Improvements in keyword search Japanese Characters within handwritten digital ink

Cheng Cheng [12], Bilan Zhu[2], Xiaorong Chen[1] and Masaki Nakagawa[2]
[1] *Guizhou University, Guiyang 550025, China*
[2]*Tokyo University of Agriculture and Technology, Tokyo 184-8588, Japan;*
chengchengowen@gmail.com

Abstract

This paper presents a revised method for keyword search from handwritten digital ink in comparison with the previous system. We adopt a search method using noise reduction. Experiments on digital ink databases show that the revised method typically improves the system's overall accuracy (f-measure) from 0.653 to 0.891.

Keywords:
Keyword search, digital ink, segmentation, character recognition

1. Introduction

In some situations, handwriting (pen input) is more effective than other input methods such as keyboard for writing down memos and ideas. A variety of pen input devices are already commercially available, e.g. tablet PC, PC Notes Taker [1], Anoto Pen [2][3], DigiMemo [4] and so on. However, software that can quickly and accurately recognize or search pen strokes, or "digital ink", from these devices is still much in demand.

In one particular usage, one is to find occurrences of a phrase, or a keyword, within handwritten digital ink. The phrase is given in some encoding (such as ASCII or Unicode), while each pen stroke of the digital ink is specified as a series of 2D coordinates. The problem is complicated because we typically have no knowledge of which of the pen strokes belong to an individual letter (or word).

Searching for phrases (word spotting) within off-line handwritten text has been studied for many years; *Manmatha et al.* used Euclidean Distance Mapping and SLH Algorithm for searching [5]. *Lu et al.* presents a document retrieval technique that is capable of searching document images without OCR [6]. *Marukawa 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 [7]. *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 [8]. *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 [9].

Their targets are mainly printed documents and hand-printed historical documents, which are often damaged by several reasons. Those for online digital ink have been less well considered. Due to the proliferation of pen-based devices, however, demands for ink search is expanding.

Between offline paradigm and online paradigm, features, suitable segmentation methods and character recognition methods differs so that search methods will also differ.

Early work was made by *Lopresti et al.* [10]. They proposed ink search at several level of representations. For character level matching they showed performance prediction based on simulated text and fuzzy string matching. For stroke level, they presented stroke level matching algorithm and its performance. They continued the former research and formulated approximate string matching and fuzzy logic [11], which is also valid for noisy text after OCR (offline paradigm). *Senda et al.* presented a method to retrieve handwritten memos with handwritten queries [12]. This employs matching at the feature level. *Oda et al.* proposed a search system for finding Japanese keywords in digital ink by employing handwritten character recognizer (HWX) [13]. They prepare candidate lattice from digital ink, which are much richer representations than just sequences of top candidates.

978-0-7695-3725-2/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDAR.2009.20

In general, ink search at low level features is language independent but often writer dependent while that at the level of recognized character level is language dependent but can be writer independent if HWX is writer independent. Accuracy and efficiency depend on features, methods, screening, indexing and so on.

We follow along HWX-based systems. In this paper we suggest new methods for the problem, and demonstrate that these methods are effective in improving the previous system by *Oda et al.* [13] .We first give an outline of the HWX-based system.

2. Basic idea of the HWX-based system

The system is given as input a keyword (i.e., a short string of Japanese character codes) and searches it in digital ink.

2.1. Before the search

Before the search is made, the system hypothetically segments the digital ink, divide an input sequence of pen strokes into segments each of which may form a single character pattern.

These segments are then considered for composite segments. That is, the system forms candidates of composite segments by combining (up to some to be determined number of) neighboring segments.

Together, these candidates form the search space, which is a lattice where each candidate is a node, and an edge exists between every two candidates which are neighboring segments. Figure 1 shows an example.

