Comparative Performance Analysis of Stroke Correspondence Search Methods for Stroke-Order Free Online Multi-Stroke Character Recognition

Wenjie Cai, Seiichi Uchida, Hiroaki Sakoe

Department of Advanced Information Technology, Kyushu University. 744, Motooka, Nishi-ku, Fukuoka-shi, 819-0395 Japan

Abstract For stroke-order free online multi-stroke character recognition, stroke-to-stroke correspondence search between an input pattern and a reference pattern plays an important role to deal with the stroke-order variation. Although various methods of stroke correspondence have been proposed, no comparative study for clarifying the relative superiority of those methods has been done before. In this paper, we firstly review the approaches for solving the stroke-order variation problem. Then, five representative methods of stroke correspondence proposed by different groups, including cube search (CS), bipartite weighted matching (BWM), individual correspondence decision (ICD), stable marriage (SM), and deviation-expansion model (DE), are experimentally compared, mainly in regard of recognition accuracy and efficiency. The experimental results on an online Kanji character dataset, showed that 99.17%, 99.17%, 96.37%, 98.54%, and 96.59% were attained by CS, BWM, ICD, SM, and DE, respectively as their recognition rates. Extensive discussions are made on their relative superiorities and practicalities.

Keywords cube search, bipartite weighted matching, individual correspondence decision, stable marriage, deviation-expansion model

1 Introduction

Online handwriting recognition is gaining renewed interest because of the new development of new pen devices and their applications. An important distinction between online and offline handwriting recognition (or OCR) is the fact that spatial and temporal information are available in the former, while only spatial information is available in the latter. For the online case, character patterns are expressed as the trajectory of pen tip motion during writing. For the offline case, the patterns are expressed as scanned images and recognized by conventional OCR techniques.

A practical online handwriting recognition system should be *stroke-order free* to cope with characters with stroke-order variation. In fact, for multiple stroke characters like Chinese characters or Japanese Kanji characters, the number of strokes of a single character often exceeds 20 and their stroke-order varies in many ways.

Although there are several researches for solving the stroke-order free recognition problem, the problem has not been solved yet. Moreover, since no extensive comparative study has been made so far, none knows the relative characteristics and performance of the past approaches. In other words, there has been no research which helps to choose the most suitable stroke-order free recognition approach for individual applications.

The main contributions of this paper are twofold.

- A review is made to clarify the relative superiority of stroke-order free recognition approaches from the viewpoints of recognition accuracy and efficiency.
- Especially for stroke correspondence approach, which is theoretically the most reliable approach among several approaches, we conduct a quantitative and qualitative evaluation of five representative methods through an experiment using 17,983 Japanese online handwriting character (i.e., Kanji) samples.

To cope with stroke-order variation, several approaches, such as multiple prototype approach, offline feature approach, and stroke correspondence approach, have been proposed. Our investigation shows that, in the past decade, the conventional multiple prototype approach was frequently reported, especially for Hidden Markov Models (HMMs) based methods. This approach is simple and prepares several prototypes with typical stroke-order variations for each class. It might be attributed to the emerging application of HMMs in online handwriting recognition during recent years, for which is short of effective strategy to cope with stroke-order variation. The offline feature approach was also intensively studied in the past decade and gave exciting recognition result. This approach converts the motion trajectory into a 2D image to avoid stroke-order variation problem. The stroke correspondence approach, which was intensively studied in 1980s and 1990s, is the traditional strategy to cope with stroke-order variation. This approach first optimizes the stroke correspondence between an input pattern and a reference pattern and then evaluates the dissimilarity between those patterns. However, few studies on stroke correspondence approach were reported during recent years.

In this paper, we focus on the stroke correspondence approach. Although the multiple prototype and offline feature approaches were frequently reported during recent years because of their promising results, the stroke correspondence approach still has the following unreplaceable characteristics:

- It is theoretically the most reliable approach among several approaches, because it can give the globally optimal or suboptimal stroke correspondence.
- It is possible to estimate the actual stroke-order of the input pattern explicitly and thus useful to analyze stroke-order variation, which is important information to identify the writer, to give a hint for more beautiful handwriting, and so on.
- The stroke correspondence approach can cope with even rare stroke-order variations of each class.
- The technique to establish the stroke correspondence is still very important because it can be applied into the various fields that need order analysis of feature sequences, such as DNA analysis, gesture recognition, etc.

Since the stroke correspondence approach is formulated as an optimization problem, its characteristics depend on its optimization criterion and optimization strategy. Several research groups have proposed promising stroke correspondence-based methods for stroke-order free online handwriting recognition, based on different formulations of the optimization problem.

In this paper, a comparative study is made to clarify the relative superiority of representative stroke correspondence-based methods from the viewpoints of recognition accuracy and efficiency. Specifically, we pick up the following five methods:

- *The cube search method* (CS) by Sakoe and his colleagues [1].
- *The bipartite weighted matching method* (BWM) by Hsieh et al. [2].
- *The individual correspondence decision method* (ICD) by Wakahara and his colleagues [3,4].
- *The stable marriage method* (SM) by Yokota et al. [5].
- *The deviation-expansion model method* (DE) by Lin et al. [6].

Note that our comparative experiments adhere to “stroke-number fixed" condition, where no stroke concatenation occurs. This is because we try to concentrate on the influence of stroke-order variations on recognition performance of those methods.

In the following, we begin with a brief review on the methods of online handwriting recognition in Section 2. In Section 3, the basic principle of stroke correspondence search and some details of the above five methods are summarized. Our test results will be presented in Section 4. Section 5 gives our analysis and an extensive discussion on the five methods, and Section 6 provides some concluding remarks.

2 Review of stroke-order free methods of online handwriting recognition

In online handwriting recognition, the problem of stroke-order variation has been studied for many years. Various methods have been published to solve the problem [7]. As shown in Fig. 1, we roughly divide the methods of dealing with stroke-order variations into three classes, i.e., multiple prototype, offline feature and stroke correspondence. Furthermore, we divide the stroke correspondence approach into two classes, i.e., optimal approach and suboptimal approach.

