Recent Results of Online Japanese Handwriting Recognition and Its Applications

Masaki Nakagawa, Junko Tokuno, Bilan Zhu, Motoki Onuma,
Hideto Oda, and Akihito Kitadai

Tokyo University of Agriculture and Technology
Naka-cho 2-24-16, Koganei, Tokyo, 184-8588, Japan
nakagawa@cc.tuat.ac.jp

Abstract. This paper discusses online handwriting recognition of Japanese characters, a mixture of ideographic characters (Kanji) of Chinese origin, and the phonetic characters made from them. Most Kanji character patterns are composed of multiple subpatterns, called radicals, which are shared among many (sometimes hundreds of) Kanji character patterns. This is common in Oriental languages of Chinese origin, i.e., Chinese, Korean and Japanese. It is also common that each language has thousands of characters. Given these characteristics, structured character pattern representation (SCPR) composed of subpatterns is effective in terms of the size reduction of a prototype dictionary (a set of prototype patterns) and the robustness to deformation of common subpatterns. In this paper, we show a prototype learning algorithm and HMM-based recognition for SCPR. Then, we combine the SCPR-based online recognizer with a compact offline recognizer employing quadratic discriminant functions. Moreover, we also discuss online handwritten Japanese text recognition and propose character orientation-free and line direction-free handwritten text recognition and segmentation. Finally, as applications of online handwritten Japanese text recognition, we show segmentation of mixed objects of text, formulas, tables and line-drawings, and handwritten text search.

1 Introduction

As PDAs, tablet PCs, and other pen-based or paper-based systems, such as the Anoto [1] and e-pens [2], spread, the demand for improving online handwriting recognition and liberating it from writing constraint is still increasing. In online handwriting recognition, both temporal information of pen tip movement and spatial shape information are available, so it can yield higher recognition accuracy than offline handwriting recognition. Moreover, online handwriting recognition provides good interaction and adaptation capability because the writer can respond to the recognition result to correct misrecognition and rejection.

The research of online handwriting recognition began in the 1960s and has been receiving intensive interest since the 1980s. The comprehensive survey before the 1990s appears in [3][4]. In recent survey papers, Plamondon, et al. mainly

D.S. Doermann and S. Jaeger (Eds.): SACH 2006, LNCS 4768, pp. 170–195, 2008.

reviewed the status of western online handwriting recognition [5] while Liu, et al. and Jaeger, et al. reviewed online Chinese and Japanese handwriting recognition [6][7]. In this paper, we mainly discuss online Japanese handwriting recognition, including our recent results.

The Japanese character set consists of various characters: numerals, symbols, Hiragana, Katakana, and Kanji characters of Chinese origin. Hiragana and Katakana are phonetic characters. The former consists of 83 characters, and the latter consists of 86 characters. Kanji characters are idiographic characters. Two classes are defined for the purpose of computer processing: JIS first level and JIS second level (JIS stands for Japanese Industrial Standard). The JIS 1st level set contains 2,965 common characters, which are necessary for reading the newspaper, and the JIS 2nd level set contains 3,390 characters less common and special characters for naming.

Most Kanji character patterns are composed of multiple subpatterns, called radicals, which are shared among many (sometimes hundreds of) Kanji character patterns. This is common in Asian languages of Chinese origin, i.e., Chinese, Korean, and Japanese. Among Kanji character patterns, some patterns are simple, consisting of a single radical, while others are complex with multiple radicals.

In the field of pattern recognition, large volumes of sample patterns are as important as recognition methods. We spent four years compiling two databases of online Japanese handwritten character patterns, named "TUAT Nakagawa Lab. HANDS-kuchibue_d-97-06" (hereafter Kuchibue_d) [8] and "TUAT Nakagawa Lab. HANDS-nakayosi_t-98-09" (hereafter Nakayosi_t) [9]. Kuchibue_d stores 11,962 character patterns from each of 120 people (1,435,440 patterns), and Nakayosi_t stores 10,403 patterns from 163 people each (1,695,689 patterns). Thus, they store more than 3 million patterns in total. About 50 institutions, including more than 10 groups from abroad, use our databases, so we will base our experiments on these databases.

The large number of Japanese character categories affects the classification techniques. In western handwriting recognition, Hidden Markov models (HMM) are successfully applied. However, they are not common in Japanese handwriting recognition because they require huge amount of training data for each character. Therefore, DP-matching comprises the core of many online Japanese handwriting recognizers with several modifications proposed [10]-[12]. These large categories also affect the size of the dictionary (a set of prototype patterns) and the recognition speed, so it has been difficult to use a powerful online recognizer or combine online/offline recognizer to improve the recognition accuracy in a small computer.

The demand to remove writing constraint for online handwriting recognition is increasing steadily as people can write more freely on enlarged surfaces of tablet PCs, electronic whiteboards, and other paper-based handwriting environments. However, segmentation and recognition of online handwritten Japanese text is challenging work, because of the large variation of character size, and people write text horizontally, vertically or even slantwise.

In addressing these problems, we present structured character pattern representation (SCPR)-based online handwriting recognition, which has significant effect for the Japanese character set of the large category size, in Section 2. Section 3 describes the combination of the SCPR-based online recognizer with a compact offline recognizer. Section 4 presents an online handwriting Japanese text recognition method liberated from constraints of line direction and character orientation. Section 5 describes some applications. Section 6 draws conclusions.

2 Character Representation

In this section, we describe a Japanese Kanji character pattern representation, which strongly relates to handwriting recognition.

In the representation for input patterns, the sequence of feature points or line segments in time series are commonly used in online Japanese handwriting recognition [10]-[17] The stroke order and the stroke directions of the input pattern are kept in these sequences. Some recognition methods employ the offline features (e.g. directional features, loops of strokes) extracted from the images of the input patterns [18]-[20]. Employing these representations, instead of the raw data of the input pattern (sampling points by digital pen), reduces the data size and noise.

Early research attempted to extract subpatterns from an input character pattern and recognize the input pattern as the composite of subpatterns, but did not succeed because subpattern extraction was difficult. Instead, many systems employ the pattern structure composed of subpatterns in a prototype representation and expand it to a sequence of feature points, or line segments, when matched with the input pattern. Shape variations or stroke order variations register as multiple alternatives into a subpattern, so they are shared among character patterns that include the subpattern in their shapes. We will see this in more detail in the following sections.

2.1 Structured Character Pattern Representation (SCPR)

Japanese Kanji characters are mostly composed of multiple subpatterns called radicals. SCPR represents a character pattern as a composite of basic subpatterns (primitive, so they are not further decomposed) and structural information of how to combine them (Figure 1) [12].

Here, we present two online handwriting recognition systems. One uses the prototype learning algorithm (PLA) and linear-time elastic matching (LTM) [12][13], and the other system is based on HMM [21]. We call the former system "Sys_LTM" and the latter "Sys_HMM". Both employ the SCPR dictionary in which prototypes of basic subpatterns (BSs) are shared among character categories, as shown in Figure 2.

