

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

185,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



---

# Online Handwritten Chinese/Japanese Character Recognition

---

Bilan Zhu and Masaki Nakagawa

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/51474>

---

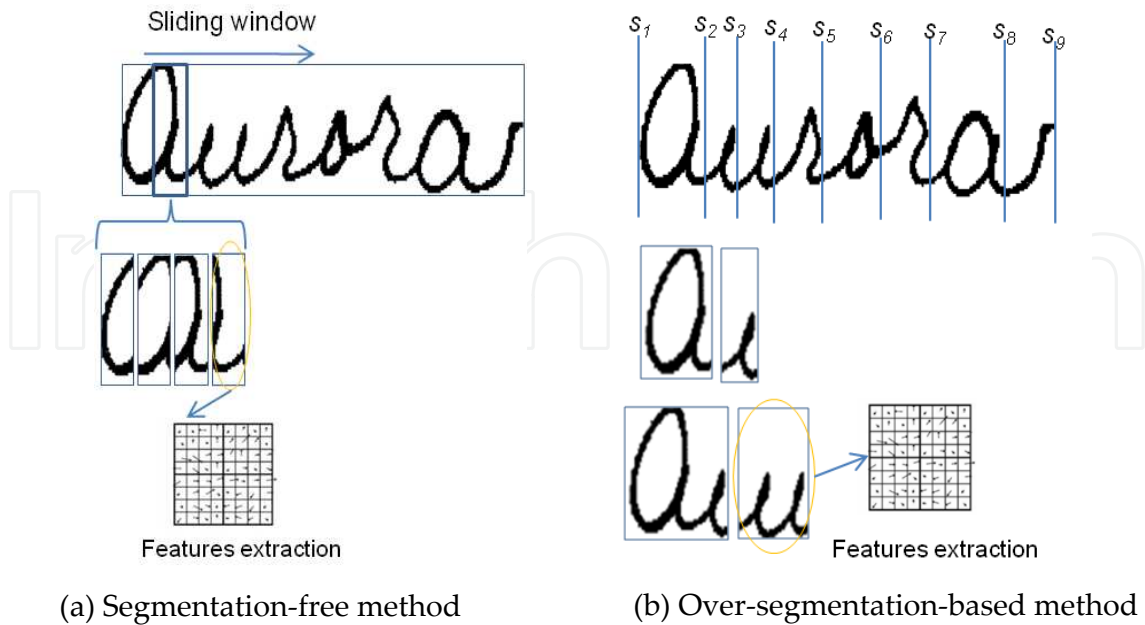
## 1. Introduction

Handwritten character pattern recognition methods are generally divided into two types: online recognition and offline recognition [1]. 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. Un-structural methods are also called statistical methods [2] or feature matching methods [3]. Structural methods are based on stroke analysis and use structural features such as sampling points, line segments and/or strokes for offline recognition [3-5] and for online recognition [6-21]. Un-structural methods use un-structural features such as directional features, gradient histogram features and projection features such as those for offline [22-24] and online recognition [25, 26], 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 [27, 28].

For online recognition, structural features are often employed with hidden Markov models (HMMs) [12-21] or Markov random field (MRF) [29, 30]. 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.



**Figure 1.** Freely written string recognition

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 [31, 32] as shown in Figure 1. Segmentation-free methods, mostly combined with hidden Markov model (HMM)-based recognition [20, 21, 33-38], 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 [39-46]. 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) [32-38] and those using over-

segmentation-based methods [39-43] 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 [2]. 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.

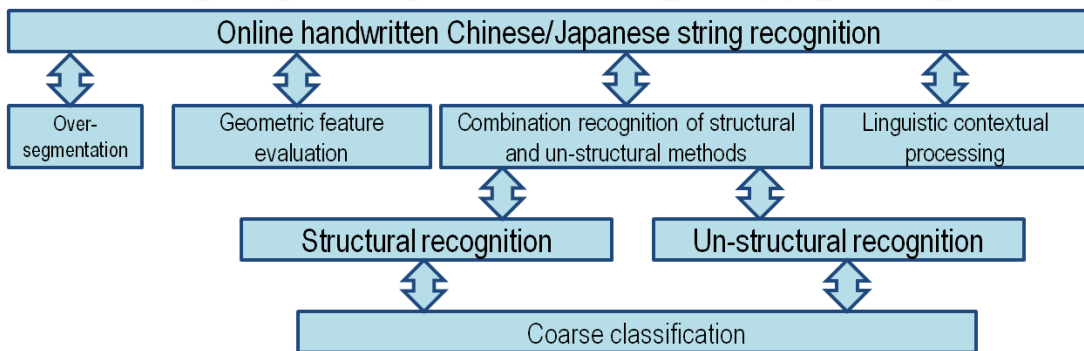
In this chapter, we describe the recent technology trends, problems and methods to solve them for the online handwritten Chinese/Japanese character recognition. The rest of this chapter is organized as follows. Section 2 presents an overview of our online handwritten string recognition system. Section 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:

