A Robust System for Online Handwritten Chinese/Japanese Character Recognition

B.L. Zhu & Masaki Nakagawa
Department of Computer and Information Science, Tokyo University of Agriculture and Technology, Tokyo, 184-0012 Japan

ABSTRACT: This paper presents a robust system for online handwritten Chinese/Japanese character recognition. Online handwritten character recognition, recognizing characters from their trajectories, will be used more widely, with the development and proliferation of pen-based or touch-based input devices such as tablet terminals, smart phones, electronic whiteboards and digital pens (e.g., Anoto pen). This paper focuses on the online handwritten Chinese/Japanese character recognition, and describes its recent technology trends, problems and methods to solve them.

KEYWORDS: Online handwriting recognition; Character recognition;

1 INSTRUCTIONS
Due to the development and proliferation of pen-based or touch-based input devices such as tablet terminals, smart phones, electronic whiteboards, and digital pens (e.g., the Anoto pen), on-line handwritten character recognition is receiving even more attention than before. Realizing online handwritten character recognition with high performance is vital, especially for applications such as the natural input of text on smart phones, to provide a satisfactory user experience.

Although character classifiers with high recognition accuracy have been reported [1], the demand for speeding up recognition is very high for portable devices as well as for desktop applications for which handwriting recognition is incorporated as one of the modules. The performance of these relatively small devices requires having a fast as possible recognition speed while maintaining high accuracy. Even for a desktop PC with relatively high device performance, recognition speed within 0.5 seconds per page and memory consumption within 10 MB are required in actual applications since those applications need more memory consumption on top of the memory consumption for a character recognizer.

Chinese, Japanese, or Korean has thousands of different categories, and their large character set is problematic not only for the recognition rate but also for the recognition speed.

Handwritten character pattern recognition methods are generally divided into two types: online recognition and offline recognition [2]. Online recognition recognizes character patterns captured from a pen-based or touch-based input device where trajectories of pen-tip or finger-tip movements are recorded, while offline recognition recognizes character patterns captured from a scanner or a camera device as two dimensional images.

Both online and offline recognition methods can be roughly divided into two categories: structural and un-structural. Structural methods are based on stroke analysis and use structural features such as sampling points, line segments and/or strokes for offline recognition [3] and for online recognition [4]. Un-structural methods use un-structural features such as directional features, gradient histogram features and projection features such as those for offline [5] and online recognition [6], which eventually achieves stroke-order independence.

Structural methods are weak at collecting global character pattern information, while they are robust against character shape variations. In contrast, un-structural methods are robust against noises but very weak against character shape variations. By combining a structural method (structural recognizer) with an un-structural method (un-structural recognizer), the recognition accuracy improves since they compensate for their respective disadvantages [7].

For online recognition, structural features are often employed with hidden Markov models (HMMs) [8] or Markov random field (MRF) [9]. However, since the un-structural features are easily extracted from an online handwritten pattern by discarding temporal and structural information, we can apply...
the un-structural method as well. Therefore, we can combine the structural and un-structural methods.

In freely written string recognition, we need to consider whether we should select segmentation-free or over-segmentation-based methods. Character segmentation of cursive handwriting is difficult due to the fact that spaces between characters are not obvious. Without character recognition cues and linguistic context, characters cannot be segmented unambiguously. A feasible way to overcome the ambiguity of segmentation is called integrated segmentation and recognition, which is classified into segmentation-free and over-segmentation-based methods [10] as shown in Figure 1. Segmentation-free methods, mostly combined with hidden Markov model (HMM)-based recognition [11], simply slice the word pattern into frames (primitive segments) and label the sliced frames, which are concatenated into characters during recognition. Such methods do not sufficiently incorporate character shape information. On the other hand, over-segmentation-based methods attempt to split character patterns at their true boundaries and label the split character patterns [12]. Character patterns may also be split within them, but they are merged later. This is called over-segmentation. It is usually accomplished in two steps: over-segmentation and path search. The string pattern is over-segmented into primitive segments such that each segment composes a single character or part of a character. The segments are combined to generate candidate character patterns (forming a segmentation candidate lattice), which are evaluated using character recognitions incorporating geometric and linguistic contexts. However, character segmentation is a hard problem to solve. When handwritten words are not easily segmented, such as with cursive writing, over-segmentation may result in misrecognition. In this case, a segmentation-free method is appropriate. For Chinese/Japanese recognition, however, segmentation is easier than in English or other western languages.

