

Markov models for offline handwriting recognition: a survey

Thomas Plötz · Gernot A. Fink

Received: 16 February 2009 / Revised: 26 June 2009 / Accepted: 31 August 2009 / Published online: 31 October 2009
© The Author(s) 2009. This article is published with open access at Springerlink.com

Abstract Since their first inception more than half a century ago, automatic reading systems have evolved substantially, thereby showing impressive performance on machine-printed text. The recognition of handwriting can, however, still be considered an open research problem due to its substantial variation in appearance. With the introduction of Markovian models to the field, a promising modeling and recognition paradigm was established for automatic offline handwriting recognition. However, so far, no standard procedures for building Markov-model-based recognizers could be established though trends toward unified approaches can be identified. It is therefore the goal of this survey to provide a comprehensive overview of the application of Markov models in the research field of offline handwriting recognition, covering both the widely used hidden Markov models and the less complex Markov-chain or n -gram models. First, we will introduce the typical architecture of a Markov-model-based offline handwriting recognition system and make the reader familiar with the essential theoretical concepts behind Markovian models. Then, we will give a thorough review of the solutions proposed in the literature for the open problems how to apply Markov-model-based approaches to automatic offline handwriting recognition.

Keywords Offline handwriting recognition · Hidden Markov models · n -Gram language models

1 Introduction

A major goal of pattern recognition research is to recreate human perception capabilities in artificial systems. As a special aspect of visual perception, the ability to read machine-printed or handwritten text is one such remarkable ability of humans that is—even today—hardly matched by machine intelligence. Since the very first efforts to achieve optical character recognition (OCR), i.e., to automatically read machine-printed texts, the research field dealing with artificial reading systems has undergone significant changes in methodology and made substantial progress toward its ultimate goal.

For example, the problem of reading machine-printed addresses in a mail-sorting machine—especially with the impressive speed of the commercial systems available—can be considered solved. The availability of more general commercial solutions for OCR demonstrates that the technology is quite mature in this field already. However, as soon as the variability in the script to be read becomes more prominent, as it is the case for degraded documents or—even more severely—for handwritten text, current technology reaches its limits. Consequently, the task of automatically reading handwriting with close-to-human performance is still an open problem and the central issue of an active field of research.

In almost all endeavors to build artificial perception systems, research focuses on methods that automatically learn from sample data. For learning models of sequential data—as text can be considered to be with some approximation—approaches based on Markovian models proved very successful, especially in the field of automatic speech recognition. Today, systems based on Markov models (MM) are also successfully used for automatic handwriting recognition (HWR). Since their first introduction into the field almost

T. Plötz (✉) · G. A. Fink
Intelligent Systems Group, Robotics Research Institute,
Technische Universität Dortmund, Otto-Hahn-Strasse 8,
44227 Dortmund, Germany
e-mail: Thomas.Ploetz@udo.edu

G. A. Fink
e-mail: Gernot.Fink@udo.edu

two decades ago, considerable progress has been made in adapting MM-based techniques to this new domain. However, in contrast to the field of automatic speech recognition, where quasi-standard procedures are established, researchers are still exploring a wide range of possibilities in applying MM-based methods to the challenging problem of reading handwritten text.

1.1 Scope

The goal of this survey is to provide a comprehensive overview of the application of Markov models in the research field of offline handwriting recognition. The term “Markov model” here refers to both the widely used hidden Markov models (HMMs) and the less complex Markov-chain models. To some limited extent general foundations of automatic HWR not explicitly related to the application of Markov models will be discussed here as well. However, for more detailed treatments of this general state-of-the-art, the reader is referred to the broader surveys, which have been published in recent years (e.g., [2, 22, 54, 99, 115]).

Techniques for automatic handwriting recognition can be distinguished as being either online or offline, depending on the particular processing strategy applied. Online recognition is performed as the text to be recognized is written. Therefore, the process of handwriting has to be captured online, e.g., using some pressure sensitive devices. They provide rich sequences of sensor data including geometrical positions of the stylus as well as temporal information about the writing process, which is *the* big advantage of online approaches. In contrast, offline recognition is performed after the text has been written. For this purpose, images of the handwriting are processed, which are captured, e.g., using a scanner or a camera.

It is commonly agreed that online handwriting recognition corresponds to the easier problem. Consequently, at least for certain application domains like pen-based input interfaces substantial progress has been achieved. In fact, commercially available products suggest that the problem of online HWR can be considered as being close to solved (cf., e.g., [98]). Various groups developing online recognizers have now already moved toward much more complicated tasks like, e.g., online sketch recognition (cf., e.g., [34] for a very impressive physics simulator). This paper emphasizes approaches addressing the challenging task of offline handwriting recognition. In order to keep the argumentation as focussed as possible, we concentrate on the most widely used variants of Markov modeling. Thus, modeling approaches that currently are rather rarely used for handwriting recognition like Markov random fields (MRF, cf., e.g., [25]) or conditional random field models (CRF, cf., e.g., [46]) are not covered by this article.

