Comparing Character Recognition Based Approach with Feature Matching Based Approach for Digital Ink Search

Cheng Cheng, Bilan Zhu and Masaki Nakagawa,
Department of Computer and Information Sciences, Tokyo University of Agriculture and Technology
E-mail: {50009834702}@st.tuat.ac.jp, {zhubilan, nakagawa}@cc.tuat.ac.jp

Abstract

This paper presents a character recognition based approach to search for a keyword in on-line handwritten Japanese text. It employs an on-line character recognizer or an off-line recognizer, produces recognition candidates and search for a keyword in the lattice of the candidates. This paper also presents a feature matching based approach employing on-line features or off-line features. We compare the above two approaches and conclude that the character recognition based approach yields superior performance compared to the feature-matching-based approach.

Keywords-digital ink search; geometric context; character recognition; feature matching

1. Introduction

Pen-input devices such as PDAs, tablet PC's and electronic whiteboards as well as touch-sensitive devices such as iPod and iPad are spreading into the world drastically and forming a new platform where a keyboard is too large for operation or not suitable for human computer interaction. On these devices, a pen-tip or a fingertip is traced and its trajectory is expressed by a time-sequence of strokes and each stroke is again a time-sequence of coordinates from pen-down to pen-up. A sequence of strokes is classified as on-line handwriting or handwritten patterns, which are often called digital ink.

Digital ink is recognized by on-line handwritten character recognition (ONHCR) or left unrecognized. Due to the proliferation of pen-input or touch-sensitive devices, accumulation of digital ink is expected so that digital ink should be searchable though it should not be necessary recognized when it is written. For instance, one is to find occurrences of a phrase or a keyword within digital ink. The phrase is given in some encoding (such as ASCII or Unicode) or in digital ink. The problem is how to find occurrences accurately and efficiently.

Early work was made by Lopresti *et al.* [1]. They proposed ink search at several level of representations. They continued this research and formulated approximate string matching and fuzzy logic [2], which is also valid for noisy text after OCR (off-line paradigm). They were followed by Senda *et al.* [3] for Japanese text and Jawahar *et al.* for Indian text [4]. They have employed feature matching; On the other hand, Zhang *et al.* and Oda *et al.* have employed ONHCR for Chinese text [6] and for Japanese text [5], respectively. They prepare candidate lattices from digital ink, which are much richer representations than just sequences of top candidates.

In general, digital 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 ONHCR is writer independent. Accuracy and efficiency depend on features, methods, screening, indexing and so on.

Not only for on-line handwriting patterns, but also searching for a word or phrase within scanned documents has been studied for many years. Often target documents are very old and damaged hand-printed or printed documents so that optical character recognition (OCR) does not work well for them [7]. In this field, often the term "word spotting" is used. Here again, the character recognition based approach [8, 9, 10, 11, 12] and the feature matching based approach have been employed [13, 14, 15]

Between offline paradigm and online paradigm, common methods and techniques are applicable although they differ since features, suitable segmentation methods and character recognition methods differ.

Focusing on on-line paradigm, i.e., digital ink search, we propose an ONHCR-based system and compare it with a feature-matching-based keyword search approach. The on-line or off-line features employed in this study are those used in most of high performance systems. The rest of this paper is organized as follows: Section 2 details the architecture of ONHCR-based digital ink search. Section 3 describes the feature-matching-based keyword search

978-0-7695-4774-9/12 $26.00 © 2012 IEEE
DOI 10.1109/ICFHR.2012.193

approach. Section 4 details the experiments and presents our consideration. Section 5 draws the conclusion.

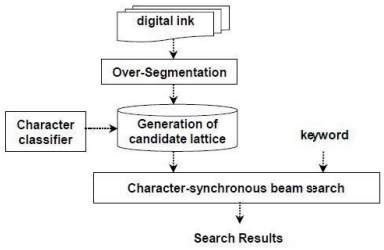


Fig. 1. Diagram of ONHCR-based keyword search

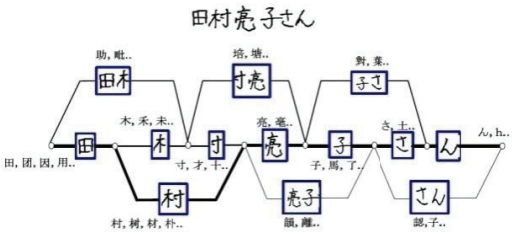


Fig. 2. Candidate lattice.