Segmentation-free offline word recognition methods using un-structural features in sliding windows combined with hidden Markov models (HMMs) [13] and those using over-segmentation-based methods [14] can also be considered as un-structural methods since they use un-structural features.

Since over-segmentation-based methods can better utilize character shapes, they are considered to outperform segmentation-free methods [15]. Moreover, over-segmentation-based methods produce less primitive segments since they attempt to find the true boundaries of character patterns as segmentation point candidates; therefore, we consider that over-segmentation-based methods are effective and more efficient compared with segmentation-free methods for Chinese/Japanese string recognition. We show our online handwritten Chinese/Japanese string recognition system in Figure 2, where an over-segmentation-based method is used.

In this chapter, we describe a robust system for online handwritten Chinese/Japanese character recognition, and the recent technology trends, problems and methods to solve them for the online handwritten Chinese/Japanese character recognition. The rest of this chapter is organized as follows. Section 2 presents an overview of our online handwritten string recognition system. Section 3 presents structural and un-structural recognitions. Section 4 describes coarse classification. Section 5 describes combination of structural and un-structural recognitions. Section 6 presents string recognition, and Section 7 draws conclusions.

2 RECOGNITION SYSTEM OVERVIEW

We process each online handwritten string pattern as follows:

(1) Segmentation candidate lattice construction.

Strokes in a string are grouped into blocks (primitive segments) in accordance with the features such as off-stroke (pen lift between two adjacent strokes) distance and overlap of bounding boxes of adjacent strokes. Each primitive segment is assumed to be a character or a part of a character. An off-stroke between adjacent blocks is called a candidate segmentation point, which can be a true segmentation point (SP) or a non-segmentation point (NSP). One or more consecutive primitive segments form a candidate character pattern. The combination of all candidate patterns is represented by a segmentation candidate lattice.
(2) Character pattern recognition.

There are thousands of categories for the Chinese/Japanese language. First, to improve the recognition speed, we reduce recognition candidates by using a coarse classifier for each online candidate character pattern. After coarse classification, we apply a structural recognizer and an un-structural recognizer to recognize the input pattern, and obtain two sets of character candidate classes from the two recognizers. Each candidate class of each set has a corresponding structural or un-structural recognition score. We combine the two sets of candidate classes considering their recognition scores to output a set of candidate classes to nominate them into the candidate lattice.

![Figure 3. Feature points extraction and labeling.](image)

(3) Search and recognition.

We apply the beam search strategy to search for the candidate lattice. During the search, the paths are evaluated in accordance with the path evaluation criterion proposed Zhu et al. [1], which combines the scores of character recognition, linguistic context, and geometric features (character pattern sizes, inner gaps, single-character positions, pair-character positions, candidate segmentation points) with the weighting parameters estimated by a genetic algorithm (GA). This method selects an optimal path as the recognition result.

3 STRUCTURAL AND UN-STRUCTURAL RECOGNITIONS

3.1 Structural Recognition

First, we include a brief description here on our structural character pattern recognition system [16]. Our system first normalizes an input online character pattern by a linear or nonlinear normalization method. An online character pattern is a sequence of strokes and a stroke is a time sequence of coordinates of pen-tip or finger-tip movements. Then, it extracts feature points by such a method as Ramner [17]. First, the start and end points of every stroke are picked up as feature points. Then, the most distant point from the straight line between adjacent feature points is selected as a feature point if the distance to the straight line is greater than a threshold value. This selection is done recursively until no more feature points are selected. The feature point extracting process is shown in Figure 3(a).