2.1 Multiple prototype approach

The multiple prototype approach is the simplest approach to cope with stroke-order variations. In this approach, several prototypes with different stroke-order variations are prepared for every class. In general, the patterns with typical stroke-order variations are collected as the prototypes. When recog-

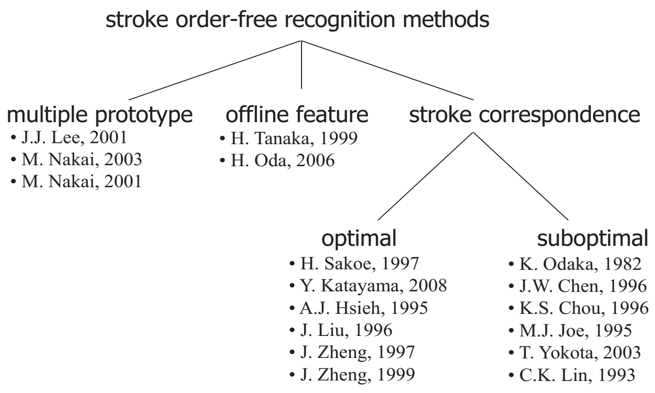


Fig. 1 Classification of methods of dealing with stroke-order variation.

nizing an input pattern, the most similar prototype is simply selected and then the input pattern is recognized as the class of the prototype.

Lee et al. [8] have proposed a method based on the multiple prototype approach using HMMs. First, each online character pattern is represented as a sequence of straight-line segments (i.e., stroke), and modeled by HMM. Then, typical stroke-orders of each class are selected by clustering the training pattern of the class. For each of the selected stroke-order, an individual HMM is prepared. Consequently, the multiple HMMs are prepared for each class. Further, for efficient representation of variations, those HMMs are combined to form a single HMM architecture, called a multiple parallel-path HMM [8].

In [9], Nakai et al. proposed a multiple prototype approach with a hierarchical structured dictionary. In this dictionary, the common subpattern stroke-orders are for reducing higher time complexity drastically. The dictionary is represented as a network and constructed hierarchically by using shared subnets which define stroke-order rules of Kanji subpatterns. Based on the dictionary network, thousands of stroke-order variations of Kanji patterns can be produced using a small number of subpattern stroke-order rules. They successfully applied it to their previous study of substroke-based HMM [10], and showed that the recognition speed was fast.

The merit of the multiple prototype approach is its simplicity. However, it is clear that it is difficult to define typical stroke-order variations for general use. In addition, it can not cope with rare stroke-order variations. Furthermore, its time complexity is large when allowing huge variations.

2.2 Offline feature approach

The offline feature approach employs “inking" process which converts an online character pattern into a 2D image. Then we can apply OCR techniques to recognize the online character patterns without dealing with stroke-order variations explicitly.

Tanaka et al. [11] converted an online character pattern into a bitmap image, and then, an offline classifier was applied to recognize it. In order to improve recognition accuracy, a classifier combination strategy is adopted, where this offline classifier and another online classifier are combined. The offline classifier works as a coarse classifier to provide candidates for the online classifier, so that the time complexity of the online classifier can be decreased.

Oda et al. [12] also proposed a combined classifier, which is composed of an online classifier and an offline classifier. For the latter classifier, the online character patterns are converted to bitmap images. In [12], in order to decrease time complexity and memory size, they proposed several approaches to reduce the dictionary sizes of the online and offline classifiers significantly, especially for the offline dictionary. Specifically, for the online classifier, they employed a structured character pattern representation dictionary, which is based on the common subpatterns of Kanji patterns, to reduce the total memory size. For the offline classifier, the modified quadratic discriminant function (MQDF2) is used, and its dictionary size is drastically reduced by reducing parameters for MQDF2.

2.3 Stroke correspondence approach

The stroke correspondence approach determines the optimal or suboptimal stroke correspondence between an input pattern and a reference pattern. By estimating the actual stroke-order of the input pattern, it is possible to eliminate the effect of stroke-order variations, even when rare stroke-order variation occurs. In addition, the stroke-order information estimated by the stroke correspondence approach can be utilized to identify the writer, to give a hint for more beautiful handwriting, and so on.

In the following, we will briefly introduce some methods based on the stroke correspondence approach. As shown in Fig. 1, those methods are classified into two classes, that is, methods for optimal stroke correspondence and the methods for suboptimal stroke correspondence. Generally speaking, the former gives more accurate correspondence with larger computations (This fact will be first proven experimentally by this paper). Note that among the following methods, CS, BWM, ICD, SM, and DE, will be detailed in Section 3, then compared experimentally and discussed in Sections 4 and 5.

In this paper, the classification between optimal and suboptimal is done whether a method can give the globally optimal solution of the stroke correspondence problem formulated later. Thus, some method becomes suboptimal if it gives an optimal solution in, for example, a constrained problem.

2.3.1 Methods for optimal stroke correspondence

In CS [1], Sakoe and Shin formulate the stroke correspondence search problem as a shortest path search problem on an N-dimensional cube graph. This cube graph is introduced to impose the bijection property on the stroke correspondence between an N-stroke input pattern and an N-stroke reference patterns. An efficient dynamic programming (DP) algorithm is used to search for the shortest path on the cube graph. Note that CS can be extended to be a stochastic stroke correspondence model [13]. In this case, the cube graph becomes an HMM.

Hsieh et al. [2] proposed a straightforward stroke correspondence method by defining the stroke matching problem as BWM, where the stroke distances are served as the weights of edges of an $N \times N$ bipartite graph. The BWM problem is to find the minimum weight perfect matching of the bipartite graph. Hsieh et al. formulated this minimization problem as integer programming (IP) problem, and applied the Hungarian algorithm to solve this IP problem.

In [14], Liu et al. represented both of the reference pattern and the input pattern as complete attributed relational graphs (ARG). In a complete ARG, its nodes and arcs describe line segments and relations between any two segments, respectively. They defined a measure for the optimal correspondence of nodes and arcs between two ARGs. The correspondence searching is formulated as a heuristic search problem in a state space tree. An A* algorithm is used to perform the heuristic search.

In [15], to overcome the shortcoming of being too strict description of primitive (strokes or line segments) relationship in popular ARG, Zheng et al. proposed so-called a fuzzy ARG (FARG). In FARG, the attributes of nodes and/or arcs are represented with fuzzy sets. The FARG is employed to define reference patterns for Chinese characters. An elastic matching algorithm is used to search the optimal stroke correspondence between an input pattern and the FARG.

In [16], Zheng et al. proposed a path-controlled HMM (PCHMM), which is an improved version of popular HMM-based methods and needs not prepare multiple HMMs for the stroke-order variations of each class. In PCHMM, only one HMM is modeled for each class, corresponding to the optimal state transition path on the state sequence space. A path controlling function is defined for directly controlling the optimal state transition path. The path controlling function can place some constraints on state sequence space, according to the optimal line segment correspondence between input and reference patterns, which is searched by using an A* algorithm.