2. ONHCR-based approach

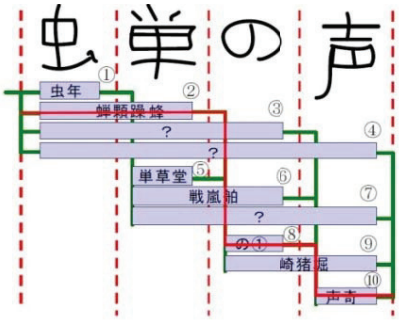
Digital ink search refers to the task of retrieving keywords from digital ink. The architecture of our retrieval system for on-line handwritten Japanese documents is shown in Fig.1. First, on-line handwritten Japanese patterns are over-segmented into primitive segments according to the features such as spatial information between adjacent strokes [16]. Then, one or more consecutive primitive segments are combined to generate candidate character patterns and each pattern is associated with several candidate character classes with scores assigned by a character recognizer [17]. The combination of all candidate patterns and character classes is represented by a segmentation and recognition candidate lattice (candidate lattice in short) as show in Fig.2. Last, the system searches the keyword into the candidate lattice, and obtains several occurrences when the resulting similarity score is higher than a certain threshold.

In the field of speech recognition and handwritten character string recognition, the time-synchronous approach and the character-synchronous approach are widely used for searching into a candidate lattice. The former expands search beams in line with time frames while the character synchronous approach dose so character by character.

Fig. 3 (a) shows our character-synchronous beam search algorithm . The algorithm works for the example shown in Fig. 3 (b). The candidate character pattern ① is the first node having three recognition candidates and it is followed by three succeeding patterns (⑤, ⑥ and ⑦). The first character of the input keyword is [虫単]. It is not in the recognition candidate classes for the pattern ① so that the search proceeds to the pattern ②. Then, [虫単] appears in the candidate classes for the pattern ②. Next, the second character [の] is compared with the pattern ⑧ and the pattern ⑨ following the character pattern ②and it is found in the candidate classes for the pattern ⑧. Third, the last character [声] is compared with the last pattern ⑩ and [声] appears in the candidate classes for the pattern ⑩. Finally, strokes from ②, ⑧ and ⑩ are outputs. The search continues from the pattern ③ but there are no more match with the keyword.

```
Algorithm 1: Search algorithm
Input: Keyword K = k1…km , candidate character patterns c1…cn
Output: a sequence of character pattern P = ps+1…ps+n
Begin initialize: i = 1 , j = 1 ;
1:for i = 1 to n
2:     Match(kj ,ci)
3:     if kj is a character class of ci
4:          ps+m = ci
5:          j++
6:          if kj is the last character of keyword
7:               output result and go to End
8:          else
9:               ci = the succeeding character pattern of ci
10:              go to Step 2
11:    else
12:         continue
13:End
```

(a) Pseudocode of the search algorithm



(b) Path search for the keyword 蝉の声

Fig. 3. Search algorithm and example.

Here, the precision is often affected by a tiny discrepancy between strokes in digital ink. This happens when the correct sequence of strokes contain a subsequence which produces a high score also as shown in Fig.4. Although the strokes in (A) produces the highest score (since "が" matches "が" better than "か"), the strokes in (B) also produces still a 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 have two or three slightly different family members: one for voiceless sound, one for voiced sound and another for p-sound.

Fig. 4. Correct stroke sequence (left) and subsequence (right) giving high scores.

We use a straight-forward and efficient solution for this problem. We first sort the results from the 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.

2.1. Similarity measure

Given a keyword, the system searches into the candidate lattice to find paths (stroke sequences) matching with the input keyword. Among the output paths, there are some noises. Therefore, it is important for the search method into the candidate lattice to reduce the amount of computation and search noises. To distinguish correct retrieval results from incorrect ones in the candidate lattice, we design a linear discriminant function to evaluate the paths. The discriminant function is defined as:

$$\boldsymbol{f(S,K)} = \frac{\mathbf{1}}{\boldsymbol{n}} \sum_{i=1}^{n} \mathrm{Rec}(s_i, c_i) \qquad (1)$$

where S is a sequence of strokes matching with the keyword, K is the input keyword (a sequence of character codes), $\boldsymbol{n}$ is the length of the input keyword, c_i is the *i-th* character of the input keyword; s_i denotes the *i-th* character pattern matched with c_i.