Then it uses a MRF model to match the feature points with the states of each character class as shown in Figure 3(b) and obtain a similarity for each character class. It then selects the character class with the largest similarity as the recognition result.

**Site**: Feature points from an input pattern \( S = \{s_1, s_2, s_3, \ldots, s_{12}\} \)

**Label**: States of a character class \( L = \{l_1, l_2, l_3, \ldots, l_{12}\} \)

**Labeling problem**: Assign labels to the sites such as \( s_j = l_1, s_2 = l_1, s_3 = l_3, \ldots, s_{11} = l_9, s_{12} = l_1 \) such that the system recognizes the input pattern by assigning labels to the sites to make the matching between the input pattern and each character class. MRF model is used to solve the labeling problem.

HMMs have been often used with online statistical structural recognition methods and offline English word recognition methods. HMMs probabilistically treat a sequence of feature vectors in writing or position order, so that they can only use the neighborhood relationships between the successively adjacent feature vectors in writing or position order (the so-called one-dimensional neighborhood relationships) and the two-dimensional neighborhood relationships, such as those among the neighboring feature vectors, on distances are not explicitly expressed. For one-dimensional neighborhood relationships, HMMs only use the state transition probabilities and unary features, but binary features are not used well. Moreover, the neighborhood relationships among more than two neighboring feature vectors, such as ternary features, cannot be used. Although some HMMs apply binary features, they only merge the binary features into the unary features and use a vector of larger dimension because HMMs do not take a new view of the binary features, which limits recognition accuracy.

The MRF model is described using an undirected graph in which a set of random variables have a Markov property, and MRFs can be used to effectively integrate information among neighboring feature vectors, such as binary and ternary features, and two-dimensional neighborhood relationships [9]. Therefore, MRFs have been widely and successfully applied to image processing [9]. However, MRFs had not been applied to online character recognition until our reports [16]. Current online handwritten character recognition tends to use HMMs (note that HMMs can be viewed as specific cases of MRFs). MRFs have more degrees of freedom than HMMs for explicitly expressing relations among multiple feature vectors.

We have proposed an MRF model for online recognition of handwritten Japanese characters [16]. We focused on an online structural method introducing temporal information into one-dimensional neighborhood relationships and compared their ef-
fects on HMMs and MRFs. Experimental results demonstrated the superiority of the method and that MRFs exhibited higher recognition accuracy than HMMs.

3.2 Un-structural recognition

For the un-structural recognizer, we do not need to transform each online character pattern to an offline character pattern (two dimensional images), and extract directional features directly from the online character pattern. From an online character pattern, we extract directional features: histograms of normalized stroke direction [6]. For coordinate normalization, we apply pseudo 2D bi-moment normalization (P2DBMN) [6]. The local stroke direction is decomposed into eight directions, and from the feature map of each direction, 8x8 values are extracted by Gaussian blurring so that the dimensionality of feature vectors is 512. To improve the Gaussianity of feature distribution, each value of the 512 features is transformed by the Box-Cox transformation (also called variable transformation). The input feature vector is reduced from 512D to nD by the Fisher linear discriminant analysis (FLDA) [10]. Then we use the nD feature vectors to create a modified quadratic discriminant function (MQDF) recognizer [18] as follows:

\[
g(x, \omega_i) = \sum_{j=1}^{k} \frac{1}{\delta j} \left[ (x - \mu_j)^2 + \frac{2}{\delta j} \left( \sum_{m=1}^{n} (x - \mu_j)^2 \right) \right] + \sum_{j=1}^{k} \log \lambda_j + (n-k) \log \delta
\]

where \( \mu_i \) is the mean vector of class \( \omega_i \), \( \lambda_{ij} \) \((j = 1, \ldots, k)\) are the largest eigenvalues of the covariance matrix and \( \delta j \) are the corresponding eigenvectors, \( k \) denotes the number of principal axes, and \( \delta \) is a modified eigenvector that is set as a constant. The value of \( \delta \) can be optimized on the training data set. However, for a convenience, we simply set it as \( \gamma \text{average} \) where \( \gamma \text{average} \) is the average of \( \lambda_{ij} \) \((i, j = 1, \ldots, n)\) for all features of all classes and \( \gamma \) is a constant that is larger than 0 and smaller than 1.