2.3.2 Methods for suboptimal stroke correspondence

ICD [3] is the simplest method to establish a suboptimal stroke correspondence. It cannot guarantee one-to-one correspondence because it relies on a simple greedy algorithm. Specifically, for each of input strokes, one reference stroke with minimum distance (called stroke distance) is selected. In [17–19], the greedy algorithm is also employed to search the stroke correspondence under different constraints. In [17], under the constraint of predefined rules of reference strokes, such as possible predefined stroke types and invariant geometric features of strokes. In [18], under the constraint of the minimum stroke distance that satisfies some conditions on feature point coordinates and slopes of strokes. In [19], under the constraint of the minimum stroke distance, for which input stroke and reference stroke to be matched must have the same number of segments. Clearly, the above methods cannot avoid one-to-many correspondence.

SM [5], which will be detailed later, is different from other methods because it uses the rank of the stroke distance instead of the value of the stroke distance for the criterion of correspondence optimization. The rank can be considered as a rough approximation of the stroke distance as discussed later. SM has a very efficient algorithm for determining the stroke correspondence under this rank criterion.

DE, proposed by Lin et al. [6] is a model to represent the reference pattern. DE contains prior knowledge of possible writing deviations. Specifically, a tree graph is constructed with limited branches for representing stroke-order variations, and then a DP algorithm is used to search for the optimal path on the tree. Because of the limitation, DE cannot deal with all possible stroke-orders and thus DE is treated as a suboptimal method.

3 Details of five representative methods based on stroke correspondence approach

In this section, we will formulate the optimal stroke correspondence problem. Then, we will pick up and detail five representative methods to solve the problem, from a viewpoint of their optimization strategy. Based on our survey of many stroke correspondence strategies, we find the optimization strategies can be clustered into such five classes, that are, two (globally) optimal methods, CS and BWM, and three suboptimal methods, ICD, SM, and DE.

3.1 General problem of determining optimal stroke correspondence

Let A denote an input pattern with N strokes,

$$\boldsymbol{A} = \boldsymbol{A}_1\boldsymbol{A}_2\ldots\boldsymbol{A}_k\ldots\boldsymbol{A}_N, \tag{1}$$

where $\boldsymbol{A}_k$ is the kth stroke and represented as a sequence of feature vectors. For example, $\boldsymbol{A}_k$ is a sequence of three-dimensional vectors, each of which is comprised of the local direction and $x - y$ coordinates of each feature point. Similarly, let B denote the reference pattern,

$$\boldsymbol{B} = \boldsymbol{B}_1\boldsymbol{B}_2\ldots\boldsymbol{B}_l\ldots\boldsymbol{B}_N. \tag{2}$$

Let $\delta(k,l)$ denote a *stroke distance* between the input stroke $\boldsymbol{A}_k$ and the reference stroke $\boldsymbol{B}_l$. Note that the dimensionalities of $\boldsymbol{A}_k$ and $\boldsymbol{B}_l$ are often different due to their different numbers of feature points. Thus, we generally cannot calculate the simple Euclidean distance between them. Instead, DP-matching distance [1,20] between a pair of strokes has been often utilized, for which the stroke distance is the sum of the distances of matched feature points determined by DP.

Consider a mapping $l = l(k)$ for representing the stroke correspondence between $\boldsymbol{A}_k$ and $\boldsymbol{B}_l$. Under the mapping $l(k)$, the stroke $\boldsymbol{A}_k$ corresponds to $\boldsymbol{B}_{l(k)}$. Thus, the *character distance* between $\boldsymbol{A}$ and $\boldsymbol{B}$ becomes $\sum_k \delta(k, l(k))$. The mapping $l(k)$ should be bijective (one-to-one) from $\{\boldsymbol{A}_k\}$ onto $\{\boldsymbol{B}_l\}$.

The *optimal stroke-order* of $\boldsymbol{A}$ can be obtained as $l(1),\ldots,l(k),\ldots,l(N)$, which will minimize the criterion $\sum_k \delta(k, l(k))$. It is very important to note that stroke correspondence methods try to minimize the criterion $\sum_k \delta(k, l(k))$ in their own ways. Some of them are very efficient suboptimal methods and the others are global optimization methods. Hereafter, we will compare five stroke correspondence-based methods.

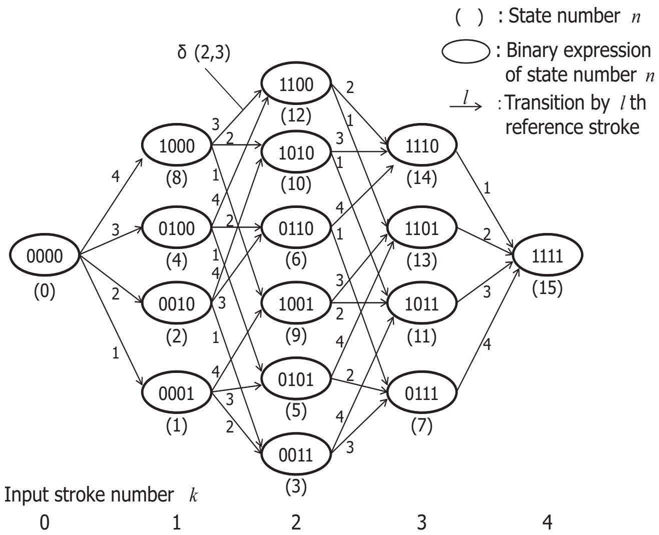


Fig. 2 Cube search graph ($N = 4$).

3.2 Cube search (CS) [1]

CS is a global optimization method for stroke correspondence search. In CS, an N-dimensional cube graph, shown in Fig. 2, is used for representing stroke correspondence; every path from the leftmost node to the rightmost node represents a stroke correspondence $l(1),\ldots,l(k),\ldots,l(N)$ and the graph can represent all (N!) possible correspondences.

Specifically, as shown in Fig. 2, the graph is comprised of 2^N nodes and each of them is indexed by an N-bit binary number; each bit position corresponds to the reference pattern stroke number l, and the bit value 1 means that this reference stroke has already been matched to some input stroke. For example, the node "1100" indicates that the first and the second input strokes corresponds to the third and the fourth reference strokes, respectively, *or*, the fourth and the third reference strokes, respectively.

CS treats the stroke correspondence problem as a global path optimization problem under the condition that the two nodes linked by an edge should have only one different bit. For example, there is an edge from the node "0100" to "1100". This edge indicates that the second input stroke corresponds to the fourth reference stroke. Clearly, any path from the leftmost node "0000" to the rightmost node "1111" will satisfy this condition and represents a bijective correspondence. Since each edge indicates that a specific input stroke corresponds a specific reference stroke, the stroke distance $\delta(k,l)$ is attached to the edge. For example, the distance $\delta(2,4)$ will be attached to the edge from the node "0100" to "1100".