In the SCPR dictionary of Sys_LTM, all the BS prototypes, as well as the character pattern prototypes, are represented by a square shape with 128 x 128

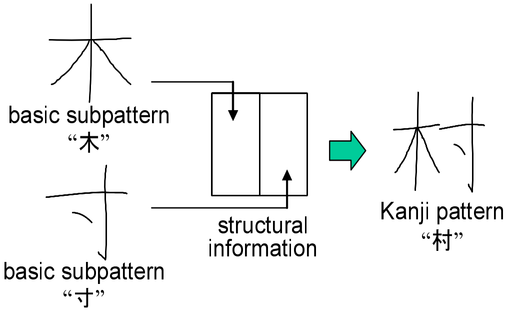


Fig. 1. SCPR

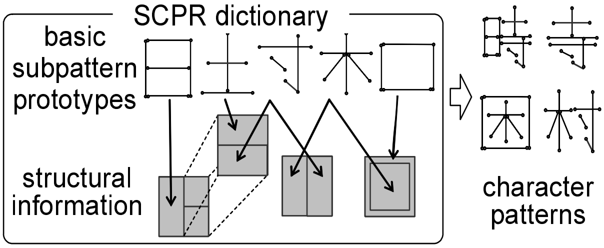


Fig. 2. SCPR dictionary

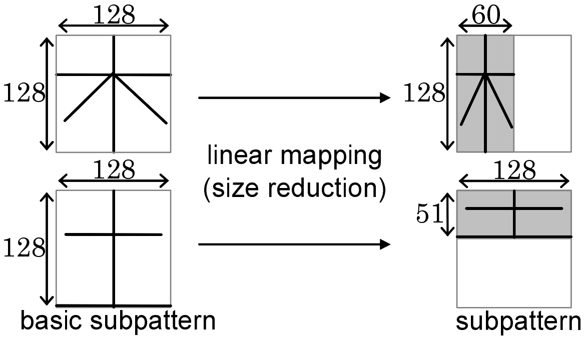


Fig. 3. Size reduction by linear mapping

resolution, and each of them is a sequence of feature points in a time series. When included in prototypes of larger subpattern prototypes or character pattern prototypes, their sizes reduce to bounding boxes in structural information through linear mapping (Figure 3). We call a result of the linear mapping a "mapped BS prototype," even if the mapping is sometimes identical (with no deformation). Hereafter, we refer to this as an MBS prototype.

The SCPR dictionary of Sys_HMM has similar structural information and links to component SCPR-based HMMs in place of BS prototypes.

SCPR provides advantages to the size reduction of the dictionary (a set of prototype patterns) and the robustness against deformation of common radicals.

2.2 Prototype Learning Algorithm (PLA)

No matter the classification method, prototypes greatly influence the performance of classifiers. PLA better approximates discrimination boundaries between different categories in a feature space [22]-[25]. Liu, et al. have shown the advantages of PLA in offline handwritten Kanji character recognition [26].

For online handwriting recognition systems in which prototypes are the sequences of feature points (e.g. Sys_LTM), we have proposed a PLA to improve the BS prototypes by moving their feature points [27]. The base learning method of our PLA is the generalized learning vector quantization (GLVQ) [25]. GLVQ updates the genuine prototype (the closest prototype in the correct class) and the rival prototype (the closest one in different classes) using learning patterns.

Recognition by LTM and PLA. Our PLA uses the correspondences between feature points that are the results of LTM in the process of recognition. In Figure 4, the dash lines show the correspondences. Although general elastic matching methods commonly generate one-to-many or many-to-one correspondences between the feature points, our method of Sys_LTM generates only one-to-one correspondences by discarding the uncertain correspondences.

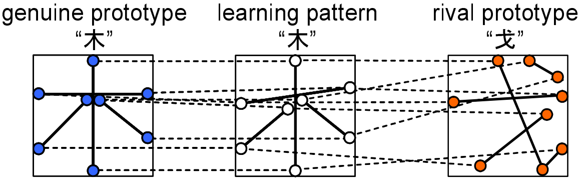


Fig. 4. Correspondences between feature points

SCPR-Based PLA. To improve the prototypes in the SCPR dictionary, we consider each prototype matched with the learning pattern as a composite of MBS prototypes. Figure 5 shows the process of our PLA, improving the feature point v in the BS prototype. Each $u(v)$ is a feature point of the MBS prototype mapped from the feature point v, and each p_l is a feature point in the learning pattern corresponding to $u(v)$. S is the bounding box size of each MBS prototype, and $G(p_l - u(v), S)$ is the function to normalize the displacement $p_l - u(v)$ by the bounding box size. Every displacement between $u(v)$ and p_l is measured and reflected in the feature point v using $G(p_l - u(v), S)$.

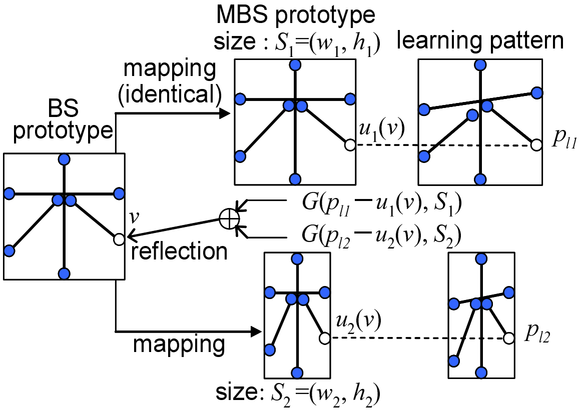


Fig. 5. Displacement reflection

We transform the formulae of GLVQ into (1) and (2) to update the feature points of the BS prototypes.

$$\begin{cases} x_i' = x_i + 4\alpha(t) l_k (1-l_k) \dfrac{G(d_j, S_j) G(x_l - x_i, w_i)}{\{G(d_i, S_i) + G(d_j, S_j)\}^2} \\ \\ y_i' = y_i + 4\alpha(t) l_k (1-l_k) \dfrac{G(d_j, S_j) G(y_l - y_i, h_i)}{\{G(d_i, S_i) + G(d_j, S_j)\}^2} \end{cases} \tag{1}$$

$$\begin{cases} x_j' = x_j - 4\alpha(t) l_k (1-l_k) \dfrac{G(d_i, S_i) G(x_l - x_j, w_j)}{\{G(d_i, S_i) + G(d_j, S_j)\}^2} \\ \\ y_j' = y_j - 4\alpha(t) l_k (1-l_k) \dfrac{G(d_i, S_i) G(y_l - y_j, h_j)}{\{G(d_i, S_i) + G(d_j, S_j)\}^2} \end{cases} \tag{2}$$

In the above formulae, (1) and (2), $u(v_i) = (x_{ui}, y_{ui})$ in the genuine prototype maps from $v_i = (x_i, y_i)$ of one BS prototype, while $u(v_j) = (x_{uj}, y_{uj})$ in the rival prototype maps from $v_j = (x_j, y_j)$ of the other BS prototypes. The feature point $p_l = (x_l, y_l)$ in the learning pattern corresponds to $u(v_i)$ and $u(v_j)$. The term $\alpha(t)$ denotes the learning rate. The other parameters l_k, μ_k, d_i, and d_j are defined as follows:

$$l_k = l_k(\mu_k) = \frac{1}{1+e^{-\zeta\mu_k}} \tag{3}$$

$$\mu_k = \frac{G(d_i, S_i) - G(d_j, S_j)}{G(d_i, S_i) + G(d_j, S_j)} \tag{4}$$

$$d_i = \|p_l - p_i\| \tag{5}$$

$$d_j = \|p_l - p_j\| \tag{6}$$

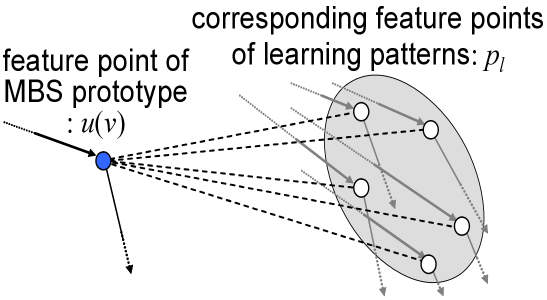


Fig. 6. Distribution of feature points

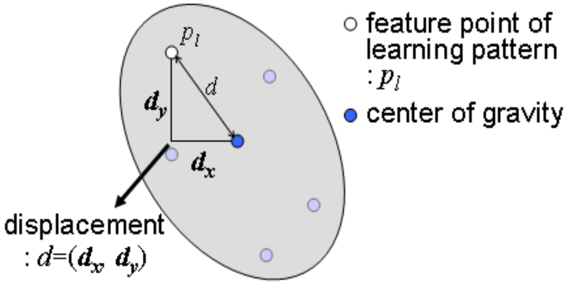


Fig. 7. Distance between the center of gravity and feature points

We obtained the distribution of the feature points in learning patterns corresponding to each feature point in MBS prototypes, as the size of the distribution (the average distance from the center of the distribution to every $l(v)$ in the distribution) shows to what degree each MBS point can move (Figure 6).