### 2.1. 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.



**Figure 2.** Online handwritten Chinese/Japanese string recognition

## 2.2. 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.

## 2.3. 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. [47], 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 [48, 49]. 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 [50]. 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  $S=\{s_1, s_2, s_3, \dots, s_{12}\}$

**Label:** States of a character class  $L=\{l_1, l_2, l_3, \dots, l_{12}\}$

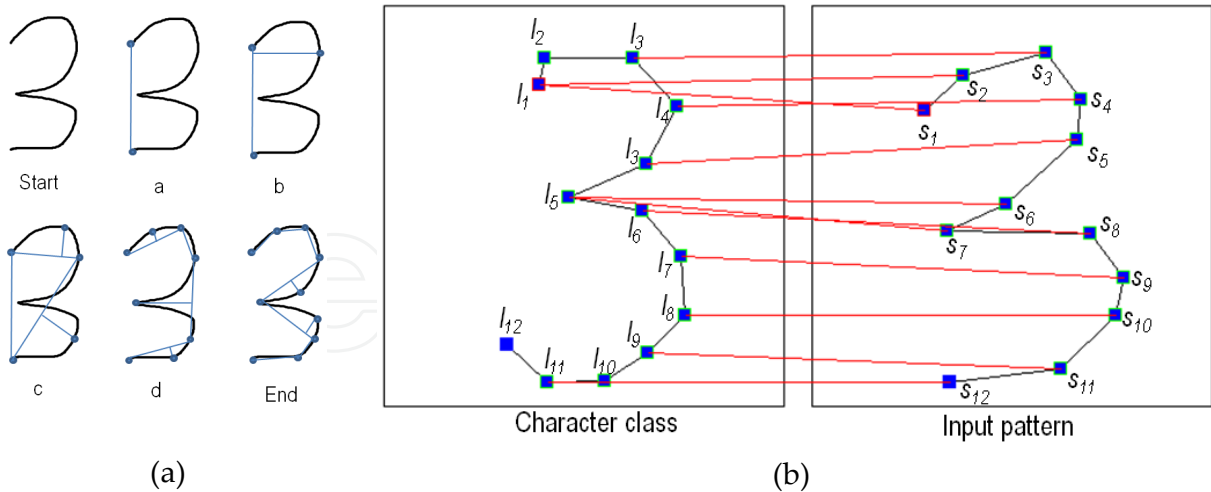
**Labeling problem:** Assign labels to the sites such as  $\{s_1=l_1, s_2=l_1, s_3=l_3, \dots, s_{11}=l_9, s_{12}=l_{11}\}$

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.

Structural methods can be further divided into template-based structural methods [3, 6-11] and statistical structural methods [4, 5, 12-21, 39, 40]. Template-based structural methods work well with handwriting recognition for user dependent systems. However, these methods do not take into account the distributions of training patterns, resulting in limited recognition accuracy. Statistical structural methods measure probabilistic primitives and/or relationships so as to better model the shape variations of input patterns [29, 30].

HMMs have been often used with online statistical structural recognition methods and offline English word recognition methods. HMMs were first described in a series of statistical papers [51] and applied to speech recognition [52-53] in the middle of the 1970s. Then, they were applied widely to online handwriting [12-21] and offline word recognition [32-38].

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 [31].



**Figure 3.** Feature points extraction and labeling

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 [54]. Therefore, MRFs have been effectively applied to stroke-



analysis-based structural offline character recognition [4, 5]. They have also been widely and successfully applied to image processing [55, 56] and online stroke classification [57]. However, MRFs had not been applied to online character recognition until our reports [48, 49]. 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.

Saon et al. [33] proposed an HMM-based offline English word recognition method that uses neighboring pixels to estimate the pixel observation probabilities and discussed its performance. However, it is still an HMM-based method, although it uses the neighborhood relationships of the recognition. Based on the advantages of MRFs, we can assume that applying MRFs instead of HMMs to integrate the information among the neighboring feature vectors can improve performance of offline English or other western word recognition using segmentation-free methods [32-38].

Since online character patterns contain temporal information on pen movements, structural methods that discard temporal information and only apply structural information can result in stroke-order independence. However, it is computationally expensive since the neighborhood relationships must be examined in two dimensions. Although the method introducing temporal information is very sensitive to stroke order variations, it is efficient in recognition speed, and combining it with an un-structural method can deal with the stroke-order variations [27, 28]. Even for the one-dimensional neighborhood relationships applying MRFs instead of HMMs to integrate the information of binary features between the successively adjacent feature vectors in writing or position order can improve performance.

