Online Recognition of Chinese Characters: The State-of-the-Art

Cheng-Lin Liu, *Member*, *IEEE*, Stefan Jaeger, and Masaki Nakagawa, *Member*, *IEEE Computer Society*

Abstract—Online handwriting recognition is gaining renewed interest owing to the increase of pen computing applications and new pen input devices. The recognition of Chinese characters is different from western handwriting recognition and poses a special challenge. To provide an overview of the technical status and inspire future research, this paper reviews the advances in online Chinese character recognition (OLCCR), with emphasis on the research works from the 1990s. Compared to the research in the 1980s, the research efforts in the 1990s aimed to further relax the constraints of handwriting, namely, the adherence to standard stroke orders and stroke numbers and the restriction of recognition to isolated characters only. The target of recognition has shifted from regular script to fluent script in order to better meet the requirements of practical applications. The research works are reviewed in terms of pattern representation, character classification, learning/adaptation, and contextual processing. We compare important results and discuss possible directions of future research.

Index Terms— Online Chinese character recognition, state-of-the-art, pattern representation, character classification, model learning, contextual processing, performance evaluation.

1 Introduction

In online character recognition, the trajectories of pen tip movements are recorded and analyzed to identify the linguistic information expressed. Owing to the availability of both temporal stroke information and spatial shape information, online character recognition is able to yield higher accuracy than offline recognition. Online recognition also provides good interaction and adaptation capability because the writer can respond to the recognition result to correct the error or change the writing style.

In recent years, new types of pen input devices and interfaces have been developed to improve the precision of trajectory capturing and the comfort of writing. Devices are available for writing on ordinary paper and wireless transmission of handwriting, for example. Powerful software is available now for analyzing and retrieving handwritten documents. This development stimulates new applications of handwriting recognition and has resulted in a renewed interest in research [114]. The applications of online recognition include text entry for form filling and message composition, personal digital assistants (PDA), computer-aided education [90], handwritten document retrieval [77], [107], etc. For handheld devices, pen input is competitive to speech input because it is insensitive to environmental noise, which is an important advantage for many applications. For desktop applications, online recognition is well-suited to text entry for large alphabets (like Oriental languages). A common feature of these applications is that they require high recognition accuracy.

The research of online character recognition started in the 1960s and has been receiving intensive interest from the 1980s. The comprehensive survey of Tappert et al. reviewed the status of research and applications before 1990 [120] and early works of online Japanese character recognition have been reviewed in [85], [132]. A recent comprehensive survey of handwriting recognition, by Plamondon and Srihari, mainly concerns western handwriting [102]. Our paper contributes a survey to online Chinese character recognition (OLCCR) since this recognition problem is very different from western handwriting recognition and it poses a special challenge.

1.1 Related Problems

Pen computing applications closely related to handwriting recognition are mathematical formula recognition [5], [148] and diagram recognition [3], where both the character classes (mostly Latin characters) and the layout are recognized. Another application is signature verification that checks whether a handwritten signature is generated by a specific writer or not. It does not necessarily identify the symbolic classes of a signature's constituent characters though. Signature verification has been reviewed in [61], [102] and recent works are reported in [44], [57]. A form of handwritten document retrieval, the so-called ink matching, does not identify the character classes either [78]. Handwritten sketch recognition is based mostly on noncharacter data and typically ignores linguistic information [82], [109]. We do not cover these problems further because they are not relevant to the methodology of OLCCR.

1.2 Characteristics of Chinese Characters

Chinese characters are used in daily communications by over one quarter of world's population, mainly in Asia. There are mainly three character sets: traditional Chinese characters, simplified Chinese characters, and Japanese

- *C.-L. Liu is with the Central Research Laboratory, Hitachi, Ltd., 1-280 Higashi-koigakubo, Kokubunji-shi, Tokyo 185-8601, Japan. E-mail: liucl@crl.hitachi.co.jp.*
- *S. Jaeger and M. Nakagawa are with the Department of Computer Science, Tokyo University of Agriculture and Technology, 2-24-16 Naka-cho, Koganei-shi, Tokyo 184-8588, Japan. E-mail: stefan@hands.ei.tuat.ac.jp, nakagawa@cc.tuat.ac.jp.*

Manuscript received 27 Sept. 2002; revised 19 May 2003; accepted 11 Aug. 2003.
Recommended for acceptance by K. Yamamoto.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 117468.

Traditional Chinese	姑蘇城外寒山寺 夜半鐘聲到客船
Simplified Chinese	姑苏城外寒山寺 夜半钟声到客船
Japanese Kanji	姑蘇城外寒山寺 夜半鐘声到客船

Fig. 1. Examples of traditional Chinese, simplified Chinese, and Japanese Kanji.

Kanji. Japanese Kanji characters have mostly identical shape to the corresponding traditional Chinese or simplified Chinese. For some Kanji characters, nevertheless, the shape is slightly different from both the traditional and simplified Chinese. Fig. 1 shows some examples of the three character sets. We can see that, among the 14 characters, four have different shapes (the last character only varies slightly, while the fourth one is treated identically in three sets).

In the mainland of China, two character sets, containing 3,755 characters and 6,763 characters, respectively, were announced as the National Standard GB2312-80 (the first set is a subset of the second one) [116]. In Taiwan, 5,401 traditional characters are included in a standard set. In both traditional and simplified Chinese, about 5,000 characters are frequently used. In Japan, 2,965 Kanji characters are included in the JIS level-1 standard and 3,390 Kanji characters are in the level-2 standard (the two sets are disjoint).

A Chinese character is an ideograph and is composed of mostly straight lines or "poly-line" strokes. Many characters contain relatively independent substructures, called radicals, and some common radicals are shared by different characters. This property can be utilized in recognition to largely reduce the size of reference model database and speed up recognition.

Chinese handwritten scripts are classified into three typical styles: regular script, fluent script, and cursive script. The intermediate styles are called fluent-regular script and fluent-cursive script, respectively. Some examples of the three typical styles are shown in Fig. 2. We can see that, in regular script, strokes are mostly straight-line segments. The fluent script has many curved strokes and, frequently, successive strokes are connected. In cursive script, some character shapes totally differ from the standard shape, so it is difficult to recognize them, even for humans.

1.3 The State-of-the-Art

Since the 1990s, the research efforts of OLCCR have been aiming at the relaxation of constraints imposed on writers to ensure successful recognition, namely, the isolation of characters and the compliance with standard shapes. For Chinese characters, the main problem in online recognition is to overcome the stroke-order and stroke-number variability. The target of OLCCR in the 1990s has shifted from regular script to fluent script, which features greater variability of stroke-order and stroke-number and occurs frequently in practical writing.

In the literature of character recognition, the regular style is also referred to as block style or hand-printed style, while the fluent style is often called "cursive" style. The current systems can recognize regular script with high accuracy, whereas the recognition of fluent style still remains unsolved and requires more intensive research efforts. The fluent script or fluent-regular script is the target of most recognition systems because people naturally write this way.

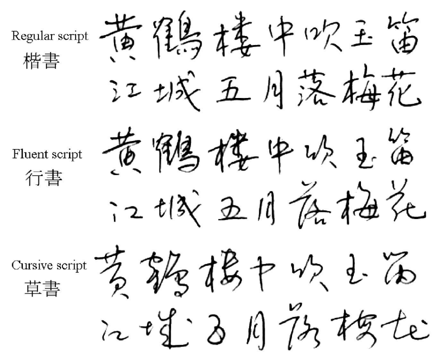

Fig. 2. Chinese writing styles: regular, fluent, and cursive.

The methods of OLCCR can be roughly divided into two categories: structural methods and statistical methods. Structural methods are based on stroke analysis. The character models of structural methods can be further divided into stroke-order dependent models and stroke-order free ones. Statistical methods mainly utilize the holistic shape information, so it is easier to achieve stroke-order independence. From an application's point of view, a recognition method can be writer-dependent or writer-independent. Writer-independent recognition is more challenging due to the diversity of writing styles. On the other hand, writer-dependent recognition allows stable recognition of cursive script due to the relative stability of personal writing styles.

1.4 Contents of the Paper

This survey will emphasize the research efforts from the 1990s. On comparing the performance of state-of-the-art methods, discussing their insufficiencies, we will suggest future research directions. The rest of this paper is organized as follows: Section 2 gives the overview of a typical OLCCR system and Section 3 briefly reviews preprocessing. Sections 4, 5, and 6 address the main tasks of character recognition, namely, pattern representation, classification (including coarse classification and fine classification), and reference model learning/adaptation, respectively. Though in an OLCCR system, the schemes of classification and learning largely depend on that of representation, a scheme in a task can still connect with multiple schemes in another task. Hence, we address the three tasks in separate sections and try to thread the connected schemes in different tasks. Section 7 addresses the contextual processing of character segmentation and recognition. Section 8 compares the performance of representative methods and Section 9 discusses the future research directions.

2 Overview of a Typical OLCCR System

A practical OLCCR system is depicted diagrammatically in Fig. 3. The input to the system is a sequence of handwritten character patterns. First, the handwriting sequence is segmented into character patterns according to the temporal and shape information. Often, the boundary between characters cannot be determined unambiguously before character recognition, so candidate character patterns are generated and recognized and the correct patterns are selected in contextual processing at the end of the process chain. The recognition of segmented (candidate) patterns involves the following steps: preprocessing, description, and

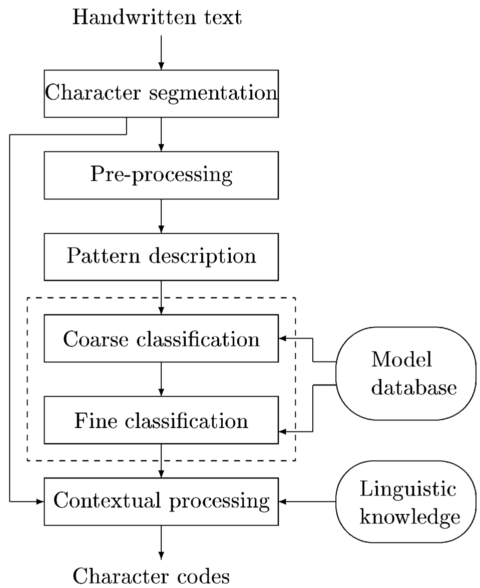

Fig. 3. Diagram of a practical OLCCR system.

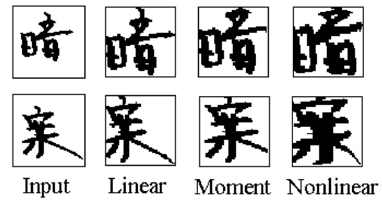

Fig. 4. Examples of linear normalization, moment normalization, and nonlinear normalization.

classification. Classification is often decomposed into coarse classification and fine classification. Pattern description is also referred to as feature extraction, which represents the input pattern either statistically by feature vectors or structurally by various levels of primitives. The model database (also called reference database or recognition dictionary) contains the reference models or classification parameters for coarse classification and fine classification.

To speed up the recognition of the large category set, a fast coarse classification procedure is commonly used to first select a small subset of candidate classes to which the input pattern is expected to belong to. Then, the input pattern is classified into one of these candidate classes in the fine classification stage. This two-stage recognition strategy has been widely adopted by now, though tree classification and multistage classification can further speed up the recognition. In contextual processing, linguistic knowledge and geometric features are used to verify the segmentation and classification results. The performance of character recognition relies largely on the quality of the model database. This database is built from heuristic knowledge, manually selected character prototypes, or from multiple sample patterns. For writer-dependent recognition, the models or parameters can adapt to the writer's style to improve the recognition performance.