To describe the size of distribution, we employed the average distance from the center of the distribution to every $l(v)$ in the distribution (Figure 7).

We employed our database, Nakayosi_t to obtain the distribution. By using the database, we estimated the relations between the bounding box size of MBS prototype $S = (w, h)$ and the distribution size D for vertical and horizontal directions as follows:

$$D_w(w) = 0.0846w + 1.7 \tag{7}$$

$$D_h(h) = 0.0539h + 3.5 \tag{8}$$

The normalization formulae using the relation are as follows.

$$G_w(x_1 - x_2, w) = (x_1 - x_2)\{D_w(128)/D_w(w)\} \tag{9}$$

$$G_h(y_1 - y_2, h) = (y_1 - y_2)\{D_h(128)/D_h(h)\} \tag{10}$$

Evaluation for SCPR-Based PLA. In the experiment, we improved the SCPR dictionary of Sys_LTM to evaluate our PLA. As the set of learning patterns, we employed all the character patterns of the JIS 1*st* level set in Nakayosi_t. We employed another database, Kuchibue_d, for evaluation. In the first step, we generated the averaged BS prototypes by learning patterns. Then, we performed our PLA. Before learning, the recognition rate of Sys_LTM with the dictionary was approximately 84.4% for the data set of evaluation. After learning, the rate improved to 89.1%.

SCPR-Based PLA for Offline Recognition. Now, we describe SCPR and the learning method for offline recognition. In general, offline recognition methods, directional features with four or eight-directional quantization, are extracted from a character pattern divided into an array of cells [20][28]. Therefore, a character (same as a prototype) is represented by a matrix of directional features. In Figure 8, $f_v(i,j)$ is the set of directional features extracted from the cell of the $i-th$ row and $j-th$ column in the character pattern.

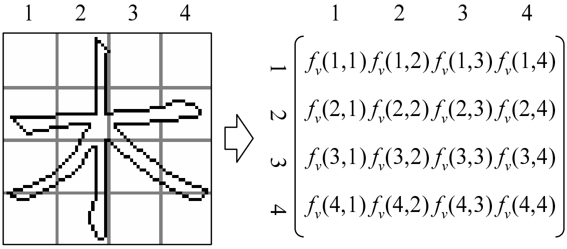

1 2 3 4
1
2
3
4
1 2 3 4
1
2
3
4
$f_v(1,1)\,f_v(1,2)\,f_v(1,3)\,f_v(1,4)$
$f_v(2,1)\,f_v(2,2)\,f_v(2,3)\,f_v(2,4)$
$f_v(3,1)\,f_v(3,2)\,f_v(3,3)\,f_v(3,4)$
$f_v(4,1)\,f_v(4,2)\,f_v(4,3)\,f_v(4,4)$

Fig. 8. Matrix of the directional features extracted from the character pattern

When the character C is composed from a set of subpatterns, we can show it as follows:

$$C = \sum_{i=1}^{N_s} A_i S_i \tag{11}$$

(N_s: number of subpatterns, A_i: linear mapping to make a subpattern, S_i: matrix of fv to describe a basic subpattern)

Subpatterns don't overlap each other in Japanese characters, so

$$F_i = A_i^{-1} C \tag{12}$$

Linear mapping transforms the directional features in a basic subpattern. Therefore, when a character pattern is learned and reflected to a subpattern, the directional features again map through inverse mapping with the directions modified. To enable mapping and inverse mapping that may change the directions of segments, the directional features must be extracted and represented finely enough.

For offline character recognition, the dimensions of features often reduce by the K-L transformation:

$$y = \psi^t x \tag{13}$$

In this case, the reduced set of features is difficult to decompose into subpatterns. However, if the reduction is small and ψ is chosen to minimize the mean square error, the original features can be approximated by

$$x \approx \psi y \tag{14}$$

This enables decomposition into subpatterns and reflection to their features. Then, we can realize structural learning for common offline character recognition methods.

2.3 SCPR-Based HMM

The HMM has an ability to model deformations of strokes and variations of sampled feature points, and has been successfully applied to western handwriting recognition [29][30]. For western characters, each character of the alphabet is typically modeled by one HMM and all the words are represented by a sequence of character HMMs. We call this approach "character HMM." However, thousands of characters comprise Asian characters of Chinese origin, so the character HMM leads to a huge amount of memory space and training data [15][32][33][35]. To address this problem, the SCPR-based HMM [31][34] has been proposed.

Recognition by SCPR-Based HMM. The SCPR-based online handwriting recognition system (Sys_HMM) consists of a feature extraction module, a SCPR dictionary mentioned in Section 2.1, SCPR-based HMMs, and a decoder shown in Figure 9. Note that off-stroke information (vector from the pen-up to the next pen-down) is not necessary when we model pen-coordinate features.

In the SCPR-based HMMs, the decoder generates HMMs for each character pattern by connecting one, or more than one, SCPR-based HMMs according to the SCPR dictionary. It then calculates the probability that an input pattern is produced from the HMMs by the Viterbi algorithm. By doing this, we can handle a large number of character patterns with a small number of HMMs.

Modeling of Pen-coordinate Features by SCPR-Based HMM. Though the pen-coordinate feature is no less important than the pen-direction feature, it has not been employed in the SCPR-based HMMs [31][34] because it tends to change when subpatterns are composed into each character pattern. We proposed SCPR-based HMMs that model both the pen-direction and pen-coordinate features.

The basic idea of our approach involves the linear mapping and its inverse mapping, presented in Section 2.1. The BS prototypes reduce to bounding boxes in structural information through a linear mapping when they are included in larger subpatterns or character patterns (Figure 10). Therefore, when a SCPR-based HMM is incorporated into a character pattern by being mapped into the bounding box, we adapt the parameters for pen-coordinate feature according to which character pattern and where incorporated. If each state of SCPR-based HMMs has a Gaussian distribution, the mean vector of the Gaussian distribution, at a state of

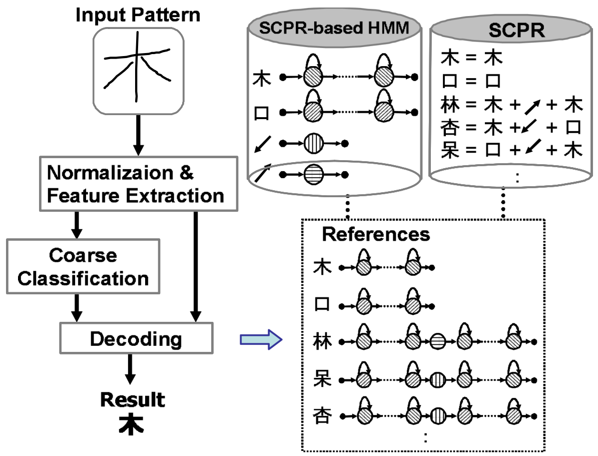


Fig. 9. Recognition by SCPR-based HMM

Character HMM: $\boldsymbol{\mu_i} = (\mu_{ix}, \mu_{iy})$, is adapted by the following equations:

$$\hat{\mu_{ix}} = \mu_{ix} \times \frac{w}{128} + sp_x \qquad (15)$$

$$\hat{\mu_{iy}} = \mu_{iy} \times \frac{h}{128} + sp_y \qquad (16)$$

where μ_{ix} and μ_{iy} denote the mean vectors of the pen-coordinate feature x and y, and (sp_x, sp_y) denotes the top-left corner and $< w, h >$ denotes $< width, height >$ of the bounding box.

In contrast, we apply the inverse of the above mapping when estimating SCPR-based HMMs. A simple idea is to enlarge the size of the bounding box of a mapped basic subpattern in a learning pattern to the normalization size. By applying the inverse mapping, we can exclude character dependency of each subpattern (difference in size and position when it appears in different character patterns) to model pen-coordinate features of the subpattern by SCPR-based HMMs. However, as handwriting usually contains noise due to hand vibration, etc., the inverse mapping may magnify these noises and reflect them into the subpattern. To avoid noise expansion, we employ the displacement normalization mentioned in Section 2.2, instead of inverse mapping each subpattern.