Cho et al. [58] proposed a Bayesian network (BN)-based framework for online handwriting recognition. BNs share similarities with MRFs. They are directional acyclic graphs and model the relationships among the neighboring feature vectors as conditional probability distributions, while MRFs are undirected graphs and model the relationships among the neighboring feature vectors as probability distributions of binary or ternary features.

Introducing weighting parameters to MRFs and optimizing them based on conditional random fields (CRFs) [59] or the minimum classification error (MCE) [60] may result in even higher recognition accuracy; CRFs have been successfully applied to online string and offline word recognition [42, 61].

We have proposed an MRF model with weighting parameters optimized by CRFs for online recognition of handwritten Japanese characters [48, 49]. We focused on an online structural method introducing temporal information into one-dimensional neighborhood relationships and compared their effects on HMMs and MRFs. The model effectively integrates unary and binary features and introduces adjustable weighting parameters to the MRFs, which are optimized according to CRF. The proposed method extracts feature points along the pen-tip trace from pen-down to pen-up and matches those feature points with states for character classes probabilistically based on this model. 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 [26]. For coordinate normalization, we apply pseudo 2D bi-moment normalization (P2DBMN) [22]. 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  $n$ D by the Fisher linear discriminant analysis (FLDA) [31]. Then we use the  $n$ D feature vectors to create a modified quadratic discriminant function (MQDF) recognizer [24] as follows:

$$g_2(\mathbf{x}, \omega_i) = \sum_{j=1}^k \frac{1}{\lambda_{ij}} [\boldsymbol{\phi}_{ij}^T (\mathbf{x} - \boldsymbol{\mu}_i)]^2 + \frac{1}{\delta} \{ \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 - \sum_{j=1}^k [\boldsymbol{\phi}_{ij}^T (\mathbf{x} - \boldsymbol{\mu}_i)]^2 \} + \sum_{j=1}^k \log \lambda_{ij} + (n - k) \log \delta \quad (1)$$

where  $\boldsymbol{\mu}_i$  is the mean vector of class  $\omega_i$ ,  $\lambda_{ij}$  ( $j = 1, \dots, k$ ) are the largest eigenvalues of the covariance matrix and  $\boldsymbol{\phi}_{ij}$  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  $\gamma \lambda_{average}$  where  $\lambda_{average}$  is the average of  $\lambda_{ij}$  ( $i, j = 1, \dots, n$ ) for all features of all classes and  $\gamma$  is a constant that is larger than 0 and smaller than 1.

According to previous works [23, 26], 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

Although character classifiers with high recognition accuracy have been reported [26, 47-49], 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 performance, a recognition speed of within 0.5 seconds per page is required in actual applications. Therefore, we need to refine the recognition scheme to improve the processing speed.

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



speed. A general approach to improving the recognition speed is to perform coarse classification, pre-classification, or candidate selection before the fine classification [62, 63].

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 [64, 65]. The confidence evaluation provides even more precise candidate selection [66]. Others use simple features different from those for fine classification [67]. Sequential (multi-stage) classifications using a partial set of features at each stage have also been used [68].

On the other hands, 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.

## 5. Combined recognition

How to combine different classifiers is an important problem in multiple classifier approaches. In Japanese character recognition, Oda et al. improved recognition performance by combining a recognizer by a structural method and that by an un-structural method using probabilistic tables to normalize the combination scores [69]. The combination method by probabilistic tables is a generative method, and applying a discriminative method such as the MCE criterion and neural network to estimate and to optimize the combination may bring about higher performance.

Liu investigated the effects of confidence transformation in combining multiple classifiers using various combination rules [70]. Kermorvant et al. constructed a neural network to combine the top rank candidates of three word recognizers [37]. The two works used the discriminative methods to estimate the combination parameters. However, when optimizing the parameters the previous works always only considered the character/word recognition performance, not the string recognition performance. In fact, real applications usually use the string recognition rather than the character recognition. The character recognition is a part of the string recognition. Therefore, when we create a character recognizer, we have to consider the string recognition performance, as done by Tonouchi [71] and Cheriet et al. [71]. The methods that only guarantee the character recognition accuracy do not necessarily provide high string recognition performance. They cannot even be applied for string recognition.

