Recognition of Cursive Roman Handwriting - Past, Present and Future

Horst Bunke
Department of Computer Science, University of Bern
Neubrückstrasse 10, CH-3012 Bern, Switzerland
bunke@iam.unibe.ch

Abstract

This paper reviews the state of the art in off-line Roman cursive handwriting recognition. The input provided to an off-line handwriting recognition system is an image of a digit, a word, or - more generally - some text, and the system produces, as output, an ASCII transcription of the input. This task involves a number of processing steps, some of which are quite difficult. Typically, preprocessing, normalization, feature extraction, classification, and postprocessing operations are required. We'll survey the state of the art, analyze recent trends, and try to identify challenges for future research in this field.

Keywords: cursive handwriting recognition, off-line handwriting, Roman script, isolated digits and characters, digit string, word recognition, word sequence recognition

1 Introduction

Automatic handwriting recognition has been a subject of research for more than 40 years [107]. On the one hand, the reading of human handwriting by machine has been considered an interesting and intellectually challenging problem in its own right. To approach, or even surpass, the performance of humans in text recognition has been a major driving force behind many research activities. On the other hand, the field has been quite important from the commercial and application oriented point of view. Automatic address reading [57, 111], bank check processing [53], and recognition of text filled in by hand on forms have been major challenges in automatic handwriting recognition research. Moreover, handwritten data has often been used to validate and test the performance of new methods developed in pattern recognition.

Since the beginning of the 1990's a significant growth of activities in handwriting recognition research has been observed. One indicator is the increasing number of publications that appeared in related journals (for a sample see the references). Another indicator is the growing interest in specialized workshops and conferences. There is no doubt that enormous progress has taken place in this area. For example, for the tasks of handwritten address reading and amount recognition on bank checks, commercial systems have become available [35, 57, 111]. Nevertheless, there is a clear need to further develop the field. All successful applications, for example, address and check reading, work in narrow domains with limited vocabularies, where task specific knowledge and constraints are available. Examples are the relation between zip code and city name in address reading, or the redundancy of courtesy and legal amount on a check. However, when it comes to general word or sentence recognition where no constraints exist and one is faced with a large, possibly open lexicon, the state of the art is quite limited and recognition rates are rather low. Yet the problem of unconstrained word and sentence recognition is important in a number of future applications, for example, the transcription of personal notes, faxes, and letters, or the electronic conversion of historical handwritten archives in the context of the creation of digital libraries [117].

The field of handwriting recognition can be divided into on-line and off-line recognition. In on-line recognition the writer is physically connected to a computer via a mouse, an electronic pen, or a touch sensitive device and his or her handwriting is recorded as a time-dependent process. By contrast, in the off-line mode handwriting is captured by means of a scanner and becomes available in form of an image without any temporal information. Also the use of cameras for capturing handwriting is becoming increasingly popular [22]. Because of the lack of temporal information, off-line recognition is considered the more difficult problem. In the current paper we will focus our attention on off-line recognition. However, it has to be noted that there

Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003)
0-7695-1960-1/03 $17.00 © 2003 **IEEE**

are close relations between the two modalities. First, many methods, particularly those that are applied after the writing has been converted into a feature vector or a sequence of feature vectors, are applicable to both types of problems. Secondly, many attempts have been reported in the literature to convert one modality into the other so as to make on-line methods available for off-line recognition tasks and vice versa. Converting on-line handwriting into the off-line format is more or less straightforward. The other direction is rather complicated, because it requires reconstructing the temporal order of the individual strokes from the image of a given handwritten text. Nevertheless a number of solutions to this problem have been reported [23, 58, 92].

A task related to handwriting recognition is automatic writer identification. It has applications in the forensic sciences [50] and in handwritten document retrieval. Writer identification can be understood as a classification problem where a word, text fragment, or text is to be assigned to one out of a number of possible writers. Another task related to handwriting recognition is automatic signature verification. Signature is regarded an important biometric feature that is very useful to establish the identity of a person [54]. We will not review writer identification and signature verification in greater detail in this paper, although they share some common subtasks with handwriting recognition, for example, preprocessing, feature extraction and classification methods. Recent work in this field has been reported in [13, 49, 84, 100]. An earlier survey can be found in [70].

In this paper we focus on the recognition of cursive Roman script only. The problem of Asian script recognition is addressed in [113]. It has to be noted, however, that some of the methods discussed in this paper can be used for either problem. There are books available that address important aspects of handwriting recognition for both Roman and Asian scripts [8, 86]. Generic classification methods that are applicable to a wide spectrum of tasks in handwriting recognition can be found in [24, 104] and similar books. Recent surveys on cursive Roman script recognition are [93, 112, 120]. Moreover, noteworthy is the collection of articles by O'Gorman and Kasturi [89].

2 State of the Art

Roman cursive handwriting recognition can be divided into the tasks of recognizing isolated characters or digits, individual words, and unconstrained text consisting of a sequence of an a priori unknown number of words. Recognition of isolated characters and digits is by far the simplest problem for which mature solutions have become available. The other two problems, word and word sequence recognition, are considerably more difficult and are still widely unsolved. In this section we will first review methods of text image preprocessing and normalization that are commonly found in any of the three tasks mentioned above. Subsequently the recognition of isolated characters and digits, individual words, and sequences of words will be discussed.

2.1 Document Image Preprocessing

In the off-line mode an image of the handwriting to be recognized is captured by a sensor, for example, a scanner or a camera. Traditionally the first processing step consists in converting the grey level image acquired by the sensor to a binary image. A number of algorithms for this step are known from the literature, see Chapter 2 of [89] and Chapter 4 of [86]. Nowadays, however, with increasing processing speed and memory capacity of modern computers, the direct use of grey-level images is becoming more and more popular.

Before or after binarization, images are often filtered to remove noise. Popular methods of noise removal in both binary and grey-level images are based on image filtering theory and mathematical morphology, see Chapter 6 of [86], and [41].

Very often the paper document is not perfectly aligned with the coordinate system of the scanner. To recover from artefacts of this kind, skew correction methods are applied. Methods for skew angle estimation are based on horizontal projection profiles, the handwriting's contour, and other quantities [4, 76, 87].

In many applications there are no constraints imposed on the writing instrument. Consequently there is considerable variation in stroke width across different input samples. To normalize the handwriting, thinning or skeletonization methods are often applied. The aim of these methods is to normalize the width of each stroke to one pixel, while maintaining the topology of the patterns under consideration; see Chapter 5 of [86] and [116].