3 Preprocessing

The preprocessing of the trajectory of input pattern directly facilitates pattern description and affects the quality of description. The preprocessing tasks of online character patterns include noise elimination, data reduction, and shape normalization. The noise in character trajectories is due to erratic hand motions and the inaccuracy of digitization. The noise reduction techniques used in most systems are basically those explained in [120]: smoothing, filtering, wild point correction, stroke connection, etc. As the quality of input devices steadily advances, trajectory noise becomes less influential and simple smoothing operations will suffice.

Data reduction can be accomplished by two approaches: equidistance sampling and line approximation (feature point detection). With equidistance sampling, the trajectory points are resampled such that the distance between adjacent points is approximately equal. The data amount of equidistance point representation is still appreciable. A higher data reduction rate can be achieved by detecting the corner points of trajectories. The corner points and the ends of a stroke trajectory are often called feature points.

Corner detection from digitized curves has been widely addressed in the shape recognition literature. The basic idea is to estimate the curvature at each point on the curve and retain the points of high curvature [105]. An alternative is polygonal approximation, which recursively finds the vertex of maximum point-to-chord distance [104]. Corner detection and polygonal approximation are complementary and can be combined to achieve better performance [138]. Line approximation of strokes has been used in many online recognition systems (e.g., [38], [55], [63], [145], [146]).

Normalization of character trajectories to a standard size is adopted in almost every character recognition system. Conventionally, the coordinates of stroke points are shifted and scaled such that all points are enclosed in a standard box (this is called linear normalization). Alternatively, by moment normalization [4], the centroid of input pattern is shifted to the center of standard box and the second-order moments are scaled to a standard value.

To alleviate the shape deformation of handwritten Chinese characters, nonlinear normalization was proposed in the 1980s and was proven efficient to improve the accuracy of offline character recognition. It was later successfully applied to online character recognition as well [28], [45], [88], [99], [100]. Nonlinear normalization reassigns the coordinates of stroke points according to the line density distribution with the aim of equalizing the stroke spacing [62], [127], [141]. For comparing the effects of normalization, Fig. 4 shows two character patterns and the results of linear, moment, and nonlinear normalization. It was shown that moment normalization can yield comparable recognition accuracy to nonlinear normalization [70]. For online patterns, the line density can be computed directly from the online trajectory [45], [88], instead of the 2D image.

4 Pattern Representation

The representation schemes of input pattern and model database are of particular importance since the classification method depends largely on them. We divide the schemes into three groups: statistical, structural, and hybrid statistical-structural. In statistical representation, the input pattern is described by a feature vector, while the model database (also called parameter database in this case) contains the classification parameters. The structural representation scheme has

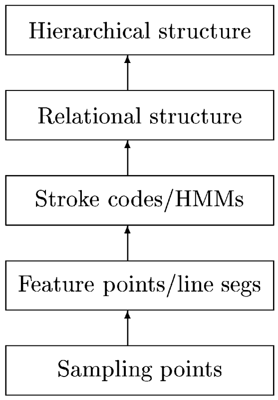

Fig. 5. Hierarchy of structural representation schemes.

long been dominating the OLCCR technology, whereas the statistical scheme and the hybrid scheme are receiving increasing attention in recent years. The statistical-structural scheme is only used for describing the reference models. It takes the same structure as the traditional structural representation, yet the structure elements (primitives) and/or relationships are measured probabilistically. Hidden Markov Models (HMMs) can be regarded as instances of the statistical-structural representation.

The structural representation schemes can be further partitioned into five levels: sampling points, feature points or line segments, stroke codes or HMMs, relational, and hierarchical. Fig. 5 shows the hierarchy of five levels. For describing the input pattern or the model database, the primitives of higher-level structure are composed of the lower-level primitives, e.g., a stroke is recorded by the constituent line segments or sampling points, while the relational structure takes strokes as primitives. We treat HMMs on the same level as stroke code representation because HMMs are frequently used to model strokes or substrokes, while a sequence of HMMs represent a radical or a character.

The feature point representation is equivalent to the line segment representation because every pair of succeeding feature points gives a line segment. In stroke code representation, the types of strokes in input pattern or reference models are specified and the reference model is represented as a sequence of strokes. The relational structure represents a character model or a radical model wherein the relationship between strokes is specified. In the hierarchical representation of model database (also called structured representation), a number of radical models are shared to construct the character models of all categories. On the other hand, the hierarchical structure of input pattern is a relational structure with radicals as primitives. The structured representation of model database is very storage efficient, while the sampling points representation is data intensive.

An OLCCR system can use different representation schemes for the input pattern and the model database, respectively. In the case of structural or statistical-structural representation, the model database is usually described at higher level than the input pattern. In recognition, for example, the input pattern is represented in point sequence or line segments, then strokes and relational structure are extracted by matching the point sequence or line segments with the higher-level primitives of reference models.

In the following, we review the detailed schemes of structural representation, statistical-structural representation, and statistical representation, respectively.

4.1 Structural Representation

We review the structural representation schemes in the order of the levels as shown in Fig. 5.

4.1.1 *Point and Line Segment Representation*

A representation with resampled points can cope well with curved strokes, though it results in large size of model database. If both the input pattern and the character prototype in the model database are represented as point sequences, they can be matched based on stroke correspondence in which the between-stroke distance is computed by aligning the points [132], [133], [134].

The feature point or line segment representation is widely adopted nowadays. It is especially suited to regular-style characters, which are composed of mostly straight-line segments. Input patterns represented as feature point sequences can be matched with character models represented as feature point sequences [13], [55], [86], higher-level primitives (such as stroke codes) [20], or hierarchical structures [60]. Analogously, input patterns represented as line segments can be matched with character models represented as line segments [18], [35], [124] or higher-level structures [16], [139].

4.1.2 *Stroke Code Representation*

Stroke code representation schemes have been adopted from the early stages of OLCCR research (e.g., [145], [146]). The strokes in regular script can be categorized into classes according to the constituent line segment sequence and each class is assigned a code or index number [63], [65]. Each stroke code has a corresponding reference model/prototype or a set of rules. A character model is then represented as a sequence of stroke codes or a relational structure with stroke codes as primitives [14], [65]. When using stroke code-based models in recognition, the stroke codes of input pattern are determined in matching with character/stroke models. In [74], [106], the input pattern is initially represented as line segments and strokes are detected using finite state automaton.

The stroke code representation of [47] is unique in that a connected stroke (a piece of trajectory containing one or multiple normal strokes) is represented as a single feature vector and stroke prototypes are designed by clustering sample patterns. However, because it does not rely on stroke decomposition, this scheme does not generalize to novel stroke shapes not contained in the learning samples.

4.1.3 *Relational Representation*

For stroke-order-free recognition, the primitives (strokes or line segments) of a character and the relationship between them are often represented by means of a relational structure, such as an Attributed Relational Graph (ARG). In an ARG, the nodes denote primitives and the arcs denote the relationships between nodes. If the attributes of nodes and/or arcs are represented with fuzzy sets, the graph is called fuzzy ARG (FARG) [6]. Fig. 6 shows an example of ARG.

Two problems arise in ARG matching of characters. First, the primitives and relation codes of the reference models must be carefully designed to tolerate the shape variation of

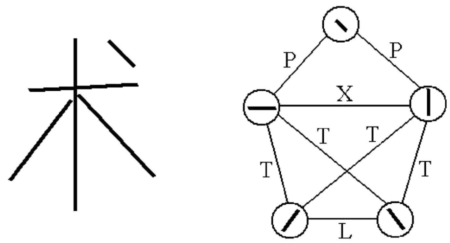

Fig. 6. An example of ARG representation (left: character, right: ARG). The relation codes "X," "T," "L," and "P" stand for intersection, end-to-line adjacency, end-to-end adjacency, and positional relation, respectively.

input patterns. Using fuzzy attributes or multiattribute relation codes can alleviate this problem [7], [10], [71]. Second, the description of input patterns in ARGs is not straightforward because it is often difficult to reliably extract the primitives and their relationships prior to ARG matching. To solve this problem, the input pattern is represented in lower-level primitives (usually line segments), which are grouped into higher-level primitives by matching them with reference ARGs [149]. Because a stroke may contain multiple line segments, using line segments instead of strokes as the primitives of ARGs can facilitate the extraction of primitives from input pattern [72], [73].

4.1.4 Structured Representation

Due to the hierarchical nature of Chinese characters, a character pattern can be described in a tree structure, with the character itself at the top level and the radicals and strokes as low-level primitives. In a structured representation of reference models, the radical models or stroke models are shared by different characters such that a character model is constructed dynamically using the constituent radicals and strokes. This strategy can largely save the storage space of model database, considering the fact that hundreds of distinct radicals are shared by thousands of characters [87].

The character models with shared radical models can be organized in a lookup table [8], [9], [38], [146], a tree structure [11], [75], [76], or a network [47]. Fig. 7 shows an example of network representation of model database. Via stroke and radical extraction from the input pattern, the character model fitting the input pattern can be retrieved by traversing the tree or network. The structured representation approach can vary in the representation scheme of radicals: line segments [75], [76], connected stroke codes [47], and hierarchies of substroke HMMs [91].

4.2 Statistical-Structural Models

In a statistical-structural representation scheme, a character model is described in a string, tree, or graph structure, with the primitives and/or relationships measured probabilistically to better model the shape variations of input patterns. In principle, any structural model can be described probabilistically by replacing the attributes of primitives and/or relationships with probability density functions (PDFs). The mean and variance of stroke and relationship attributes in [76] are connected to PDF representations. Gaussian PDFs have been used for describing the distributions of feature points [20] and stroke attributes [150]. This direction has not been explored adequately yet.

The Hidden Markov Model (HMM) is a directed graph with nodes and between-node transitions measured

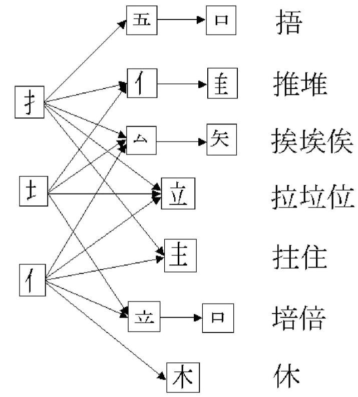

Fig. 7. Network representation of characters with shared radical models. Every path corresponds to a character as shown in the rightmost column.

probabilistically. HMMs have been used in speech recognition since the 1970s [103] and have been applied to western character recognition since the 1980s. Only in recent years have HMMs been applied to Chinese characters. Generally, left-right HMMs are used to model the sequences of points or line segments for substrokes [91], [92], [121], strokes, radicals [48], or whole characters [117], [143]. Since the character-based HMM is stroke-order dependent, multiple models are often generated for the characters with stroke-order variations or large shape variations [117], [143]. Using stroke-based or substroke-based HMMs, the character models can be constructed hierarchically and the stroke-order variations can be represented in a variation network [93].

A constrained ergodic HMM, named path controlled HMM (PCHMM), was proposed to overcome the stroke-order variation of online characters [150]. The successor attribute matrix (SAM) of [58] is similar to ergodic HMMs since it estimates the transition probabilities between strokes.

4.3 Statistical Representation

In the statistical recognition approach, we are mainly concerned with the representation of input patterns (basically in feature vectors). The model database contains the classification parameters, which can be estimated by standard statistical techniques [22], [25]. We will review the statistical classification techniques in Section 5.

The feature vector representation of character patterns enables stroke-order and stroke-number free recognition by, for example, mapping the pattern trajectory into a 2D image and extracting so-called offline features [28]. In this context, various feature extraction techniques in offline character recognition [32], [128] can be applied to online recognition as well.