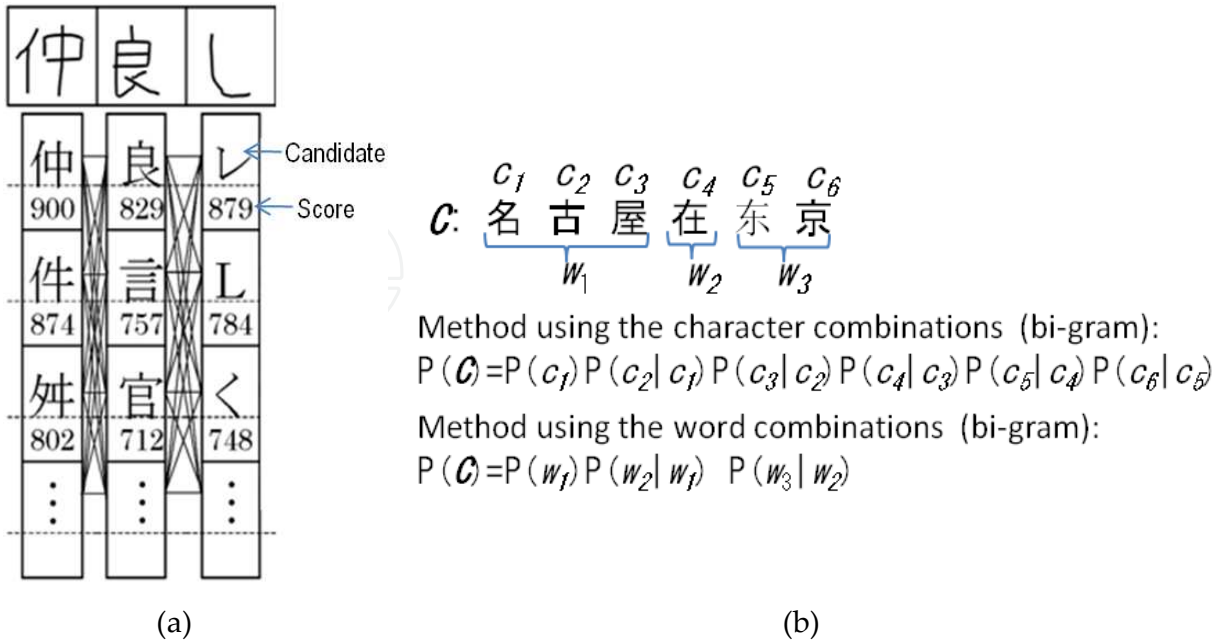
On the other hand, we have to point out that introducing more parameters for a discriminative method dose not bring about higher performance, since we have only a limited amount of samples for training. However, previous works tended to introduce too many parameters for a discriminative method.

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 [28]. 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.



**Figure 4.** Character recognition candidate lattice and linguistic contextual processing methods

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  $\{c_1, c_2, \dots\}$  or a sequence of words  $\{w_1, w_2, \dots\}$ .

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 [73].

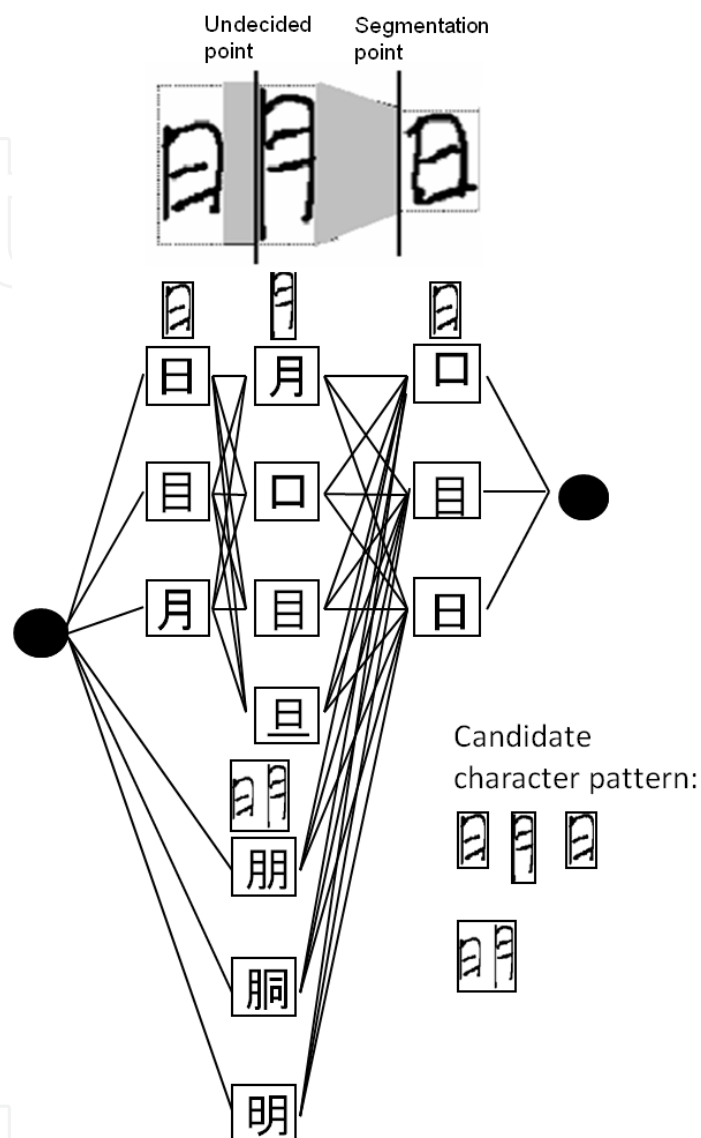
Recognition of a limited vocabulary such as addresses, person names, dates and departments is important issues in string recognition. Applying a general-purpose contextual processing method such as tri-gram to recognize specific words would cause misrecognitions. Applying domain specific methods to recognize specific words into the vocabulary set can improve recognition rate significantly [45, 74].