A feature that varies from one individual to another is the slant of the handwriting. Training a classifier on writing that hasn't been slant corrected may require significantly more effort and training data. Consequently, slant removal is an operation that is found in almost any handwriting recognition system. Methods for slant estimation are depending on the considered task. For isolated character and digit recognition, often the angle of the principle axis is used as an estimation [42]. For word or word sequence recognition it is common to approximate the handwriting's contour by straight line segments and use the average or median direction of these straight segments as an estimate of the slant [9, 122]. Other methods have been reported in [4, 60, 61, 105].

Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003)
0-7695-1960-1/03 $17.00 © 2003 IEEE

2.2 Recognition of Isolated Characters

The task of isolated character recognition is usually cast as a pattern classification problem, where an unknown input pattern is to be assigned to one out of a number of given classes. Most approaches to isolated character recognition follow the traditional paradigm of pattern recognition. There are three main processing steps carried out in sequential order, namely, preprocessing and normalization, feature extraction, and classification. Typical preprocessing and normalization steps have been described in Section 2.1. A large number of features for isolated character recognition and corresponding extraction methods have been proposed in the literature. They include moments and quantities derived from series expansion (see Chapter 7 in [86]), features based on projection profiles and on the contour [42], as well as structural features such as endpoints, fork points, holes, length, shape, or curvature of individual strokes that occur as part of the character under consideration [115]. Also features extracted via principal component analysis from the set of pixels in an image [30] and the raw pixel matrix itself have been used.

Once a feature vector has been extracted from the image of a handwritten character, it is fed into a classifier. Pattern classification has a long history and almost all generic classifiers that were proposed at some time have been applied to isolated character recognition. Concrete examples include nearest- or k-nearest-neighbor classifier [42, 109], Bayes classifier [66, 104], polynomial classifier [30], neural network [42, 66, 73], and support vector machine [1, 72, 123]. Also structural classifiers which use string or graph representations of the characters to be classified have been proposed, see [12, 72], Chapters 12 and 13 in [86], and [18, 75].

It has been widely debated which classifier is best for handwritten digit recognition. A number of studies have been published where different methods have been compared to each other [72]. From these studies it can be concluded that there is no 'universally best' classifier. Which method is to be preferred over another depends not only on the classification performance, but also on a number of additional factors, including, among others, the number of free parameters of the classifier, the size of the available training set and the time needed for training.

2.3 Cursive Word Recognition

One possible approach to word recognition is to segment the given input word into a sequence of characters and then recognize each individual character using one of the methods described in Section 2.2. It turns out, however, that the extraction of isolated characters from a word is extremely difficult without knowing the word's identity. Hence one is confronted with a 'chicken-and-egg' problem (also known as Sayre's Paradox [103]): If the identity of the word were known, its segmentation into individual characters would be feasible. But to know the word's identity, we need to segment it first. To overcome this dilemma, different approaches to word recognition have been proposed. They all try to cope, in the one or the other way, with the segmentation problem. Three prominent examples, known as the holistic, split-and-merge, and segmentation-free approach, respectively, will be discussed below.

2.3.1 Holistic Methods

Holistic methods have been proposed to completely bypass the difficult problem of segmenting a word into its individual characters. Here the image of the given word is considered as an entity in its whole and it is attempted to classify it, given a dictionary of possible words. Features that characterize a word as a whole are loops, ascenders, descenders, face-up and face-down valleys, as well as local and global shape descriptors. Usually these features are ordered so as to reflect their left-to-right order of occurrence in the word under consideration. Then the features of an unknown input word are matched against known prototypes. For a recent survey on holistic word recognition, including pointers to the related literature, see [77].

The holistic approach is limited in that it can't deal properly with a large number of classes. It has been successful, however, as a method for lexicon reduction. Hence holistic classifiers are suitable to be used in conjunction with one of the two other approaches discussed below.

2.3.2 Segmentation Based Approaches

Segmentation based approaches try to segment a given word into smaller entities. However, as it is extremely difficult, if not impossible, to segment a given word into its individual characters without knowing the word's identity, they usually split a word into entities that don't necessarily correspond to exactly one character each, and they consider a number of possible segmentation alternatives at the same time. Typically, an oversegmentation of the given input word is attempted. That is, the image of a character that occurs within a word may be broken into several constituents, also called graphemes. At the same time the segmentation procedure avoids merging two adjacent characters, or parts of two adjacent characters, into the same constituent. A large number of heuristics for achieving such kind of segmentation have been reported in the literature. For surveys see [11, 110].

Once the given input word has been transformed, through segmentation, into a sequences of graphemes, $(g_1, g_2, \ldots, g_n)$, all possible combinations of adjacent graphemes, up to a maximum number M, are considered and fed into a recognizer for isolated characters. Typically it is supposed that the recognizer not only returns an ordered list of class names, but also renders a confidence for each class. Once all possible combinations of graphemes have been classified, a search procedure is applied that finds, based on the confidence values returned by the classifier, the best sequence of characters matching the input word image. Typically, dynamic programming [110] or some A*-type search algorithm [43], is used. The search procedure is often run under the control of a dictionary of legal words. As an alternative, the dictionary may be used in a postprocessing phase.

The procedures sketched in the previous paragraph provide just a generic framework. Many instances of this generic procedure have been reported in the literature [4, 14, 28, 73].

An advantage of segmentation based word recognition schemes as discussed above is that the problem is reduced to isolated character recognition - a problem for which a number of quite mature algorithms have become available. On the other hand, segmentation and grapheme recombination are both based on heuristic rules that are derived by human intuition. The development of automatic procedures that are able to learn segmentation rules from training data and automatically infer the parameters that guide the search for fitting the optimal character hypotheses is still an open problem.

2.3.3 HMM Based Recognition

The third main approach to cursive handwritten word recognition is based on hidden Markov models (HMMs). For all technical details of HMMs we refer the reader to [94]. For a survey of HMMs in cursive handwriting recognition see [67]. HMMs qualify as a suitable tool for cursive script recognition for a number of reasons. First they are stochastic models able to cope with noise and shape variations that occur in handwriting. Next, the number of tokens, or feature vectors, representing an unknown word may be of variable length. This is a fundamental requirement in cursive handwriting recognition because the lengths of the individual input words exhibit a great degree of variation. Moreover, using an HMM-based approach, the segmentation problem, which is extremely difficult and error-prone, can be avoided. Finally there are standard algorithms known from the literature for both training and recognition using HMMs. These algorithms are reasonably fast and can be implemented with ordinary effort. Also software packages including all necessary modules for training and decoding have become available [17, 129].

When using HMMs for a classification problem an individual HMM is usually constructed for each pattern class. For each sequence of feature vectors extracted from the input pattern, the likelihood that this sequence was produced by a particular HMM can be computed. Eventually, the class with the highest likelihood value is chosen as the recognition result. In word recognition systems with a small vocabulary, it is possible to build an individual HMM for each word. But for large vocabularies this method is not applicable any longer because of lack of sufficient training data. In this case HMMs are built on a character basis and character models are concatenated to word models. In this way the training data are more intensively used [82].