Consequently, the optimal stroke correspondence problem is organized as a minimum distance path problem. The objective function is the summation of $\delta(k,l)$ along the correspondence $l = l(k)$, and the obtained minimum summation is used as character distance. Therefore, the problem is formulated as,

$$D(A,B) = \min_{l(1),\ldots,l(k),\ldots,l(N)} \left[\sum_{k=1}^{N} \delta(k, l(k)) \right]. \tag{3}$$

An efficient DP algorithm is used to search for the "globally" minimum distance path on the cube graph. The recurrence equation of DP is formulated as,

$$G(n) = \min_{m} \left[G(m) + \delta(k,l) \right], \tag{4}$$

where $G(n)$ is the minimum cumulative distance to the state n and there should be the relation that the binary number n has k "1"-bits and two binary numbers m and n are different only at the lth bit. The character distance $D(\boldsymbol{A},\boldsymbol{B})$ is obtained as $G(n)$ at the final state n ="1111". The time complexity of CS is $O(N \cdot 2^{N-1})$.

Since the computational complexity of the original CS is an exponential order of N, we need to introduce some technique to reduce the complexity for practice (especially, for dealing with a multiple stroke character with large N). The *beam search* acceleration technique or pruning technique has been often used at a cost of global optimality. In this paper, we define a pruning threshold of each row k as the summation of the minimum cumulative distance of states in row k and an empirical value. The CS method with beam search (abbreviated as CSBS) provides suboptimal solution with far less computations than CS. Note that if we can consider that each multiple stroke character is comprised of *radicals*, we can expect that there is few stroke-order confusion in different radicals and then reduce computations drastically [21].

3.3 Bipartite weighted matching (BWM) [2]

BWM is also an optimal method for stroke correspondence search. In [2], Hsieh et al. modeled the stroke correspondence problem as a bipartite weighted graph matching problem. Figure 3 shows an example of bipartite weighted graph for determining the optimal stroke correspondence. The kth left vertex stands for the kth input stroke $\boldsymbol{A}_k$, whereas the lth right vertex stands for the lth reference stroke $\boldsymbol{B}_l$. Let the weight of edge $(\boldsymbol{A}_k, \boldsymbol{B}_l)$ be the stroke distance $\delta(k,l)$. A matching in a graph is a set of edges, no two of which share a vertex. If every vertex $\boldsymbol{A}_k \in \boldsymbol{A}$ is incident to an edge in the set, then the matching is perfect.

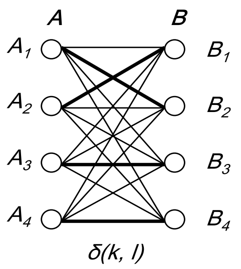


Fig. 3 A weighted bipartite graph ($N = 4$). A perfect matching is marked by thick lines.

BWM problem is to find the globally optimal perfect matching such that the sum of the weights (i.e. stroke distances) of matching edges is minimum. Under the assumption that the stroke distances are given, Hsieh et al. [2] formulated this minimization problem by the following IP:

$$D(A,B) = \min_{x_{11},\ldots,x_{kl},\ldots,x_{NN}} \left[\sum_{k=1}^{N} \sum_{l=1}^{N} \delta(k,l) x_{kl} \right]. \tag{5}$$

where x_{kl} is a binary value (0 or 1) and satisfy $\sum_{k=1}^{N} x_{kl} = 1$ for all l and $\sum_{l=1}^{N} x_{kl} = 1$ for all k. Note that $x_{kl} = 1$ when $\boldsymbol{A}_k$ is matched with $\boldsymbol{B}_l$.

In [2], this IP problem is solved by the Hungarian method [22–24], the well-known primal-dual algorithm. By applying the Hungarian method, the algorithm solves the N strokes correspondence problem in $O(N^3)$ computations.

3.4 Individual correspondence decision (ICD) [3, 4]

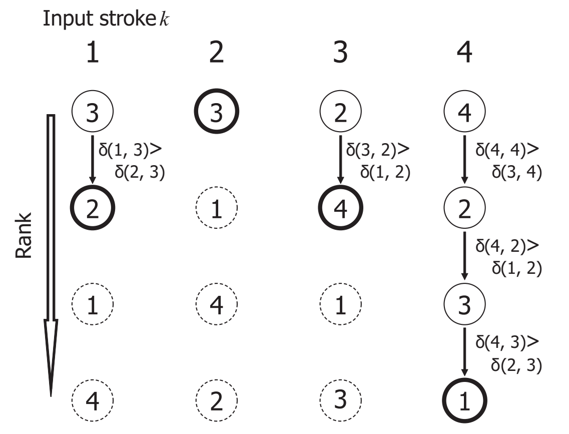
ICD is a suboptimal method for stroke correspondence search. In ICD, the stroke $\boldsymbol{B}_l$ corresponding to the stroke $\boldsymbol{A}_k$ is determined independently. First, an $N \times N$ stroke distance matrix whose element equals $\delta(k,l)$ is defined. Figure 4 (a) shows an example of the matrix. Then, the minimum value in each row k of the matrix is searched for. That is, for the kth input stroke, the lth reference stroke with the minimum stroke distance from k is determined as the corresponding stroke (e.g., in Fig. 4 (a), ③ = $l(1)$). Consequently, the minimized character distance $D(\boldsymbol{A},\boldsymbol{B})$ can be described as

$$D(\boldsymbol{A},\boldsymbol{B}) = \sum_{k=1}^{N} \min_{l} \delta(k, l(k)). \tag{6}$$

There is no guarantee that the resulting stroke correspondence becomes one-to-one as shown in the example of Fig. 4 (a). The time complexity of ICD is $O(N^2)$.

Input stroke k	Reference stroke l			
	①	②	③	④
1	**62**	14★	$\underline{13}$	87
2	61	139	$\underline{\mathbf{10}}$★	86
3	87	$\underline{\mathbf{25}}$	97	36★
4	158★	64	97	$\underline{\mathbf{42}}$

(a) The stroke distance table of the example. The stroke correspondences by ICD and SM are indicated by underlines and ★, respectively. The optimal one-to-one correspondence is indicated by boldface.



(b) Finding the stroke correspondence by SM ($N = 4$).

Fig. 4 An example of finding the stroke correspondence.