## 6.2. Freely written string recognition

With pen-based or touch-based input devices of large writing areas such as tablet PCs, Pad PCs, electronic 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 [31]. Based on the advantages of over-segmentation-based method, we apply it for our recognition system.



**Figure 5.** Character segmentation-recognition candidate lattice

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).

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 [75] and normalization-based ones [71]. 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 [47]. 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 [47]. Denote  $\mathbf{X} = x_1 \dots x_m$  as successive candidate character patterns of one path, and every candidate character pattern  $x_i$  is assigned a candidate class  $C_i$ . Then  $f(\mathbf{X}, \mathbf{C})$  is the score of the path  $(\mathbf{X}, \mathbf{C})$  where  $\mathbf{C} = C_1 \dots C_m$ . The path evaluation criterion is expressed as follows:

$$f(\mathbf{X}, \mathbf{C}) = \sum_{i=1}^m \left\{ \sum_{h=1}^6 [\lambda_{h1} + \lambda_{h2}(k_i - 1)] \log P_h + \lambda_{71} \log P(g_{j_i} | SP) + \lambda_{72} \sum_{j=j_i+1}^{j_i+k_i-1} \log P(g_j | NSP) \right\} + m\lambda \quad (2)$$

where  $P_h$ ,  $h=1, \dots, 6$ , stand for the probabilities of  $P(C_i | C_{i-2}C_{i-1})$ ,  $P(b_i | C_i)$ ,  $P(q_i | C_i)$ ,  $P(p^u_i | C_i)$ ,  $P(x_i | C_i)$ , and  $P(p^b_i | C_{i-1}C_i)$ , respectively.  $b_i$ ,  $q_i$ ,  $p^u_i$ , and  $p^b_i$  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  $x_i$ .  $\lambda_{h1}$ ,  $\lambda_{h2}$  ( $h=1 \sim 7$ ) 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  $P(x^{str}_i | C_i)$ ,  $P(x^{un-str}_i | C_i)$  where  $x^{str}_i$  denotes the structural features of  $x_i$ ,  $x^{un-str}_i$  denotes the un-structural features of  $x_i$ ,  $P(x^{str}_i | C_i)$  is estimated by the score of the structural recognizer and  $P(x^{un-str}_i | C_i)$  is estimated by the score of the un-structural recognizer. The path evaluation criterion is changed as follows:

$$f^1(\mathbf{X}, \mathbf{C}) = \sum_{i=1}^m \left\{ \sum_{h=1}^7 [\lambda_{h1} + \lambda_{h2}(k_i - 1)] \log P_h + \lambda_{81} \log P(g_{j_i} | SP) + \lambda_{82} \sum_{j=j_i+1}^{j_i+k_i-1} \log P(g_j | NSP) \right\} + m\lambda \quad (3)$$

where  $P_h$ ,  $h=1, \dots, 7$ , stand for the probabilities of  $P(C_i | C_{i-2}C_{i-1})$ ,  $P(b_i | C_i)$ ,  $P(q_i | C_i)$ ,  $P(p^u_i | C_i)$ ,  $P(x^{str}_i | C_i)$ ,  $P(x^{un-str}_i | C_i)$ , and  $P(p^b_i | C_{i-1}C_i)$ , respectively.  $\lambda_{h1}$ ,  $\lambda_{h2}$  ( $h=1 \sim 8$ ), 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. Conclusion

This chapter described the recent trends in online handwritten Chinese/Japanese character recognition and our recognition system. 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.

## Author details

Bilan Zhu and Masaki Nakagawa

*Department of Computer and Information Sciences, Tokyo University of Agriculture and Technology, Tokyo, Japan*

## 8. References

- [1] R. Plamondon and S.N. Srihari (2000) On-line and off-line handwriting recognition: a comprehensive survey. *IEEE Trans. PAMI*. 22(1): 63-82.
- [2] C.-L. Liu, S. Jaeger and M. Nakagawa (2004) On-line recognition of Chinese characters: the state of the art. *IEEE Trans. PAMI*. 26(2): 198-213.
- [3] C.-L. Liu, I.-J. Kim and J. H. Kim (2001) Model-based stroke extraction and matching for handwritten Chinese character recognition. *Pattern Recognition*. 34(12): 2339-2352.
- [4] J. Zeng and Z.-Q. Liu (2005) Markov random fields for handwritten Chinese character recognition. *Proc. 8th ICDAR*: 101-105.
- [5] J. Zeng and Z.-Q. Liu (2008) Markov random field-based statistical character structure modeling for handwritten Chinese character recognition. *IEEE Trans. PAMI*. 30(5): 767-780.