In order to optimize recognition performance, the HMMs have to be fitted to the considered problem. In particular the number of states, the possible transitions, and the feature vector probability distributions have to be chosen. Because of the linear, left-to-right direction of handwriting, a linear transition structure is often adopted (i.e. the state transition probabilities are chosen in such a way that a linear left-to-right ordering of the states is imposed). The feature distributions are usually assumed to be Gaussian or mixtures of Gaussians. To adjust the remaining free parameters of an HMM, i.e. the transition probabilities and the parameters of the feature probability distributions, Baum-Welch training, a special version of the EM-algorithm is often used [94].

Features are normally extracted in a left-to-right scan over the word to be recognized. Often a sliding window is used. The features describe structural properties, such as ascenders, descenders, loops or cusps, or they are derived from the greylevel distribution of the pixels in the window. Examples of HMM-based word recognizers can be found in [7, 27, 34, 57, 64, 68, 81, 127]. As an extension of pure HMM-based recognition, some authors have proposed to use HMMs together with neural networks in hybrid systems [65, 97, 105]

A special case of cursive words is digit string. The recognition of digit strings is needed in address reading (zip code) and bank check processing (courtesy amount). This problem is simpler than word recognition from the point of view that there exist only ten different character classes rather than 26 (or 52 if both small and capital letters occur). On the other hand, as almost any sequence of digits may represent a courtesy amount on a check or almost any digit can follow another digit in a zip code, much weaker lexical constraints are applicable in digit string recognition than in word recognition. Except for the utilization of lexical constraints, methods for the recognition of digit strings are similar to those used for cursive word recognition. Examples of systems specifically developed for digit string recognition are [15, 43, 71].

Finally we want to point out that there is another group of successful approaches to word recognition, which are inspired by human perception. Some representative references are [19, 25, 74].

2.4 Cursive Word Sequence Recognition

In its most general form, cursive handwriting recognition requires the transcription of some handwritten text that consists of a sequence of words, for example, a phrase, a sentence, or a whole essay, where the text may occupy a line, several lines, or a whole page. Similarly to the task of word recognition where it has been attempted to segment a word into its constituent characters and then to recognize the individual characters, it has been proposed to segment a line of text into isolated words and then to recognize these words using one of the methods discussed in Section 2.3. Various segmentation procedures have been proposed in this context. Many of them are based on analyzing the distances between connected components. From the theoretical point of view the problem can be seen as a classification or clustering task, where a space between two consecutive connected components is to be assigned to the class 'within-word' or 'between-words'. For more details of the segmentation and the corresponding word recognition procedures see [4, 60, 61, 69, 81, 105].

The problem of segmenting a line of text into words is usually easier than the problem of segmenting a word into its constituent characters. Nevertheless, there are cases where complicated ambiguities arise. Consequently, segmentation-free methods based on HMMs have been proposed for the task of word sequence recognition. The principal idea is to concatenate character models to word models and word models to word sequence models. In this way the segmentation of a word sequence into individual words is delivered as a by-product of the recognition process, like the segmentation of a word into its constituent characters is delivered as a by-product of HMM-based word recognition. This technique has been successfully used in some systems [82, 122].

In cursive word recognition, a lexicon of legal words is usually assumed. This lexicon significantly decreases the number of possible character sequences to be taken into account by the recognizer. In unconstrained natural text recognition, no direct analogy exists. However, linguistic knowledge can be exploited in order to restrict the number of possible combinations of words in a handwritten sentence. One of the most popular methods to incorporate linguistic knowledge into a recognizer is word N-grams [55, 98]. A word N-gram is a sequence of words of length N with an associated probability of occurrence. N-gram probabilities are usually estimated from natural language corpora. They are utilized in the recognition process by weighting the word confidence values returned by the recognizer with the corresponding N-gram probabilities. In [80] it has been shown that N-gram models are suitable to decrease the perplexity in the recognition process, i.e. to narrow down the average number of possible successors of a word within a sentence. It is possible to utilize the N-gram probabilities in a postprocessing phase or to directly integrate them into the recognition process [82]. Often there is not enough training data available for robust estimation of N-gram probabilities if $N > 3$ (it may be even a problem for $N = 3$ or $N = 2$). However in this case back-off modeling and similar techniques can be applied [98]. An alternative is to use N-grams at the level of syntactic word classes [98]. Here each word gets assigned a syntactic tag, such as *noun, verb, adverb* etc., and N-grams relate to the syntactic classes rather than individual words. Under this approach N-grams can be estimated more robustly, but the constraints imposed by the language model on possible sequences of words become weaker.

Another approach to integrating linguistic knowledge into the recognition process is based on language syntax and syntactic parsing [47, 51, 131]. Here the output of the recognizer is constrained to those sequences of words that constitute syntactically correct sentences of the underlying language. Similar ideas have been reported for the recognition of legal amounts on bank checks [21, 59]. For this application it is easy to construct the underlying grammar by hand (for example, a grammar that defines all legal amounts up to one million). However, for natural languages it is still a challenge to infer a grammar automatically from a body of given text.

3 Recent Trends

Many of the methods discussed in Section 2 have become state-of-the-art in cursive handwriting recognition. Despite a certain level of maturity has been reached, there is still an urgent need to further improve the available recognition technology. In this section we'll discuss a few trends that have emerged recently. Of course, the selection is based on the author's subjective opinion and no claim of completeness is made.

3.1 Databases and Performance Evaluation

The availability of large amounts of data for training and robust testing is a fundamental prerequisite for building a handwriting recognition system. Furthermore, with more and more recognition methods becoming available, the comparison and benchmarking of these methods is becoming increasingly important. Consequently, the acquisition of standard databases has become an issue of great concern in the handwriting research community. Since both the collection of the data and the preparation of the ground truth, i.e. the ASCII transcription of the handwritten text, are expensive and time consuming tasks, it is highly desired to reuse existing databases as much as possible. This also facilitates the direct comparison of different recognition algorithms.

A survey of existing databases for handwriting recognition research covering the state of the art until about 1996 has been provided in [39]. Databases included in this survey are CEDAR [52], NIST [126] and CENPARMI [115]. Moreover there are databases for the online domain [40] and for Asian characters [101]. A database that contains complete handwritten sentences has been described in [105]. This database has been acquired in the course of developing an HMM-based sentence recognizer. All sentences have been written by the same writer. In [119] a publicly available database is described that contains both on-line and off-line data of handwritten isolated characters, digits and cursive words. A database containing essays written by students is described in [26]. Furthermore a new database designed to support research on bank check processing has been presented in [21].