Evaluation for SCPR-Based HMMs. We use only Kanji categories in the JIS 1*st* level set in Kuchibue_d. Patterns from 60 writers were used for training, and those from the remaining 60 writers were used for testing.

First, we evaluated the effect of positional features and directional features upon the latter alone. Table 1 demonstrates the importance of both the directional and positional features. They have increased more than eight points, from 83.6 to 92.3%.

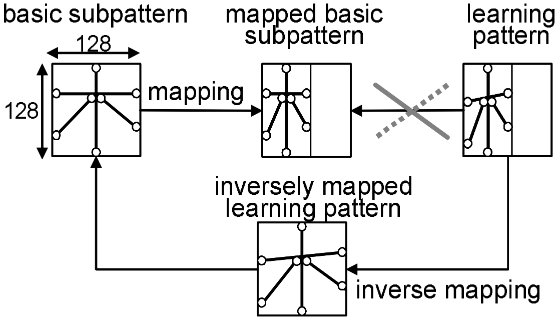


Fig. 10. Mapping and inverse mapping

Table 1. Effect of positional features and directional features

Features	N-best cumulative recognition rate (%)			
	1	-2	-3	-10
directional features	83.6	88.4	90.0	92.7
direct. f + positional f	92.3	95.1	95.9	96.9

Second, we compared the proposed SCPR-based HMMs and the conventional character HMMs with respect to memory size. The former requires 30.68 M bytes, and the latter needs 3.81 M bytes.

3 Classifier Combination

This section describes the combined recognizer, composed of the SCPR-based online recognizer and the offline recognizer. An online handwritten pattern easily converts to an offline pattern by discarding temporal information, so we can apply the offline method. Although the online method is more robust against stroke catenation, running strokes, and deformation of character patterns when compared to the offline method, the offline method is free from stroke-order variation and robust to duplicated strokes when people write two or more times. Therefore, by combining the online method with the offline method, the recognition accuracy improves because they can compensate for their disadvantages. Several attempts have been made to combine the two methods [36][37]. In Japanese character recognition, Tanaka, et al., showed the initial attempt to combine online and offline classifiers [38] while Okamoto, et al., showed the combination in the feature level, i.e., added online features to offline features in an offline recognition scheme [20]. It seems that classifier combination is more flexible than feature combination, as we can employ the most suitable classification method for each set of features.

In this paper, we show succeeding research after [38] to improve recognition accuracy, while increasing recognition speed and reducing memory size. The memory requirements for the offline prototype dictionary are significantly larger than that for the online one. Especially, the dictionary size depends on the number of categories, so the combined recognizer for Japanese characters is difficult for a small computer, such as a PDA. We also propose a compact combined recognizer, composed of the SCPR-based online recognizer and an offline recognizer with a small prototype dictionary.

3.1 Combination Process

To combine the online recognition and the offline recognition, a given online character pattern converts to a bitmap image, then online recognition and offline recognition are processed in parallel. We employ Sys_LTM as mentioned in Section 2.1, as the online recognition method and Modified Quadratic Discriminant Function (MQDF2) [39] as the offline recognition method.

Combination Rule. Various possibilities can combine outputs from multiple classifiers. Kittler, et al. present many combination schemes, such product rule, sum rule, min rule, max rule, median rule, and majority voting [40]. We employ the sum-rule, in which the total score of a combined classifier is the addition of all classifiers. The sum rule is denoted as follows:

$$\begin{gathered} AssignX \rightarrow C_j \ \text{ if} \\ \sum_{i=1}^{R} P(C_j|fv_i) = \max_{k=1}^{N_{cc}} \sum_{i=1}^{R} P(C_k|fv_i) \end{gathered} \tag{17}$$

where N_{cc} denotes the number of character categories, R denotes the number of classifiers, f_v denotes a feature vector extracted from the input pattern X, C_k denotes the prototype and $P(C_k|fv_i)$ denotes the probability that C_k occurs when fv_i is given.

Evaluation Score Normalization. Recognition results, produced by each recognizer, are pairs of a candidate character and an evaluation score that represents similarity or distance. However, each recognizer outputs a different type of evaluation score. The evaluation score of our online recognizer shows similarity. The higher the score, the more likely the candidate is. However, our offline recognizer employs distance. The lower the score, the higher the likelihood is. To combine these recognizers, we apply our likelihood normalization approach [41][42].

3.2 Small Offline Prototype Dictionary

MQDF2 for the offline recognition is given as:

$$\begin{aligned} & g_2(x, \omega_i) \\ & = \sum_{j=1}^{m} \frac{1}{\lambda_{ij}} [\varphi_{ij}^T (x - \mu_i)]^2 \\ & + \frac{1}{\delta} \{ \|x - \mu_i\|^2 - \sum_{j=1}^{m} [\varphi_{ij}^T (x - \mu_i)]^2 \} \\ & + \sum_{j=1}^{m} \log \lambda_{ij} + (n - m) \log \delta \end{aligned} \tag{18}$$

Table 2. Recognition rates of the combined recognizer (%)

Online	Offline		Combined	
	9.7MB	91.8MB	9.7MB	91.8MB
87.2	83.2	86.9	91.4	92.2

where μ is the mean vector, φ is the eigen vector, λ is eigen value, and δ is a modified eigen vector, n is the number of dimension, and m is the number of λ.

While the SCPR dictionary of our online recognizer is only 150KB, the prototype dictionary of the offline recognizer is about 90MB. To reduce the total size of memory, we propose a small prototype dictionary for the offline recognizer by reducing the parameters for MQDF2.

The size of the offline prototype dictionary: S_i is calculated from the size of each parameter $\{s_\mu, s_\varphi, s_\lambda, s_\delta\}$ where s_x denotes the size for the parameter x as follows.

$$S_i = N_{cc} \times \{n \times (s_\mu + m \times s_\varphi) + m \times s_\lambda + s_\delta\} \tag{19}$$

In this study, we make two extreme sizes of dictionaries. One is 9.7MB whose n is 100 and m is 10. The other is 91.8MB whose n is 256 and m is 40. Each parameter $\{s_\mu, s_\varphi, s_\lambda, s_\delta\}$ requires 16 bits in both the dictionaries. The size of the SCPR dictionary in the online recognizer is 150KB, so the dictionary size of the combined recognizer is almost the same as in the offline recognizer. Hereafter, we call the combined recognizer which employs the 9.7MB dictionary "Sys_9.7MB" and the 91.8MB dictionary "Sys_91.8MB".

3.3 Evaluation of the Combined Recognizer

We trained the online recognizer using Nakayosi_t and the offline recognizer using ETL9B [43] written by 200 participants, each composed of 3,036 character patterns; JEITA-HP [44] written by 580 participants, each composed of 3,306 character patterns; NTT-AT [45] written by 51 participants, each composed of 1,237 character patterns; and Nakayosi_t. We also used Nakayosi_t for normalizing evaluation scores.

The recognition rates appear in Table 2. These rates show no significant difference between the Sys_9.7MB and the Sys_91.8MB, though the correct recognition accuracy of the offline recognizer with 91.8MB dictionary is higher than the offline recognizer with Sys_9.7MB by 3.7 points.

We also compare the processing time of each recognizer on a Pentium IV 3.06 GHz processor with 512MB RAM (Table 3). The Sys_9.7MB outperforms the Sys_91.8MB.

Table 3. Average processing time per character (ms)

Online	Offline		Combined	
	9.7MB	91.8MB	9.7MB	91.8MB
3.32	6.6	18.5	10.6	22.5

Table 4. Recognition rates of the combined recognizer employing bi-gram model (%)

Online	Offline		Combined	
	9.7MB	91.8MB	9.7MB	91.8MB
91.2	97.0 %	97.7	98.6	98.6

Combined Recognizer with Context Postprocessing. We show the recognition accuracy of our combined recognizer with context postprocessing. In this experiment, a character bi-gram model performs as a simple stochastic language model. Given a sequence of character patterns $X = X_1 X_2 \ldots X_i \ldots X_N$, The problem is to find the character string $C = C_1 C_2 \ldots C_i \ldots C_N$ to maximize the probability $P(C|X)$. Using the Bayes rule:

$$P(C|X) = \frac{P(C) \cdot P(X|C)}{P(X)} \tag{20}$$

The term $P(X|C)$ shows the probability of C written as X. The term $P(C)$ shows the context likelihood.