Although offline HWR shows parallels to classical optical character recognition (OCR), i.e., the analysis of machine printed text, the scope of this paper is limited to handwriting recognition. If not absolutely necessary, we do not cover specialties of OCR applications. The recognition of handwriting data is addressed, which exhibits *unconstrained* writing style in mainly Roman or Arabic scripts.¹ More precisely, the recognition of non-alphabetic scripts (like Kanji) is not covered by this survey. Reasonable exceptions, where certain substantial similarities to alphabetic scripts recognition exist, will be discussed though.

1.2 History

The application of Markov models has a fairly long history in various domains. At the beginning of the past century, the Russian mathematician Andrej Andrejewitsch Markov first applied such a type of statistical model for the analysis of character sequences [83]. Honoring his fundamental developments, statistical models that share the same basic properties were named after Markov.

The basic technology of hidden Markov models—the most prominent variant of Markov models—was originally used for speech recognition applications. Fundamental developments in the late 1960s started to allow the robust and efficient automatic analysis of time-dependent signal data for real life scenarios (cf., e.g., [5, 6, 120]). After substantial research efforts, Markovian models can today be considered the state-of-the-art technology in the area of automatic speech recognition (cf., e.g., [128]). In the last 20+ years, this concept has also been transferred to the domain of handwriting recognition—initially, however, with limited success only (cf., e.g., [69]). According to the literature in the early and mid-1990s, research activities in the area of Markov-model-based handwriting recognition increased substantially. Apparent reasons for this were the grown commercial interest in the topic and that, probably due to this reason, more researchers—often from the speech recognition domain—turned toward the field. Consequently, the number of publications increased significantly (cf., e.g., [23, 65, 107]). Today, the field can be considered being mature and powerful recognition systems exist (see Sect. 5).

Markov models are used for both online and offline handwriting recognition. Especially relevant for the latter, the application of the sliding window principle can be understood as an important ‘milestone’ for successful Markov-model-based handwriting recognition [65, 105]. Its application allows for the effective linearization of

¹ Interestingly, techniques for the recognition of Latin and Arabic script are more similar than one might expect when only visually comparing documents written in, e.g., English or an Arabic language [104].

handwriting data, which can be considered as *the* pre requisite for the successful application of Markov models for offline recognition.

During the last few years, research w.r.t. Markov-model-based handwriting recognition has made substantial progress. Comparable to their application in the speech recognition domain, certain procedures have been established now serving as quasi-standard for offline HWR-based on Markov models. Just to mention two examples practically all current recognizers are derived by applying (variants of) Baum–Welch training on sample data (see Sect. 3.3.1 for details) resulting in most cases in (semi-) continuous output models. They are usually based on Gaussian mixture densities. Furthermore, and again similar to speech processing, the integration of Markov-chain models into HWR systems is very promising. According to recent publications, combined recognizers integrating both HMMs as script appearance models and statistical n -grams as language models represent one of the latest trends in handwriting recognition (cf., e.g., [99, 117]).

1.3 Applications

Markov-model-based approaches are today widely used in the application field of automatic offline HWR. In addition to their use in academic research, they also play an important role in commercial applications in industrial contexts.

Although certain publications regarding Markov-model-based recognition of isolated characters exist (cf., e.g., [21, 56, 57, 95]), it is at least questionable whether the use of these models is appropriate for such data. Instead, the approach shows its strength especially for sequences. Thus, at least words should be addressed to benefit from the properties of Markov models. The actual recognition is either performed for isolated words or for connected words. The latter is the more complicated but the more realistic use case since, e.g., full sentences can be treated without relying on prior—successful—segmentation. Consequently, actual document analysis based on text recognition becomes possible.

Certainly one of the most important applications is address reading, e.g., for mail sorting. In recent years, tremendous efforts have been directed toward this issue (cf., e.g., [17, 68, 97, 114]). It has resulted in powerful recognition systems, which are successfully applied by major postal service companies (cf., e.g., [54, 90, 103]).

Another important application field of Markov-model-based handwriting recognition is the automatic processing of bank cheques and official forms as regularly considered by insurance companies, banks, governmental organizations, etc. Various recognizers for different languages have been developed and are applied (cf., e.g., [91, 126]).

In addition to the aforementioned “killer applications” of Markov models for automatic HWR, numerous further application fields exist. Just to mention some examples, digit recognition (with the special case of analyzing touching digits) and offline signature verification play important roles for legal issues (cf., e.g., [31, 30, 64]). Markov models have also been used for explicit segmentation of handwritten texts at word level [86, 129]. Furthermore, a special case of text recognition recently gained importance: whiteboard reading. By means of either a special infrared / ultrasonic tracking device for online recognition or a video camera for offline recording, images from a whiteboard containing handwriting data are captured. Markov models are used in both cases for text recognition (cf., e.g., [74, 124]).