A fairly large database containing full sentences is the IAM database described in [83]. This database is similar to the one described in [105] in that it is build up from sentences contained in the LOB corpus [56]. However, it is significantly larger than [105] and includes texts from multiple writers. The version of the database described in [83] contains 82,227 instances of handwritten words distributed over 9,285 lines of text produced by approximately 400 writers. The underlying lexicon includes 10,841 different words. Because all texts come from the LOB corpus, which is electronically available, it is possible to automatically generate various kinds of language models. This property makes the database interesting for the development of recognizers that use linguistic knowledge beyond the lexicon level. Originally, the database was designed so as to support the development of a text line recognizer. Consequently, the basic units in the database are complete lines of text. However, a set of segmentation tools have been developed meanwhile that allow splitting a line of text into individual words [83]. Moreover a novel word segmentation procedure that makes use of the ASCII ground truth has been described in [130]. On a subset of about 3,700 lines this tool achieved a correct word segmentation rate of 98%. Hence it can be expected that a fully segmented version of this database will be available soon.

The IAM database was instrumental in the development of a number of handwriting recognition systems at the University of Bern. However it has also been used by other research groups [5, 60, 122] The database is still being enlarged and freely available upon request.[1]

3.2 Synthetic Training Data

All methods for handwritten character, word or sentence recognition need to be trained. As a rule of thumb, the larger the training set, the better is the recognition performance of the system. This empirical finding has been confirmed in a number of experiments [10, 99, 109]. However, the acquisition of training data is a tedious and expensive process with clear limitations.

In the area of machine printed character recognition it was proposed to use synthetic data for training. A number of successful activities in this direction have been reported in the literature. Using a degradation model, Baird successfully constructed a Tibetan OCR system using training data that was initialized with real images but augmented with synthetic variations [2]. Based on the same degradation model, a full-ASCII, 100-typeface classifier was developed using exclusively synthetic training data [29]. A recent review on document image degradation models and their use in synthetic data generation of machine printed character recognition can be found in [3]. A system for machine printed Arabic OCR that was trained on synthetic data only is described in [78].

Recently similar ideas were proposed in the field of handwriting recognition. In [10, 85] the synthetic generation of isolated characters has been described. The generation of synthetic handwritten words and sentences has been described in [38]. The basic idea is to use image templates consisting of n-tuples of characters (with $n = 1, 2, 3$) and to concatenate them to generate words and word sequences from a given ASCII text. A similar approach was adopted in [44]. However, while the aim in [38] was to produce naturally looking handwritten notes from ASCII text for personal communication, the method described in [44] has been tested in conjunction with an HMM-based recognizer for handwritten word sequences. A number of different alternatives in synthetic handwriting generation, with varying degree of complexity, have been explored. Under the most elaborate model, the system trained exclusively on synthetic data reached a recognition rate comparable to that of the same system trained with natural handwriting only.

In [118] a geometrical distortion model for complete lines of handwritten text was proposed. The model is

[1] Please contact the author or Matthias Zimmermann at zimmerma@iam.unibe.ch. See also: http://www.iam.unibe.ch/ zimmerma/iamdb/iamdb.html

based on a combination of periodic functions that control the strength of geometrical distortion, affecting the slant, width and height of the writing, the shape of the baseline, and other parameters. A number of experiments with an HMM-based recognizer have been described. From these experiments it can be concluded that the use of synthetic training data can improve the recognition performance, particularly when the training data has been produced by a small population of writers and a writer independent recognition task is considered (i.e. the writers who rendered the test samples are not represented in the training set).

Finally it has to be noted that the use of synthetic data is not limited to enlarging the training set. It may be meaningful as well to generate synthetic data for extensively testing a system under various conditions. Synthetic data has also been used in the recognition phase, making a classifier insensitive to perturbations that naturally occur in handwriting [42]. Further methods for synthetic handwriting generation have been reported in [91, 106, 124].

3.3 Multiple Classifier Systems

Recently it has been shown that systems incorporating multiple classifiers have the potential of improved classification accuracy over single classifiers in difficult classification tasks [63]. Particularly in handwriting recognition the use of multiple classifier systems has been advocated by many authors. Examples include [31, 33, 88, 108, 114].

In order to actually build a multiple classifier system one needs a number of basic classifiers first. Very often the design of these basic classifiers is guided by intuition and heuristics. For example, different sets of features and/or different classification algorithms may be used [31]. Sometimes, different sources of information, which are redundant or partly redundant to each other, are exploited, for example, zip code and city name in postal address reading [57], or legal and courtesy amount in bank check processing [53, 59]. Recently a number of procedures for classifier generation, called ensemble creation methods, have been proposed in the field of machine learning. They are characterized by the fact that they produce several classifiers out of one base classifier automatically. Prominent examples are Bagging [6], Adaboost [32] and random subspace method [45]. For a summary of these methods see [20]. Applications to the recognition of cursive words are described in [36, 37].

Once a number of classifiers have been generated, an appropriate procedure has to be defined to combine their outputs in order to derive the ensembles' final result. Many methods for classifier combination have been proposed in the literature [96]. They depend on the type of output produced by the individual classifiers. If the output is only the best-ranked class then majority voting can be applied [95]. More sophisticated combination schemes look at dependencies between classifiers in the so-called behavior-knowledge space [48]. If the classifiers' output is a ranked list of classes, Borda count or related methods can be applied [46]. In the most general case, a classifier outputs a score value for each class. Then the sum, product, maximum, minimum, or the median of the scores of all classifiers can be computed for each class and the class with the highest value is output as the ensembles' classification result [62]. It is also possible to weight each classifier according to its recognition performance and then apply a combination rule. This strategy has been adopted in Adaboost [32]. More sophisticated combination procedures use the score values output by the individual classifiers as input for a trainable classifier, for example a neural network that acts as a combiner [33]. Another interesting approach is to view the selection of the individual classifiers of the ensemble, including their weights and perhaps even the combination procedure, as an optimization problem and find the solution by means of evolutionary search procedures [108].

All classifier combination rules discussed above are not applicable if each classifier of the ensemble outputs a sequence of class names rather than just a single class name. Such a situation typically occurs in word sequence recognition. Because of segmentation errors it can not be assumed that the sequences produced by the different classifiers are all of the same length. Therefore, some synchronization mechanism is needed. It has been proposed to use dynamic programming techniques in order to optimally align the individual classifiers' outputs. However this topic is still under research and only a few solutions have been reported in the literature [79, 125, 128].

4 Outlook and Conclusions

The focus of attention in handwriting recognition research has been gradually shifting from isolated character recognition to more complex tasks, such as recognition of words and unconstrained text. Some level of maturity has been reached for isolated characters and digits, but recognition rates in word and word sequence recognition are still rather low. There is no doubt that more research is needed in these areas, particularly as there are some interesting potential applications, examples of which include the automatic reading of personal notes and communications, and the transcription of handwritten archives in the advent of digital libraries.