3.5 Stable marriage (SM) [5]

According to the formulation of Section 3.1, SM is a suboptimal searching method. The principle of SM is often explained by the situation that N men and N women have expressed mutual preferences (each man must say how he feels about each of the N women and vice versa). The SM problem is to find a set of N stable marriages according to the preferences [25]. For stable marriages, we should avoid unstable marriage, which is the situation that in two married couples (M_1, F_1) and (M_2, F_2), M_1 prefers F_2 to F_1 and F_2 prefers M_1 to M_2.

Yokota et al. [5] applied SM to the stroke correspondence problem, where input strokes $\{\boldsymbol{A}_k\}$ and reference strokes $\{\boldsymbol{B}_l\}$ stand for men and women, respectively. A natural way to express the preferences is the stroke distance $\delta(k, l)$. The process of determining SM for Fig. 4 (a) is illustrated in Fig. 4 (b). For example, when $k = 1$, the reference stroke ③ is first selected as the candidate stroke and matched with the input stroke 1. Then, when $k = 2$, since $\delta(2, 3) < \delta(1, 3)$, the input stroke 2 will match with the reference stroke ③ and the input stroke 1 has to select the lower order of reference stroke ②.

The most important point is that SM does not use the value $\delta(k, l)$ directly but uses the rank of the preference for establish the correspondence. This fact can be confirmed by the fact that if we transform the distance values in Fig. 4 (a) into the rank (the resulting table is called “preference list"), the resulting correspondence is the same. Because of this property, SM is treated as a suboptimal method in this paper. Using $l(k)$ by the above method, the character distance is given as $\sum_k \delta(k, l(k))$. The time complexity of SM is $O(N^2)$.

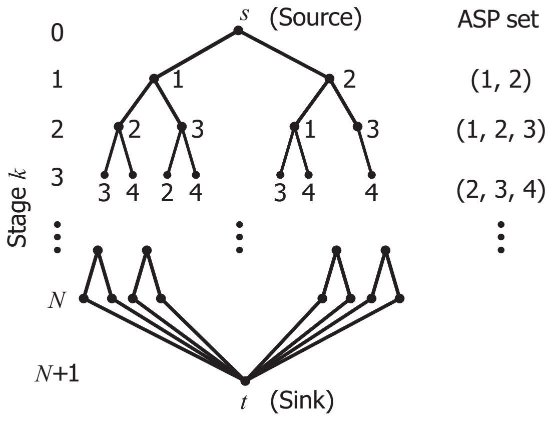


Fig. 5 Tree for DE.

3.6 Deviation-expansion model (DE) [6]

In this paper, DE is also viewed as a suboptimal method because it practically can deal with only a limited number of stroke-orders as follows. Lin et al. [6] proposed DE as a general concept of stroke correspondence model. Unfortunately, in [6], they gave no solution algorithm except for a very limited case, where stroke-order deviation is limited to the two directly adjacent orders. The correspondence algorithm for this limited case is briefly given below.

Figure 5 shows an example of the tree for DE. In DE, the corresponding stroke for k is limited as $l \in \{k-1, k, k+1\}$. This set is called adjacency stroke position (ASP). The branch of the tree is constrained by ASP. A stroke distance $\delta(k, l)$ is assigned at each node of the tree. In [6], the minimum distance path on the tree is searched for by DP[1]. Since the tree branches exponentially, the time complexity of DE is $O(2^N)$.

3.7 Comparison of computational complexity

The time complexities of the above five methods are summarized in Table 1. It is clear that ICD, SM, and BWM require polynomial computations and have the relative superiority over the other methods. Note that CSBS has a parameter (the

[1] Intuitively speaking, this is a “reversed" tree search where the searching process starts from leaves and ends in the root.

Table 1 Performance summary of the methods of stroke correspondence.

Methods	Solution	Time complexity	Recognition rate(%)	Computation time per character(ms)	Rates of character samples with perfect stroke correspondences (%): Correct stroke -order samples	Rates of character samples with perfect stroke correspondences (%): Incorrect stroke -order samples
DM	—	None	93.38	< 0.1	100.00	0.00
CS [1]	Optimal	$O(N \cdot 2^{N-1})$	99.17	553.8	98.34	95.75
CSBS [1]	Suboptimal	Depends on pruning threshold	99.15	43.5	98.32	95.53
BWM [2]	Optimal	$O(N^3)$	99.17	6.4	98.34	95.75
ICD [3]	Suboptimal	$O(N^2)$	96.37	0.4	68.55	54.02
SM [5]	Suboptimal	$O(N^2)$	98.54	2.2	83.67	73.66
DE [6]	Suboptimal	$O(2^N)$	96.59	1212.9	98.62	47.79

$*$ Note that the direct matching method (DM), which establishes stroke correspondence directly by the original stroke-order, is also listed here for emphasizing the importance of stroke correspondence optimization.

threshold for beam search) and its computational complexity depends on the parameter. Note that their actual computation times will be compared experimentally in the next section.

4 Comparative experiment and evaluation

4.1 Experimental setup

In order to compare the performance of the above methods, we conducted a comparative recognition experiment on a workstation (with a 1.7GHz CPU). In Section 4.2, we will first observe the accuracy of the stroke correspondence by the methods. We also observe the recognition accuracy by the methods.

Since we want to analyze the performances of the methods on two sets of correct and incorrect stroke-order respectively, a dataset with known stroke-orders of characters was employed, rather than a common dataset like the well-known Kuchibue database [26] or the recent CASIA database [27]. A total of 17,983 Kanji character patterns (882 categories, 1-20 strokes) written by 30 persons were used in the dataset. The stroke-order of each pattern was confirmed detailedly, and 14,786 patterns were written with correct stroke-order, 3,197 patterns were written with incorrect stroke-order. Figure 6 shows some Kanji samples of this dataset. Again, we adhered to exclude variations in stroke-numbers for understanding only the effect of stroke-order variations in stroke correspondence. Accordingly, each character in the dataset was written with its correct (standard) stroke-number.

Each stroke ($\boldsymbol{A}_k$ or $\boldsymbol{B}_l$) was represented as a sequence of the $x - y$ coordinate feature and the (quantized) directional feature. The stroke distance $\delta(k, l)$ were calculated by DP-matching and all of the methods used the same $\delta(k, l)$ for optimizing the stroke correspondence.