1.4 Structure

The remainder of this article is organized as follows. In the following section, we will first give a qualitative overview of the architecture of a typical HWR system based on Markov models. Key references will be given for only those aspects that are not integral parts of an MM-based system and will be treated in more detail in subsequent sections. The concepts and algorithms behind the MM-based recognition paradigm will then be presented in Sect. 3. Subsequently, in Sect. 4, we will review the different methods applied and solutions proposed for solving the key problems in MM-based HWR systems. Markov-model-based handwriting recognition has already become a mature research field with important applications in both academic and industrial context. For these purposes, integrated recognition systems have been and are still being developed. Section 5 provides an overview of such major recognition systems including their particular key features.

Finally, in Sect. 6, we will give a concluding discussion on the state-of-the-art of Markov-model-based handwriting recognition. We will present the latest technological trends, some remarks on benchmarking and reporting results, and a short discussion of future challenges we identified for the field of MM-based HWR.

2 General architecture

Any handwriting recognition system is usually embedded into a larger document analysis framework, e.g., a mail-sorting machine. Necessarily, as a first step of document understanding a digital image of the document to be analyzed needs to be captured by some device, e.g., a scanner or a camera. As, in general, the captured image may show, besides the desired text, other document structures (e.g., tables, figures, or images) or even non-document parts of the scene background, the relevant document elements (e.g., text lines or

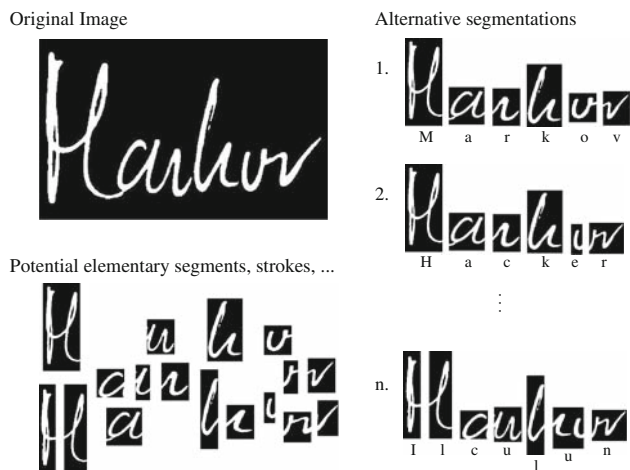


Fig. 1 Example of classical HWR approach relying on explicit segmentation and subsequent classification (inspired by [55])

paragraphs) need to be segmented² from the captured input image. As soon as regions of handwritten text have been identified in the input image, these can be further segmented into text-line or even word images. These are then subject to a number of preprocessing steps that aim at reducing the variability in the appearance of the writing by applying a sequence of normalization operations.

A classical HWR system would then proceed by attempting to segment the normalized text into candidates of primitives (e.g., strokes or characters, see Fig. 1). The potential primitives would then be classified, and the best segmentation result would be selected—usually according to some heuristic. In contrast to this approach, MM-based HWR systems can avoid carrying out segmentation³ at the level of character sequences and subsequent classification separately. The only requirement for their application is that the data to be processed must be representable as a sequence of items *without* making decisions about potential segmentations.

Therefore, after normalization text-line images are converted to a sequential representation and local features are computed. The feature vector sequences obtained are then fed to an MM-based decoder that produces a hypothesis for the segmentation *and* classification of the analyzed portion

² Especially, when processing machine-printed documents where it is usually clear that the document image only shows the document to be analyzed, this initial segmentation of relevant document structures is referred to as layout analysis (cf., e.g., [80]).

³ As any document analysis system needs to extract relevant textual items, e.g., words or lines, from the document image prior to recognition, several preprocessing steps are necessary. These perform tasks which can also be termed “segmentation.” Though there are first approaches to perform, e.g., line separations using HMMs [78], in this respect, traditional and MM-based systems are still quite similar. Therefore, in this article, we focus on the segmentation at the level of character or word sequences where MM-based approaches can show their strengths.

of handwritten text—usually, as a sequence of word or character hypotheses.

The complete pipeline of processing steps applied in a state-of-the-art HWR system based on Markov models is shown in Fig. 2. It largely follows a standard pattern recognition approach comprising preprocessing, feature extraction, and classification (cf., e.g., [39]). What is special for MM-based HWR systems is that a serialization of the patterns to be recognized is carried out prior to feature extraction and that segmentation and classification are achieved in an integrated manner by the joint decoding of HMMs and n -gram models.

2.1 Text-line extraction

In most applications of HWR, it is ensured that the document-capturing process delivers an image of the desired document only. Within this document image, paragraphs of text need to be localized and lines of text need to be extracted. The methods applied usually rely on the assumption that handwritten text is oriented approximately horizontally and is organized in line structures. Individual lines of text can then be extracted by, for example, analyzing horizontal projection histograms or by applying probabilistic segmentation techniques (cf. [71]). A good overview of algorithms for text-line extraction is given in [72].