One very important issue in promoting research in handwriting recognition is the acquisition of publicly available databases of large size. In the past databases were often kept private and considered proprietary assets. Particularly for the development of enhanced algorithms for word and unconstrained text recognition, there is an urgent need for more, larger, and more diverse databases that can be used by everybody working in the field. Another challenge is to coordinate database acquisition activities so as to arrive at a common format, common test protocols, and logical and physical links between different databases. It is furthermore interesting to explore to which degree the inclusion of synthetically generated data in these databases could be useful for handwriting recognition.

It can be expected that a significant percentage of future systems for cursive handwriting recognition will be personal systems serving a single user. Such systems will perform best when they are trained with data provided by the future user. However it can be quite cumbersome for an individual to provide a sufficiently large body of handwritten samples to train a system. A possible way out of the dilemma is to use data from a general pool of writers in a first training phase and to adjust the system in a second phase, using some carefully chosen samples provided by the future user [121]. However, the question of writer-dependent vs. writer-independent performance as well as adjusting an existing system to some particular writing style has been rarely addressed in the literature, although it may be a great practical relevance in the near future.

There is no doubt that many improvements in cursive handwritten word and word sequence recognition are due to the application of HMMs. It is interesting to note that most of the HMM technology that is used in handwriting recognition today has been adopted, without any modifications except for preprocessing and feature extraction, from speech recognition, although speech is a one-dimensional signal while handwriting is intrinsically two-dimensional. There has been surprisingly little work on developing two-dimensional HMMs or two-dimensional HMM-like stochastic models [16, 90, 102]. A major obstacle in developing such methods is surely their complexity. However with an steadily increasing power of modern computers and the potential of synthetic training data generation, two-dimensional stochastic models seem a very promising way to improve current handwriting recognition methodology.

The ultimate goal of handwriting recognition is to have machines which can read any text with the same recognition accuracy as humans but at a faster rate [74]. There are many activities currently going on to bring the state of the art closer to that goal, in particular the ones discussed in Section 3. Eventually, however, in order to reach the performance of humans in handwritten text reading, we must aim not only at the *transcription*, but also at the *understanding* of the given text. This includes syntactic and semantic text analysis. Consider, as an analogy, the scenario of a human reader who is faced with the task of transcribing a handwritten text: a) on some subject he or she is familiar with, in his or her native language; b) on some subject he or she is not familiar with, in his or her native language; c) in a foreign language he or she doesn't understand. Clearly, his or her recognition performance will deteriorate as we move from a) to b), and from b) to c). However, in cursive handwriting recognition (and also in machine printed OCR), we are still at a stage that is comparable to c). Very few attempts have been reported in the literature to integrate methods from natural language parsing and text understanding into a recognizer [47, 51, 131]. However there are such methods available from natural language understanding. Natural language processing techniques, and machine tanslation [98] are very promising to improve the recognition performance of today's handwriting recognition procedures. In addition they would naturally lead to tools for content based search and retrieval in the context of archives of handwritten texts.

Acknowledgment

The author wants to thank his PhD students S. Günter, T. Varga and M. Zimmermann for valuable contributions to this paper. Further thanks are due to U. Miletzki for useful hints. Financial support by the Swiss National Science Foundation (20-52087.97) is gratefully acknowledged. Additional funding was provided through the Swiss National Science Foundation NCCR program "Interactive Multimodal Information Management (IM2)" in the individual project "Scene Analysis".

References

[1] N. Ayat, M. Cheriet, L. Remaki, and C. Suen. KMOD - a new support vector machine kernel with moderate decreasing for pattern recognition. In *6th Int. Conference on Document Analysis and Recognition*, pages 434–438, 2001.

[2] H. Baird. Document image defect models. In H. Baird, H. Bunke, and K. Yamamoto, editors, *Structured Document Image Analysis*, pages 546–556. Springer, 1992.

[3] H. Baird. The state of the art in document image degradation modeling. In *4th IAPR Workshop on Document Analysis Systems*, pages 1–13, 2000.

[4] R. Bozinovic and S. Srihari. Off-line cursive script word recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(1):68–83, Jan. 1989.

[5] A. Brakensiek, J. Rottland, A. Kosmala, and G. Rigoll. Off-line handwriting recognition using

various hybrid modeling techniques and character n-grams. In *7th Int. Workshop on Frontiers in Handwriting Recognition*, pages 343–352, 2000.

[6] L. Breiman. Bagging predictors. *Machine Learning*, 2:123–140, 1996.

[7] H. Bunke, M. Roth, and E. Schukat-Talamazzini. Off-line handwriting recognition using hidden Markov models. *Pattern Recognition*, 55(1):75–89, 1995.

[8] H. Bunke and P. Wang, editors. *Handbook of Character Recognition and Document Image Analysis*. World Scientific, 1997.

[9] T. Caesar, J. M. Gloger, and E. Mandler. Preprocessing and feature extraction for a handwriting recognition system. In *2nd Int. Conference on Document Analysis and Recognition*, pages 408–411, 1993.

[10] J. Cano, J.-C. Perez-Cortes, J. Arlandis, and R. Llobet. Training set expansion in handwritten character recognition. In T. Caelli, A. Amin, R. Duin, M. Kamel, and D. de Ridder, editors, *Structural, Syntactic and Statistical Pattern Recognition*, pages 548–556. LNCS 2396, Springer, 2002.

[11] R. Casey and E. Lecolinet. A survey of methods and strategies in character segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18:690–706, 1996.

[12] S. Cha, Y. Shin, and S. Srihari. Approximate stroke sequence matching algorithm for character recognition and analysis. In *5th Int. Conference on Document Analysis and Recognition*, pages 53–56, 1999.

[13] S.-H. Cha and S. Srihari. Multiple feature integration for writer verification. In *7th Int. Workshop on Frontiers in Handwriting Recognition*, pages 333–342, 2000.

[14] W. Chen, P. Gader, and H. Shi. Lexicon driven handwritten word recognition using optimal linear combinations of order statistics. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21:77–82, 1999.

[15] Y.-K. Chen and J.-F. Wang. Segmentation of single- or multiple-touching handwritten numeral string using background and foreground analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22:1304–1317, 2000.

[16] A. Choisy and A. Belaid. Cross-learning in analytic word recognition without segmentation. *Int. Journal on Document Analysis and Recognition*, 4:281–289, 2002.

[17] R. Collobert, S. Bengio, and J. Mariéthoz. Torch: a modular machine learning software library. IDIAP-RR 46, IDIAP, 2002.