The so-called direction feature [144], which is widely used in offline character recognition [50], [51], is being used in online recognition [28], [45], [88]. In offline recognition, the skeleton or contour pixels of the character image are classified into either four (horizontal, vertical, diagonal, and antidiagonal) or eight directions and stored in respective directional planes. Each directional plane is compressed by grid partitioning or blurring (spatial filtering and sampling) [67]. In [41], we named the resulting features "histogram features," which is motivated by the fact that they describe the number of occurrences for each direction. For online recognition, directional features can be extracted directly from the online trajectory [45]. A "direction-change" feature characterizing the temporal information was proposed to enhance the recognition performance of direction feature [99], [100].

5 Character Classification

In this section, we first review the coarse classification techniques. For fine classification, we categorize the techniques into three groups: structural matching, probabilistic matching, and statistical classification.

5.1 Coarse Classification

Coarse classification can be accomplished by class set partitioning or dynamic candidate selection. In class set partitioning, the character classes are divided into disjoint or overlapping groups. The input pattern is first assigned to a group or multiple groups and then, in fine classification, the input pattern is compared in detail with the classes in the group(s). In dynamic candidate selection, a matching score (similarity) is computed between the input pattern and each class and a subset of classes with high scores is selected for detailed classification. The average number of candidates can be significantly reduced without loss of precision via selecting a variable number of candidates by confidence evaluation [68].

For coarse classification based on class set partitioning, the groups of classes are determined in the classifier design stage using clustering or prior knowledge. Class grouping can be based on overall character structure [66], basic stroke substructure [12], stroke sequence [14], and statistical or neural classification [79]. Partitioning into overlapping groups can reduce the risk of excluding the true class of input pattern.

Dynamic candidate selection avoids the training process of class set partitioning. The character classes are ordered according to a matching score based on simple structural features or statistical features. For instance, the number of strokes or line segments of the input pattern can be used to filter out the unlikely classes [7], [65], [132], [133]. For efficient filtering, the bounds of stroke number depend on the character class and writing quality [29]. As to other features, the matching score is computed by string matching [124], peripheral feature matching [19], voting of structural features [126], or feature vector matching [28], [81], [88], [94], [99], [139]. The matching score of coarse classification can also be combined with that of fine classification to improve the final accuracy [119], [139].

In coarse classification by feature vector matching, the distance measure, such as city block distance and Euclidean distance, can be computed very efficiently and the efficiency can be further improved by dimensionality reduction and combining class-specific features [81]. In structural matching, candidate classes can also be selected via radical detection [16], [60], [76]. A detected radical excludes all the classes not containing the radical from fine classification. The radical detection approach is tightly connected to structural matching and will be addressed later.

5.2 Structural Matching

In fine classification by structural matching, the input pattern is matched with the structural model of each (candidate) class and the class with the minimum matching distance is taken as the recognition result. We divide the structural matching methods into four categories: DP (dynamic programming) matching, stroke correspondence, relational matching, and knowledge-based matching. DP matching works on ordered sequences and, hence, is stroke-order dependent. Stroke correspondence is different from relational matching in that it does not consider the interstroke relationship. Connected to these approaches, some general strategies are: hierarchical matching and deformation methods.

Hierarchical matching can improve the speed of structural recognition. When stroke codes or radical models are shared by different characters, the classification can be performed by a decision tree [11] or a network [47]. When the strokes or radicals of input pattern have been identified, the character recognition is reduced to traversing a path in the tree/network. However, the accuracy of classification is limited by the identification of strokes or radicals in the input pattern, which is not a trivial task. Therefore, instead of deterministic traversals, measuring the likelihood of paths and search with backtrack is helpful to improve the recognition performance.

Deformation techniques are useful to improve the matching similarity by deforming the character prototype or the input pattern. Based on the stroke correspondence, the deformation vector field (DVF) between the input pattern and the prototype can be computed and the prototype is iteratively deformed by local affine transformation (LAT) to fit the input pattern [131]. A noniterative stroke-based affine transformation (SAT) decomposes the DVF of each stroke incorporating the relationship between successive strokes [134]. In another work, a so-called parabola transformation was proposed to deform the character prototype based on attributed string matching of feature point sequences [13].

5.2.1 DP Matching

DP matching finds the ordered correspondence between the symbols (primitives) of two strings with the aim of minimizing the edit (Levinstein) distance. The DP matching of point sequences is also referred to as dynamic time warping (DTW). Attributed string matching refers to the matching of sequences of attributed primitives. In online character recognition, feature points or line segments are often taken as the primitives of sequence representation [55], [88], [124].

The search space of DP matching is represented in a rectangular grid with two diagonal corners denoting empty matching (start) and complete matching (goal), respectively. In a path from start to goal, the transition between neighboring grid points corresponds to symbol deletion, insertion, or substitution. A generalization of attributed string matching can merge multiple primitives in one string to match with one primitive in another string [123]. By imposing constraints onto potential grid transitions, the search speed can be largely improved with little loss of accuracy (e.g., [86], [88]).

DP matching is a mature technique, but the performance of recognition depends strongly on the selection of primitives and the definition of the between-primitive distance measure. For dealing with stroke-order variations, a character class needs multiple prototypes.

5.2.2 Stroke Correspondence

Based on the stroke correspondence between the input pattern and a character prototype, the character matching distance is computed as the sum of between-stroke distances. The correspondence can be found by reordering the input strokes or prototype strokes using domain specific rules [31], [74], [145]. The rules include, e.g., the precedence orders of different types of strokes. The compilation of rules, however, is laborious.

A simple noniterative technique, such as the interstroke distance matrix (ISDM) [96], [132] and the like [60], works well but does not necessarily find one-to-one correspondence. Wakahara et al. proposed a heuristic one-to-one stroke correspondence algorithm and an advanced selective linkage method to cut connected input strokes and allow matching with multiple prototype strokes [133]. This approach deals with both stroke-order variation and stroke-number variation.

Stroke correspondence was also solved by DP search in a hypercube and stroke connection was resolved by pairing an input stroke with multiple prototype strokes in multilayer cube search [112]. Interstroke relationships and the prior knowledge of stroke order variations can be incorporated to improve the matching performance [113].

5.2.3 Relational Matching

Relational matching is the search for a correspondence between two sets of elements under the constraint of relationship. It can be formulated as a consistent labeling problem and be solved using heuristic search [95], [135] or relaxation labeling [30], [36]. Relaxation labeling is computationally efficient, while heuristic search is flexible in terms of incorporating various knowledge sources and constraints.

A well-known heuristic search algorithm, the A* search, has been used for ARG (attributed relational graph) matching in [122], [136] and for OLCCR in [71], [72], [150]. The graph matching was also accomplished by finding the maximal cliques in the association graph [7], [10]. Relaxation labeling was used for graph matching of OLCCR in [149]. Utilizing the local invariance of stroke-order within a radical, a method combines DP and relaxation for radical detection and residual matching, respectively [139]. Relational matching was also formulated as an assignment problem (AP) with relational constraints [35] or a two-layer AP with a layer computing between-stroke distance [73]. Though the AP can be efficiently solved using the Hungarian method [101], the incorporation of relational constraints is not so flexible as in heuristic search.

As DP matching and stroke correspondence do, relational matching can be applied to the matching of either a holistic pattern or a radical. The advantage of relational matching over DP matching lies in the stroke-order independence, while the advantage over stroke correspondence is that the constraint of relationship improves the accuracy of matching. Relational matching is more computationally expensive than both DP matching and stroke correspondence, however.

5.2.4 Knowledge-Based Matching

The prior knowledge of character structure and writing appears as heuristic rules or constraints. The constraints can be used to efficiently reduce the search space of structural matching [71], [72], [113], [139], while rule-based methods have been used for stroke reordering [31], [74], radical detection [75], [76], and character matching [8], [9], etc. The rules represent the knowledge of basic strokes allowed for a character and the invariant geometric features of strokes [8], [9], for example. A knowledge-based system was also proposed for discriminating similar characters which were output as candidate classes by a base recognition system [16], [23].

We regard the deviation-expansion (D-E) model of [16], [65] as a kind of prior knowledge because it prespecifies the ranges of stroke-order variations and stroke-number variations. The variations of a character class are expanded in a D-E tree and the optimal matching with an input pattern is found with DP search [65] or A* search [16].

For constructing knowledge-based systems, the acquisition and organization of knowledge bases is not trivial and is sometimes laborious. Nevertheless, using simple heuristics to assist search-based matching is beneficial. Some heuristics, such as the stroke-order statistics, can be obtained from character samples [93], [113]. On the other hand, building special modules or heuristic rules for discriminating similar characters is helpful to improve the overall recognition accuracy.

5.3 Probabilistic Matching

Using probabilistic attributes in representing structural models and computing matching distance helps improve the tolerance of shape deformations. An example is to use stroke-type probability table in calculating the between-stroke distance [106], while, by modeling the prototype strokes as Gaussian density functions, the matching score of an input pattern and a character model becomes the joint probability of constituent strokes [20], [150], which is computed by grouping the feature points or line segments of the input pattern into strokes by attributed string matching or heuristic search.

In HMM-based recognition, the task of recognition is to decode the observation sequence (points or line segments) into the most probable state sequence. This is usually accomplished by using a dynamic programming procedure called Viterbi decoding [103], as practiced in [117], [143] when representing character models holistically as HMMs.

By representing radicals and interradical ligatures in HMMs and sharing them for all character classes in a network [48], the input sequence (8-direction codes) can be decoded into the concatenation of radical and ligature HMMs using the level building algorithm [84]. In recognition based on substroke HMMs, the recognition is also performed by search through an interconnected network [91], [92], [121]. In matching with a character model represented by path-controlled HMM (PCHMM), the line segment sequence of the input pattern is reordered to correspond to the optimal state sequence using A* search [150].

5.4 Statistical Classification

Various statistical techniques [43] are applicable for classification when describing the input pattern as a feature vector. Unlike that in coarse classification, simple classifiers are used to achieve high-speed; fine classification usually employs sophisticated classifiers to achieve high accuracy.

Kawamura et al. [45] achieved a fairly high recognition accuracy in OLCCR using a multiple similarity measure [37], which is similar to the subspace method [97] in that each class is represented as a linear subspace. The multiple similarity method, the subspace method, and the modified quadratic discriminant function (MQDF) [50] are quadratic classifiers. The MQDF is a smoothed version of QDF, obtained under the assumption of multivariate Gaussian density for each class [25]. The quadratic classifiers yield high accuracies, but are expensive regarding storage and computation. On the other hand, a simple metric, like the Euclidean distance between the input feature vector and a prototype vector, gives fast recognition. Designing multiple prototypes by clustering for each class can improve the accuracy, whereas the prototype learning by learning vector quantization (LVQ) [56], [69], [125] leads to significant improvement.

Closely related to statistical techniques are neural networks. For large category set classification, divide-and-conquer strategies, i.e., partitioning the category set into groups of classes, followed by an appropriate organization of multiple neural networks, may yield high performance. There have not been many works of OLCCR using neural networks. A successful example was reported by Matic et al. [79].

6 Model Learning and Adaptation

The quality of the model database influences the recognition performance. For statistical or neural classification, the database contains parameters (prototype vectors, subspace vectors, connecting weights, etc.), which can be estimated from samples using well-known estimation and learning techniques [22], [25], [43]. For structural matching, the database contains the structural character/radical models. The learning of structural models from samples is not trivial because the learning patterns of a class have a different number of primitives and the primitives do not correspond.