2.2 Preprocessing

Images of handwritten text lines or words usually vary with respect to baseline orientation (frequently also referred to as skew), slant angle, and size of the handwriting. Therefore, almost all HWR systems apply preprocessing operations that attempt to normalize the appearance of the writing with respect to these three aspects (cf., e.g., [53]). Usually, in some phase of preprocessing, the word or text line images are binarized separating dark ink pixels from the document’s background (cf., e.g., [113]).

To compensate for an unknown baseline orientation or skew angle, either the baseline orientation itself or a suitable approximation thereof is usually computed and subsequently compensated by appropriately rotating the line image. The methods applied are somewhat script dependent. For example, computing a baseline estimate by interpolating local contour minima (cf. [13]) works well for Roman script, but will be likely to fail for Arabic. More widely applicable is the implicit estimation of the baseline orientation by maximizing the entropy of the horizontal projection histogram (cf., e.g., [119]).

The slant angle, i.e., the tilt of individual handwritten characters with respect to the vertical, is compensated by applying a shear transform to the text-line image. The crucial part here is the reliable estimation of the slant angle. Many

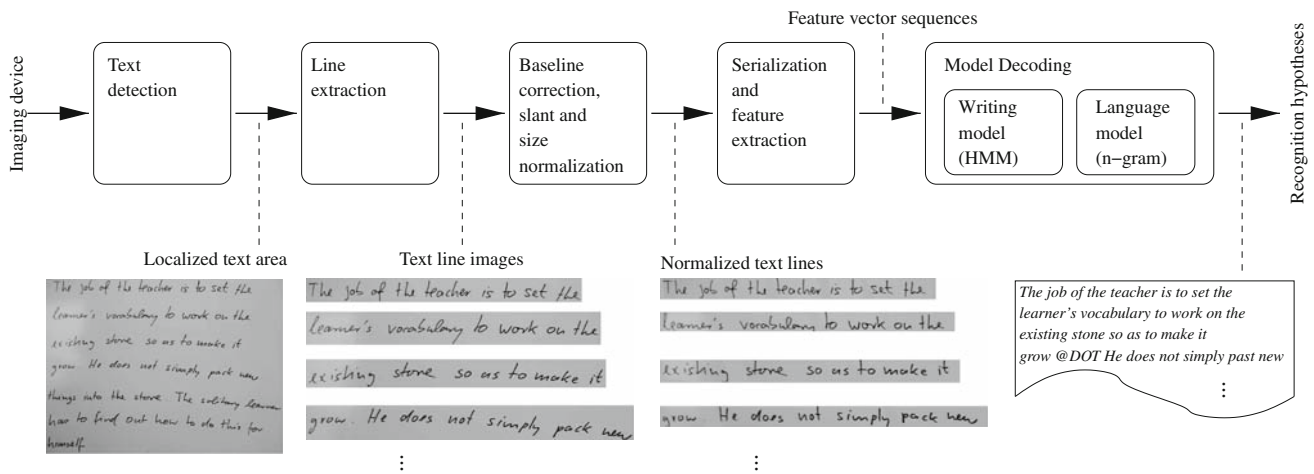


Fig. 2 Schematic overview of a typical architecture for an HMM-based handwriting recognition system. Example whiteboard data, preprocessing, and recognition results shown are taken from [125]

successful methods rely on gradient information extracted from the text-line image (cf., e.g., [9,38]), which is often binarized in advance.

No generally accepted methodology exists so far for the normalization of the apparent size of handwriting in image data. HWR systems that compute the baseline and other writing lines often rely on an estimate of the so-called core size, i.e., the size of lower-case letters in Roman script (cf., e.g., [26,84]). However, as size normalization is quite crucial, any error in estimating the writing lines will lead to an inappropriate size normalization and consequently to the almost inevitable failure of the subsequent recognition process. A quite robust method for normalizing the size of Roman script was proposed in [79] which uses an estimate of average character width.

2.3 Serialization and feature extraction

MM-based recognizers require the data to be analyzed to be sequentially or temporally ordered. Therefore, a suitable technique for converting two-dimensional images of handwritten words or text lines to a sequential representation is an integral part of any MM-based offline HWR system. The most prominent and widely used method for serialization of text-line images—the so-called sliding-window approach—was first and independently proposed by researchers at Daimler-Benz research center for offline handwriting recognition [65] and at BBN for offline recognition of machine-printed text [105]. It consists of sliding a small analysis window, which is usually only a few pixels wide (i.e., much narrower than a character image), along the text-line image, i.e., in the direction of the writing. Thus, small vertical stripes are extracted from the text-line image, which usually overlap to some degree (see Fig. 3). This sequence of image stripes forms the basis of the subsequent feature extraction.