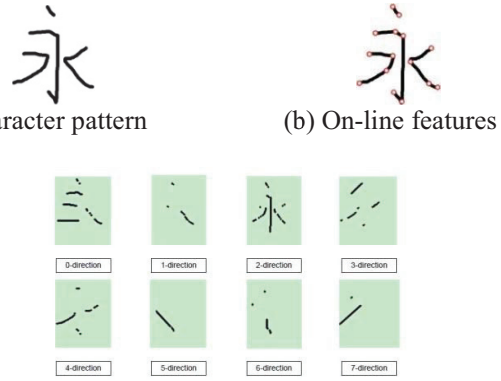
The term $Rec(s_i, c_i)$ is an output score of character recognizer on a character pattern s_i to a character class c_i. It is given by an on-line character recognizer or an off-line character recognizer.

The details are described in [17]. Here we only show their performances in Table 1.

Table 1. Cumulative character recognition rates in Kondate

recognizer	Top rec.rate	5th cumulative	10th cumulative
On-line	78.30	95.11	97.95
Off-line	79.99	96.81	98.46

The on-line character recognizer employs Markov Random Field (MRF) models. We normalize an input character pattern linearly to a standard size while keeping the horizontal-vertical ratio and extract feature points from each stroke so that the start and end points are selected and the most distant point from the straight line between adjacent feature points is selected while the distance is greater than a threshold [18] as shown in Fig. 5 (b). Then, we match feature points of each model with those of an input pattern by a Dynamic Time Warping (DTW) algorithm. We employ MRF to represent the character models and evaluate the similarities between the input pattern and the character models.



(a) Character pattern (b) On-line features

(c) Off-line features

Fig. 5. Feature extraction

On the other hand, the off-line character recognizer employs Modified Quadratic Discriminant Function (MQDF) [19]. We apply pseudo 2D bi-moment normalization (P2DBMN) [20] to an input pattern, apply the Gaussian blurring and extract 8-directional features in 8 x 8 regions as shown in Fig. 5 (c). To improve the Gausssianity, we apply the Box-Cox transformation to each feature. Then, we employ the Fisher linear discriminant analysis (FLDA) to reduce 512 dimensional features to 160 features. MQDF measures the similarities between the input pattern and the prototypes.

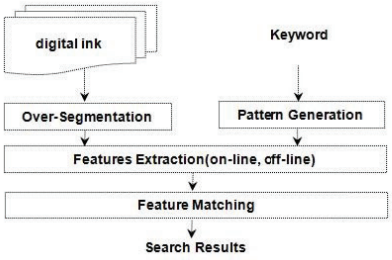


Fig. 6. Diagram of feature-matching-based approach

3. Feature-matching-based approach

In this paper, we compare the proposed ONHCR-based approach for digital ink search with feature-matching-based keyword search approach. Fig. 6 shows its diagram. We will describe each component in Fig.6 as follows.

In query pattern generation, a search keyword is input either from a keyboard as character code or from a tablet as digital ink. Two approaches are noted as code-based and ink-based, respectively. In this paper, we focus on the code-based approach where each character of the keyword is converted to a standard on-line or off-line pattern and their sequence is prepared for matching.

As for the over-segmentation of the target digital ink and the feature extraction, we employ the same methods described in Section 2. Fig. 7 shows an example of over-segmentation.

In feature matching, the keyword pattern is directly matched with digital ink by block shift. Given a keyword of $\boldsymbol{m}$ characters, we extract *candidate regions* from the digital ink starting from the current (initially first) primitive segment up to $\boldsymbol{n}$ consecutive primitive segments, where we consider $\boldsymbol{n}$ as wide as [$\boldsymbol{m}$, 5*$\boldsymbol{m}$] so that we do not miss the occurrence of the keyword pattern. The search algorithm computes a similarity score between the keyword pattern and all *candidate regions,* and produces the location of the candidate pattern if its matching score is less than a certain threshold. Then, it shifts the target regions by one segment and repeat this process.

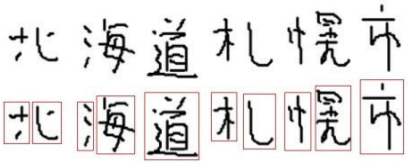


Fig. 7 An example of over-segmentation

We represent the keyword pattern as $\boldsymbol{X}$ and the *candidate regions* as $\boldsymbol{S_j}$ where j is an index of candidate regions $[j = 1, \dots, t]$.