Following the bi-gram model, the probability $P(C)$ is given as:

$$P(C) = P(C_1) \prod_{i=1}^{N-1} P(C_{i+1}|C_i) \tag{21}$$

where N denotes the number of character patterns in a text string, and C_i denotes each character pattern. The uni-gram probability $P(C_1)$ is assumed to be independent from characters. In our study, the character bi-gram language model was trained with the ASAHI newspaper text corpus “CD-HIASK’93”.

The recognition results in Table 4 show that the rate of Sys_9.7MB equals the rate of Sys_91.8MB and raises to 98.6%.

4 Online Handwritten Japanese Text Recognition

According to the increasing size of writing surface of pen input devices, demand for online handwritten text recognition is growing. Due to the difference between Japanese and western languages and handwriting, handwritten recognition differs naturally. In this section, we describe some problems of handwritten Japanese text recognition and approaches to tackle those problems.

4.1 Problems

Generally, on large writing surfaces Asian languages of Chinese origin are often written horizontally, vertically, or even diagonally.

Most previous publications and systems assumed only horizontal lines of text [46][47], while we attempted to relinquish any writing constraint from online text input. We proposed a method to recognize mixtures of horizontal, vertical, and slanted lines of text with assuming normal character orientation [48]. Then, we attempted handwriting recognition with characters rotated like in handwritings typical of whiteboards [49].

As mentioned before, Japanese text includes various sizes of character patterns ranging from so-called “half-width” characters like numbers and symbols, Kana characters and Kanji characters of only one radical in the middle, to those consisting of multiple radicals. Moreover, handwriting magnifies even the size variations, as shown in Figure 11. Some characters may be several times longer and/or wider than others.

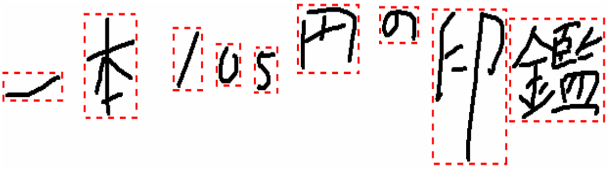


Fig. 11. An example of Japanese handwritten text

Many Japanese characters can be divided into multiple character patterns. For example, the patterns shown in Figure 12(a) can be read as either C_1, a character in itself, or as the two consecutive characters C_2C_3. The correct one is determined by the characters (or strings) preceding and/or following it. In the example of Figure 12(b), the character C_4 follows, which causes the pattern of Figure 12(a) to be read as C_1. In Figure 12(c), the characters C_5C_6 follow, which causes the pattern to be read as two characters C_2C_3. This example shows how the position of character segmentation can differ even for the same handwritten pattern, depending on the context, so it is difficult to segment characters deterministically on the basis of geometrical features alone.

In the next section, we present an enhanced method to recognize online handwriting of arbitrary line directions and character orientations as well as their mixtures.

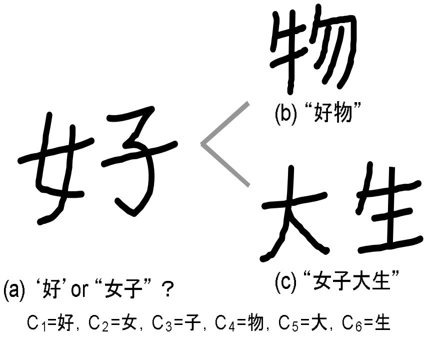


Fig. 12. An example of segmentation ambiguity

4.2 Flow of Processing

We first need to define terminology. Character orientation specifies the direction of a character from its top to bottom, while line direction designates the writing direction of a sequence of characters until it changes (Figure 13). A text line is a piece of text separated by new-line or 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 (Figure 14). The line direction and the character orientation are independent.

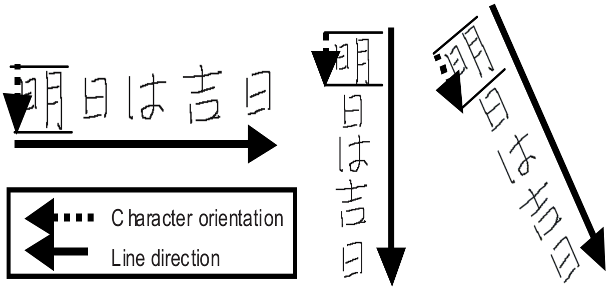


Fig. 13. Line direction and character orientation

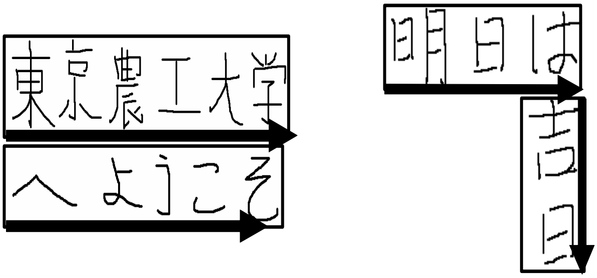


Fig. 14. Text line element and line direction

In the following subsections, we explain the procedure of our online recognition system of handwritten Japanese text, free from character orientation, line direction, and any writing format constraint.

Separation of Handwriting into Text Line Elements. First, we estimate the average character size from all the strokes written on a tablet by measuring the length of the longer side of the bounding box for each stroke, sorting the lengths from all the strokes, and taking the average of the larger 1/3 of them. The estimated average character size decides the threshold for separating written text into text line elements.

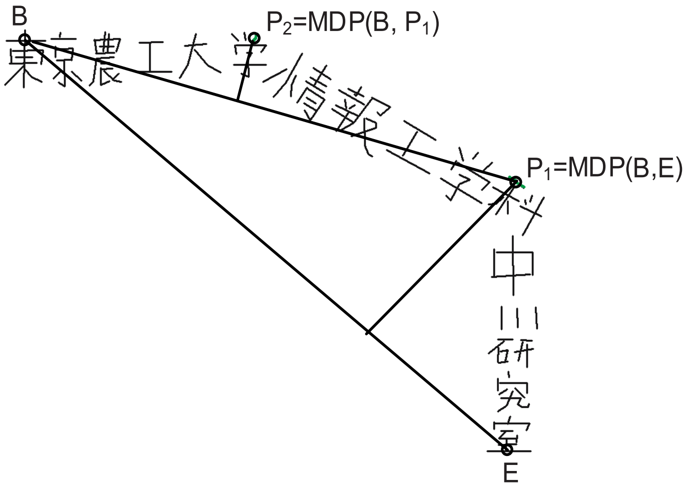


Fig. 15. Detection of directional changing points

Next, we separate freely written text into text lines by a large off-stroke from a previous line to a new line. Then, we separate each text line into text line elements by the changing points of line direction.

To detect the changing points of line direction, we employ a recursive procedure similar to that used to detect corner points [50]. Among a series of coordinates of the centers for the bounding boxes of strokes forming a handwritten text line, it finds the most distant point (MDP) from the straight line connecting the starting, and ending points of the series of coordinates. If the distance is larger than the threshold, then we apply the same procedure to the straight line from the starting point to the MDP, and from the MDP to the ending point, with the result of detecting multiple points of directional change, as shown in Figure 15, with B as the beginning point and E as the ending point. Thus a text line segmented by large space is further segmented into text line elements having different line directions.

Here, it is worth noting that points detected might be within character patterns rather than between characters, as P_1 and P_2 in Figure 15. We will address the problem in a later section, determining the best segmentation points while recognizing handwritten text.