According to previous works [6], the best un-structural recognition performance is obtained when \( n \) is about 160 and \( k \) is about 50 for the MQDF recognizer. When combining structural and un-structural recognizers and then combining them with linguistic context and geometric features for the string recognition, we have found the best combination performance is obtained when \( n \) is about 90 and \( k \) is about 10 for the MQDF recognizer. Therefore, we take \( n \) as 90 and \( k \) as 10, respectively.

4 COARSE CLASSIFICATION

A general approach to improving the recognition speed is to perform coarse classification, pre-classification, or candidate selection before the fine classification [19].

The coarse classification typically uses simpler classification algorithms or fewer features in order to achieve a better speed than does the fine classification. It is used to select a relatively small subset of candidates out of a very large set quickly. The fine classification would then be used on these candidates to match an input pattern so that the whole recognition time is reduced. Current approaches for coarse classification typically use distance measures that are simpler than those for fine classification [20]. The confidence evaluation provides even more precise candidate selection [21]. Others use simple features different from those for fine classification [22]. Sequential (multi-stage) classifications using a partial set of features at each stage have also been used [23]. A common characteristic of these methods is that the entire candidate selection process is performed during the recognition process. We call these “dynamic approaches.”

In contrast to dynamic approaches, prototypes may be organized prior to the search itself so that the search is performed on a subset of prototypes. We could mention a number of methodologies that vary slightly in how the data is organized. The simplest ones are proposals for ordered spaces and tree structures. The search on pre-structured spaces aims particularly at alleviating problems with search costs. As a result, recognition is accelerated. We refer to these methods as static approaches. In previous research within our group, we applied a static approach named “structuring search space” (SSS) as an off-line recognizer of handwritten Japanese characters [24]. In SSS, prototypes are organized by unsupervised clusters, and their centroids, for example, are used to represent the group. During recognition, an input pattern is first compared with all of the clusters’ centroids, and second, only the clusters with centroids similar to the input are used as search space. This method was extended to a two-layered search space [25] and named the “layered search spaces (LSS) method”. Being different from the approach by Tseng et al. [26] and that by Fujimoto et al. [27], the LSS method works in the original feature space for fine classification and, therefore, only has to assume one distance space. Waizumi et al. also presented a multilayer search space construction method for character class prototypes that uses LVQ [28].

In general, the previous methods for both the dynamic and static approaches have applied rather intuitive ideas for selecting simple recognition functions such as Euclidean distance or fewer features to speed up recognition. Therefore, they have to take a larger number of output candidates to maintain recognition rates, which causes recognition time to increase and limits the effects on speedup.
We have presented a systematic method for constructing an efficient and robust coarse classifier that uses a genetic algorithm (GA) for the online recognition of handwritten Japanese characters [29]. The method creates 243 basic recognizers with different classification costs and different classification accuracies by controlling the parameters of feature extraction and discriminant functions as well as the LSS method. It then uses these basic recognizers to construct a robust coarse classifier. It constructs a sequential cascade of basic recognizers and reduces the candidates after each basic recognizer. The parameters are estimated by using a GA so as to optimize the holistic character recognition performance. The coarse classifier thus developed follows the dynamic approach although it integrates the statistically tuned LSS. Experimental results demonstrated the superiority of our method.