It is noteworthy that many previous works avoided the problem of model learning. Instead, they built the structural models manually using prior knowledge (e.g., [7], [16], [65], [76], [73], [139]) or used carefully written character patterns as prototypes (e.g., [132], [133]). Usually, model learning proceeds by iteratively adjusting the parameters of the structural models on matching the learning patterns with the models. The adjusted models can give higher recognition accuracy than the initial, manually built models.

6.1 Mean Prototype Learning

Usually, a "mean" prototype of the learning patterns for each class gives good recognition performance. Based on an initial structural model and the correspondence between this model and every learning pattern, the means and variances (or PDFs) of the structural attributes can be computed and the mean prototype can be updated. The mean prototype (described in mean structural attributes) is updated iteratively until the mean attributes converge. This iterative procedure is a generalized version of the EM (expectation-maximization) algorithm [21]. It has been used to learn the mean prototypes of point sequences [133], feature points [18], FARG attributes [149], and the parameters of PCHMM [150].

6.2 HMM Learning

In HMM learning, the parameters (initial probabilities, transition probabilities, and emission probabilities) are generally estimated on learning patterns using an EM procedure called Baum-Welch algorithm [103]. If the states can be partitioned artificially or have explicit physical meanings, the probabilities can be calculated by counting the frequencies of events, such as for the discrete HMMs of [143].

In the situation that an HMM represents a primitive (stroke, substroke, etc.) instead of a holistic pattern, partitioning the primitives of learning patterns is necessary for HMM learning. This can generally be accomplished by level building Viterbi decoding [84]. Currently, some research works use manually partitioned primitives in HMM learning [48], [91].

6.3 Multiprototype Learning

Clustering techniques have been used to design multiple stroke prototypes and character prototypes from learning patterns. If the strokes or characters are represented as feature vectors, the clustering can be accomplished by well-known statistical clustering algorithms like the k-means algorithm. Statistical clustering has been used to design stroke prototypes in [142]. The clustering of structural patterns, however, must be based on the structural matching between patterns.

For stroke-order dependent recognition methods, multiple prototypes per class are necessary to absorb stroke-order variations. In a simple scheme, each class initially has a single prototype and, when the matching distance between a learning pattern and the prototype exceeds a threshold, a new prototype is added [55], [143]. Iterative clustering analogous to k-means has been used to learn multiple structural prototypes for alphabetic characters [59], [130] and Chinese characters [1]. In [1], each class initially has one prototype and the number of prototypes is adjusted in the clustering process. Another approach designs prototypes by selection from samples according to the proximity between the samples of a class [106].

Learning vector quantization (LVQ) [56], primarily for feature vectors, was generalized to adjust structural prototypes and applied to alphabetic recognition [59], [130] and Chinese character recognition [1]. For LVQ of structural patterns, an elastic matching algorithm is embedded to correspond the primitives of the prototype and the learning pattern. Based on the deformation vectors between the corresponding primitives of prototype and learning pattern, the prototype is drawn either toward the learning pattern or apart from the learning pattern with the aim of reducing the number of misclassifications. Since LVQ adjusts prototypes discriminatively, it can give higher classification accuracy than clustering.

6.4 Structured Learning

In structured model database, the common radical models shared by different classes can be generated and adjusted according to the learning patterns that contain this radical. However, the automated learning of radical models has not been adequately addressed.

In an interactive learning approach, radical prototypes are generated incrementally upon request when a part of the input pattern or the pattern as a whole mismatch the prototypes [87]. An approach uses LVQ to adjust the radical prototypes discriminatively [53]. It describes radical prototypes as line segments spanned in a square box. When constructing a character model, the constituent radical prototypes are rescaled to the actual sizes and aspect ratios. On matching the character model with an input pattern, the deformation vectors of a constituent radical are scaled back to square box and used to adjust the radical prototype. It was shown that the adjusted radical prototypes outperform the mean radical prototypes [53].

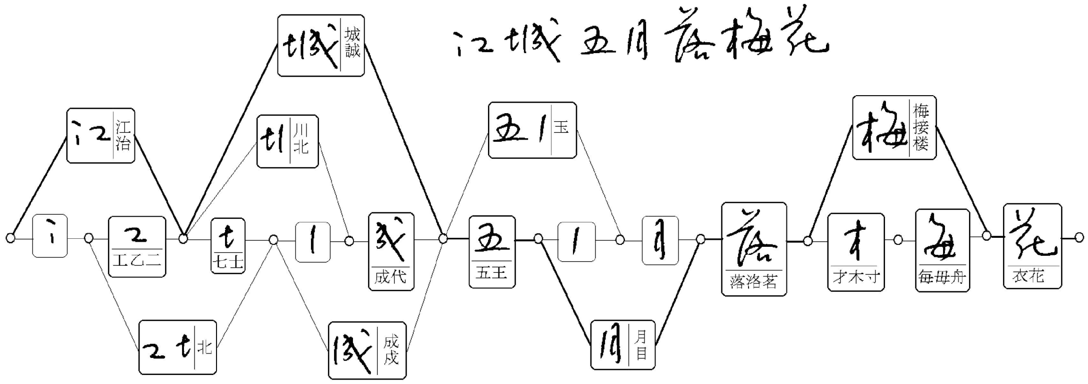

Fig. 8. A segmentation-recognition candidate network. The thick line denotes the most plausible segmentation path.

6.5 Writer Adaptation

For writer-dependent recognition, the model database is initially designed on the samples of multiple writers (this is to overcome the insufficient amount of writer-specific samples) and then adapted to the patterns of a specific user. Writer adaptation improves the recognition accuracy for the user, particularly when the user's writing style is novel or the writing environment changes [52].

For alphanumeric recognition, an adaptation method includes three adaptation modes: adding new prototypes, deactivating confusing prototypes, and reshaping existing prototypes [59], [130]. The classification is performed using the k-nearest neighbor (k-NN) rule on DTW (dynamic time warping). The input pattern is added into the prototype set when one of the k-NN prototypes belongs to wrong class. If two prototypes are close to the input pattern but belong to different classes, they are deactivated. Prototype reshaping works in a similar way to LVQ. It takes place when the input pattern is slightly different from the nearest prototype. A similar method was used in online Japanese character recognition with special strategies to control the adding, deletion, and reshaping of prototypes [2].

A simple interactive adaptation method shows promise in online Japanese character recognition in [147]. When the recognition of an input pattern gives multiple candidate classes and the best candidate class is not correct, the input pattern is added as a prototype or an existing prototype is replaced with the average of the input pattern and the prototype.

The adaptation of HMMs has been widely addressed in the speech recognition literature. In the adaptation of substroke HMMs for OLCCR, the mean vectors of emission probability densities are adjusted based on the segmentation of HMM states [91].

7 Contextual Processing

Most efforts in OLCCR research have dealt with feature extraction and classification/matching of character patterns. Typically, the classification/matching module provides not only a unique class index, but also the scores (probabilities, similarities, or distances) to a number of candidate classes. The linguistic context can provide valuable information for selecting the optimal class from this set of candidates. In addition, the geometric features of character patterns (size, location, aspect ratio, etc.) are useful to segment a handwriting sequence into single characters. For page-based handwritten text recognition, the integration of segmentation and recognition in contextual processing has particular importance because, often, the characters cannot be segmented unambiguously prior to recognition [24], [83].

The linguistic processing of character recognition results after segmentation is usually referred to as postprocessing. Based on the candidate classes given by the character recognizer, additional candidates can be added according to the statistics of confusion between characters in order to reduce the risk of excluding the true class [140]. The selection of final classes from candidate classes is based on the linguistic knowledge represented in word dictionaries, character-based n-grams [64], [88], [115], or word-based n-grams [39], [118], [137], [140]. The word-based n-gram is generally based on the syntactic/semantic classes (e.g., parts of speech) of words. Its use in linguistic processing involves the segmentation of text into words, usually by morphological analysis using a lexicon [118], [137]. The adaptation of writer-specific linguistic dictionaries is beneficial for writer-dependent recognition [40]. Linguistic postprocessing can significantly improve the accuracy of handwritten text recognition, e.g., halved character-level error rates were reported in OLCCR [39], [140].

The ambiguities in segmentation are generally solved by generating candidate character patterns and verifying the candidate patterns using geometric features, recognition results, and linguistic knowledge. The candidate character patterns can be represented in a candidate network whose edges denote combinations of segments that constitute the candidate patterns [83]. Each candidate pattern is assigned several candidate classes, together with their scores. Each path in the network represents a segmentation of the handwritten input and its recognition result. The most plausible path can be found by DP search. Linguistic knowledge is either used to verify the paths [34], [83] or numerically incorporated into the path scores [26], [98], [110], [111].

Fig. 8 shows an example of candidate segmentation-recognition network. On assigning candidate classes to each candidate pattern, the edge of the candidate pattern should be split into multiple edges corresponding to multiple candidate classes. The path with the minimum cost or maximum likelihood defines the result, i.e., the optimal segmentation with the pattern classes. The likelihood of a path integrates both the probability that a candidate pattern is indeed a character and the compatibility of the recognized character string with the language model [26], [110], [111].

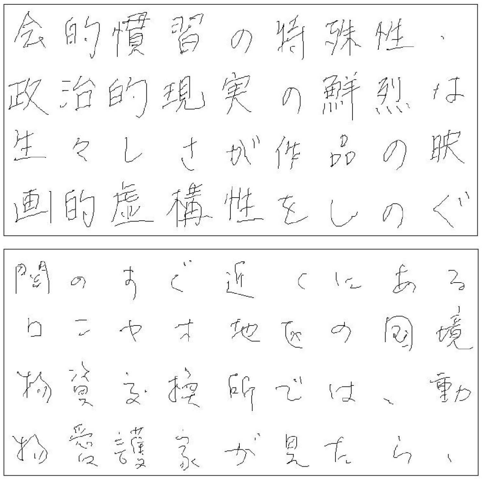

Fig. 9. Online patterns of TUAT Kuchibue database.

8 Performance Evaluation

Comparing the performance of different recognition systems is not easy because the systems are usually optimized for different applications and are thus adapted to different character sets and writing styles. However, in the experimental stage, the settings of recognizers are controllable and the performance can be compared by testing on a common sample set. We introduce online sample databases available for system design and performance evaluation and then compare recognition results reported in the literature.

8.1 Databases

Several recent research works promote the sharing of samples, such as the UNIPEN project [27] and the TUAT (Tokyo University of Agriculture and Technology) databases [89]. The online sample databases of TUAT are gaining popularity in Japan for system design and evaluation [42], [80], [89]. Two TUAT databases are available: The Nakayosi database is usually used for classifier learning, while the Kuchibue database is used for evaluation. Many researchers have reported results on the Kuchibue database [1], [53], [88], [99], [119], [143].

In the TUAT databases, most patterns were collected by writing natural sentences taken from a Japanese newspaper. The writing style was not constrained so that most of the characters were written fluently. Some people habitually write in regular style, though. The samples of both databases were collected using two types of digitizing tablets (WACOM and MUTOH). For either database, the writing box on display was set to 60×60 dots, corresponding to physical size of $1.7 \times 1.7\text{cm}^2$ (WACOM) or $1.43 \times 1.43\text{cm}^2$ (MUTOH). The samples of Kuchibue were recorded in display coordinates while those of Nakayosi were recorded in tablet coordinates, with the resolution 13 times higher (WACOM) or 9.45 times higher (MUTOH) than the display coordinates. Fig. 9 shows some typical sample patterns of the Kuchibue database.