Estimation and Normalization of Character Orientation. When Japanese characters are written, principal pen movement within real strokes remains the same as the character orientation. This happens because Japanese characters, especially Kanji, are composed of downward and rightward strokes. Therefore, if we take the histogram of displacement direction of pen-tip coordinates, we see two peaks, as shown in Figure 16.

Let us assume the intensity of the histogram at the angle θ as $f(\theta)$. Then, take the θ that makes the max value of $f(\theta)\times f(\theta+\pi/2)$ as the character orientation

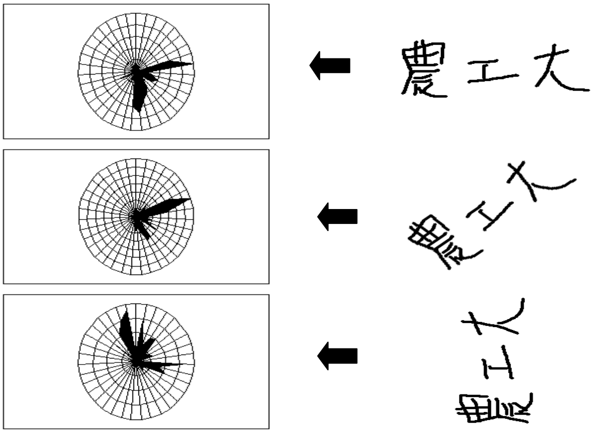


Fig. 16. Two peaks in pen movement direction

for each text line element. To make the peak detection more robust, we take the convolution of $f(\theta)$ and a Gauss function to blur the peak, so it works for slanting characters with rightward strokes and a slightly upward inclination.

After estimating the character orientation, the text line element can be normalized by rotating it.

Hypothetical Segmentation for Each Text Line Element. We hypothetically segment a text line element, after the character orientation normalization, into character patterns using geometric features.

Hypothetical segmentation depends on character orientation and line direction. After normalizing character orientation, it depends on line direction of a text line element, as shown in Figure 17. Note that a segmentation hypothesis often occurs within character patterns, and it differs even for the same character pattern depending on the line direction. The quantization can be finer, but the 4-directional quantization (shown in Figure 17) is adequate and effective to prevent a text line element from being segmented excessively. When the line direction is classified, downward or upward (rightward or leftward), a considerable gap projected on the vertical axis (horizontal axis) or a long off-stroke to the quantized line direction are employed as candidates for segmentation. Strokes or off-strokes to the opposite direction merge their crossing strokes with the result that hypotheses on segmentation can be decreased, which is then effective to accelerate the text recognition and increase the recognition rate.

After the quantization of line direction, we extract multi-dimensional features, such as distance and overlap between adjacent strokes, from each off-stroke and apply the SVM to the extracted features to produce segmentation point candidates [51]. Character size may vary among text line elements, so we estimate the character size again for every text line element. Then, we normalize the extracted multi-dimensional features by the re-estimated character size for each text line element. Each off-stroke is classified into segmentation point, non-segmentation

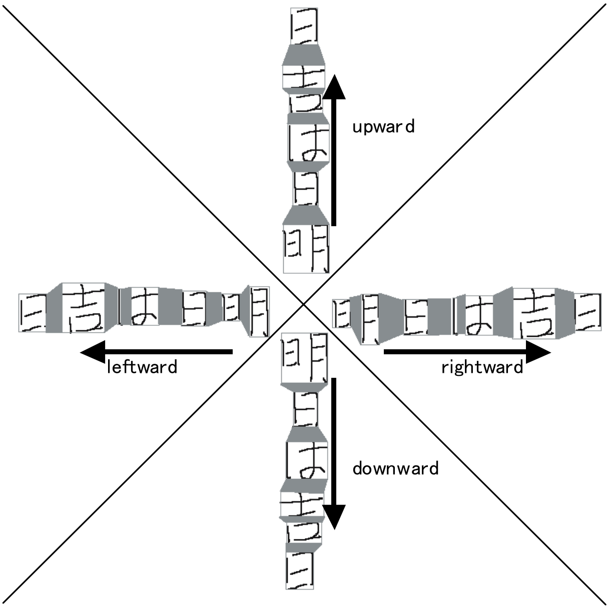


Fig. 17. Quantization of line direction

point, and undecided point, according to the features. A segmentation point should be between two characters, and a non-segmentation point appears within a character pattern. An undecided point is one which cannot be judged as segmented or non-segmented. A segmentation unit bounded by two adjacent segmentation points is assumed to be a character pattern. An undecided point is treated in two ways: as a segmentation point or a non-segmentation point. When treated as a segmentation point, it extracts a segmentation unit.

Construction of Candidate Lattice. A candidate lattice is constructed for each text line, where each arc denotes a segmentation point and each node denotes a character recognition candidate produced by character recognition for each segmentation unit, as shown in Figure 18.

Scores associate with each arc or node following the stochastic model of evaluating the likelihood composed of character segmentation, character recognition, character pattern structure, and context.

The Viterbi search is implemented into the candidate lattice for a handwritten text line and the best segmentation and recognition is determined.

Segmentation of a text line into text line elements by an MDP should not be decisive. Wrong segmentation, within a character pattern, into two text line elements and rotation of the segmented text line elements to normalize character orientation may damage their recognition, as shown in Figure 19. To avoid this problem, we produce multiple alternatives of text line segmentation by choosing the segmentation point among candidates around the MDP. The range of the

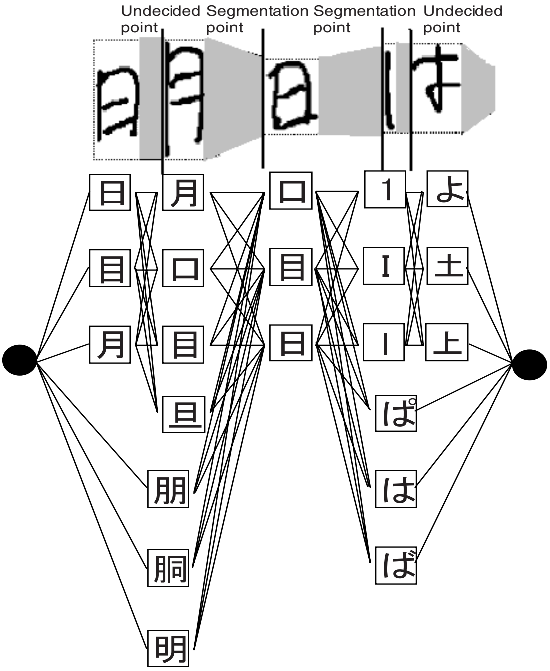
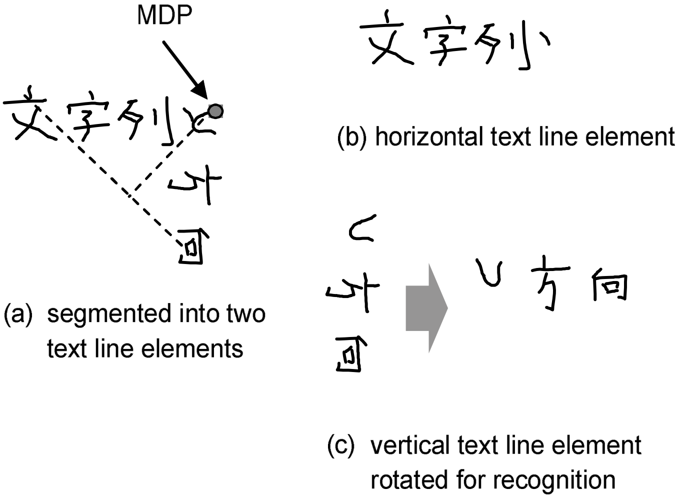


Fig. 18. Candidate lattice



(a) segmented into two text line elements

(b) horizontal text line element

(c) vertical text line element rotated for recognition

Fig. 19. Problem of segmentation by an MDP

segmentation point's perturbation can be confined within the average character size before or after the MDP. According to each alternative segmentation point, the two text line elements at both sides of the segmentation point are rotated with character orientations normalized when necessary and recognized. Then, the Viterbi search chooses the best segmentation point.