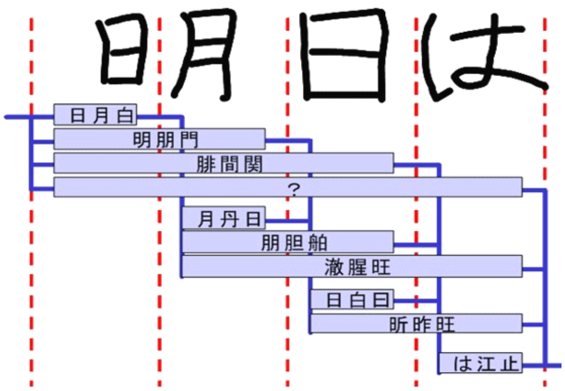
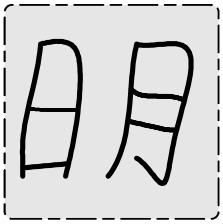


Figure 1. Search space from the digital ink "明日は".

A recognition engine is used to identify the possible characters for each candidate. Each of these characters is given a score corresponding to how well it matches the strokes in the segment. For example, the segment for the digital ink "明" in Figure 1 is recognized to be one of the following characters: "明", "朋", "門" (for their scores see Figure 2).



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

Figure 2. Recognition candidates for pattern "明".

2.2. Executing the search

Given a search keyword, the system searches the lattice for paths which match the input keyword. For each such path found the system outputs the positions of the start and end strokes of the path.

3. Improved search methods

The method in the [13] for searching the generated candidate lattice is the Viterbi search. Recall and precision of the search depends on the correctness of the candidate lattice.

In order to both increase the precision of its output as well as speed up the search, it is desirable to prune bad candidates from the search space. Two ways of pruning candidates were employed. The following candidates are pruned:

i) Candidates whose ranks in character recognition are below a threshold ***Tr***.

ii) Candidates whose recognition scores are below a threshold score ***Ts***.

However, the precision of the Viterbi search in the [13] is often affected by a tiny discrepancy between strokes in digital ink. This happens when the correct set of strokes contain a subset which produces a high score as well. For example consider the two high scoring matches for the keyword "明 日 が" in Figure 3.

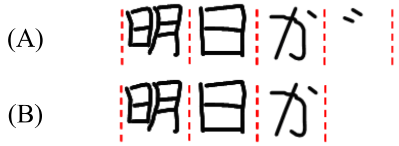


Figure 3. (A) The correct set of strokes for the keyword "明 日 が"; (B) A subset of the correct set of strokes which give a high score as well.

Although the strokes in (A) produces the highest score (since “が” matches “が” better than “か”), the strokes in (B) also produces a significantly high score for the input keyword. Since both (A) and (B) are output by the search, (B) reduces the search’s overall precision. This effect is especially noticeable in the case of the Japanese language since many Japanese hiragana/katakana characters each consist of two slightly different versions: one for voiceless sound and one for voiced sound.

We use a straight-forward and efficient solution for this problem. We first sort the results from the Viterbi search by their scores. Then, starting from the result with the highest score, we remove all the results with segments of over a certain length that overlaps it.

4. Experiment and evaluation

4.1. Criterions for search

We evaluate the search system by counting the *f-measure*. The *f*-measure is defined by the formula (1).

$$f = \frac{2}{\frac{1}{r} + \frac{1}{p}} \qquad (1)$$

In Formula (1), $\boldsymbol{r}$ is called *recall*, $\boldsymbol{p}$ is called *precision* and they are expressed by the formulas (2)-(3). The recall rate expresses the tolerance of the system to search losses, while the precision rate expresses the system’s tolerance to search noises. The *f-measure* expresses the overall performance of the search method.

$$r = \frac{\textit{Number of correct search}}{\textit{Number of search keywords in target data}} \quad (2)$$

$$p = \frac{\textit{Number of correct search}}{\textit{Number of searched items (include noise)}} \quad (3)$$

4.2. Databases used for experiments

We employ the database “TUAT Nakagawa Lab.HANDS-Kondate_t_bf-2001-11” (hereafter we refer to as Kondate) and the database “TUAT Nakagawa Lab. HANDS-kuchibue_d-97-06” (hereafter we refer to as Kuchibue) in our experiment and evaluation for the search system. Kondate is a set of on-line handwritten text patterns written by 100 participants with each composed of 2,425 character patterns written by a single participant (all of the character patterns in meaningful context). 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).