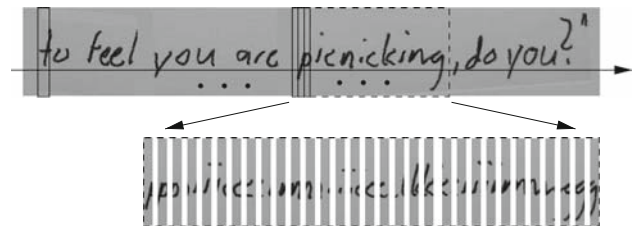


Fig. 3 Example of the sliding-window approach: Above, the text line to be analyzed is shown with the implicit sequencing direction assumed and some of the overlapping analysis windows superimposed. Below, the extracted image stripes are shown for part of the text line

Feature extraction methods building on the sliding-window analysis-framework usually compute some local characteristics of the extracted image stripes as, e.g., image moments or—after binarization of the image data—simple geometrical properties. Taken together, the sliding-window technique and the feature extraction applied to each window compute a local feature representation of the text-line image to be analyzed. The procedure can principally be compared to the short-time analysis and feature computation as applied for the purpose of speech recognition, though it lacks a clear justification from a signal-theoretic point of view.

2.4 Modeling and decoding

For reasons that will be detailed in the following section, recognition systems based on Markov models use two distinct modeling components. The appearance of the handwriting as defined by its local characteristics captured in the feature representation is described by hidden Markov models. We will refer to this modeling component as the *writing model*. The second modeling component describes long term sequencing restrictions within the data, i.e., on the level of

word or character sequences. This so-called language model is usually realized by Markov chain models.

The writing model usually defines basic models for elementary units like, e.g., characters. By simple combination operations, more complex model structures for, e.g., arbitrary sequences of words from a given lexicon, can be constructed from these basic entities quite easily. In such a complex model, the language model component provides a probabilistic model for long-term dependencies.

In combination, the writing and the language model form a powerful statistical description of handwriting. The parameters of the models can be estimated automatically on sample data. The so-called decoding of the integrated model—i.e., the search for the optimal path through the combined state space—provides the optimal segmentation *and* classification of the data in an integrated framework.

3 Markov model concepts: the essence

For the analysis of sequential data, the use of hidden Markov models (HMMs) as statistical models can be considered the state-of-the-art. In combination with Markov-chain models that describe restrictions of possible hypotheses sequences, powerful classification systems can be realized. For online recognition, handwriting data is of sequential type by its nature since it is recorded as the text is written and, thus, corresponds to time series. Images of handwriting, as processed by offline recognizers, can be transferred into a sequential representation by moving a sliding window along the particular text lines (cf. Sect. 2.3). Thus, Markov models is a perfect fit for handwriting recognition.

The following summary of the theoretical concepts behind Markov models is mainly adopted from the argumentation in [51]. An abridged, tutorial-style version thereof with focus on handwriting recognition can be found in [50]. The interested reader will find an in-depth treatment of MM-based pattern recognition methods in [47].

3.1 Recognition paradigm

Markov-model-based recognition approaches are, generally, based on the assumption of a statistical model for the generation of the data to be analyzed. A sequence of symbols \mathbf{w} —characters or words—generated by some source, i.e., the writing process, is coded into a signal representation (for offline HWR, this means images of handwritten text) and later observed as a sequence of feature vectors \mathbf{X} . Formally, the goal of the recognition process is then to find the sequence $\hat{\mathbf{w}}$ that maximizes the posterior probability $P(\mathbf{w}|\mathbf{X})$ of the symbol sequence given the data.

$$\begin{aligned}\hat{\mathbf{w}} &= \operatorname{argmax}_{\mathbf{w}} P(\mathbf{w}|\mathbf{X}) = \operatorname{argmax}_{\mathbf{w}} \frac{P(\mathbf{w})P(\mathbf{X}|\mathbf{w})}{P(\mathbf{X})} \\ &= \operatorname{argmax}_{\mathbf{w}} P(\mathbf{w})P(\mathbf{x}|\mathbf{w})\end{aligned}\quad (1)$$

When applying Bayes' rule $P(\mathbf{w}|\mathbf{X})$ can be rewritten into a form, where the two modeling components of typical Markov-model-based recognition systems become manifest. $P(\mathbf{w})$ denotes the language model probability for the sequence of symbols \mathbf{w} . Technically, stochastic n -gram models represent the usual realization of language models. $P(\mathbf{X}|\mathbf{w})$ represents the probability of observing the sequence of symbols as features \mathbf{X} according to the writing model, namely the HMM.

The fundamental advantage of Markov-model-based recognizers is that they do not require an explicit segmentation of the data prior to its classification. The recognition is thus performed in a *segmentation-free* manner, which means that segmentation and classification are integrated. Thus, the application of Markov models to presegmented input data, as described in some publications, appears to make sense only for very rare special cases, if any.

3.2 Hidden Markov models

3.2.1 Definition

Hidden Markov models describe a two-stage stochastic process with hidden states and observable outputs. The first stage represents a discrete stochastic process, which produces a series of random variables that take on values from a discrete set of states. This process is *stationary*, which means that its statistical properties do not change over time, and it is also *causal* and *simple*. The last two properties taken together restrict the dependency of the probability distributions of states generated by the random variables to be dependent on the immediate predecessor state only. The Markov process is then said to be of first order.

$$P(s_t|s_1, s_2, \dots, s_{t-1}) = P(s_t|s_{t-1})$$

Basically, this first stage represents a finite state automaton, which behaves probabilistically. In the second stage, then at every time t , an output O_t is generated depending on the current state s_t only:

$$P(O_t|O_1 \dots O_{t-1}, s_1 \dots s_t) = P(O_t|s_t)$$

Since only these outputs O_t , and not the associated internal states s_t , can be observed, the overall model is referred to as *hidden* Markov model.