Model of Free Format Recognition. We made a model and recently formalized it for online handwritten Japanese text recognition free from line-direction constraint and writing format constraint, such as character writing boxes or ruled lines [42][52]. The model evaluates the probability of character segmentation, character recognition, character pattern structure, and context. The likelihood of character pattern structure considers the plausible height, width, and inner gaps within a character pattern that appears in Chinese characters composed of multiple subpatterns.

The problem involves finding the character string $C = C_1C_2...C_i...C_N$ to maximize the likelihood $L(C|X)$ that a handwritten text line pattern X is recognized as the character string C. After several steps of approximations and modifications, we arrive at the following formula:

$$\begin{aligned} &L(C|X) \\ &= \log P(C_1) + \sum_{i=1}^{N-1} \log P(C_{i+1}|C_i) \\ &+ \sum_{i=1}^{N} (\log P(X_i|St_i, C_i) + \log P(St_i/\bar{C}|C_i)) \\ &+ \sum_{i=1}^{N-1} (\log P(gap_i/\bar{C}|C_i, C_{i+1})) \end{aligned} \tag{22}$$

where,

N : number of characters in C.
$P(C_{i+1}|C_i)$: probability that a character C_{i+1} follows C_i (bi-gram probability).
$P(X_i|St_i, C_i)$: probability that a character C_i is written in a structure St_i and represented by the stroke sequence X_i.
$\bar{C}$: average size of the character sequence C.
$P(St_i/\bar{C}|C_i)$: probability that a character C_i is written in a structure St_i.
$P(gap_i/\bar{C}|C_i, C_{i+1})$: probability that an outer gap : gap_i appears between C_i and C_{i+1}.

In the right-hand side of the above equation, the second term considers context likelihood in terms of a bi-gram, the third term relats to character recognition likelihood, the fourth and fifth terms evaluate character pattern structure likelihood and outer gap likelihood, respectively.

5 Application

In this section, we show two types of online handwriting applications: segmentation and recognition of mixed text, formulas, tables, and line-drawings; and handwritten text search.

5.1 Segmentation and Recognition of Mixed Objects

As the writing area of pen input devices grows, users can easily write text, mathematical formulae and figures on the screen. It is one of the most important benefits of pen interfaces that people can write these objects by a single pen without switching the device, mode, or software, and without any writing restriction such as grids or boxes. However, it requires the difficult task to separate online handwritten patterns into Japanese text, figures, and mathematical formulas. We approached this problem earlier [53]. Recently, we take a probabilistic approach to this problem and employed stroke features, stroke crossings, and stroke densities. Moreover, we partially applied segmentation by recognition. Although the current recognizer for formulae is only a prototype, we have achieved about 81% correct segmentation for all the strokes in Kondate_t, the newly prepared database of mixed patterns [54]. Figure 20 shows an example of separating mixed objects. Our new approach is generally better but less effective in distinguishing figures from other components.

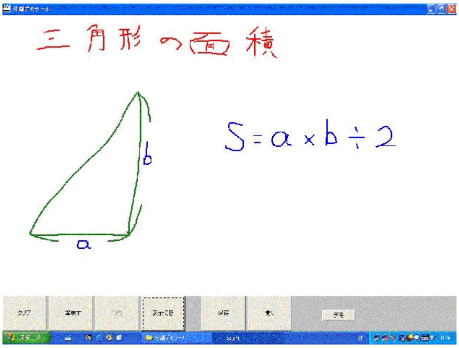
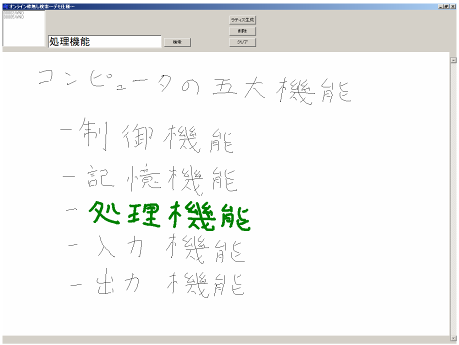


Fig. 20. Separation of handwriting into text, formula and line drawing

5.2 Online Handwritten Text Search

As various pen input devices become popular, online handwritten text will be accumulated. Without a search method, however, accumulated online handwritten text can not be utilized effectively. Search of online handwritten text by employing a pattern matching method without character recognition was reported in [55]. Lopresti, et al. proposed a stroke search method, "Script Search Algorithm," which searches a long handwritten text pattern and finds approximate patterns of a keyword [56]. Also, we have proposed a method for writing-box-free online Japanese handwritten text search, based on the online Japanese handwritten text recognition, mentioned in Section 4 [57]. It searches for a target keyword in the candidate lattice composed of candidate segmentations and candidate characters, as shown in Figure 18, which has been generated beforehand by the background process of online handwriting recognition. Figure 21 shows an example of online handwritten text search. As the performance of recognizer improves, the performance of search upgrades.



処理機能

コンピュータの五大機能

-制御機能

-記憶機能

-処理機能

-入力機能

-出力機能

Fig. 21. Searching a keyword from free format handwritten text

6 Conclusion

We presented online handwriting recognition of Japanese characters. Most Kanji character patterns are composed of multiple subpatterns called radicals, and these subpatterns are shared among many character patterns, so we have employed structured character pattern representation (SCPR). SCPR is effective in terms of the size reduction of the prototype dictionary and the robustness to deformation of common subpatterns. Then, we described a prototype learning algorithm and HMM-based recognition for SCPR. We combined the SCPR-based online recognizer with a compact offline recognizer. Moreover, we also presented online handwritten Japanese text recognition free from character orientation and line direction constraints. Finally, we showed some applications of online handwritten Japanese text recognition.

Acknowledgments

This research is being partially supported by Grant-in-Aid for Scientific Research under the contract number (B) 17300031 and the MEXT fund for Promoting Research on Symbiotic Information Technology.

References

1. http://www.anoto.com
2. http://www.e-pen.com
3. Tappert, C.C., et al.: The State of the Art in On-Line Handwriting Recognition. IEEE, Trans. PAMI 12(8), 787–808 (1990)
4. Nakagawa, M.: Non-Keyboard Input of Japanese Text - On-line Recognition of Handwritten Characters as the Most Hopeful Approach. J. Information Processing 13(1), 15–34 (1990)
5. Plamondon, R., et al.: On-Line and Off-Line Handwriting Recognition: a Comprehensive Survey. IEEE Trans. PAMI 22(1), 63–82 (2000)
6. Jaeger, S., et al.: The State of the Art in Japanese On-line Handwriting Recognition Compared to Techniques in Western Handwriting Recognition. IJDAR 6(2), 75–88 (2003)
7. Liu, C.L., et al.: On-Line Recognition of Chinese Characters: the State of the Art. IEEE Trans. PAMI 26(2), 198–213 (2004)
8. Nakagawa, M., et al.: On-line Handwritten Character Pattern Database Sampled in a Sequence of Sentences without any Writing Instructions. In: Proc. 4th ICDAR, pp. 376–381 (1997)
9. Nakagawa, M., et al.: Collection of On-line Handwritten Japanese Character Pattern Databases and Their Analysis. IJDAR 7(1), 69–81 (2004)
10. Shin, J., et al.: Stroke Correspondence Search Method for Stroke-Order and Stroke-Number Free On-Line Character Recognition - Multilayer Cube Search (in Japanese). IEICE Trans J82-D-II(2), 230–239 (1999)
11. Kobayashi, M., et al.: RAV (Reparameterized Angular Variations) Algorithms for Online Handwriting Recognition. IJDAR 3(3), 181–191 (2001)
12. Nakagawa, M., et al.: Robust and Highly Customizable Recognition of On-line Handwritten Japanese Characters. Proc. 13th ICPR 3, 269–273 (1996)
13. Nakagawa, M., et al.: A Linear-Time Elastic Matching for Stroke Number Free Recognition of On-Line Handwritten Characters. In: Proc. 4th IWFHR, pp. 48–56 (1994)
14. Yokota, T., et al.: An On-line Cuneiform Modeled Handwritten Japanese Character Recognition Method Free from Both the Number and Order of Character Strokes (in Japanese). Trans. IPSJ 44(3), 980–990 (2003)
15. Takahashi, K., et al.: A Fast HMM Algorithm for On-Line Handwritten Character Recognition. In: Proc.4th ICDAR, pp. 369–375 (1997)
16. Ishigaki, K., et al.: A Top-down On-line Handwritten Character Recognition Method via the Denotation of Variation. In: Proc. ICCPCOL, pp. 141–145 (1998)
17. Rammer, U., et al.: An Iterative Procedure for the Polygonal Approximation of Plane Closed Curves. CGIP 1, 244–256 (1972)
18. Kawamura, A., et al.: On-line Recognition of Freely Handwritten Japanese Characters Using Directional Feature Densities. Proc. 11th ICPR 2, 183–186 (1992)
19. Hamanaka, M., et al.: On-line Japanese Character Recognition Experiments by an Off-line Method. In: Proc. 2nd ICDAR, pp. 204–207 (1993)
20. Okamoto, M., et al.: On-line Handwritten Character Representation using Directional Features and Direction-Change Features (in Japanese). Journal of IEE Japan 119(3), 358–366 (1999)
21. Tokuno, J., et al.: Pen-Coordinate Information Modeling by SCPR-based HMM for On-line Japanese Handwriting Recognition. In: Proc. 18th ICPR. WED-P-II-1, vol. 3 (2006)