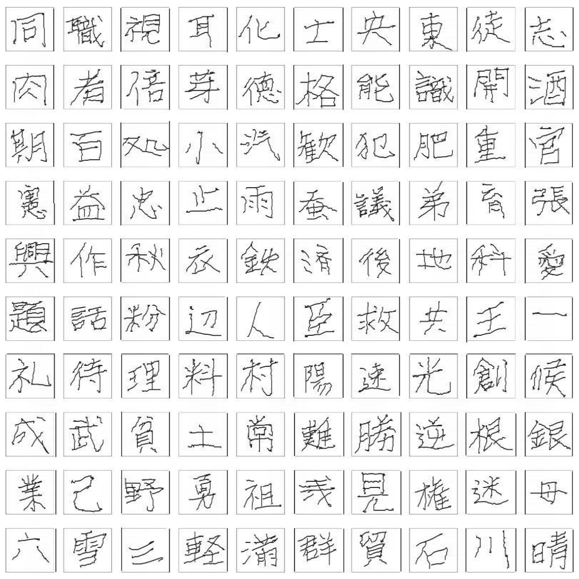


Fig. 6 Samples of our dataset.

4.2 Experimental results

The accuracies of the stroke correspondence by the five methods are shown in Table 1. Specifically, this is the rate of characters with perfect stroke correspondences. The test patterns used were written in not only correct stroke-order but also incorrect stroke-order.

For emphasizing the importance of stroke correspondence optimization, the results of the direct matching method (DM), are also listed in Table 1. DM establishes stroke correspondence directly by the original stroke-order; that is, $k = l(k)$. Accordingly, DM cannot deal with stroke-order variations.

Table 1 also shows the experimental results showing recognition rates and computation times for each of the five methods. Note that 39ms for calculating local distances $\{\delta(k, l)\}$ were excluded from the computation times in the table because this 39ms were common for all of the methods.

Table 2 The error rates of character recognition depending on N.

N	♯ of categories	Character error rate (%)						
		DM	**CS**	**CSBS**	**BWM**	**ICD**	**SM**	**DE**
1	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	10	8.39	2.80	2.80	2.80	2.80	2.80	2.80
3	19	9.79	2.29	2.29	2.29	3.54	2.29	3.33
4	45	7.51	1.34	1.34	1.34	3.75	1.43	3.04
5	65	10.64	1.47	1.47	1.47	3.31	1.47	4.59
6	65	8.93	0.98	1.04	0.98	3.65	1.50	5.28
7	79	8.27	1.18	1.12	1.18	5.01	1.91	4.16
8	94	9.61	0.76	0.76	0.76	4.55	1.42	4.92
9	86	4.40	0.77	0.77	0.77	3.19	1.43	2.97
10	80	6.11	0.46	0.53	0.46	4.25	1.53	3.19
11	77	4.80	0.47	0.47	0.47	2.47	1.47	3.80
12	86	3.72	0.31	0.25	0.31	4.72	1.51	2.52
13	51	2.99	0.43	0.43	0.43	2.56	1.28	1.18
14	46	3.55	0.39	0.66	0.39	2.37	0.79	0.92
15	27	2.83	0.00	0.00	0.00	2.12	0.24	0.94
16	16	0.42	0.00	0.00	0.00	0.84	0.00	0.00
17	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18	11	6.38	0.71	0.71	0.71	3.55	2.13	2.84
19	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20	3	2.13	0.00	0.00	0.00	0.00	0.00	0.00

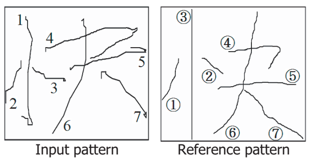
To analysis the influence of the number of input strokes on recognition rate, the error rates depending on the number of input strokes N were evaluated for each of the methods and shown in Table 2. Table 2 also shows the number of character categories on N in our dataset.

4.3 Performance analysis

Table 1 shows that CS (including CSBS) and BWM could provide accurate stroke correspondence for both correct and incorrect stroke-order input patterns. In particular, for input patterns with incorrect stroke-order, CS (including CSBS) and BWM show remarkable superiority than the other methods.

Figure 7 (b) gives examples of stroke correspondences between an input pattern "快"(cheerful) and a reference pattern "快" shown in Fig. 7 (a), by CS, BWM, ICD, SM, and DE. Figure 7 (c) shows the stroke distance table between the two patterns. In this case, as shown in Fig. 7 (b), CS and BWM gave perfect stroke correspondences. The stroke correspondence of ICD was not one-to-one, and SM and DE also gave incorrect stroke correspondences. Based on Fig. 7 (c), SM gave the incorrect correspondences of input strokes 4, 5, and 7, because SM only determined their correspondences locally by iteratively comparing the preferences in their ranks, and could not avoid correction of the errors by considering global optimality. DE could not give the correct correspondence because the stroke-order variation in the input pattern had exceeded the expected variation range (±1).

Table 1 shows that all stroke correspondence search meth-



(a) An input pattern "快"(cheerful), a reference pattern "快", and their stroke-orders.

Correspondence of reference stroke	Input stroke							Character distance
	1	**2**	**3**	**4**	**5**	**6**	**7**	
CS	$\underline{\text{③}}$	$\underline{\text{①}}$	$\underline{\text{②}}$	$\underline{\text{④}}$	$\underline{\text{⑤}}$	$\underline{\text{⑥}}$	$\underline{\text{⑦}}$	92
BWM	$\underline{\text{③}}$	$\underline{\text{①}}$	$\underline{\text{②}}$	$\underline{\text{④}}$	$\underline{\text{⑤}}$	$\underline{\text{⑥}}$	$\underline{\text{⑦}}$	92
ICD	$\underline{\text{③}}$	$\underline{\text{①}}$	$\underline{\text{②}}$	$\underline{\text{④}}$	④	$\underline{\text{⑥}}$	⑤	75
SM	$\underline{\text{③}}$	$\underline{\text{①}}$	$\underline{\text{②}}$	⑦	④	$\underline{\text{⑥}}$	⑤	150
DE	①	③	$\underline{\text{②}}$	$\underline{\text{④}}$	$\underline{\text{⑤}}$	$\underline{\text{⑥}}$	$\underline{\text{⑦}}$	131

(b) Stroke correspondences and character distances given by each method for the character pair of Fig. 7 (a). The correct correspondence is indicated by the underline.

Input stroke	Reference stroke						
	①	②	③	④	⑤	⑥	⑦
1	15	44	10	95	96	38	90
2	3	21	37	70	62	72	61
3	13	4	65	29	18	70	42
4	98	63	121	23	59	79	98
5	74	35	128	10	23	90	62
6	37	43	20	68	66	7	63
7	49	37	75	27	18	53	22

(c) Stroke distance table between the character pair of Fig. 7 (a).

Fig. 7 Stroke correspondence examples by the five methods.

ods can give higher recognition accuracy than DM and proves the importance of stroke correspondence search. Table 1 also shows that CS (including CSBS) and BWM can obtain higher recognition accuracy, and ICD, SM, and BWM can obtain higher recognition speed. On the other hand, although CSBS is slower than ICD, SM, and BWM, its recognition speed is also fast and practical. DE is very slower than the other methods as expected from its exponential computations, $O(2^N)$.