The specifications of both TUAT databases are given in Table 1. The Kuchibue database contains the patterns of 120 writers: 11,962 patterns per writer covering 3,356 categories. Excluding the JIS level-2 Kanji characters, there are 11,951 patterns for 3,345 categories (including 2,965 Kanji characters and 380 non-Kanji symbols), which are frequently used in recognition experiments. The Nakayosi database covers a larger category set than the Kuchibue database.

TABLE 1
Specifications of Kuchibue and Nakayosi Databases

Database	#category	#writer	#samples per writer
Kuchibue	3,356	120	11,962
Nakayosi	4,438	163	10,403

8.2 Comparison of Results

The performance of a recognition system is measured in terms of recognition accuracy, memory requirement, and recognition speed. In general, statistical methods offer high speed under large memory requirements, while structural methods (including statistical-structural models) have lower speed but smaller memory. The large memory of statistical methods is due to the high dimensionality of feature representation and, particularly, the class-specific subspaces of quadratic classifiers. Structural methods are computationally expensive because the matching of character structures is a combinatorial problem.

Table 2 lists some recognition results (experimented with at least 1,000 categories, without linguistic processing) reported since 1990. The recognition methods are roughly classified into statistical (statis) or structural (struct) methods. HMM-based methods and some special structural methods are singled out. The method of Zheng et al. [150] unifies statistical stroke models and PCHMMs. Velek et al. combined four (statistical) offline recognizers and three (structural) online recognizers [129]. Table 2 also gives the test writing style, the numbers of learning samples and test samples of each recognizer, if available.

The recognition rates in Table 2 reflect the combined performance of pattern representation, learning, and classification. The recognition accuracy depends on the number of learning samples, especially for statistical methods. Some structural methods use a carefully written character prototype or a manually built model for each class, yet achieve high accuracy by knowledge-based or flexible matching. Both statistical and structural methods can benefit from learning with large sample sets. Some successful examples are Kawamura et al. [45], Wakahara et al. [133], [134], Rowley et al. [106], and Velek et al. [129].

The comparison between statistical methods and structural methods is a debate in the character recognition community. The selection of classifier should take into account the trade-off between the recognition accuracy and the model database size. Statistical classification can yield very high accuracy if a sophisticated classifier is trained with large number of samples. For instance, among the classifiers of Velek et al. [129], a statistical classifier gives higher accuracy than the structural ones, but its model database consumes over 30MB memory. On the other hand, a structural classifier can give fairly high accuracy with very small storage, e.g., the model database of Akiyama et al. [2] consumes only 166KB memory. Structural methods have potential to give higher accuracies relying on probabilistic structure modeling and learning with samples.

TABLE 2
Recognition Results Reported in the Literature

Source	Method	#category	Style	#learning	#test	Rec. rate
Liu'91 [75]	struct	6,763	flu-regular	N/A	N/A	90%
Kawamura'92 [45]	statis	2,965	careful/free	380 PC	20 PC	94.51/91.78%
Lin'93 [65]	struct	5,400	regular	1 PC	10 PC	87.4%
Liu'93 [76]	struct	13,000	flu-regular	N/A	N/A	93%
Hamanaka'93 [28]	statis	1,064	regular	54,028	52,944	95.1%
Chou'94 [16]	struct	5,401	regular	3 PC	17 PC	94.88%
Wakahara'95 [133]	struct	2,980	careful/free	120 PC	36 PC	97.6/94.1%
Lay'96 [60]	struct	5,401	regular	N/A	5 PC	96.35%
Kim'96 [47]	struct	1,800	free	4 PC	6 PC	93.13%
*Nakagawa'96 [88]	struct	3,345	fluent	N/A	11951 PW×30	80–90%
Chou'96 [18]	struct	5,401	flu-regular	5 PC	15 PC	93.4%
Wakahara'97 [134]	struct	2,980	careful/free	120 PC	36 PC	98.4/96.0%
Kim'97 [48]	HMM	1,800	free	4 PC	6 PC	90.3%
Zheng'97 [149]	FARG	3,755	regular	N/A	6 PC	98.8%
Xiao'97 [139]	struct	3,755	flu-regular	35 PC	3 PC	93.9
Nambu'98 [94]	struct	3,942	flu-regular	200 PC	200 PC	89.7%
Kuroda'99 [58]	statis	1,000	regular	25 PC	10 PC	94.34%
*Okamato'99 [99]	statis	3,345	fluent	11951 PW×40	11951 PW×41	86.32%
*Yasuda'99 [143]	HMM	3,057	fluent	10038 PW×100	10038 PW×20	85.89%
*Tanaka'99 [119]	combined	3,356	fluent	Nakayosi	Kuchibue	87.6%
Zheng'99 [150]	sta-struct	3,755	mixed	100 PC	7 PC	95.52%
*Akiyama'00 [1]	struct	3,345	fluent	6690 PW×150	11951 PW×3	88.58%
Shin'02 [113]	struct	2,965	regular	90 PC	24 PC	99.28%
*Kitadai'02 [53]	struct	3,345	fluent	9309 PW×163	11951 PW×120	87.2%
Tokuno'02 [121]	HMM	1,016	fluent	50,986	42,718	92.0%
Velek'02 [129]	combined	3,036	fluent	3,669,089	54,927	94.14%
Nakai'02 [92]	HMM	1,016	fluent	34 PC	34 PC	93.1%
Matic'02 [79]	neural	4,400	N/A	80 PC	20 PC	97.3%
Rowley'02 [106]	struct	6,847	natural	5 million	85,655	94.45%

PC: per category
PW: per writer
*Tested on TUAT Kuchibue database

Our purpose in collecting recognition rates from the literature is both to investigate the status of performance and to compare the recognition methods. We did not collect the recognition rates of commercial products in the market considering that the advertised rates are estimated in different environments and their recognition methods are rarely publicized. We believe it is currently possible to achieve a very high recognition rate, above 98 percent on regular scripts. However, on fluent or fluent-regular scripts, it is difficult to achieve a recognition rate above 90 percent (see the results on Kuchibue database). Though linguistic processing can largely reduce the error rate, the remaining recognition errors still bring high inconvenience to the user. Assuming linguistic processing resolves half of recognition errors and our target of final correct rate is 99 percent (even this rate is not enough), then the character recognizer should provide an accuracy above 98 percent. For free handwriting recognition, it is very hard to reach this target.

9 Future Directions

The gap between the technical status and the required performance indicates that the problem of OLCCR is not solved yet and it leaves us research opportunities. To reach the goal of totally free handwriting recognition, we should seriously reconsider the methods and find ways to significantly improve the recognition performance. The significant improvement relies on the integration of multiple approaches and the joint effects of all processing steps. In the following, we will discuss the research directions in respect to pattern representation, classification, learning/adaptation, and contextual processing, respectively.

In the area of representation, both the feature vector representation and the traditional structural representation have apparent insufficiencies which can be overcome by the hybrid statistical-structural representation. In OLCCR, only a few works have modeled the PDFs of stroke attributes [20], [150]. Recently, modeling both strokes and interstroke relationships probabilistically has been tried in online numeral recognition [15] and offline Chinese character recognition [49]. A hybrid model can also describe characters structurally with statistical radical/stroke models (HMMs, PDFs, or discriminant functions) as primitives. Global feature vector representation schemes can also be improved by informative feature extraction, automatic feature transformation and selection. Currently, the so simple direction feature (histogram feature) performs fairly well, but we are sure that it can be surpassed. The feature transformation and selection, pertaining to statistical pattern recognition [43], is effective to improve the classification performance.

Improvements in classification are possible with both statistical classification and structural matching approaches and, particularly, the integration of multiple classifiers. In desktop applications, sophisticated statistical classifiers can yield high recognition accuracies and the accuracy can be further improved by discriminative learning (like the LVQ). The recognition accuracy of structural matching can be improved by modeling the structural models probabilistically. The computation cost of structural matching can be lowered by exploring a variety of prior knowledge, such as the partial invariance of stroke-order and stroke connection. Multiple classifiers can be combined in cascade or in parallel to improve the speed and/or accuracy. The cascaded combination can not only speed up the recognition, but also improve the accuracy by employing sophisticated classifiers (such as neural networks, support vector machines, and rule-based methods) for discriminating similar characters. The parallel combination of multiple complementary classifiers can improve the classification accuracy [33], [54] and its promise in OLCCR has been justified in [119], [129].

The recognition performance depends on the quality of the model database and can benefit from learning with large sample sets. The potential high performance of structural methods has been hindered by the lack of efficient learning methods. The adjustment of structural parameters starting from manually built models and the structural prototype selection from samples have been attempted, but the automatic construction of structural models has not been reported in the literature of OLCCR. The discriminative learning of structural parameters is able to yield higher recognition accuracy than the EM-like learning, but has not been widely adopted. The task of learning also lies in the acquisition of knowledge base for rule-based classification or for assisting structural matching, which was often accomplished manually. For the adaptation of writer-dependent models, the previous works have used ad hoc methods to add, delete, or adjust structural prototypes. This problem needs to be considered more closely and principled methods are hoped to better deal with it.

With regard to the contextual processing of OLCCR, layout analysis and text line extraction techniques are to be developed to meet with the needs of page-based input. The performance of integrated segmentation-recognition can be improved by evaluating the paths in the candidate network probabilistically, rather than heuristically. This requires the probabilistic representation of linguistic knowledge, the transformation of geometric features and character recognition scores into probabilities. Another contextual feature, called style consistency, as has been utilized in numeral recognition [46], [108], is helpful to reduce errors in writer-independent recognition. In a group of character patterns given by the same writer, the correlation between the character shapes can be explored to clear the confusions such as similar shapes given by different writers belonging to different classes. The style consistency can be utilized in the classifier design or postprocessing stage.

ACKNOWLEDGMENTS

The authors are grateful to the anonymous reviewers for their invaluable comments and suggestions, which led to improvements to the presentation quality of this paper.

REFERENCES

[1] K. Akiyama and K. Ishigaki, "A Method of Generating High Quality Online Character Recognition Dictionary Based on Training Samples," Technical Report of IEICE, PRMU99-235 (2000-02), (in Japanese).

[2] K. Akiyama et al., "An Adaptation Method Based on Template Cache for Online Character Recognition," Technical Report of IEICE, PRMU2000-210 (2001-03), (in Japanese).

[3] D. Blostein, "General Diagram-Recognition Methodologies," *Graphics Recognition Methods and Applications,* R. Kasturi and K. Tombre eds. pp. 107-122, Springer, 1995.

[4] R.G. Casey, "Moment Normalization of Handprinted Character," *IBM J. Research Development,* vol. 14, pp. 548-557, 1970.

[5] K.-F. Chan and D.-Y. Yeung, "Error Detection, Error Correction and Performance Evaluation in On-Line Mathematical Expression Recognition," *Pattern Recognition,* vol. 34, no. 8, pp. 1671-1684, 2001.

[6] K.P. Chan and Y.-S. Cheung, "Fuzzy-Attribute Graph with Application to Chinese Character Recognition," *IEEE Trans. Systems Man, and Cybernetics,* vol. 22, no. 1, pp. 153-160, 1992,

[7] J.W. Chen and S.Y. Lee, "On-Line Handwritten Chinese Character Recognition via a Fuzzy Attribute Representation," *Image and Vision Computing,* vol. 12, no. 10, pp. 669-681, 1994.

[8] J.-W. Chen and S.-Y. Lee, "A Hierarchical Representation for the Reference Database of On-Line Chinese Character Recognition," *Advances in Syntactic and Structural Pattern Recognition,* P. Perner, P. Wang, and A. Rosenfeld, eds., pp. 351-400, Springer, 1996.