22. Kohonen, T., et al.: Improved Versions of Learning Vector Quantization. Proc. IJCNNI 1, 545–550 (1990)
23. Geva, S., et al.: Adaptive Nearest Neighbor Pattern Recognition. IEEE Trans. NN 2(2), 318–322 (1991)
24. Juang, B.-H., et al.: Discriminative Learning for Minimum Error Classification. IEEE Trans. SP 40(12), 3043–3054 (1992)
25. Sato, A., et al.: A Formulation of Learning Vector Quantization Using a New Misclassification Measure. In: Proc. 14th ICPR, pp. 322–325 (1998)
26. Liu, C.-L., et al.: Evaluation of Prototype Learning Algorithms for Nearest-neighbor Classifier in Application to Handwritten Character Recognition. Pattern Recognition 34, 601–615 (2001)
27. Kitadai, A., et al.: Prototype Learning for Structured Pattern Representation Applied to On-line Recognition of Handwritten Japanese Characters (Published Online). IJDAR, 1433–2825 (2007)
28. Liu, C.-L., et al.: Preprocessing and Statistical /Structural Feature Extraction for Handwritten Numeral Recognition Process of Handwriting Recognition, pp. 161–168. World Scientific, Singapore (1997)
29. Hu, J., et al.: HMM based On-line Handwriting Recognition. IEEE Trans. PAMI 18(10), 1039–1045 (1996)
30. Starner, T., et al.: On-Line Cursive Handwriting Recognition Using Speech Recognition Methods. Proc. ICASSP 5, 125–128 (1994)
31. Kim, H.-J., et al.: On-line Recognition of Handwritten Chinese Characters based on Hidden Markov Models. Pattern Recognition 30(9), 1489–1499 (1997)
32. Ito, H., et al.: An On-line Handwritten Character Recognition Method based on Hidden Markov Model (in Japanese). Technical Report of IEICE PRMU97-85, 95–100 (1997)
33. Okamoto, D., et al.: An HMM Implementation for On-line Handwriting Recognition Based on Pen-coordinate Feature and Pen-direction Feature. In: Proc. 8th ICDAR, pp. 26–30 (2005)
34. Nakai, M., et al.: Sub-stroke Approach to HMM-based On-line Kanji Handwriting Recognition. In: Proc. 6th ICDAR, pp. 491–495 (2001)
35. Cho, S.-J., et al.: Bayesian Network Modeling of Hangul Characters for On-line Handwriting Recognition. In: Proc. 7th ICDAR, pp. 207–211 (2003)
36. Manke, S., et al.: Combining Bitmaps with Dynamic Writing Information for On-line Handwriting Recognition. Proc. 13th ICPR, 596–598 (1994)
37. Vinciarelli, A., et al.: Combining Online and Offline Handwriting Recognition. In: Proc. 7th ICDAR, pp. 844–848 (2003)
38. Tanaka, H., et al.: Hybrid Pen-Input Character Recognition System Based on Integration of Online-Offline Recognition. In: Proc. 5th ICDAR, pp. 209–212 (1999)
39. Kimura, F., et al.: Modified Quadratic Discriminant Functions and the Application to Chinese Character Recognition. IEEE Trans. PAMI 9(1), 149–153 (1987)
40. Kittler, J., et al.: On Combining Classifiers. IEEE Trans. PAMI 20(3), 222–239 (1998)
41. Velek, O., et al.: A New Warping Technique for Normalizing Likelihood of Multiple Classifiers and its Effectiveness in Combined On-Line/Off-Line Japanese Character Recognition. In: Proc. 8th IWFHR, pp. 177–182 (2002)
42. Nakagawa, M., et al.: A Model of On-line Handwritten Japanese Text Recognition Free from Line Direction and Writing Format Constraints. IEICE Trans E88-D(8), 1815–182 (2005)
43. Saito, T., et al.: On the Data Base ETL9 of Handprinted Characters in JIS Chinese Characters and Its Analysis. Trans. IEICE J68-D(4), 757–764 (1985)

44. Kawatani, T., et al.: Handwritten Kanji Recognition with Determinant Normalized Quadratic Discriminant Function. Proc. 15th ICPR 2, 343–346 (2000)
45. http://www.ntt-at.com/products_e/jwords/
46. Fukushima, T., et al.: On-line Writing-box-free Recognition of Handwritten Japanese Text Considering Character Size Variations. Proc. 15th ICPR 2, 359–363 (2000)
47. Senda, S., et al.: A Maximum-Likelihood Approach to Segmentation-Based Recognition of Unconstrained Handwriting Text. In: Proc. of 6th ICDAR, pp. 184–188 (2001)
48. Inamura, Y., et al.: An On-line Writing-box-free and Writing-direction Free Recognition System for Handwritten Japanese Text (in Japanese). Technical Report of IEICE PRMU100-37, 17–24 (2000)
49. Nakagawa, M., et al.: On-line Handwritten Japanese Text Recognition Free from Constrains on Line Direction and Character Orientation. In: Proc. 7th ICDAR, pp. 519–523 (2003)
50. Senda, S., et al.: Box-free Online Character Recognition Integrating Confidence Values of Segmentation (in Japanese). Technical Report of IEICE PRMU98-138, 17–24 (1998)
51. Zhu, B., et al.: Segmentation of On-Line Handwritten Japanese Text Using SVM for Improving Text Recognition. In: Bunke, H., Spitz, A.L. (eds.) DAS 2006. LNCS, vol. 3872, pp. 208–219. Springer, Heidelberg (2006)
52. Nakagawa, M., et al.: A Formalization of On-line Handwritten Japanese Text Recognition Free from Line Direction Constraint. Proc. 17th ICPR 2, 359–362 (2004)
53. Machii, K., et al.: On-line Text/Drawings Segmentation of Handwritten Patterns. In: Proc. 2nd ICDAR, pp. 710–713 (1993)
54. Mochida, K., et al.: Separating Figures, Mathematical Formulas and Japanese Text from Free Handwriting in Mixed On-Line Documents. IJPRAI 18(7), 1173–1187 (2004)
55. Senda, S., et al.: MemoPad: Software with Functions of Box-free Japanese Character Recognition and Handwritten Query Search (in Japanese). Technical Report of IEICE PRMU99-75, 85–90 (1999)
56. Lopresti, D., et al.: On the Searchability of Electronic Ink. Proc. of 4th IWFHR, 156–165 (1994)
57. Oda, H., et al.: A Search Method for On-line Handwritten Text Employing Writing-box-free Handwriting Recognition. In: Proc. 9th IWFHR, pp. 545–550 (2004)