In Table 2, CS and BWM show the same lowest error rates of character recognition, and the poor correspondence accuracy of SM, DE, and ICD caused many errors. It is observed that, for all of the methods, there exists a tendency that the error reduces as N increases (the sharper variation at $N = 18$ is just due to the small number of input patterns at there). It seems reasonable to conclude that the information conveyed by a character pattern increases as the stroke-number increases, and sufficient information is available to discriminate characters. With CS and BWM, error rates monotonically reduce and keep the lowest error rate at each N. Especially at large values of N, CS and BWM show remarkable superiority. It means that, for the complex input Kanji with large stroke-number, CS and BWM can utilize the increased information more efficiently than ICD, SM, and DE.

For the above methods, we can summarize the experimental results as follows.

- Since CS and BWM can obtain the globally optimal stroke correspondence (from all possible correspondences), they can give the same optimal stroke correspondence and obtain the same highest recognition accuracy, especially for difficult input patterns, that is, patterns with incorrect stroke-order and/or large stroke-number.
- There is only a slight difference in perfect stroke correspondence rate, as well as recognition rate, between CSBS and CS. In contrast, there is a drastic reduction in computation time by CSBS. This indicates that, although CSBS is a suboptimal method by the introduction of beam search (pruning), it is possible to find the optimal solution in many cases.
- ICD cannot give accurate stroke correspondence as shown in Table 1. This also results in the poor recognition rate 96.37%. Those facts indicate that the stroke correspondence problem is difficult to solve by a simple greedy optimization method.
- SM could not provide accurate stroke correspondence. This is because only SM employs a rank criterion instead of the distance criterion. The rank criterion sometimes underestimates large difference and overestimates small difference, and thus is insufficient to evaluate the stroke correspondence.
- DE could not provide correct stroke correspondence results for incorrect stroke-order input. This is because that DE cannot deal with stroke-order variation beyond the range permit of model assumption. Unfortunately, it is practically impossible enlarge the range because DE graph becomes huge and then its computation becomes intractable.

From above results, clearly, for the performance of recognition accuracy, relative superiority of CS and BWM over ICD, SM, and DE are established.

5 Discussions

Since the above mentioned methods can give different performances in different conditions of stroke-order and stroke-number of input patterns, they have their own appropriate characters. For the optimal methods of CS and BWM, it is reasonable to employ them to recognize the characters with large stroke-number, such as Chinese character or Japanese Kanji (or employ them in the other pattern recognition fields, eg., the recognition of gestures with complex and variable orders of bodily motion). For the suboptimal methods of ICD and SM, it is reasonable to employ them to recognize the characters (or other kinds of patterns) with small stroke-numbers, such as Roman and Japanese Kana. However, application of ICD and SM to the Indian characters like Bangla [28] will not be appropriate even though they have small stroke-number. This is because the shapes of Bangla character strokes are very similar to each other, globally optimal stroke correspondence will be necessary to keep the higher recognition accuracy than ICD and SM. CS and BWM are also advised for the characters with large alphabet, such as Indian characters, Kanji characters or subpatterns of Kanji. Since the recognition accuracy always decreases as the alphabet increases, the optimal methods of CS and BWM are needed to keep the higher recognition accuracy, for the large alphabet characters. On the other hand, considering the recognition speed (or computational complexity), the ICD, SM, and BWM are advised, especially for those simple recognition systems (e.g., portable data terminal). Table 3 shows our recommendation of the application conditions for each of the methods.

The above mentioned methods of stroke correspondence approach also can combine with the methods of multiple pro-

Table 3 The application condition for each method.

Methods	**Application condition**			
	♯ of strokes	**Alphabet size**	**Speed**	**Ex. of character script**
CS (CSBS)	Large / Small	Large	Middle	Chinese, Kanji, Bangla
BWM	Large / Small	Large	Fast	Chinese, Kanji, Bangla
ICD	Small	Small	Fast	Roman, Kana
SM	Small	Small	Fast	Roman, Kana
DE	—	—	Slow	—

totype and offline feature approach to decrease the total time complexity and improve the recognition accuracy. CS and BWM give stroke correspondence with higher accuracy, between the input pattern and the reference pattern of a class. Their stroke correspondence results can help the methods of multiple prototype approach to pick up some more possible prototypes so as to decrease the complexity, rather than all of the prototypes of the class, especially for the class with large stroke-number. On the other hand, for methods of offline feature approach, it seems reasonable to employ the information of one-to-one stroke correspondence, given by above methods like CS, BWM, and SM, to evaluate the image character distances and improve the recognition accuracy.

6 Conclusion

In this paper, we have reviewed the methods for solving the stroke-order variation problem. Among various approaches for dealing with this problem, we have focused on the stroke correspondence search methods. Especially, five representative methods — CS, BWM, ICD, SM, and DE, were discussed and compared to clarify their relative superiority. From the viewpoints of not only recognition accuracy but also stroke correspondence accuracy, the five methods have been experimentally compared on the same test set. According to the experimental results under the stroke-number fixed condition, performance superiorities of CS and BWM over ICD, SM, and DE were established. This indicates that global optimal solution under a stroke distance-based criterion is necessary for higher stroke correspondence accuracy. A detailed discussion on the five methods showed that these methods could be applied to different conditions of character recognition tasks, and combined with other approaches for dealing with stroke-order variation problem.

In our future work, we will test the above mentioned methods on different types of datasets with different languages, to further prove and deepen our analysis. Meanwhile, we should study how to apply the stroke correspondence technique into other fields that need order analysis of feature sequences, such as DNA analysis and gesture recognition.