- [6] Y.-T. Tsay and W.-H. Tsai (1993) Attributed string matching by split-and-merge for on-line Chinese character recognition. *IEEE Trans. PAMI*. 15(2): 180-185.
- [7] M. Nakagawa and K. Akiyama (1994) A linear-time elastic matching for stroke number free recognition of on-line handwritten characters. *Proc. 4th IWFHR*: 48-56.
- [8] J. Liu, W.-K. Cham and M. M. Y. Chang (1996) Stroke order and stroke number free on-line Chinese character recognition using attributed relational graph matching. *Proc. 13th ICPR*. 3: 259-263.
- [9] T. Wakahara and K. Okada (1997) On-line cursive kanji character recognition using stroke-based affine transformation. *IEEE Trans. PAMI*. 19(12): 1381-1385.
- [10] J.-P. Shin and H. Sakoe (1999) Stroke correspondence search method for stroke-order and stroke-number free on-line character recognition—multilayer cube search. *Trans. IEICE Japan*. J82-D-II (2): 230-239.
- [11] A. Kitadai and M. Nakagawa (2002) A learning algorithm for structured character pattern representation used in on-line recognition of handwritten Japanese characters. *Proc. 8th IWFHR*: 163-168.
- [12] M. Nakai, N. Akira, H. Shimodaira and S. Sagayama (2001) Substroke approach to HMM-based on-line kanji handwriting recognition. *Proc. 6th ICDAR*: 491-495.
- [13] M. Nakai, T. Sudo, H. Shimodaira and S. Sagayama (2002) Pen pressure features for writer-independent on-line handwriting recognition based on substroke HMM. *Proc. 16th ICPR*. 3: 220-223.
- [14] M. Nakai, H. Shimodaira and S. Sagayama (2003) Generation of hierarchical dictionary for stroke-order free kanji handwriting recognition based on substroke HMM. *Proc. 7th ICDAR*: 514-518.
- [15] J. Tokuno, N. Inami, S. Matsuda, M. Nakai, H. Shimodaira and S. Sagayama (2002) Context-dependent substroke model for HMM based on-line handwriting recognition. *Proc. 8th IWFHR*: 78-83.
- [16] K. Takahashi, H. Yasuda and T. Matsumoto (1997) A fast HMM algorithm for on-line handwritten character recognition. *Proc. 4th ICDAR*: 369-375.
- [17] H. Yasuda, K. Takahashi and T. Matsumoto (1999) On-line handwriting recognition by discrete HMM with fast learning. *Advances in Handwriting Recognition*. S.-W. Lee: World Scientific. pp. 19-28.
- [18] Y. Katayama, S. Uchida and H. Sakoe (2008) HMM for on-line handwriting recognition by selective use of pen-coordinate feature and pen-direction feature. *Trans. IEICE Japan*. J91-D (8): 2112-2120.
- [19] H. J. Kim, K. H. Kim, S. K. Kim and F. T.-P. Lee (1997) on-line recognition of handwritten Chinese characters based on hidden Markov models. *Pattern Recognition*. 30(9): 1489-1499.
- [20] S. Jaeger, S. Manke, J. Reichert and A. Waibel (2001) Online handwriting recognition: the Npen++ recognizer. *IJDAR*. 3(1): pp.69-180.
- [21] M. Liwicki and H. Bunke (2006) HMM-based on-line recognition of handwritten whiteboard notes. *Proc. 10th IWFHR*: 595-599.