[18] L. Cordella, P. Foggia, C. Sansone, and M. Vento. Subgraph transformation for inexact matching of attributed relational graphs. *Computing*, (Supplement 12):43–52, 1997.

[19] M. Cote, E. Lecolinet, M. Cheriet, and C. Suen. Building a perception based model for reading cursive script. In *3rd Int. Conference on Document Analysis and Recognition*, volume 2, pages 898–901, 1995.

[20] T. Dietterich. Ensemble methods in machine learning. In J. Kittler and F. Roli, editors, *1st Int. Workshop on Multiple Classifier Systems*, pages 1–15, 2000.

[21] G. Dimauro, S. Impedovo, R. Modugno, and G. Pirlo. A new database for research on bank-check processing. In *8th Int. Workshop on Frontiers in Handwriting Recognition*, pages 524–528, 2002.

[22] D. Doermann. Progress in camera-based document image analysis. In *7th Int. Conference on Document Analysis and Recognition*, 2003.

[23] D. Doermann and A. Rosenfeld. Recovery of temporal information from static images of handwriting. *Int. Journal of Computer Vision*, 15:143–164, 1995.

[24] R. Duda, P. Hart, and D. Stork. *Pattern Classification, 2nd Edition*. John Wiley and Sons, Inc., 2001.

[25] S. Edelman, T. Flash, and S. Ullmann. Reading cursive handwriting by alignment of letter prototypes. *Int. Journal of Computer Vision*, 5:303–331, 1990.

[26] D. Elliman and N. Sherkat. A truthing tool for generating a database of cursive words. In *6th Int. Conference on Document Analysis and Recognition*, pages 1255–1262, 2001.

[27] C. Farouz, M. Gilloux, and J.-M. Bertille. Handwritten word recognition with contextual hidden Markov models. In S.-W. Lee, editor, *Advances in Handwriting Recognition*, pages 183–192. World Scientific, 1999.

[28] J. Favata. Offline general handwritten word recognition using an approximate BEAM matching algorithm. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23:393–398, 2001.

[29] R. Fossey and H. Baird. A 100 font classifier. In *1st Int. Conference on Document Analysis and Recognition, St. Malo, France*, 1991.

[30] J. Franke. Isolated handprinted digit recognition. In *[8]*, pages 103–121.

[31] J. Franke, L. Lam, R. Legault, C. Nadal, and C. Suen. Experiments with the CENPARMI database combining different classification approaches. In *3rd Int. Workshop on Frontiers in Handwriting Recognition*, pages 305–311, 1993.

[32] Y. Freud and R. Shapire. A decision theoretic generalization of online learing and and application to boosting. *Journal on Computer and Systems Sciences*, 55:119–139, 1997.

[33] P. Gader, M. Mohamed, and J. Keller. Fusion of handwritten word classifiers. *Pattern Recognition Letters*, 17:577–584, 1996.

[34] M. Gilloux, M. Leroux, and J.-M. Bertille. Strategies for handwritten words recognition using hidden Markov models. In *2nd Int. Conference on Document Analysis and Recognition*, pages 299–302, 1993.

[35] N. Gorski, V. Anisimov, E. Augustin, O. Baret, and S. Maximor. Industrial bank check processing: the A2iA check reader. *Int. Journal on Document Analysis and Recognition*, 3:196–206, 2001.

[36] S. Guenter and H. Bunke. Creation of classifier ensembles for handwritten word recognition using feature selection algorithms. In *8th Int. Workshop on Frontiers in Handwriting Recognition*, pages 183–188, 2002.

[37] S. Guenter and H. Bunke. Ensembles of classifiers for handwritten word recognition. *Int. Journal on Document Analysis and Recognition*, 2003.

[38] I. Guyon. Handwriting synthesis from handwritten glyphs. In *5th Int. Workshop on Frontiers in Handwriting Recognition*, pages 309–312, 1996.

[39] I. Guyon, R. Haralick, J. Hull, and I. Phillips. Data sets for OCR and document image understanding research. In *[8]*, pages 780–799.

[40] I. Guyon, L. Schomaker, R. Plamondon, M. Liberman, and S. Janet. Unipen project of on-line data exchange and benchmarks. In *12th Int. Conference on Pattern Recognition*, pages 29–33, 1994.

[41] T. Ha and H. Bunke. Image processing methods for document image analysis. In *[8]*, pages 1–47.

[42] T. Ha and H. Bunke. Off-line handwritten numeral recognition by perturbation method. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(5):535–539, 1997.

[43] T. Ha, M. Zimmermann, and H. Bunke. Off-line handwritten numeral string recognition by combining segmentation-based and segmentation-free methods. *Pattern Recognition*, 31(3):257–272, 1998.

[44] M. Helmers and H. Bunke. Generation and use of synthetic training data in cursive handwriting recognition. In *Proc. 1st Iberian Conf. on Pattern Recognition and Image Analysis*, 2003.

[45] T. Ho. The random subspace method for constructing decision forests. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20:832–844, 1998.

[46] T. Ho, J. Hull, and S. Srihari. Decision combination in multiple classifier systems. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16:66–75, 1994.

[47] T. Hong and J. Hull. Text recognition enhancement with a probabilistic lattice chart parser. In *2nd Int. Conference on Document Analysis and Recognition*, pages 222–225, 1993.

[48] T. Huang and C. Suen. Combination of multiple experts for the recognition of unconstrained handwritten numerals. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17:90–94, 1995.

[49] T. Huang and C. Suen. Off-line signature verification using structural feature correspondence. *Pattern Recognition*, 35:2467–2477, 2002.

[50] R. Huber and A. Headrick. Handwriting identification: Facts and fundamentals. *CRC Press*, 1999.

[51] J. Hull. A hidden Markov model for language syntax in text recognition. In *11th Proc. on Int. Conference on Pattern Recognition*, volume B, pages 124–127, 1992.

[52] J. Hull. A database for handwritten text recognition research. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(5):550–554, May 1994.

[53] S. Impedovo, P. Wang, and H. Bunke, editors. *Automatic Bankcheck Processing*. World Scientific, 1997.

[54] A. Jain, L. Hong, and S. Pankanti. Biometric identification. *CACM*, 43:91–98, 2002.

[55] F. Jelinek. *Statistical Aspects of Speech Recognition.* MIT Press, 1998.

[56] S. Johansson, G. Leech, and H. Goodluck. *Manual of Information to accompany the Lancaster-Oslo/Bergen Corpus of British English, for use with digital Computers*. Department of English, University of Oslo, Oslo, 1978.

[57] A. Kaltenmeier, T. Caesar, J. Gloger, and E. Mandler. Sophisticated topology of hidden markov models for cursive script recognition. In *2nd Int. Conference on Document Analysis and Recognition*, pages 139–142, 1993.