[9] J.-W. Chen and S.-Y. Lee, "On-Line Handwriting Recognition of Chinese Characters via Rule-Based Approach," *Proc. 13th Int'l Conf. Pattern Recognition,* vol. 3, pp. 220-224, 1996.

[10] J.W. Chen and S.Y. Lee, "On-Line Chinese Character Recognition via a Representation of Spatial Relationships between Strokes," *Int'l J. Pattern Recognition and Artificial Intelligence,* vol. 11, no. 3, pp. 329-357, 1997.

[11] K.J. Chen, K.K. Li, and Y.L. Chang, "A System for On-Line Recognition of Chinese Characters," *Int'l J. Pattern Recognition and Artificial Intelligence,* vol. 2, pp. 139-148, 1988.

[12] R.-H. Chen, C.-W. Lee, and Z. Chen, "Preclassification of Handwritten Chinese Characters Based on Basic Stroke Substructures," *Proc. Fourth Int'l Workshop Frontiers in Handwriting Recognition,* pp. 176-184, 1994.

[13] W.-T. Chen and T.-R. Chou, "A Hierarchical Deformation Model for On-Line Cursive Script Recognition," *Pattern Recognition,* vol. 27, no. 2, pp. 205-219, 1994.

[14] Z. Chen, C.-W. Lee, and R.-H. Cheng, "Handwritten Chinese Character Analysis and Preclassification Using Stroke Structural Sequence," *Proc. 13th Int'l Conf. Pattern Recognition,* vol. 3, pp. 89-93, 1996.

[15] S.J. Cho and J.H. Kim, "Bayesian Network Modeling of Strokes and Their Relationships for On-Line Handwriting Recognition," *Proc. Sixth Int'l Conf. Document Analysis and Recognition,* pp. 86-90, 2001.

[16] K.-S. Chou et al, "Knowledge Model Based Approach in Recognition of On-Line Chinese Characters," *IEEE J. Selected Areas Comm.,* vol. 12, no. 9, pp. 1566-1575, 1994.

[17] K.-S. Chou, K.-C. Fan, and C.-K. Lin, "A Knowledge Based Approach to the Recognition of On-Line Confusing Chinese Characters," *Proc. Fourth Int'l Workshop Frontiers in Handwriting Recognition,* pp. 185-194, 1994.

[18] K-S. Chou, K.-C. Fan, and T.-I. Fan, "Radical-Based Neighboring Segment Matching Method for On-Line Chinese Character Recognition," *Proc. 13th Int'l Conf. Pattern Recognition,* vol. 3, pp. 84-88, 1996.

[19] K.-S. Chou, K.-C. Fan, and T.-I. Fan, "Peripheral and Global Features for Use in Coarse Classification of Chinese Characters," *Pattern Recognition,* vol. 30, no. 3, pp. 483-489, 1997.

[20] T.-R. Chou and W.T. Chen, "A Stochastic Representation of Cursive Chinese Characters for On-Line Recognition," *Pattern Recognition,* vol. 30, no. 6, pp. 903-920, 1997.

[21] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Royal Statistical Soc. Series B.,* vol. 39, no. 1, pp. 1-38, 1977.

[22] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification,* second ed. Wiley Interscience, 2001.

[23] K.-C. Fan, C.-K. Lin, and K.-S. Chou, "Confusion Set Recognition of On-Line Character Recognition by Artificial Intelligence Techniques," *Pattern Recognition,* vol. 28, no. 3, pp. 303-313, 1995.

[24] T. Fujisaki, T.E. Chefalas, J. Kim, C.C. Tappert, and C.G. Wolf, "On-Line Run-On Character Recognizer: Design and Performance," *Character and Handwriting Recognition: Expanding Frontiers,* P.S.P. Wang ed., pp. 123-136, World Scientific, 1992.

[25] K. Fukunaga, *Introduction to Statistical Pattern Recognition,* second ed. Academic Press, 1990.

[26] T. Fukushima and M. Nakagawa, "On-Line Writing-Box-Free Recognition of Handwritten Japanese Text Considering Character Size Variations," *Proc. 15th Int'l Conf. Pattern Recognition,* vol. 2, pp. 359-363, 2000.

[27] I. Guyon, L. Schomaker, R. Plamondon, M. Liberman, and S. Janet, "UNIPEN Project of On-Line Data Exchange and Recognizer Benchmarks," *Proc. 12th Int'l Conf. Pattern Recognition,* vol. 2, pp. 29-33, 1994.

[28] M. Hamanaka, K. Yamada, and J. Tsukumo, "On-Line Japanese Character Recognition Experiments by an Off-Line Method Based on Normalization-Cooperated Feature Extraction," *Proc. Third Int'l Conf. Document Analysis and Recognition,* pp. 204-207, 1993.

[29] M. Hamanaka and K. Yamada, "On-Line Character Recognition Adaptively Controlled by Handwriting Quality," *Proc. Seventh Int'l Workshop Frontiers in Handwriting Recognition,* pp. 23-32, 2000.

[30] R.M. Haralick, "The Consistent Labeling Problem: Part I," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 1, no. 2, pp. 173-184, 1979.

[31] Y. Hidai, K. Ooi, and Y. Nakamura, "Stroke Re-Order Algorithm for On-Line Hand-Printed Character Recognition," *Proc. Eighth Int'l Conf. Pattern Recognition,* vol. 2, pp. 934-936, 1986.

[32] T.H. Hilderbrand and W. Liu, "Optical Recognition of Chinese Characters: Advances since 1980," *Pattern Recognition,* vol. 26, no. 2, pp. 205-225, 1993.

[33] T.K. Ho, J.J. Hull, and S.N. Srihari, "Decision Combination in Multiple Classifier Systems," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 16, no. 1, pp. 66-75, Jan. 1994.

[34] C. Hong, G. Loudon, Y. Wu, and R. Zitserman, "Segmentation and Recognition of Continuous Handwriting Chinese Text," *Proc. Int'l Conf. Computer Processing of Oriental Languages,* pp. 630-633, 1997.

[35] 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.

[36] R.T. Hummel and S.W. Zucker, "On the Foundations of Relaxation Labeling Processes," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 5, no. 3, pp. 267-287, 1983.

[37] T. Iijima, H. Genchi, and K. Mori, "A Theory of Character Recognition by Pattern Matching Method," *Proc. First Int'l Joint Conf. Pattern Recognition,* pp. 50-56, 1973.

[38] K. Ishigaki and T. Morishita, "A Top-Down Online Handwritten Character Recognition Method via the Denotation of Variation," *Proc. Int'l Conf. Computer Processing of Chinese and Oriental Languages,* pp. 141-145, 1988.

[39] N. Itoh, "Japanese Language Model Based on Bigrams and Its Application to On-Line Character Recognition," *Pattern Recognition,* vol. 28, no. 2, pp. 135-143, 1995.

[40] N. Iwayama and K. Ishigaki, "Adaptive Context Processing in On-Line Handwritten Character Recognition," *Proc. Seventh Int'l Workshop Frontiers in Handwriting Recognition,* pp. 469-474, 2000.

[41] S. Jaeger, C.-L. Liu, and M. Nakagawa, "The State of the Art in Japanese On-Line Handwriting Recognition Compared to Techniques in Western Handwriting Recognition," *Int'l J. Document Analysis and Recognition,* vol. 6, no. 2, pp. 75-88, 2003.

[42] S. Jaeger and M. Nakagawa, "Two On-Line Japanese Character Databases in UNIPEN Format," *Proc. Sixth Int'l Conf. Document Analysis and Recognition,* pp. 566-570, 2001.

[43] A.K. Jain, R.P.W. Duin, and J. Mao, "Statistical Pattern Recognition: A Review," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 1, pp. 4-37, Jan. 2000.

[44] A.K. Jain, F.D. Griess, and S.D. Connell, "On-Line Signature Verification," *Pattern Recognition,* vol. 35, no. 12, pp. 2963-2972, 2002.

[45] A. Kawamura et al., "On-Line Recognition of Freely Handwritten Japanese Characters Using Directional Feature Densities," *Proc. 11th Int'l Conf. Pattern Recognition,* vol. 2, pp. 183-186, 1992.

[46] T. Kawatani, "Character Recognition Performance Improvement Using Personal Handwriting Characteristics," *Proc. Third Int'l Conf. Document Analysis and Recognition,* pp. 98-103, 1995.

[47] H.J. Kim, J.W. Jung, and S.K. Kim, "On-Line Chinese Character Recognition Using ART-Based Stroke Classification," *Pattern Recognition Letters,* vol. 17, pp. 1311-1322, 1996.

[48] H.J. Kim, K.H. Kim, S.K. Kim, and F.T.-P. Lee, "On-Line Recognition of Handwritten Chinese Characters Based on Hidden Markov Models," *Pattern Recognition,* vol. 30, no. 9, pp. 1489-1499, 1997.

[49] I.-J. Kim and J.H. Kim, "Statistical Character Structural Modeling and Its Application to Handwritten Chinese Character Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 11, pp. 1422-1436, 2003.

[50] F. Kimura, K. Takashina, S. Tsuruoka, and Y. Miyake, "Modified Quadratic Discriminant Functions and the Application to Chinese Character Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 9, no. 1, pp. 149-153, Jan. 1987.

[51] F. Kimura, T. Wakabayashi, S. Tsuruoka, and Y. Miyake, "Improvement of Handwritten Japanese Character Recognition Using Weighted Direction Code Histogram," *Pattern Recognition,* vol. 30, no. 8, pp. 1329-1337, 1997.

[52] Y. Kimura, K. Odaka, A. Suzuki, and M. Sano, "Analysis and Evaluation of Dictionary Learning on Handy Type Pen-Input Interface for Personal Use," *Trans. IEICE Japan,* vol. J84-D-II, no. 3, pp. 509-518, 2001.

[53] A. Kitadai and M. Nakagawa, "A Learning Algorithm for Structural Character Pattern Representation Used in On-Line Recognition of Handwritten Japanese Characters," *Proc. Eighth Int'l Workshop Frontiers in Handwriting Recognition,* pp. 163-168, 2002.

[54] J. Kittler, "On Combining Classifiers," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 3, pp. 226-239, Mar. 1998.

[55] M. Kobayashi et al., "RAV (Reparameterized Angular Variations) Algorithms for Online Handwriting Recognition," *Int'l J. Document Analysis and Recognition,* vol. 3, no. 3, pp. 181-191, 2001.

[56] T. Kohonen, "The Self-Organizing Map," *Proc. IEEE,* vol. 78, no. 9, pp. 1464-1480, 1990.

[57] Y. Komiya, T. Ohishi, T. Matsumoto, "A Pen Input On-Line Signature Verifier Integrating Position, Pressure and Indication Trajectories," *IEICE Trans. Information and Systems,* vol. E84-D, no. 7, pp. 833-838, 2001.

[58] K. Kuroda, K. Harada, and M. Hagiwara, "Large Scale On-Line Handwritten Chinese Character Recognition Using Successor Method Based on Stochastic Regular Grammar," *Pattern Recognition,* vol. 32, no. 8, pp. 307-1315, 1999.

[59] J. Laaksonen, V. Vuori, and E. Oja, "Adaptation of Prototype Sets in On-Line Recognition of Isolated Handwritten Latin Characters," *Advances in Handwriting Recognition,* S.-W. Lee ed., pp. 489-497, World Scientific, 1999.

[60] S.-R. Lay et al., "On-Line Chinese Character Recognition with Effective Candidate Radical and Candidate Character Selection," *Pattern Recognition,* vol. 29, no. 10, pp. 1647-1659, 1996.

[61] F. Leclerc and R. Plamondon, "Automatic Signature Verification: the State of The Art—1989-1993," *Int'l J. Pattern Recognition and Artificial Intelligence,* vol. 8, no. 3, pp. 643-660, 1994.