- [22] C.-L. Liu and K. Marukawa (2005) Pseudo two-dimensional shape normalization methods for handwritten Chinese character recognition. *Pattern Recognition*. 38(12): 2242-2255.
- [23] C.-L. Liu (2006) High accuracy handwritten Chinese character recognition using quadratic classifiers with discriminative feature extraction. *Proc. 18th ICPR*. 2: 942-945.
- [24] F. Kimura (1987) Modified quadratic discriminant function and the application to Chinese characters. *IEEE Trans. PAMI*. 9 (1): 149-153.
- [25] A. Kawamura, K. Yura, T. Hayama, Y. Hidai, T. Minamikawa, A. Tanaka and S. Masuda (1992) On-line recognition of freely handwritten Japanese characters using directional feature densities. *Proc. 11th ICPR*. 2: 183-186.
- [26] C.-L. Liu and X.-D. Zhou (2006) Online Japanese character recognition using trajectory-based normalization and direction feature extraction. *Proc. 10th IWFHR*: 217-222.
- [27] H. Tanaka, K. Nakajima, K. Ishigaki, K. Akiyama and M. Nakagawa (1999) Hybrid pen-input character recognition system based on integration of on-line and off-line recognition. *Proc. 5th ICDAR*: 209-212.
- [28] B. Zhu, J. Gao and M. Nakagawa (2011) Objective function design for MCE-based combination of on-line and off-line character recognizers for on-line handwritten Japanese text recognition. *Proc. 11th ICDAR*: 594-599.
- [29] T.-R. Chou and W. T. Chen (1997) A stochastic representation of cursive Chinese characters for on-line recognition. *Pattern Recognition*. 30(6): 903-920.
- [30] J. Zheng, X. Ding, Y. Wu and Z. Lu (1999) Spatio-temporal unified model for on-line handwritten Chinese character recognition. *Proc. 5th ICDAR*: 649-652.
- [31] M. Cheriet, N. Kharm, C.-L. Liu and C. Y. Suen (2007) *Character recognition systems - A guide for students and practitioners*. Hoboken, New Jersey: John Wiley & Sons, Inc.
- [32] M. Mohamed and P. Gader (1996) Handwritten word recognition using segmentation-free Hidden Markov Model and segmentation-based dynamic programming techniques. *IEEE Trans. PAMI*, 18(5): 548-544.
- [33] G. Saon and A. Belaid (1997) Off-line handwritten word recognition using a mixed HMM-MRF approach. *Proc. 4th ICPR*. 1: 118-122.
- [34] D. Guillevis and C. Y. Suen (1997) HMM word recognition engine. *Proc. 4th ICPR*. 2: 544-547.
- [35] S. Gunter and H. Bunke (2004) HMM-based handwritten word recognition: on the optimization of the number of states, training iterations and gaussian components. *Pattern Recognition*, 37: 2069-2079.
- [36] J.A. Rodriguez and F. Perronnin (2008) Local gradient histogram features for word spotting in unconstrained handwritten documents. *Proc. 1st ICFHR*: 7-12.
- [37] C. Kermorvant, F. Menasri, A.-L. Bianne, R. Al-Hajj, C. Mokbel and L. Likforman-Sulem (2010) The A2iA-Telecom ParisTech-UOB system for the ICDAR 2009 handwriting recognition competition. *Proc. 12th ICFHR*: 247-252.
- [38] T. Hamamura, B. Irie, T. Nishimoto, N. Ono and S. Sagayama (2011) Concurrent optimization of context clustering and GMM for offline handwritten word recognition using HMM. *Proc. 11th ICDAR*: 523-527.

- [39] F. Kimura, M. Sridhar and Z. Chen (1993) Improvements of lexicon-directed algorithm for recognition of unconstrained hand-written words. Proc. 2nd ICDAR: 18-22.
- [40] G. Kim and V. Govindaraju (1997) A lexicon driven approach to handwritten word recognition for real-time applications. IEEE Trans. PAMI, 19(4): 366-379.
- [41] J.T. Favata (2001) Offline general word handwritten word recognition using an approximate BEAM matching algorithm. IEEE Trans. PAMI, 23(9): 1009-1021.
- [42] S. Shetty, H. Srinivasan and S. Srihari (2007) Handwritten word recognition using conditional random fields. Proc. 9th ICDAR: 1098-1102.
- [43] U. Pal, R.K. Roy and F. Kimura (2011) Handwritten street name recognition for Indian postal automation. Proc. 11th ICDAR: 483-487.
- [44] M. Koga, R. Mine, H. Sako and H. Fujisawa (1999) Lexical search approach for character-string recognition. In: S.-W. Lee and Y. Nakano editors. Document Analysis Systems: Theory and Practice. Springer . pp. 115-129.
- [45] C.-L. Liu, M. Koga and H. Fujisawa (2002) Lexicon-driven segmentation and recognition of handwritten character strings for Japanese address reading. IEEE Trans. PAMI, 24(11): 1425-1437.
- [46] M. Koga, R. Mine, H. Sako and H. Fujisawa (2003) A lexicon driven approach for printed address phrase recognition using a trie dictionary. Trans. IEICE Japan. J86-D (2): 1297-1307.
- [47] B. Zhu, X.-D. Zhou, C.-L. Liu, M. Nakagawa (2010) A robust model for on-line handwritten Japanese text recognition. IJDAR. 13(2): 121-131.
- [48] B. Zhu and M. Nakagawa (2011) A MRF model with parameters optimization by CRF for on-line recognition of handwritten Japanese characters. Proc. Document Recognition and Retrieval XVIII (DRR) that is part of IS&T/SPIE Electronic Imaging. San Jose. CD-ROM.
- [49] B. Zhu and M. Nakagawa (2011) On-line handwritten Japanese characters recognition using a MRF model with parameter optimization by CRF. Proc. 11th ICADR: 603-607.
- [50] U. Ramer (1972) An iterative procedure for the polygonal approximation of plan closed curves. Computer Graphics and Image Processing. 1: 244-256.
- [51] L. E. Baum, T. Petrie, G. Soules and N. Weiss (1970) A maximization technique in the statistical analysis of probabilistic functions of Markov chains. Annals of Mathematical Statistics. 41(1): 164-171.
- [52] J. K. Baker (1975) The dragon system – an overview. IEEE Trans. Acoustic, Speech and Signal Processing. ASSP-23. 1: 24-29.
- [53] Lawrence R. Rabiner (1989) A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE, 77(2): 257-286.
- [54] S. Z. Li (2001) Markov random field modeling in image analysis. Tokyo: Springer.
- [55] C. Wolf, D. Doerman (2002) Binarization of low quality text using a Markov random field model. Proc. 16th ICPR. 3: 160 -163.
- [56] T. Obafemi-Ajayi, G. Agam and O. Frieder (2008) Efficient MRF approach to document image enhancement. 19th ICPR: 1-4.