In summary, a *first-order* hidden Markov model λ is formally defined as consisting of:

- a finite set of states $\{s | 1 \leq s \leq N\}$,
- a matrix of state transition probabilities⁴ $\mathbf{A} = \{a_{ij} | a_{ij} = P(s_t = j | s_{t-1} = i)\}$,
- a vector of start probabilities $\pi = \{\pi_i | \pi_i = P(s_1 = i)\}$, and
- state-specific output probability distributions $\{b_j(O_t) | b_j(O_t) = p(O_t | s_t = j)\}$ for discrete emissions or $\{b_j(\mathbf{x}) | b_j(\mathbf{x}) = p(\mathbf{x} | s_t = j)\}$ for continuous modeling, respectively (see below).

3.3 Modeling emissions

Depending on the type of input data, the output elements generated per state can be either symbolic—i.e., of discrete type—or continuous. The latter representation is better suited for handwriting recognition purposes, as usually real-valued vectors \mathbf{x} from some high-dimensional feature-space \mathbb{R}^N , which is derived from the original handwriting (image) data, are processed. Consequently, the probability distributions $p(\mathbf{x} | s_t = j)$ of the statistical outputs of the model need to be able to define continuous distributions over \mathbb{R}^N . Since no general parametric families of such distributions are known, in the continuous case, probability distributions are usually approximated via state-specific mixtures of Gaussians:

$$p(\mathbf{x} | s_t = j) \hat{=} \sum_{k=1}^{\infty} c_{jk} \mathcal{N}(\mathbf{x} | \mu_{jk}, \mathbf{C}_{jk}) \approx \sum_{k=1}^M c_{jk} \mathcal{N}(\mathbf{x} | \mu_{jk}, \mathbf{C}_{jk})$$

where $\mathcal{N}(\mathbf{x} | \mu_{jk}, \mathbf{C}_{jk})$ denotes a Gaussian normal distribution with mean vector μ_{jk} and covariance matrix \mathbf{C}_{jk} , and c_{jk} represents the prior probability of the k -th mixture.

As for continuous HMMs, the number of parameters is drastically increased with respect to the discrete case, several techniques were developed to reduce the number of parameters by jointly using parts of the model. Such methods are usually referred to as the *tying* of parameters. The most well-known of these approaches are the so-called *semicontinuous* HMMs—also frequently referred to as *tied-mixture models* [62, 63]. In such models, only a single set of component densities is used to construct all state-specific output probability densities:

$$b_j(\mathbf{x}) = \sum_{k=1}^M c_{jk} \mathcal{N}(\mathbf{x} | \mu_k, \mathbf{C}_k)$$

3.3.1 Algorithms

The attractiveness of HMMs is to a large extent justified by the fact that efficient algorithms for estimating the model parameters as well as for decoding the model on new data

⁴ For practical applications, the actual model topology—i.e., the connectivity between states of a certain model—is usually limited using specific, non-ergodic model architectures (e.g., linear or Bakis type).

exist. Decoding corresponds to the integrated segmentation and classification of the associated data.

A variant of the well-known expectation maximization (EM) technique [37], namely the so-called Baum–Welch algorithm is commonly used for training the model. The method applies an iterative growth transformation to the model parameters such that the generation probability of the data given the model is improved:

$$P(\mathbf{O} | \hat{\lambda}) \geq P(\mathbf{O} | \lambda)$$

Here, $\hat{\lambda}$ denotes the adapted HMM derived from the previous model λ by applying one re-estimation step to the parameters. Model training is iterated until convergence is reached, i.e., until $P(\mathbf{O} | \hat{\lambda}) - P(\mathbf{O} | \lambda) \leq \epsilon$ for some small threshold ϵ .

The basis of model decoding is formed by the so-called Viterbi algorithm, which is used to—in the statistical sense—“infer” the hidden state sequence \mathbf{s}^* that with maximum probability generates an available sequence of outputs given the model:

$$\mathbf{s}^* = \underset{\mathbf{s}}{\operatorname{argmax}} P(\mathbf{O}, \mathbf{s} | \lambda)$$

As states can be associated with basic segmentation units—for offline HWR usually this corresponds to characters—decoding yields the segmentation of the data considered on the basis of the current model.

3.3.2 Practical issues

The efficiency in both evaluating and decoding the model arises from the fact that HMMs store only one internal state as context for future actions, which is also called the Markov property. Therefore, computations necessary to obtain the production probability $P(\mathbf{O} | \lambda)$ and the optimal state sequence \mathbf{s}^* can be performed in a dynamic programming style with linear complexity in the length of the sequence considered and quadratic complexity in the number of model states. Still, the algorithms are usually not efficient enough in practice. Hence, especially for decoding model pruning strategies like the beam-search algorithm [77] are applied.