[62] S.-W. Lee and J.-S. Park, "Nonlinear Shape Normalization Methods for the Recognition of Large-Set Handwritten Characters," *Pattern Recognition,* vol. 27, no. 7, pp. 895-902, 1994.

[63] X. Li and N.S. Hall, "Corner Detection and Shape Classification of On-Line Handprinted Kanji Strokes," *Pattern Recognition,* vol. 26, no. 9, pp. 1315-1334, 1993.

[64] M.-Y. Lin and W.-H. Tsai, "A New Approach to On-Line Chinese Character Recognition by Sentence Contextual Information Using the Relaxation Technique," *Proc. Int'l Conf. Computer Processing of Chinese and Oriental Languages,* pp. 131-134, 1988.

[65] C.-K. Lin, K.-C. Fan, and F.T. Lee, "On-Line Recognition by Deviation-Expansion Model and Dynamic Programming Matching," *Pattern Recognition,* vol. 26, no. 2, pp. 259-268, 1993.

[66] T.-Z. Lin and K.-C. Fan, "Coarse Classification of On-Line Chinese Characters via Structure Feature-Based Method," *Pattern Recognition,* vol. 17, no. 10, pp. 1365-1377, 1994.

[67] C.-L. Liu, Y.-J. Liu, and R.-W. Dai, "Preprocessing and Statistical/Structural Feature Extraction for Handwritten Numeral Recognition," *Progress of Handwriting Recognition,* A.C. Downton and S. Impedovo eds., pp. 161-168, World Scientific, 1997.

[68] C.-L. Liu and M. Nakagawa, "Precise Candidate Selection for Large Character Set Recognition by Confidence Evaluation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 6, pp. 636-642, June 2000.

[69] C.-L. Liu and M. Nakagawa, "Evaluation of Prototype Learning Algorithms for Nearest Neighbor Classifier in Application to Handwritten Character Recognition," *Pattern Recognition,* vol. 34, no. 3, pp. 601-615, 2001.

[70] C.-L. Liu, H. Sako, and H. Fujisawa, "Handwritten Chinese Character Recognition: Alternatives to Nonlinear Normalization," *Proc. Seventh Int'l Conf. Document Analysis and Recognition*, pp. 524-528, 2003.

[71] J. Liu, W.-K. Cham, and M.M.Y. Chang, "Online Chinese Character Recognition Using Attributed Relational Graph Matching," *IEE Proc. Vision Image Signal Processing,* vol. 143, no. 2, pp. 125-131, 1996.

[72] 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," *Proc. 13th Int'l Conf. Pattern Recognition,* vol. 3, pp. 259-263, 1996.

[73] J.Z. Liu, K. Ma, W.K. Cham, and M.M.Y. Chang, "Two-Layer Assignment Method for Online Chinese Character Recognition," *IEE Proc. Vision Image Signal Processing,* vol. 147, no. 1, pp. 47-54, 2000.

[74] Y.J. Liu and J.W. Tai, "Structural Approach to On-Line Chinese Character Recognition," *Proc. Ninth Int'l Conf. Pattern Recognition,* pp. 808-810, 1988.

[75] Y.-J. Liu and J.W. Tai, "An On-Line Chinese Character Recognition System for Handwritten in Chinese Calligraphy," *From Pixels to Features III: Int'l Workshop Frontiers in Handwriting Recognition,* pp. 69-80, 1991.

[76] Y.-J. Liu, L.-Q. Zhang, and J.W. Tai, "A New Approach to On-Line Handwriting Chinese Character Recognition," *Proc. Second Int'l Conf. Document Analysis and Recognition,* pp. 192-195, 1993.

[77] D. Lopresti and G. Wilfong, "Cross-Domain Searching Using Handwritten Queries," *Proc. Seventh Int'l Workshop Frontiers in Handwriting Recognition,* pp. 3-12, 2000.

[78] M.Y. Ma, P.S.P. Wang, D.P. Lopresti, and J. Crisman, "Semantic Matching of Free-Format Chinese Handwriting," *Proc. Int'l Conf. Computer Processing of Oriental Languages,* pp. 107-112, 1997.

[79] N. Matic, J. Platt, and T. Wang, "QuickStroke: An Incremental On-Line Chinese Handwriting Recognition System," *Proc. 16th Int'l Conf. Pattern Recognition,* vol. 3, pp. 435-437, 2002.

[80] K. Matsumoto, T. Fukushima, and M. Nakagawa, "Collection and Analysis of On-Line Handwritten Japanese Character Patterns," *Proc. Sixth Int'l Conf. Document Analysis and Recognition*, pp. 496-500, 2001.

[81] K. Matsumoto and M. Nakagawa, "Improvement of a Coarse Classification Method for On-Line Recognition of Handwritten Japanese Characters," Technical Report of IEICE, PRMU2001-273 (2002-03), (in Japanese).

[82] S. Müller, E. Eickeler, and G. Rigoll, "Multimedia Database Retrieval Using Hand-Drawn Sketches," *Proc. Fifth Int'l Conf. Document Analysis and Recognition,* pp. 289-292, 1999.

[83] H. Murase, "Online Recognition of Free-Format Japanese Handwritings," *Proc. Ninth Int'l Conf. Pattern Recognition,* pp. 1143-1147, 1988.

[84] C.S. Myers and L.R. Rabiner, "A Level Building Dynamic Time Warping Algorithm for Connected Word Recognition," *IEEE Trans. Acoustic, Speech, and Signal Processing,* vol. 29, no. 2, pp. 284-297, 1981.

[85] M. Nakagawa, "Non-Keyboard Input of Japanese Text—On-Line Recognition of Handwritten Characters as the Most Hopeful Approach," *J. Information Processing,* vol. 13, no. 1, pp. 15-34, 1990.

[86] M. Nakagawa and K. Akiyama, "A Linear-Time Elastic Matching for Stroke Number Free Recognition of On-Line Handwritten Characters," *Prof. Fourth Int'l Workshop Frontiers in Handwriting Recognition,* pp. 48-56, 1994.

[87] M. Nakagawa and L.V. Tu, "Structural Learning of Character Patterns for On-Line Recognition of Handwritten Japanese Characters," *Advances in Structural and Syntactic Pattern Recognition,* P. Perner, P. Wang, and A. Rosenfeld, eds., pp. 180-188, Springer-Verlag, 1996.

[88] M. Nakagawa et al., "Robust and Highly Customizable Recognition of On-Line Handwritten Japanese Characters," *Proc. 13th Int'l Conf. Pattern Recognition,* vol. 3, pp. 269-273, 1996.

[89] M. Nakagawa et al., "On-Line Handwritten Character Pattern Database Sampled in a Sequence of Sentences without Any Writing Instructions," *Proc. Fourth Int'l Conf. Document Analysis and Recognition,* pp. 376-381, 1997.

[90] M. Nakagawa, K. Akiyama, T. Oguni, and N. Kato, "Handwriting-Based User Interfaces Employing On-Line Handwriting Recognition," *Advances in Handwriting Recognition,* S.-W. Lee, ed., pp. 578-587, World Scientific, 1999.

[91] M. Nakai, N. Akira, H. Shimodaira, and S. Sagayama, "Substroke Approach to HMM-Based On-Line Kanji Handwriting Recognition," *Prof. Sixth Int'l Conf. Document Analysis and Recognition,* pp. 491-495, 2001.

[92] M. Nakai, T. Sudo, H. Shimodaira, and S. Sagayama, "Pen Pressure Features for Writer-Independent On-Line Handwriting Recognition Based on Substroke HMM," *Proc. 16th Int'l Conf. Pattern Recognition,* vol. 3, pp. 220-223, 2002.

[93] M. Nakai, H. Shimodaira, and S. Sagayama, "Generation of Hierarchical Dictionary for Stroke-Order Free Kanji Handwriting Recognition Based on Substroke HMM," *Proc. Seventh Int'l Conf. Document Analysis and Recognition,* pp. 514-518, 2003.

[94] H. Nambu, T. Kawamata, F. Maruyama, and F. Yoda, "On-Line Handwriting Chinese Character Recognition: Comparison and Improvement to Japanese Kanji Recognition," *Proc. 14th Int'l Conf. Pattern Recognition,* vol. 2, pp. 1145-1149, 1998.

[95] N.J. Nillson, *Principles of Artificial Intelligence.* Springer-Verlag, 1980.

[96] K. Odaka, T. Wakahara, and I. Masuda, "Stroke Order Free On-Line Character Recognition," *Trans. IECE Japan,* vol. J65-D, no. 6, pp. 679-686, 1982, (in Japanese).

[97] E. Oja, *Subspace Method of Pattern Recognition.* Research Studies Press, 1983.

[98] M. Okamoto, H. Yamamoto, K. Sawada, and K. Yamamoto, "On-Line Handwriting Character String Separation Method Using Network Expression," *Proc. 13th Int'l Conf. Pattern Recognition,* vol. 4, pp. 422-425, 1996.

[99] M. Okamoto and K. Yamamoto, "On-Line Handwriting Character Recognition Using Direction-Change Features that Consider Imaginary Strokes," *Pattern Recognition,* vol. 32, no. 7, pp. 1115-1128, 1999.

[100] M. Okamoto and K. Yamamoto, "On-Line Handwritten Character Recognition Method Using Directional Features and Clockwise/Counter-Clockwise Direction-Change Features," *Proc. Fifth Int'l Conf. Document Analysis and Recognition,* pp. 491-494, 1999.

[101] C.H. Papadimitrious and K. Steiglitz, *Combinatorial Optimization—Algorithms and Complexity.* Prentice Hall, 1982.

[102] R. Plamondon and S.N. Srihari, "On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 1, pp. 63-82, Jan. 2000.

[103] L.R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. IEEE,* vol. 77, no. 2, pp. 257-286, 1989.

[104] U. Ramer, "An Iterative Procedure for the Polygonal Approximation of Plane Closed Curves," *Computer Graphics and Image Processing,* vol. 1, pp. 244-256, 1972.

[105] A. Rosenfeld and E. Johnston, "Angle Detection on Digital Curves," *IEEE Trans. Computers,* vol. 22, pp. 875-878, 1973.

[106] H.A. Rowley and M. Goyal, and J. Bennet, "The Effect of Large Training Set Size on Online Japanese Kanji and English Cursive Recognizers," *Proc. Eighth Int'l Workshop Frontiers in Handwriting Recognition,* pp. 36-40, 2002.

[107] G. Russel, M.P. Perrone, Y. Chee, and A. Ziq, "Handwritten Document Retrieval," *Proc. Eighth Int'l Workshop Frontiers in Handwriting Recognition,* pp. 233-238, 2002.

[108] P. Sarkar and G. Nagy, "Style Consistency in Isogenous Patterns," *Proc. 15th Int'l Conf. Pattern Recognition,* vol. 2, pp. 859-862, 2000.

[109] L. Schomaker, L. Vuurpijl, and E. de Leau, "New Use for the Pen: Outline-Based Image Queries," *Proc. Fifth Int'l Conf. Document Analysis and Recognition,* pp. 293-296, 1999.

[110] S. Senda, M. Hamanaka, and K. Yamada, "Box-Free Online Character Recognition Integrating Confidence Values of Segmentation, Recognition and Language Processing," Technical Report of IEICE, PRMU98-138 (1998-12) (in Japanese).

[111] S. Senda and K. Yamada, "A Maximum-Likelihood Approach to Segmentation-Based Recognition of Unconstrained Handwriting Text," *Proc. Sixth Int'l Conf. Document Analysis and Recognition,* pp. 184-188, 2001.