- [57] X.-D. Zhou and C.-L. Liu (2007) Text/non-text ink stroke classification in Japanese handwriting based on Markov random fields. *Proc. 9th ICDAR*: 377-381.
- [58] S.-J. Cho and J. H. Kim (2004) Bayesian network modeling of strokes and their relationships for on-line handwriting recognition. *Pattern Recognition*, 37(2): 253-264.
- [59] J. Lafferty, A. McCallum, and F. Pereira (2001) Conditional random fields: probabilistic models for segmenting and labeling sequence data. *Proc 18th ICML*: 282-289.
- [60] B.-H. Juang and S. Katagiri (1992) Discriminative learning for minimum error classification. *IEEE Trans. Signal Processing*, 40(12): 3043-3054.
- [61] X.-D. Zhou, C.-L. Liu and M. Nakagawa (2009) Online handwritten Japanese character string recognition using conditional random fields. *Proc. 10th ICDAR*: 521-525.
- [62] S. Mori, K. Yamamoto and M. Yasuda (1984) Research on machine recognition of handprinted characters, *IEEE Trans. PAMI*. 6: 386-405.
- [63] T. H. Hildebrandt, W. T. Liu (1993) Optical recognition of handwritten Chinese characters: Advances since 1980. *Pattern Recognition*. 26 (2): 05-225.
- [64] T. Wakabayashi, Y. Deng, S. Tsuruoka, F. Kimura, Y. Miyake (1995) Accuracy improvement by nonlinear normalization and feature compression in handwritten Chinese character recognition. *Technical Report of IEICE Japan*. PRU. 95(43): 1-8.
- [65] N. Sun, M. Abe, Y. Nemoto (1995) A handwritten character recognition system by using improved directional element feature and subspace method. *Trans. IEICE Japan*. J78-D-2 (6): 922-930.
- [66] C.-L. Liu and M. Nakagawa (2000) Precise candidate selection for large character set recognition by confidence evaluation, *IEEE Trans. PAMI*, 22 (6): 636-642.
- [67] T. Kumamoto, K. Toraichi, T. Horiuchi, K. Yamamoto, H. Yamada (1991) On speeding candidate selection in handprinted chinese character recognition. *Pattern Recognition*. 24 (8): 793-799.
- [68] C.-H. Tung, H.-J. Lee, J.-Y. Tsai (1994) Multi-stage pre-candidate selection in handwritten Chinese character recognition systems. *Pattern Recognition*. 27 (8): 1093-1102.
- [69] H. Oda, B. Zhu, J. Tokuno, M. Onuma, A. Kitadai, M. Nakagawa (2006) A compact on-line and off-line combined recognizer. *Proc 10th IWFHR*: 133-138.
- [70] C.-L. Liu (2005) Classifier combination based on confidence transformation. *Pattern Recognition*: 38(1):11-28.
- [71] C.-L. Liu, H. Sako, H. Fujisawa (2004) Effects of classifier structures and training regimes on integrated segmentation and recognition of handwritten numeral strings. *IEEE Trans. PAMI*. 26(11): 1395-1407.
- [72] Y. Tonouchi (2010) Path evaluation and character classifier training on integrated segmentation and recognition of online handwritten Japanese character string. *Proc. 12th ICFHR*: 513-517.
- [73] M. Nagata (1998) A Japanese OCR error correction method using character shape similarity and statistical language Model. *Trans. IEICE Japan*. J81-D-2 (11): 2624-2634.

- [74] B. Zhu and M. Nakagawa (2011) Trie-lexicon-driven recognition for on-line handwritten Japanese disease names using a time-synchronous method. Proc. 11<sup>th</sup> ICDAR: 1130-1134.
- [75] M. Nakagawa, B. Zhu and M. Onuma (2005) A model of on-line handwritten Japanese text recognition free from line direction and writing format constraints. Trans. IEICE Japan. E88-D (8): 1815-1822.