For on-line feature-matching-based approach, the keyword pattern $\boldsymbol{X}$ is a concatenated sequence of feature points of standard on-line patterns for constituent characters of the keyword. On the other hand, each *candidate region* is linearly normalized to the size of the keyword pattern as shown in Fig. 8 and feature points are extracted from each *candidate region*. Then, $\boldsymbol{X}$ consisting of l_A feature points and $\boldsymbol{S_j}$ consisting of l_B feature points are represented as $\{p_1, p_2 \dots p_{l_A}\}$ and $\{q_1, q_2 \dots q_{l_B}\}$, respectively, where $p_i (i \le l_A)$ and $q_j (i \le l_B)$ are the feature point coordinates of $\boldsymbol{X}$ and $\boldsymbol{S_j}$. Then, the similarity score between the keyword pattern and each *candidate region* is measured by recurrence equation in Eq. (2), where $d(i,j)$ is the Euclidean distance between p_i and q_j.

$$\boldsymbol{f(X, S_j)} = \boldsymbol{DTW(i,j)} = \boldsymbol{min}\begin{pmatrix} \boldsymbol{DTW(i-1,j)} \\ \boldsymbol{DTW(i-1,j-1)} \\ \boldsymbol{DTW(i,j-1)} \end{pmatrix} + \boldsymbol{d(i,j)} \qquad \textbf{(2)}$$

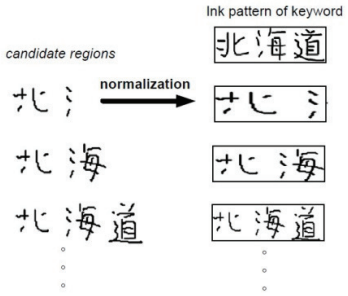


Fig. 8 Matching strategy of on-line feature-matching-based method

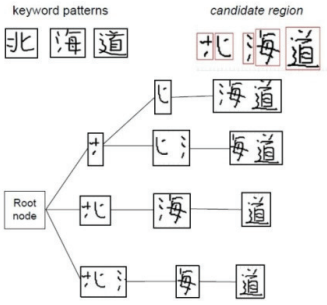


Fig. 9 Matching strategy of off-line feature-matching-based method

For off-line feature-matching-based approach, unlike the above on-line approach, the keyword pattern $\boldsymbol{X}$ is a split sequence of standard off-line features of constituent characters for the keyword. The keyword pattern is matched with all paths which belong to a *candidate region* by a tree match strategy and the highest score among them is output. Fig. 9 shows the tree match strategy. Each path has *m candidate patterns* where each *candidate pattern* is composed of a primitive segment or multiple consecutive segments. The off-line feature values same as those described in 2.1 are extracted from each *candidate pattern.* Then, $\boldsymbol{X}$ consisting of m character patterns is represented as $\{\boldsymbol{v_1}, \boldsymbol{v_2} \dots \boldsymbol{v_m}\}$, while $\boldsymbol{S_j} = \{\boldsymbol{S_j^i}\}$ denotes a set of paths in a *candidate region,* where $\boldsymbol{i}$ is the index of path and each $\boldsymbol{S_j^i}$ consists of m *candidate patterns* and represented as $\{\boldsymbol{w_1}, \boldsymbol{w_2} \dots \boldsymbol{w_m}\}$. Here, $\boldsymbol{v_i}$ and $\boldsymbol{w_i}$ are d-dimensionality feature vectors denoted as $[v_i^1, v_i^2, \dots v_i^d]$ and $[w_i^1, w_i^2, \dots w_i^d]$. Then, the similarity score is calculated by Eq. (3), where $||\ \ ||$ is the Euclidean metric.

$$f(X, S_j^i) = \sum_{i=1}^{m} ||v_i - w_i|| \quad (3)$$

In the feature matching-based-approach (both on-line and off-line), some output results may have overlap and the one of lower score is pruned in the same way as the ONHCR-based approach.

4. Experiments and evaluation

4.1. Sample pattern databases

To evaluate the performance of the proposed keyword search approach, we employ "HANDS-Kondate" on-line handwritten text database (in brief, Kondate). Kondate is written by 100 participants including 13,680 lines of text. We perform a 5-fold cross validation. The 13,680 text lines are split up into five blocks, consisting of 2,736 text lines each. Four blocks are used for training, and the remaining one for testing. On the other hand, the feature-matching-based keyword search approach does not need to be trained so that we use the test data for the ONHCR-based approach also to evaluate the performance of the feature-matching-based approach. The keyword set to test includes 48 keywords composed of two characters, 81 composed of three characters , 51 composed of four characters,25 composed of five characters and 13 composed of six characters, which totally appear 4,495 times, 6,315 times , 3,728 times,1,798 times and 842 times respectively in the test data of digital ink.