In almost all current implementations of HMM-based recognizers (negative), logarithmic representations of probabilities are used. Thus, products of probabilities are turned into sums. Due to the reduced dynamic range of these additive costs, computations involving very small probabilities become numerically feasible even if such quantities are accumulated within large model architectures or for extremely long observation sequences.

3.4 N-gram models

In addition to the analysis of local context within sequential data, which is covered by HMMs, it is desirable in many applications to be able to describe long-term dependencies within the statistical modeling framework. For HWR applications, restrictions concerning the potential co-occurrences of subsequent characters or words (depending on the modeling basis) cannot be captured reasonably using HMMs alone. This is where Markov-chain models come into play.

3.4.1 Definition

Markov-chain models can be used to statistically describe the probability of the occurrence of entire symbol sequences. Formally speaking (cf. Eq. 1) the probability $P(\mathbf{w})$ of a sequence of symbols $\mathbf{w} = w_1, w_2, \dots, w_T$ is calculated. In order to make things mathematically tractable, $P(\mathbf{w})$ is first factorized using Bayes' rule according to

$$P(\mathbf{w}) = P(w_1)P(w_2|w_1) \dots P(w_T|w_1, \dots, w_{T-1}) \\ = \prod_{t=1}^T P(w_t|w_1, \dots, w_{t-1})$$

Since the context dependency increases arbitrarily with the length of the symbol sequence, in practice the “history” of a certain symbol is limited:

$$P(\mathbf{w}) \approx \prod_{t=1}^T P(w_t | \underbrace{w_{t-n+1}, \dots, w_{t-1}}_{n \text{ symbols}})$$

This means that the probability of the complete sequence is defined on the basis of the conditional probabilities of some symbol—or word— w_t occurring in the context of its $n - 1$ predecessor words $w_{t-n+1}, \dots, w_{t-1}$. Markov-chain models are therefore often referred to as n -gram models.

3.4.2 Algorithms

For the evaluation of n -gram models on unknown data, usually the perplexity \mathcal{P}

$$\mathcal{P}(\mathbf{w}) = \frac{1}{\sqrt[|\mathbf{w}|]{P(\mathbf{w})}} = \frac{1}{\sqrt[T]{P(w_1, w_2, \dots, w_T)}} \\ = P(w_1, w_2, \dots, w_T)^{-\frac{1}{T}}$$

is exploited as the evaluation criterion. Formally, the perplexity of some unseen data \mathbf{w} is the cross-entropy between the symbol distribution defined by the probabilistic model and the one defined empirically by the data. The smaller the perplexity the better the n -gram model is able to predict the unseen data.

Parameter estimation for stochastic language models, i.e., training of n -gram models, is based on the determination of n -gram occurrences c in sample data. Conditional probabilities are calculated as the ratios of n -gram—or event—counts $c(w_1, w_2, \dots, w_n)$ and those obtained for the respective contexts $c(w_1, w_2, \dots, w_{n-1})$.

$$P(w_n|w_1, w_2, \dots, w_{n-1}) = \frac{c(w_1, w_2, \dots, w_n)}{c(w_1, w_2, \dots, w_{n-1})}$$

3.4.3 Practical issues

Even for moderate sizes of n (e.g., two for bi-gram models or three for tri-gram models), most n -gram events necessary for deriving robust statistical estimates will not be observed in a typical set of training data due to its limited size. If performing naive training as described in the previous section, conditional probabilities for events not observed in the training data (so-called unseen events) will erroneously be determined as being zero. However, zero-probabilities are only valid in very rare cases. Certain events being unseen generally needs to be attributed to the fact that too few samples are available for parameter estimation. Therefore, for robust estimation of n -gram models, it is of fundamental importance to appropriately smooth the raw probability estimates. Note that typically not some, but most n -gram counts will be zero.

Thus, in practical applications, n -gram counts are modified and some “probability mass” for unseen events is gathered, e.g., by certain discounting techniques. The resulting zero-probability is then redistributed to unseen events according to a more general distribution. Widely used examples of smoothing techniques are Backing-Off and Interpolation (cf., e.g., [27]).

3.5 Combination of writing and language model

As HMMs and n -gram models are quite similar to each other, they can be combined rather easily into an integrated model (cf. Eq. 1). However, as HMM and n -gram models usually describe the data on widely different levels of granularity—i.e., in units of words for the language model and in sub-character units for the writing model—the different scores need to be combined in a weighted manner:

$$P(\mathbf{w})^\rho P(\mathbf{X}|\mathbf{w})$$

The optimum weight ρ for a certain model configuration needs to be determined experimentally in practice. Sometimes, an additional bias term is also used to control the number of word or character hypotheses generated.