[58] Y. Kato and M. Yasuhara. Recovery of drawing order from single stroke handwriting images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, (22):938–949, 2000.

[59] G. Kaufmann and H. Bunke. Automated reading of check amounts. *Pattern Analysis and Applications*, 3:132–141, 2000.

[60] E. Kavallieratou, N. Fakotakis, and G. Kokkinakis. An unconstrained handwriting recogntion system. *Int. Journal on Document Analysis and Recognition*, 4:226–242, 2002.

[61] G. Kim, V. Govindaraju, and S. Srihari. An architecture for handwritten text recognition systems. *Int. Journal on Document Analysis and Recognition*, 2(1):37–44, July 1999.

[62] J. Kittler, R. Duin, and M. Hatef. On combining classifiers. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20:226–239, 1998.

[63] J. Kittler and F. Roli. *Multiple Classifier Systems.* LNCS 2096, Springer, 2001.

[64] S. Knerr, E. Augustin, O. Baret, and D. Price. Hidden Markov model based word recognition and its application to legal amount reading from checks. *Computer Vision and Image Understanding*, 70:404–419, 1998.

[65] A. L. Koerich, Y. Leydier, R. Sabourin, and C. Y. Suen. A hybrid large vocabulary handwritten word recognition system using neural networks and hidden Markov models. In *8th Int. Workshop on Frontiers in Handwriting Recognition*, pages 99–104, 2002.

[66] U. Kressel and J. Schürmann. Pattern classification techniques based on function approximation. In *[8]*, pages 49–78.

[67] A. Kundu. Handwritten word recognition using hidden Markov model. In *[8]*, pages 157–182.

[68] A. Kundu, Y. He, and M.-Y. Chen. Alternatives to variable duration HMM in handwriting recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20:1275–1280, 1998.

[69] B. Lazzerini, F. Marcelloni, and L. Reyneri. Beatrix: A self-learning system for off-line recognition of handwritten texts. *Pattern Recognition Letters*, 18(6):583–594, June 1997.

[70] F. Leclerc and R. Plamondon. Automatic signature verification: The state of the art 1989-1993. In R. Plamondon, editor, *Progress in Automatic Signature Verification*, pages 13–19. World Scientific, 1994.

[71] C.-L. Liu, H. Sako, and H. Fujisawa. Integrated segmentation and recognition of handwritten numerals: Comparison of classification algorithms. In *8th Int.*

Workshop on Frontiers in Handwriting Recognition, pages 369–374, 2002.

[72] C.-L. Liu, H. Sako, and H. Fujisawa. Performance evaluation of pattern classifiers for handwritten character recognition. *Int. Journal on Document Analysis and Recognition*, 4:191–204, 2002.

[73] J. Liu and P. Gader. Neural networks with enhanced outlier rejection ability for off-line handwritten word recognition. *Pattern Recognition*, 35:2061–2071, 2002.

[74] G. Lorette. Handwriting recognition or reading? what is the situation at the dawn of the 3rd millenium? *Int. Journal on Document Analysis and Recognition*, 2:2–12, 1999.

[75] S. Lucas and A. Amiri. Recognition of chain-coded handwritten character images with scanning n-tuple method. *Electronic Letters*, 31:2088–2089, 1995.

[76] S. Madhvanath and V. Govindaraju. Local reference lines for handwritten phrase recognition. *Pattern Recognition*, 32(12):2021–2028, 1999.

[77] S. Madhvanath and V. Govindaraju. Local reference lines for handwritten phrase recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, (23):149–164, 2001.

[78] V. Maergner and M. Pechwitz. Synthetic data for arabic OCR system development. In *6th Int. Conference on Document Analysis and Recognition*, pages 1159–1163, 2001.

[79] U.-V. Marti and H. Bunke. Use of positional information in sequence alignment for multiple classifier combination. In *[63]*, pages 388–398.

[80] U.-V. Marti and H. Bunke. On the influence of vocabulary size and language models in unconstrained handwritten text recognition. In *6th Int. Conference on Document Analysis and Recognition*, pages 260–265, 2001.

[81] U.-V. Marti and H. Bunke. Text line segmentation and word recognition in a system for general writer independent handwriting recognition. In *6th Int. Conference on Document Analysis and Recognition*, pages 159–163, 2001.

[82] U.-V. Marti and H. Bunke. Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 15:65–90, 2001.

[83] U.-V. Marti and H. Bunke. The IAM-database: an English sentence database for off-line handwriting recognition. *Int. Journal on Document Analysis and Recognition*, 5:39–46, 2002.

[84] U.-V. Marti, R. Messerli, and H. Bunke. Writer identification using text line based features. In *6th Int. Conference on Document Analysis and Recognition*, pages 101–105, 2001.

[85] M. Mori, A. Suzuki, A. Siho, and S. Ohtsuka. Generating new samples from handwritten numerals based on point correspondence. In *7th Int. Workshop on Frontiers in Handwriting Recognition*, pages 281–290, 2000.

[86] S. Mori, H. Nishida, and H. Yamada. *Optical Character Recognition*. John Wiley and Sons, Inc., 1999.

[87] M. Morita, J. Facon, F. Bortolozzi, S. Garnes, and R. Sabourin. Mathematical morphology and weighted least squares to correct handwriting baseline skew. In *5th Int. Conference on Document Analysis and Recognition*, pages 430–433, 1999.

[88] N. Nishimura, K. Kobayashi, M. Maruyama, and Y. Nakano. Off-line character recognition using HMM by multiple directional feature extraction and voting with bagging algorithm. In *5th Int. Conference on Document Analysis and Recognition*, pages 49–52, 1999.

[89] L. O'Gorman and R. Kasturi, editors. *Document Image Analysis*. IEEE Computer Society Press, 1995.

[90] H.-S. Park, B.-K. Sui, J. Moon, and S.-W. Lee. A 2-D HMM method for offline handwritten character recognition. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 15:91–105, 2001.

[91] R. Plamondon and W. Guerfali. The generation of handwriting with delta-lognormal synergies. *Biological Cybernetics*, 78:119–132, 1998.

[92] R. Plamondon and C. Privitera. The segmentation of cursive handwriting: An approach based on off-line recovery of the motor-temporal information. *IEEE Transaction on Image Processing*, 8:80–91, 1999.

[93] R. Plamondon and S. Srihari. On-line and off-line handwriting recognition: A comprehensive survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22:63–84, 2000.

[94] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, Feb. 1989.

[95] A. Rahmann, H. Alam, and M. Fairhurst. Multiple classifier combination for character recognition: Revisiting the majority voting system and its variations. In D. Lopresti, J. Hu, and R. Kashi, editors, *5th Int. Workshop on Document Analysis Systems*, pages 167–178. LNCS 2423, Springer, 2002.