For training the character recognizers, we use "HANDS-Nakayosi" database (in brief, Nakayosi) [21]. Nakayosi is a set of on-line handwritten character patterns written by 163 participants with each contributing 11,962 character patterns.

We generate keyword patterns of individual characters from (HANDS-TEHON) database which includes 7,722 character patterns of correct stroke number and order (Kanji: 7116, Kana: 169, Roman characters: 52, Numerals and Symbols: 385).

We evaluate the performance of the search methods by f-*measure*

$$\text{f-}measure = \frac{2}{1/r + 1/p} \quad (4)$$

$$r = \frac{Number\ of\ correct\ search}{Number\ of\ search\ keywords\ in\ target\ data} \quad (5)$$

$$p = \frac{Number\ of\ correct\ search}{Number\ of\ searched\ items\ (include\ noise)} \quad (6)$$

where r is recall and p is precision defined in Eq.(5) and Eq.(6), respectively. The recall rate measures the tolerance to search errors, while the precision rate measures the tolerance to search noises. The *f-measure* is an overall performance of the search system.

4.1. Results and discussion

We evaluate the proposed approach for digital ink search in a series of experiments. The first experiment is to investigate how the different recognizers affect the performance. We test the on-line and the off-line recognizers. Table 2 shows the performance of the two recognizers. As longer the keyword is, as higher *f-measure* is probably because the larger amount of information is employed for search.

The second experiment compares the ONHCR-based approach with the feature-matching-based approach, which employs the same features as the former. Table 4 shows the performance by the latter.

As in Table 2, as longer the keyword is, as higher *f-measure* is.

Table 2. Performance of ONHCR-based approach

Keyword length	online			offline		
	r	p	f	r	p	f
2	.7243	.7468	.7354	.7511	.7494	.7502
3	.8000	.8165	*.8082*	*.8371*	.8150	.8259
4	.8555	.8539	.8547	.8718	.8433	.8573
5	.9156	.8753	.8950	.9296	.8731	.9004
6	.9158	.9079	.9119	.9575	.8762	.9151

Table 3. Performance of feature-matching-based approach

Keyword length	online			offline		
	r	p	f	r	p	f
2	.5412	.5456	.5434	.6071	.6359	.6212
3	.7393	.7449	.7421	.6925	.6976	.6950
4	.7374	.7524	.7448	.7329	.7505	.7416
5	.8003	.7977	.7990	.8416	.8518	.8467
6	.8610	.8631	.8621	.8214	.8501	.8355

By comparing Table 2 and Table 3, we can see that both of the ONHCR-based keyword search methods (on-line and off-line features) have higher performance than the feature-matching-based keyword search methods. This is probably because the ONHCR-based approach reflects deformation models in terms of discriminant functions such as MRF, MQDF or else, while the feature-matching-based approach cannot exploit a prior knowledge to pattern matching and only employ geometrical or shape measures of similarity.

On the other hand, the ONHCR-based keyword search approach does not support languages which are not assumed, and it requires a large amount of training patterns. If neither the language nor the alphabet is known, the feature-matching-based approach might be the only option available.

In this paper, we have excluded the ink-based feature-matching based approach, but ink-based methods may produce better search performance for his/her own handwriting.

5. Conclusion

In this paper, we have proposed the character recognition based approach for keyword search in on-line handwritten Japanese text and compared it with the feature matching based approach.

Whether on-line features or off-line features are employed, the character recognition based approach is superior to the feature matching based approach probably because the character recognition based approach reflects deformation models in terms of discriminant functions while the feature-matching-based approach cannot exploit a prior knowledge to pattern matching and only employ geometrical or shape measures of similarity. In western historical documents, however, often feature matching based methods have been reported, which contradicts with our results. The difference is two points: language and document quality. Therefore, we should like to test on low quality ink patterns which might be generated by degradation models.

On the other hand, the character recognition based approach assumes the language to be searched and it requires a large amount of training patterns. If neither the language nor the alphabet is known, the feature-matching-based approach might be the only option available.