Oda et al. used 57 Kuchibue sets (i.e. 578,664 character patterns) to discover the optimal $\boldsymbol{Ts}$ and $\boldsymbol{Tr}$. We use these same values in all our experiments, including those on Kondate.

We use all of the Kondate databases for our tests. For Kuchibue, we use only 95 sets of character patterns which have the same format (i.e. 4 lines on a single page, with 9 characters on each line).

We use the database “TUAT Nakagawa Lab.HANDS-nakayosi t-98-09” (hereafter we refer to as Nakayosi) to train the character recognition engines. Nakayosi is a set of on-line handwritten text patterns written by 163 participants with each composed of 10,403 character patterns written by a single participant (7,376 character patterns in meaningful context, 3,027 character patterns without context). In addition, we used 50 Kondate sets to train the segmentation.

4.3. Evaluation results

In [13] Oda *et al.* tested their system with the Kuchibue database. The results they obtained are as shown in Table 1. Under the same environment, we tested our system. We show these results in Table 2.

Table 1: *Oda et al’s Kuchibue database results [1]*

Length	*Recall*	*Precision*	*f-measure*
2	83.3%	76.9%	0.799
3	88.7%	89.2%	0.890
4	89.5%	94.1%	0.917

Table 2: *Our Kuchibue database evaluation result*

Length	*Recall*	*Precision*	*f-measure*
2	92.0%	88.2%	0.900
3	93.6%	94.1%	0.937
4	94.4%	96.3%	0.952

Then, we tested our system and Oda *et al.* system with the Kondate databases. These results are shown in Table 3 and Table 4, respectively. They show that our methods significantly improved Oda *et al.* system’s performance.

Table 3: *Oda et al.’s Kondate database results*

Length	*Recall*	*Precision*	*f-measure*
2	59.9%	53.2%	0.564
3	68.0%	66.5%	0.672
4	65.1%	65.4%	0.653

Table 4: *Our Kondate database evaluation result*

Length	*Recall*	*Precision*	*f-measure*
2	77.4%	74.2%	0.756
3	87.1%	86.3%	0.867
4	88.7%	89.8%	0.891

5. Conclusions and Future Research

In this paper, we suggested the use of a few methods to increase the accuracy of finding Japanese keywords in digital ink. Applying the methods on the earlier system showed that the methods increased the accuracy of the system.

For future work, it would be interesting to perform such keyword searches without the generation of a candidate lattice prior to the search.

7. References

[1] http://www.pc-notetaker.com/

[2] http://www.acreo.se/upload/Publications/Proceedings/OE00/00-KAURANEN.pdf

[3] Kauranen (o.J.), "The ANOTO pen - Why light scattering matters". International Workshop on Microfactories 2004

[4] http://www.acecad.com.tw/

[5] R. Manmatha, C. Han and E. Riseman: "Word Spotting: A New Approach to Indexing Handwriting" In: Proc. of the IEEE Computer Vision and Pattern Recognition Conference, San Francisco, CA, June 1996

[6] S. Lu, L. Linlin and C. L. Tan, "Document image retrieval through word shape coding", IEEE Trans. PAMI, Vol.30, no.11, pp.1913-1918, Nov 2008.

[7] K. Marukawa, H. Fujisawa and 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.

[8] M. Ohta, A. Takasu and 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.

[9] 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.

[10] D. lopresti and A. Tomkins. "On the searchability of electronic ink" Proc. 4th International Workshop on Frontiers in Handwriting Recognition. Pp.156-165 Dec. 1994

[11] D. Lopresti and J. Zhou, "Retrieval Strategies for Noisy Text," Proc. 5th Annual Symposium on Document Analysis and Information Retrieval, April 15-17, pp. 255-269,1996

[12] S. Senda, Y. Matsukawa, M. Hamanaka and 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.

[13] H. Oda, A. Kitadai, M. Onuma and M. Nakagawa: "A search method for on-line handwritten text employing writing-box-free handwriting recognition", Proc. 9th International Workshop on Frontiers in Handwriting Recognition (IWFHR), Tokyo, Japan, pp. 545-550,2004.