

Figure 9. Relationship between recall rate and precision rate (keyword of 4 characters).

We can draw the following conclusions from the experiment.

- There is a trade-off between the recall rate and the precision rate.
- The recall rate increases while the precision rate decreases, as the search keyword becomes longer.
- The F measure with Ts is higher than the F measure with Tr for all the lengths of the search keyword.

Table 1 and Table 2 show the candidate pruning threshold, the recall rate and the precision rate when the F measure is the highest for each length of the keyword.

Table 1. Candidate pruning threshold Tr and training result.

Length	Rank	Recall	Precision	F measure
2	2nd	84.8%	65.8%	0.741
3	3rd	86.9%	85.9%	0.864
4	6th	88.3%	93.0%	0.906

Table 2. Candidate pruning threshold Ts and training result.

Length	Score	Recall	Precision	F Measure
2	820	79.9%	75.0%	0.774
3	771	86.1%	88.4%	0.872
4	687	88.5%	94.2%	0.912

4.4. Evaluation results

We obtained the F measure in the evaluation of searching into the whole testing sets by using the candidate pruning thresholds obtained in the previous experiment. We used the same search keywords as the previous experiment in the chapter 4.3. Table 3 and Table 4 show results.

Table 3. Candidate pruning threshold Tr and evaluation result.

Length	Rank	Recall	Precision	F measure
2	2nd	88.0%	67.6%	0.764
3	3rd	89.3%	86.6%	0.879
4	6th	89.4%	93.2%	0.912

Table 4. Candidate pruning threshold Ts and evaluation result.

Length	Score	Recall	Precision	F Measure
2	820	83.3%	76.9%	0.799
3	771	88.7%	89.2%	0.890
4	687	89.5%	94.1%	0.917

We can see that the F measure with Ts is higher than the F measure with Tr in all the lengths of the search keyword the same as the experiment of searching into the whole training sets.

The time to search for a keyword into each set was 0.02 second on a Pentium IV 3.06 GHz processor with 512MB RAM.

5. Conclusion

We proposed a search method for on-line handwritten text patterns employing a recognition engine. The proposed method reduces an amount of computation and search noises by generating a candidate lattice from on-line handwritten text patterns before the search process. It has achieved the recall rate 89.4%, the precision rate 93.2% and the F measure 0.912 for testing sets. It takes 0.02 second to search for a keyword into writing-box-free on-line handwritten text patterns about 10,000 characters. There remains work on experiments and evaluation employing true writing-box-free on-line handwritten text patterns, fine tuning of candidate pruning threshold for each writer and pruning of a candidate lattice using context.

References

[1] K. Marukawa, H. Fujisawa, Y.Shima, "Evaluation of Information Retrieval Methods with Output of Character Recognition Based on Characteristic of Recognition Error (Japanese)," Trans IEICE, vol. J79-D-II, no.5, pp.785-794, May, 1996.

[2] M. Ota, A. Takasu, J. Adachi "Full-text search Methods for OCR-recognized Japanese Text with Misrecognized Characters (Japanese)," Trans. IPSJ, vol.39, no.3, pp.625-635, Mar, 1998.

[3] T. Imagawa, Y. Matsukawa, K. Kondo, T. Mekata, "A document image retrieval technique using each character recognition reliability (Japanese)," TECHNICAL REPORT OF IEICE PRMU99-72, pp.63-68, 1999.

[4] Daniel Lopresti and Jiangying Zhou, "Retrieval Strategies for Noisy Text," Proceedings of FifthAnnual Symposium on Document Analysis and Information Retrieval, April 15-17, 1996, pp. 255-269.

[5] S. Senda, Y. Matsukawa, M. Hamanaka, K. Yamada, "MemoPad: Software with functions of Box-free Japanese Character Recognition and Handwritten Query Search (Japanese)," TECHNICAL REPORT OF IEICE, PRMU99-75 pp.85-90, Sept, 1999

[6] Daniel Lopresti, Andrew Tomkins. "On the searchability of electronic ink," Proceedings of the Fourth International Workshop on Frontiers in Handwriting Recognition , pp.156-165, Dec. 1994.

[7] Masaki Nakagawa and Motoki Onuma, "On-line handwritten Japanese text recognition free from Constrains on Line Direction and Character Orientation," Proc. 7th International Conference on Document Analysis and Recognition, pp.519-523, Edinburgh, Scotland, Aug 2003

[8] M. Nakagawa, K. Akiyama, "A Linear-time elastic matching for stroke number free recognition of on-line handwritten characters," Proc. 4th IWFHR, pp.48-56, Dec. 1994.

[9] M. Nakagawa, T. higashiyama, Y. Yamanaka, S. Sawada, L. Higashigawa, and K. Akiyama, "On-line handwritten character pattern database sampled in a sequence of sentences without any writing instructions," Proc. 4th ICDAR, vol.1, pp.376-381, Aug. 1997.