5 COMBINED RECOGNITION

We have applied a discriminative method for MCE to optimize the parameters for combinations of structural and un-structural recognizers with a linear or nonlinear function for online handwritten Japanese string recognition [7]. To introduce an effective set of parameters, we applied a $k$-means method to cluster the parameters of all character categories into groups, and for categories belonging to the same group, we introduced the same weight parameters. We investigated how to construct the function and how to introduce effective parameters for discriminative methods under the condition of a limited amount of samples for classifier training. We designed the objective functions of parameter optimization so as to optimize the string performance. Moreover, we used GA to estimate super parameters such as the number of clusters, initial learning rate, and maximum learning times as well as the sigmoid function parameter for the MCE optimization. Experimental results demonstrated the superiority of our method.

6 STRING RECOGNITION

6.1 Linguistic contextual processing

String recognition applies not only character recognition, but also linguistic contextual processing. As shown in Figure 4 (a), by character recognition, each candidate character pattern is associated with a number of candidate classes with confidence scores. The combination of all character classes is represented by a character recognition candidate lattice. The linguistic contextual processing evaluates the combinations from character classes to character classes. By searching the candidate lattice by the Viterbi algorithm, the optimal path with maximum score gives the final result of string recognition.

Linguistic contextual processing methods can be roughly divided into two classes: methods using the character combinations and methods using the word combinations. As shown in Figure 4 (b), the linguistic contextual processing evaluates the probability $P(C)$ of the string $C$ that comprises a sequence of characters $\{c_1, c_2, \ldots\}$ or a sequence of words $\{w_1, w_2, \ldots\}$.

![Figure 4. Character recognition candidate lattice and linguistic contextual processing methods.](image)

The methods using the character combinations evaluate the probability of the character combinations for each string candidate. We can use the appearance probability of only one character (unigram), bi-gram of two characters, tri-gram of three characters and generally called $n$-gram of $n$ characters. The tri-gram is smoothed to overcome the imprecision of training with insufficient text by combining unigram, bi-gram and tri-gram using a linear function with weighting parameters.

In our experiment, under the condition with character writing boxes, using bi-gram improved the character recognition rate by 5 points from 92.9%, and using tri-gram improved the character recognition rate by one point. Moreover, under the condition without character writing boxes, using bi-gram improved the character recognition rate by 10 points from 81.3%, and using tri-gram improved the character recognition rate by 3 points.

The methods using the word combinations first divide string into words by morphological analysis, and then evaluate the probability of the word combinations for each string candidate. We can also use the appearance probability of only one word (unigram), bi-gram of two words, tri-gram of three words and generally called $n$-gram of $n$ words. Although the methods have some problems such as unknown words and word dictionary memory, Nagata et al. have presented it could save more than 2/3 misrecognitions in a handwriting OCR simulation by dealing with unknown words [30].

We apply the methods using the character combinations for our system.

6.2 Freely written string recognition

With pen-based or touch-based input devices of large writing areas such as tablet PCs, Pad PCs, elec-
tronic whiteboards and digital pens, people tend to write text continuously with little constraints. This urges the need of handwritten string recognition. Compared to isolated character recognition, handwritten string recognition faces the difficulty of character segmentation because characters cannot be reliably segmented before they are recognized. Moreover, in continuous handwriting, characters tend to be written more cursively.

The integrated segmentation and recognition to overcome the ambiguity of segmentation, is dichotomized into segmentation-free method and over-segmentation-based method as shown in Fig. 1 [10]. Based on the advantages of over-segmentation-based method, we apply it for our recognition system.

By character recognition, each candidate character pattern is associated with a number of candidate classes with confidence scores. The combination of all candidate patterns and character classes is represented by a character segmentation-recognition candidate lattice, where each arc denotes a segmentation point and each node denotes a character class assigned to a candidate pattern as shown in Figure 5. The segmentation paths and corresponding string classes in the candidate lattice are evaluated by combining the scores of candidate characters and between-character compatibilities (geometric and linguistic contexts).