[96] A. Rahmann and M. Fairhurst. Multiple expert classification: A new methodology for parallel decision fusion. *Int. Journal on Document Analysis and Recognition*, pages 40–55, 2000.

[97] G. Rigoll. Combination of hidden Markov models and neural networks for hybrid statistical pattern recognition. In H. Bunke and A. Kandel, editors, *Hybrid Methods in Pattern Recognition*, pages 113–143. World Scientific, 2002.

[98] R. Rosenfeld. Two decades of statistical language modeling: Where do we go from here? *Proc. of the IEEE*, 88:1270–1278, 2000.

[99] H. Rowly, M. Goyal, and J. Bennet. The effect of large training set sizes on online Japanese Kanji and English cursive recognizers. In *8th Int. Workshop on Frontiers in Handwriting Recognition*, pages 36–40, 2002.

[100] H. Said, G. Peake, T. Tan, and K. Baker. Personal identification based on handwriting. *Pattern Recognition, Pergamon Press*, 33:149–160, 2000.

[101] T. Saito, H. Yamada, and K. Yamamoto. On the data base ETL 9 of handprinted characters in JIS chinese characters and its analysis. *IEICE Transactions*, J68-D(4):757–764, 1985.
[102] G. Saon. Cursive word recognition using a random field based hidden Markov model. *Int. Journal on Document Analysis and Recognition*, 1:199–208, 1999.
[103] K. M. Sayre. Machine recognition of handwritten words: A project report. *Pattern Recognition*, 5(3):213–228, 1973.
[104] J. Schürmann. *Pattern Classification - A Unified View of Statistical and Nerual Approaches*. John Wiley and Sons, Inc., 1996.
[105] A. Senior and A. Robinson. An off-line cursive handwriting recognition system. *IEEE Trans. on Patten Analysis and Machine Intelligence*, 20(3):309–321, Mar. 1998.
[106] S. Setlur and V. Govindaraju. Generating manifold samples from a handwritten word. *Pattern Recognition Letters*, 15:901–905, 1994.
[107] J.-C. Simon. Off-line cursive word recognition. *Proc. of the IEEE*, 80(7):1150–1161, July 1992.
[108] K. Sirlantzkis, M. Fairhurst, and M. Hoque. Genetic algorithm for multiple classifier configuration: A case study in character recognition. In *[63]*, pages 99–108.
[109] S. Smith. Handwritten character classification using nearest neighbour in large databases. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16:915–919, 1994.
[110] M. Sridar and F. Kimura. Segmentation-based cursive handwriting recognition. In *[8]*, pages 123–156.
[111] S. Srihari. Handwritten address interpretation: a task of many pattern recognition problems. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 14:663–674, 2000.
[112] T. Steinherz, E. Rivlin, and N. Intrator. Off-line cursive word recognition - a survey. *Int. Journal on Document Analysis and Recognition*, 2:90–110, 1999.
[113] C. Suen, S. Mori, S. Kim, and C. Leung. Analysis and recognition of Asian scripts - the state of the art. In *7th Int. Conference on Document Analysis and Recognition*, 2003.
[114] C. Suen, C. Nadal, R. Legault, T. Mai, and L. Lam. Recognition of handwritten numerals based on the concept of multiple experts. In *1st Int. Workshop on Frontiers in Handwriting Recognition*, pages 131–144, 1990.
[115] C. Suen, C. Nadal, R. Legault, T. Mai, and L. Lam. Computer recognition of unconstrained handwritten numerals. *Proc. of the IEEE*, 7(80):1162–1180, 1992.
[116] C. Suen and P. Wang, editors. *Thinning Methodologies for Pattern Recognition*. World Scientific, 1994.
[117] C. Tomai, B. Zhang, and V. Govindaraju. Transcript mapping for historic handwritten document images. In *8th Int. Workshop on Frontiers in Handwriting Recognition*, pages 413–418, 2002.
[118] T. Varga and H. Bunke. Generation of synthetic training data for an HMM-based handwriting recognition system. In *7th Int. Conference on Document Analysis and Recognition*, 2003.
[119] C. Viard-Gaudin, P. Lallican, S. Knerr, and P. Binter. The IRESTE on/off (ironoff dual handwriting database). In *5th Int. Conference on Document Analysis and Recognition*, pages 455–458, 1999.
[120] A. Vinciarelli. A survey on off-line cursive word recognition. *Pattern Recognition, Pergamon Press*, 35:1433–1446, 2002.
[121] A. Vinciarelli and S. Bengio. Writer adaptation techniques in HMM based off-line cursive script recognition. *Pattern Recognition Letters*, 23:905–916, 2002.
[122] A. Vinciarelli and J. Luettin. Off-line cursive script recognition based on continuous density HMM. In *7th Int. Workshop on Frontiers in Handwriting Recognition*, pages 493–498, 2000.
[123] F. Wang, . Vuurpijl, and L. Schomaker. Support vector machines for the classification of western handwritten capitals. In *7th Int. Workshop on Frontiers in Handwriting Recognition*, pages 167–176, 2000.
[124] J. Wang, C. Wu, Y.-Q. Xu, H.-Y. Shum, and L. Ji. Learning-based cursive handwriting synthesis. In *8th Int. Workshop on Frontiers in Handwriting Recognition*, pages 157–162, 2002.
[125] W. Wang, A. Brakensiek, and G. Rigoll. Combination of multiple classifiers for handwritten word recognition. In *8th Int. Workshop on Frontiers in Handwriting Recognition*, pages 117–122, 2002.
[126] R. Wilkinson, J. Geist, S. Janet, P. Grother, C. Burges, R. Creecy, B. Hammond, J. Hull, N. Larsen, T. Vogl, and C. Wilson. The first census optical character recognition systems conf. #NISTIR 4912, The U.S. Bureau of Census and the National Institute of Standards and Technology, Gaithersburg, MD, 1992.
[127] A. E. Yacoubi, M. Gilloux, R. Sabourin, and C. Suen. An HMM-based approach for off-line unconstrained handwritten word modeling and recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21:752–760, 1999.
[128] X. Ye, M. Cheriet, and C. Suen. A framework for combining numeric string recognizers. In *6th Int. Conference on Document Analysis and Recognition*, pages 716–720, 2001.
[129] S. Young, J. Jansen, J. Odell, D. Ollason, and P. Woodland, editors. *The HTK Book*. Entropic, 1999.
[130] M. Zimmermann and H. Bunke. Automatic segmentation of the IAM off-line handwritten English text database. In *16th Int. Conference on Pattern Recognition*, volume 4, pages 35–39, 2002.
[131] M. Zimmermann and H. Bunke. Parsing n-best lists of handwritten sentences. In *7th Int. Conference on Document Analysis and Recognition*, 2003.