[112] J.-P. Shin and H. Sakoe, "Stroke Correspondence Search Method for Stroke-Order and Stroke-Number Free On-Line Character Recognition—Multilayer Cube Search," *Trans. IEICE Japan,* vol. J82-D-II, no. 2, pp. 230-239, 1999.

[113] J.-P. Shin, "Optimal Stroke-Correspondence Search Method for On-Line Character Recognition," *Pattern Recognition Letters,* vol. 23, pp. 601-608, 2002.

[114] J. Subrahmonia and T. Zimmerman, "Pen Computing: Challenges and Applications," *Proc. 15th Int'l Conf. Pattern Recognition,* vol. 2, pp. 60-66, 2000.
[115] C.Y. Suen, "N-Gram Statistics for Natural Language Understanding and Text Processing," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 1, no. 2, pp. 164-172, 1979.
[116] J.W. Tai, "Some Research Achievements on Chinese Character Recognition in China," *Character and Handwriting Recognition: Expanding Frontiers,* P.S.P. Wang, ed., pp. 199-206, World Scientific, 1992.
[117] K. Takahashi, H. Yasuda, and T. Matsumoto, "A Fast HMM Algorithm for On-Line Handwritten Character Recognition," *Proc. Fourth Int'l Conf. Document Analysis and Recognition,* pp. 369-375, 1997.
[118] K. Takeuchi and Y. Matsumoto, "Japanese OCR Correction Using Stochastic Morphological Analyzer and Probabilistic N-Gram Word Model," *Int'l J. Computer Processing of Oriental Languages,* vol. 13, no. 1, pp. 69-82, 2000.
[119] H. Tanaka et al., "Hybrid Pen-Input Character Recognition System Based on Integration of Online-Offline Recognition," *Proc. Fifth Int'l Conf. Document Analysis and Recognition,* pp. 209-212, 1999.
[120] C.C. Tappert, C.Y. Suen, and T. Wakahara, "The State of the Art in On-Line Handwriting Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 12, no. 8, pp. 787-808, Aug. 1990.
[121] J. Tokuno et al., "Context-Dependent Substroke Model for HMM-Based On-Line Handwriting Recognition," *Proc. Eighth Int'l Workshop Frontiers in Handwriting Recognition,* pp. 78-83, 2002.
[122] W.-H. Tsai and K.-S. Fu, "Subgraph Error-Correcting Isomorphisms for Syntactic Pattern Recognition," *IEEE Trans. Systems, Man, and Cybernetics,* vol. 13, no. 1, pp. 48-62, 1983.
[123] W.-H. Tsai and S.S. Yu, "Attributed String Matching with Merging for Shape Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 7, no. 4, pp. 453-462, 1985.
[124] Y.-T. Tsay and W.-H. Tsai, "Attributed String Matching by Split-and-Merge for On-Line Chinese Character Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, no. 2, pp. 180-185, Feb. 1993.
[125] M.-K. Tsay, K.-H. Shyu, and P.-C. Chang, "Feature Transformation with Generalized LVQ for Handwritten Chinese Character Recognition," *IEICE Trans. Information and Systems,* vol. E82-D, no. 3, pp. 687-92, 1999.
[126] C.-C. Tseng, B.-S. Jeng, and K.-S. Chou, "Candidate Selection in On-Line Chinese Character Recognition System Using Voting Scheme," *J. Information Science and Eng.,* vol. 15, no. 3, pp. 451-462, 1999.
[127] J. Tsukumo and H. Tanaka, "Classification of Handprinted Chinese Characters Using Non-Linear Normalization and Correlation Methods," *Proc. Ninth Int'l Conf. Pattern Recognition,* pp. 168-171, 1988.
[128] M. Umeda, "Advances in Recognition Methods for Handwritten Kanji Characters," *IEICE Trans. Information and Systems,* vol. 79-D, no. 5, pp. 401-410, 1996.
[129] O. Velek, S. Jaeger, and M. Nakagawa, "A New Warping Technique for Normalizing Likelihood of Multiple Classifiers and Its Effectiveness in Combined On-Line/Off-Line Japanese Character Recognition," *Proc. Eighth Int'l Workshop Frontiers in Handwriting Recognition,* pp. 177-182, 2002.
[130] V. Vuori, J. Laaksonen, E. Oja, and J. Kangas, "Experiments with Adaptation Strategies for a Prototype-Based Recognition System for Isolated Handwritten Characters," *Int'l J. Document Analysis and Recognition,* vol. 3, no. 3, pp. 150-159, 2001.
[131] T. Wakahara, "On-Line Cursive Script Recognition Using Local Affine Transformation," *Proc. Ninth Int'l Conf. Pattern Recognition,* pp. 1133-1137, 1988.
[132] T. Wakahara, H. Murase, and K. Odaka, "On-Line Handwriting Recognition," *Proc. IEEE,* vol. 80, no. 7, pp. 1181-1194, 1992.
[133] T. Wakahara et al., "On-Line Cursive Kanji Character Recognition as Stroke Correspondence Problem," *Proc. Third Int'l Conf. Document Analysis and Recognition,* pp. 1059-1064, 1995.
[134] T. Wakahara and K. Okada, "On-Line Cursive Kanji Character Recognition Using Stroke-Based Affine Transformation," *IEEE Trans Pattern Analysis and Machine Intelligence,* vol. 19, no. 12, pp. 1381-1385, Dec. 1997.
[135] P.H. Winston, *Artificial Intelligence,* third ed. Addison Wesley, 1992.
[136] A.K.C. Wong, M. You, and S.C. Chan, "An Algorithm for Graph Optimal Monomorphism," *IEEE Trans. Systems, Man, and Cybernetics,* vol. 20, no. 3, pp. 628-636, 1990.
[137] P.-K. Wong and C. Chan, "Postprocessing Statistical Language Models for a Handwritten Chinese Character Recognizer," *IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics,* vol. 29, no. 2, pp. 286-290, 1999.
[138] W.-Y. Wu and M.-J. Wang, "Detecting the Dominant Points by the Curvature-Based Polygonal Approximation," *CVGIP: Graphical Models and Image Processing,* vol. 55, no. 2, pp. 79-88, 1993.
[139] X.-H. Xiao and R.-W. Dai, "On-Line Handwritten Chinese Character Recognition Directed by Components with Dynamic Templates," *Proc. Int'l Conf. Computer Processing of Oriental Languages '97,* pp. 89-94, 1997.
[140] R. Xu, D. Yeung, W. Shu, and J. Liu, "A Hybrid Post-Processing System for Handwritten Chinese Character Recognition," *Int'l J. Pattern Recognition and Artificial Intelligence,* vol. 16, no. 6 pp. 657-679, 2002.
[141] H. Yamada, K. Yamamoto, and T. Saito, "A Nonlinear Normalization Method for Handprinted Kanji Character Recognition—Line Density Equalization," *Pattern Recognition,* vol. 23, no. 9, pp. 1023-1029, 1990.
[142] K. Yamasaki, "Automatic Prototype Stroke Generation Based on Stroke Clustering for On-Line Handwritten Japanese Character Recognition," *Proc. Fifth Int'l Conf. Document Analysis and Recognition,* pp. 673-676, 1999.
[143] H. Yasuda, K. Takahashi, T. Matsumoto, "On-Line Handwriting Recognition by Discrete HMM with Fast Learning," *Advances in Handwriting Recognition,* S.-W. Lee ed., pp. 19-28, World Scientific, 1999.
[144] M. Yasuda and H. Fujisawa, "An Improvement of Correlation Method for Character Recognition," *Trans. IEICE Japan,* vol. J62-D, no. 3, pp. 217-224, 1979.
[145] P.J. Ye, H. Hugli, and F. Pellandini, "Techniques for On-Line Chinese Character Recognition with Reduced Writing Constraints," *Proc. Seventh Int'l Conf. Pattern Recognition,* pp. 1043-105, 1984.
[146] E.F. Yhap and E.C. Greanias, "An On-Line Chinese Character Recognition System," *IBM J. Research Development,* vol. 25, no. 3, pp. 187-195, 1981.
[147] T. Yokota, S. Kuzunuki, K. Gunji, and N. Hamada, "User Adaptation in Handwriting Recognition by An Automatic Learning Algorithm," *Proc. Ninth Int'l Conf. Human-Computer Interaction,* pp. 455-459, 2001.
[148] R. Zanibbi, D. Blostein, and J.R. Cordy, "Recognizing Mathematical Expressions Using Tree Transform," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 11, pp. 1455-1467, Nov. 2002.
[149] J. Zheng, X. Ding, and Y. Wu, "Recognizing On-Line Handwritten Chinese Character via FARG Matching," *Proc. Fourth Int'l Conf. Document Analysis and Recognition,* pp. 621-624, 1997.
[150] J. Zheng, X. Ding, Y. Wu, and Z. Lu, "Spatio-Temporal Unified Model for On-Line Handwritten Chinese Character Recognition," *Proc. Fifth Int'l Conf. Document Analysis and Recognition,* pp. 649-652, 1999.

Cheng-Lin Liu received the BS degree in electronic engineering from Wuhan University, the ME degree in electronic engineering from Beijing Polytechnic University, the PhD degree in pattern recognition and artificial intelligence from the Institute of Automation, Chinese Academy of Sciences, in 1989, 1992, and 1995, respectively. He was a postdoctoral fellow at the Korea Advanced Institute of Science and Technology (KAIST) and, later, at Tokyo University of Agriculture and Technology from March 1996 to March 1999. He then became a research staff member at the Central Research Laboratory, Hitachi, Ltd. and now is a senior researcher. His research interests include pattern recognition, artificial intelligence, image processing, neural networks, machine learning, and, especially, applications to character recognition and document processing. He is a member of the IEEE and IEEE Computer Society.

Stefan Jaeger graduated from the University of Kaiserslautern, Germany, in computer science in 1994 and received the PhD degree in computer science from Albert-Ludwigs University, Freiburg, Germany, in 1998. From 1994 to 1998, he worked as a PhD student at the Daimler-Benz Research Center, Ulm, Germany, where he was engaged in offline cursive handwriting recognition for postal mail sorting. His PhD thesis addressed the problem of recovering dynamic information from static handwritten word images and was awarded the Dissertation Prize from the German Research Centers for Artificial Intelligence in 1999. In 1998, he joined the Interactive Systems Laboratories jointly located at Carnegie Mellon University, Pittsburgh, Pennsylvania, and at the University of Karlsruhe, Germany, where he was a research staff member responsible for online handwriting recognition and pen-computing. Since November 2000, he has been working as a senior researcher in the Department of Computer, Information, and Communication Sciences at Tokyo University of Agriculture and Technology, Japan. His research interests include pattern recognition, artificial intelligence, document analysis, handwriting recognition, and machine learning.

Masaki Nakagawa received BSc and MSc degrees from the University of Tokyo in 1977 and 1979, respectively. During the academic year 1977-1978, he followed the computer science course at Essex University in England, and received the MSc degree with distinction in computer studies in July 1979. He received the PhD degree in information science from the University of Tokyo in December 1988. From April 1979, he has been working at Tokyo University of Agriculture and Technology. Currently, he is a professor of media interaction in the Department of Computer, Information, and Communication Sciences. For the past 10 years, he has been collaborating with industry to advance pen-based human interactions composed of online handwriting recognition, pen-based interfaces, and applications, especially educational applications on an interactive electronic whiteboard. He has been serving for several committees of the Japanese government on Industry and Universities partnership and those on IT-oriented and IT-supported learning. He is a member of the IEEE Computer Society.