Figure 5. Character segmentation-recognition candidate lattice

In over-segmentation based string recognition, how to evaluate the candidate characters (lying on paths in the candidate lattice) is a key issue. A desirable criterion should make the path of correct segmentation have the largest score. Unlike HMM-based recognition that classifies a unique sequence of feature vectors (each for a frame) on a string, the candidate lattice of over-segmentation has paths of different lengths, each corresponding to a different sequence of feature vectors, thus the comparison of different paths cannot be based on the Bayesian decision theory as for HMM-based recognition. Instead, candidate character recognition and context scores are heuristically combined to evaluate the paths. Such heuristic evaluation criteria can be divided into summation-based ones [31] and normalization-based ones [32]. A summation criterion is the summation of character-wise log-likelihood or the product of probabilistic likelihood. Since the likelihood measure is usually smaller than one, the summation (product) criterion is often biased to paths with fewer characters, and so, tends to over-merge characters. On the other hand, the normalized criterion, obtained by dividing the summation criterion by the number of segmented characters (segmentation length), tends to over-split characters.

To solve the problems, we have proposed a robust context integration model for online handwritten Japanese string recognition [1]. By labeling primitive segments, the proposed method can not only integrate the character shape information into recognition by introducing some adjustable parameters, but also is insensitive to the number of segmented character patterns because the summation is over the primitive segments. Experimental results demonstrated the superiority of our proposed string recognition model.

We include a brief description here on our recognition model [1]. Denote \( X = x_1 \ldots x_m \) as successive candidate character patterns of one path, and every candidate character pattern \( x_i \) is assigned a candidate class \( C_i \). Then \( f(X,C) \) is the score of the path \((X,C)\) where \( C = C_1 \ldots C_m \). The path evaluation criterion is expressed as follows:

\[
f(X,C) = \sum_{i=1}^{m} \left[ \sum_{h=1}^{6} (\lambda_h \cdot (k-1)) \log P_{hi} + \sum_{x} \log P_{g_i} \right] + m\lambda
\]

where \( P_{hi}, h=1,\ldots,6 \), stand for the probabilities of \( P(C_i|C_{i-2}C_{i-1}) \), \( P(b_i|C_i) \), \( P(q_i|C_i) \), \( P(p_i^0|C_i) \), \( P(p_i|x_i|C_i) \), and \( P(p_i^b|C_i,C_i) \), respectively. \( b_i \), \( q_i \), \( p_i^0 \), and \( p_i^b \) are the feature vectors for character pattern sizes, inner gaps, single-character positions, and pair-character positions, respectively. \( g_i \) is the between-segment gap feature vector. \( P(C_i|C_{i-2},C_{i-1}) \) is the tri-gram probability. \( k_i \) is the number of primitive segments contained in the candidate character pattern \( x_i \). \( \lambda_{h1}, \lambda_{h2} (h=1\sim7) \) and \( \lambda \) are the weighting parameters estimated by GA. \( P(x_i|C_i) \) is estimated by the combination score of the structural and un-structural recognizers. We can also divide it into two parts \( P(x_i^{str}|C_i) \), and \( P(x_i^{str}|C_i) \) where \( x_i^{str} \) denotes the structural features of \( x_i \). \( x_i^{str} \) denotes the un-structural features of \( x_i \). \( P(x_i^{str}|C_i) \) is estimated by the score of the structural recognizer and \( P(x_i^{str}|C_i) \) is estimated by the score of the un-structural recognizer. The path evaluation criterion is changed as follows:

\[
f(X,C) = \sum_{i=1}^{m} \left[ \sum_{h=1}^{7} (\lambda_h \cdot (k-1)) \log P_{hi} + \sum_{x} \log P_{g_i} \right] + m\lambda
\]
(h=1~8), and \(\lambda\) are the weighting parameters estimated by GA. By the path evaluation criterion, we re-estimate the combination of the structural and un-structural recognizers.