References

1. H. Sakoe, and J. Shin, A Stroke Order Search Algorithm for Online Character Recognition, Research Reports on Information Science and Electrical Engineering of Kyushu University, Vol.2, No.1, pp. 99-104, 1997 (in Japanese).
2. A.J. Hsieh, K.C. Fan, and T.I. Fan, Bipartite Weighted Matching for On-line Handwritten Chinese Character Recognition, Pattern Recognition, Vol. 28, No. 2, pp. 143-151, 1995.
3. K. Odaka, T. Wakahara, and I. Masuda, Stroke Order Free On-line Handwritten Character Recognition Algorithm, IEICE Transactions on Information and Systems, Vol. J65-D, No. 6, pp. 679-686, 1982 (in Japanese).
4. T. Wakahara, H. Murase, and K. Odaka, On-Line Handwriting Recognition, Proceedings of the IEEE, Vol. 80, No. 7, pp. 1181-1194, 1992.
5. T. Yokota *et al.*, An On-line Cuneiform Modeled Handwritten Japanese Character Recognition Method Free from Both the Number and Order of Character Strokes, Transactions of Information Processing Society of Japan, Vol.44, No.3, pp. 980-990, 2003 (in Japanese).
6. C.K. Lin, K.C. Fan, and F.T.P. Lee, On-line Recognition by Deviation-expansion Model and Dynamic Programming Matching, Pattern Recognition, Vol. 26, No. 2, pp. 259-268, 1993.
7. C.L. Liu, S.Jaeger, and M. Nakagawa, Online Recognition of Chinese Characters: The State-of-the-Art, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 26, No. 2, pp. 198-213, 2004.
8. J.J. Lee, J. Kim, and J.H. Kim, Data-driven design of HMM topology for online handwriting recognition, International Journal of Pattern Recognition and Artificial Intelligence, Vol. 15, No. 1, pp. 107-121, 2001.
9. M. Nakai, H. Shimodaira and S. Sagayama, Generation of Hierarchical Dictionary for Stroke-order Free Kanji Handwriting Recognition Based on Substroke HMM, Proceedings Seventh International Conference on Document Analysis and Recognition, pp. 514-518, 2003.
10. M. Nakai, N. Akira, H. Shimodaira, and S. Sagayama, Substroke Approach to HMM-Based On-Line Kanji Handwriting Recognition, Proceedings Sixth International Conference on Document Analysis and Recognition, pp. 491-495, 2001.
11. H. Tanaka, et al., Hybrid Pen-Input Character Recognition System Based on Integration of Online-Offline Recognition, Proceedings Fifth

International Conference on Document Analysis and Recognition, pp. 209-212, 1999.
12. H. Oda, et al., A Compact On-line and Off-line Combined Recognizer, Proceedings 10th International Workshop on Frontiers in Handwriting Recognition, pp. 133-138, 2006.
13. Y. Katayama, S. Uchida, and H. Sakoe, A New HMM for On-Line Character Recognition Using Pen-Direction and Pen-Coordinate Features, Proceedings 19th International Conference on Pattern Recognition, 2008.
14. J. Liu, W.K. Cham, and M.M.Y. Chang, Stroke Order and Stroke Number Free On-Line Chinese Character Recognition Using Attributed Relational Graph Matching, Proceedings 13th International Conference on Pattern Recognition, vol. 3, pp. 259-263, 1996.
15. J. Zheng, X. Ding, and Y. Wu, Recognizing On-Line Handwritten Chinese Character via FARG Matching, Proceedings Fourth International Conference on Document Analysis and Recognition, pp. 621-624, 1997.
16. J. Zheng, X. Ding, Y. Wu, and Z. Lu, Spatio-Temporal Unified Model for On-Line Handwritten Chinese Character Recognition, Proceedings Fifth International Conference on Document Analysis and Recognition, pp. 649-652, 1999.
17. J.W. Chen and S.Y. Lee, On-Line Handwriting Recognition of Chinese Characters via Rule-Based Approach, Proceedings 13th International Conference on Pattern Recognition, vol. 3, pp. 220-224, 1996.
18. K.S. Chou, K.C. Fan, and T.I. Fan, Radical-Based Neighboring Segment Matching Method for On-Line Chinese Character Recognition, Proceedings 13th International Conference on Pattern Recognition, vol. 3, pp. 84-88, 1996.
19. M. J. Joe. and H. J. Lee, A Combined Method on the Handwritten Character Recognition, Proceedings Third International Conference on Document Analysis and Recognition, pp. 112-115, 1995.
20. H. Sakoe, and S. Chiba, Dynamic Programming Algorithm Optimization for Spoken Word Recognition, IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP-26, No. 1, pp. 43-49, 1978.
21. W. Cai, S. Uchida, and H. Sakoe, An Efficient Radical-Based Algorithm for Stroke-Order-Free Online Kanji Character Recognition, Proceedings 18th International Conference on Pattern Recognition, Vol. 2, pp. 986-989, 2006.
22. H. W. Kuhn, The Hungarian Method for the Assignment Problem, Naval Research Logistics Quarterly, Vol. 2, pp. 83-97, 1955.
23. J. Munkres, Algorithms for the Assignment and Transportation Problems, Journal of the Society for Industrial and Applied Mathematics, Vol. 5, No. 1, pp. 32-38, 1957.
24. C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, Englewood Cliffs, New Jersey, 1982.
25. R. Sedgewick, *Algorithms*, Addison-Wesley, second edition, pp. 499-504, 1988.
26. M. Nakagawa, et al., Collection of on-line handwritten Japanese character pattern databases and their analysis, International Journal on Document Analysis and Recognition, Vol. 7, No. 1, pp. 69-81, 2004.
27. C.L. Liu, F. Yin, D.H. Wang, and Q.F. Wang, CASIA Online and Offline Chinese Handwriting Databases, Proceedings 11th International Conference on Document Analysis and Recognition, pp. 37-41, 2011.
28. K. Roy, N. Sharma, T. Pal, and U. Pal, Online Bangla Handwriting Recognition System, International Conference on Advances in Pattern Recognition, pp.117-122, World Scientific publishing Co., 2007.

Received B.E. and M.E. degrees from Wuhan University, China in 1990 and 1996 respectively, and Dr. Eng. degree from Kyushu University, Japan in 2012. From April 2007, he has been working at O-RID Company, Japan. His research interests include character recognition and image processing. He is a member of the IEEE Computer Society.

Received B.E., M.E., and Dr. Eng. degrees from Kyushu University in 1990, 1992 and 1999, respectively. From 1992 to 1996, he joined SECOM Co., Ltd., Tokyo, Japan where he worked on speech processing. Currently, he is a professor at Faculty of Information Science and Electrical Engineering, Kyushu University. His research interests include pattern recognition and image processing. He received 2002 IEICE PRMU Research Encouraging Award, MIRU2006 Nagao Award (best paper award), 2007 IAPR/ICDAR Best Paper Award, and 2009 IEICE Best Paper Award. Dr. Uchida is a member of IEEE and IPSJ.

Received the B.E. degree from Kyushu Institute of Technology in 1966, and the M.E. and D.E. degrees from Kyushu University in 1968 and 1987, respectively. In 1968, he joined NEC Corporation and engaged in speech recognition research. In 1989, he left NEC Corporation to become a Professor of Kyushu University. He is now a Professor Emeritus of Kyushu University.