References

[1] D. Lopresti and A. Tomkins. On the Searchability of Electronic Ink. Proc. International Workshop on Frontiers in Handwriting Recognition. Taipei, Taiwan. pp. 156-165, 1994.

[2] D. Lopresti and J. Zhou. Retrieval Strategies for Noisy Text. Proc. Annual Symposium on Document Analysis and Information Retrieval. Washington, pp. 255-269, 1996.

[3] 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 on IEICE, PRMU99-75. pp. 85-90, 1999.

[4] C. V. Jawahar, A. Balasubramanian, M. Meshesha, and A. M. Namboodiri. Retrieval of Online Handwriting by Synthesis and Matching. Pattern Recognition. pp. 1445-1457, 2009.

[5] H. Oda, A. Kitadai, M. Onuma and M. Nakagawa. A Search Method for On-line Handwritten Text Employing Writing-Box-Free Handwriting Recognition. Proc. International Workshop on Frotiers in Handwriting Recognition. Tokyo, pp. 157-162, 2004.

[6] H. Zhang, D. H. Wang and C. L. Liu. Keyword Spotting from Online Chinese Handwritten Documents Using One-Vs-All Trained Character Classifier. Proc. International Conference on Frotiers in Handwriting Recognition. Kolkata, India, pp. 271-276, 2010.

[7] A. Antonacopoulos and A. Downton. Special Issue on the Analysis of Historical Documents, International Journal on Document Analysis and Recognition. pp. 75-77, 2007.

[8] 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, pp. 785-794, 1996.

[9] M. Ota, A. Takasu and J. Adachi. Full-Text Search Methods for OCR-Recognized Japanese Text with Misrecognized Characters (Japanese). Trans. IPSJ, pp. 625-635, 1998.

[10] T. Imagawa, Y. Matsukawa, K. Kondo and T. Mekata. A Document Image Retrieval Technique using Each Character Recognition Reliability (Japanese). Technical report of IEICE PRMU99-72, pp. 63-68, 1999.

[11] H. Cao, A. Bhardwaj and V. Govindaraju. A Probabilistic Method for Keyword Retrieval in Handwritten Document Images. Pattern Recognition. pp. 3374-3382, 2009.

[12] A. Fischer, A. Keller, V. Frinken and H. Bunke. Lexicon-Free Handwritten Word Spotting Using Character HMMs, Pattern Recognition Letters. 2011

[13] T. M. Rath and R. Manmatha. Word Spotting for Historical Documents. International Journal on Document Analysis and Recognition. Vol. 9, No. 2, pp. 139-152, 2007.

[14] T. Konidaris, B. Gatos, K. Ntzios, I. Pratikakis, S. Theodoridis and S. J. Perantonis. Keyword-Guided Word Spotting in Historical Printed Documents using Synthetic Data and User Feedback. International Journal on Document Analysis and Recognition. pp 167-177, 2007.

[15] B. Zhang, S. N. Srihari, and C. Huang. Word Image Retrieval using Binary Features, Document Recognition and Retrieval XI. SPIE. San Jose pp. 45-53, 2004.

[16] B. Zhu and M. Nakagawa. Segmentation of On-line Freely Written Japanese Text using SVM for Improving Text Recognition. IEICE Trans. pp. 105-113, 2008.

[17] B. Zhu and M. Nakagawa. On-line Handwritten Japanese Characters Recognition Using a MRF Model with Parameter Optimization by CRF, Proc. International Conference on Document Analysis and Recognition, Beijing, pp. 603-607, 2011

[18] U. Rammer. An Iterative Procedure for the Polygonal Approximation of Plane Closed Curves. Comput. Graph. Image Process. Pp. 244-256, 1972.

[19] F. Kimura, K. Takashina, S. Tsuruoka and Y. Miyake. Modified Quadratic Discriminant Functions and the Application to Chinese Character Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence. pp. 149-153, 1987.

[20] C. L. Liu and X. D. Zhou. Online Japanese Character Recognition using Trajectory-Based Normalization and Direction Feature Extraction. Proc. International Workshop on Frontiers in Handwriting Recognition. La Baule, France, pp. 379-384, 2006.

[21] M. Nakagawa and K. Matsumoto. Collection of On-line Handwritten Japanese Character Pattern Databases and their Analysis. International Journal on Document Analysis and Recognition. Vol. 7, No. 1, pp. 69-81, 2004.