7 EXPERIMENTS

We investigated the results for our recognition systems using on-line Chinese handwriting databases (CASIA-OLHWDB1.0-1.2, 2.0) [33] and on-line Japanese handwriting databases (TUAT Nakayosi/Kuchibue, Kondate) [1, 34].

We trained the structural and un-structural character recognizers, the weighting parameters for combination, and geometric scoring functions using CASIA-HWDB1.0-1.2 for the Chinese recognition (7,356 classes, including 7,184 Chinese characters and 172 symbols), and using Nakayosi/Kuchibue for the Japanese recognition (4,443 classes, including 4,060 Chinese characters, 169 Kana and 211 symbols). For scorning linguistic context, we prepared a tri-gram table for Chinese recognition from a corpus of People’s Daily, and that for Japanese from the 1993 and 2002 volumes of the Asahi and Nikkei newspapers, respectively. For training the weight parameters and evaluating the performances of character string recognitions, we use the database CASIA-OLHWDB2.0 (420 writters, 20,573 text lines) for Chinese recognition and the database Kondate (100 writters, 13,685 text lines) for Japanese recognition.

We used 336 people’s Chinese text lines and 75 people’s Japanese text lines for training the classifiers for the candidate segmentation point probability and the weighting parameters of the path evaluations. The performance test for the string recognizer used the text lines of the remaining 84 persons of CASIA-OLHWDB2.0 and 25 persons of Kondate. Table 1 shows the results, where the numbers without any parentheses or brackets are the recognition rates, those shown in parentheses are the average character recognition time and those shown in brackets are the memory costs. The experiments were implemented on an Intel(R) Xeon(R) CPU W5590 @ 3.36 GHz 3.36 GHz (2 processors) with 12 GB memory.

Table 1. Recognition results.

<table>
<thead>
<tr>
<th>System</th>
<th>Chinese</th>
<th>Japanese</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
<td>90.00%(25ms)[246MB]</td>
<td>92.99%(20ms)[20MB]</td>
</tr>
</tbody>
</table>

For the Japanese recognition system, we build two compact structural/un-structural recognizers by compressing them using vector quantization (VQ) techniques [35, 36] and we did not built compact recognizers for the Chinese recognition system because of the time limitation. Therefore, the memory cost for the Japanese recognition system is lower than that for the Chinese recognition system. From the results we can see that we have realized the online handwritten Chinese/Japanese character recognitions with high performances that can be applied to actual applications. The Japanese recognition system exhibited higher recognition accuracies than the Chinese recognition system. We consider that it is because Chinese characters have simpler shapes than Japanese characters and Chinese people writes characters more fluently and more cursively than Japanese people resulting in low recognition accuracies.

8 CONCLUSION

This paper described the recent trends in online handwritten Chinese/Japanese character recognition and our recognition systems. We apply an over-segmentation-based method for our recognition system where the paths are evaluated in accordance with our path evaluation criterion, which combines the scores of character recognition, linguistic context, and geometric features (character pattern sizes, inner gaps, single-character positions, pair-character positions, candidate segmentation points) with the weighting parameters estimated by GA. We combine structural and un-structural methods to recognize each character pattern so that the recognition accuracy improves.

Improving recognition performance is the aim of our future work. This can be achieved by incorporating more effective geometric features, exploiting better geometric context likelihood functions and weighting parameter learning methods, and improving the accuracy of character recognizer. To speed up recognition and reduce memory size is another dimension of our future work. We should consider effective methods to remove invalid patterns from the lattice.

ACKNOWLEDGMENT

This work is being supported by the Basic Research Program (C) of Scientific Research Fund funded by Japan Society for the Promotion of Science (JSPS) / Ministry of Education, Culture, Sports, Science & Technology (MEXT) for “Research on Unified Model and Method for Recognizing Multilingual Handwriting from Touch Based Input Device”.

REFERENCES