Furthermore, as n -gram models span considerably longer contexts than HMMs, the search procedures used for integrated model decoding also become more complex (cf., e.g., [47, Chap. 12]). Unfortunately, there is no formal way to

predict the performance improvement to be expected from the use of a language model. However, when comparing recognition results achieved by applying language models with different perplexities on the test data, the word error rates—according to a rule-of-thumb—will be roughly proportional to the square root of the perplexities. A substantial violation of that relation always indicates a problem with the integration of the language model evaluation into the overall decoding procedure.

4 Markov-model-based handwriting recognition

The attractiveness of Markov models for various pattern recognition applications is mainly reasoned by the clear and reliable statistical framework they are based on. Efficient algorithms for parameter estimation and model evaluation exist, which is an important prerequisite for their practical use in real-world applications.

The popularity of Markov models also for handwriting recognition is based on these very arguments. However, recognizers that can be applied *successfully* to real handwriting recognition tasks require substantially more know-how than the basic concepts as described in the previous section. In the following we will discuss respective practical issues including reviews of the particular state of the art as described in the literature.⁵

4.1 Segmentation-free versus segmentation-based recognition

The process of automatic handwriting recognition can be considered as a classical pattern classification task. Sensory data is automatically assigned to those pattern classes to which it most likely belongs. Thereby, the evaluation of stochastic models is performed on the level of distinguished basic modeling units from some limited inventory (words, characters, or graphemes). Because of its creation process, handwritten script corresponds to time-series data. Characters are written one after another, thereby exhibiting mutual dependencies and touching each other. Consequently, handwriting recognition especially needs to address segmentation issues.

The importance of the segmentation aspect is—independent of actually focusing on Markov-model-based handwriting recognition or on some other recognition approach—considered in the vast majority of related document analysis literature. Unfortunately, so far no consistent

terminology has been established that is commonly used by the community. Since segmentation can be considered at various levels of document analysis certain clarification is necessary to avoid misunderstandings. As an example, it needs to be clarified whether segmentation is considered at the document-, word-, or character-level—which does not always become clear in the particular argumentations.

In this respect, Arica and Yaman-Vural in [2] discuss a kind of taxonomy very thoroughly. Comparable arguments can, for example, also be found in [22, 108, 115], just to mention some examples. They discriminate between external and internal segmentation. The first is considered as being the most critical part of document analysis in general. By subdividing a document into text and non-text regions external segmentation is a necessary step prior to the offline HWR process. Markov models have meanwhile also been applied successfully for even more general document analysis tasks w.r.t. layout segmentation (cf., e.g., [78]). However, external segmentation is not directly related to the actual handwriting recognition process and will thus not be covered by this survey. As the basis for offline HWR systems, in this survey, we assume words or text lines (not characters) that have successfully been isolated by means of some appropriate layout analysis technique.

The direct application of the classical pattern classification approach to isolated words, i.e., word-based recognition of handwritten script, leads to so-called holistic approaches (cf., e.g., [22] and the references therein). The captured image of some word to be recognized is considered as an “entity in its whole” [22] and based on some lexicon the actual recognition is performed. Despite its attractiveness, due to its simplicity, this procedure has one serious drawback. In most realistic applications, not enough sample data will be available for robust modeling, which usually prevents its application to real-world recognition tasks.

The alternative to the aforementioned holistic word-based recognition approaches is the analysis of handwritten script at the level of individual characters, i.e., character-based modeling. Based on a limited set of building blocks—character models—word-models are created by concatenation of these basic units.⁶ Character-based modeling corresponds to the standard approach for state-of-the-art MM-based handwriting recognition. There are, however, two variants of the basic procedure.

In the first case, an explicit segmentation of the word image into smaller units, usually the characters it consists of, is performed. In the literature, respective approaches are referred to as segmentation-based or as relying on some explicit segmentation. It is commonly agreed that it is extremely difficult, if not impossible, to correctly

⁵ Unfortunately, in the literature, the important technical aspects of the modeling and algorithms used are quite frequently not given in sufficient detail. Therefore, we concentrate on those publications that make this information explicit.

⁶ Note that the recognition of isolated characters corresponds to some very special application, which lies not in the focus of this survey.

segment a word into its characters without knowing the word itself. Consequently, the basic dilemma of such procedures (cf., e.g., [1, 3, 67, 68, 92]) is that the subsequent classification step will be doomed if this explicit segmentation step fails. This problem is especially critical when processing noisy input data—as handwriting often is. Certain approaches have been proposed to alleviate the strict dependency of segmentation-based HWR on *correctly* subdividing words into the characters they consist of. As one example, explicit over-segmentation is performed (cf., e.g., [70]) and based on some alignment technique (as, e.g., dynamic programming), the optimal “segmentation path” through the word to be recognized is extracted [44, 58, 91, 110]. Alternatively, multiple segmentation solutions are generated by variants of the segmentation technique and the “best” solution w.r.t. the overall recognition results is chosen [21].

By means of the aforementioned segmentation-based procedures, Markov models have been applied rather successfully to handwriting recognition tasks. Various strategies were developed that allows one to cope with the critical dependency of such recognition systems on reasonable segmentation results. However, if processing presegmented data one of the fundamental strengths of Markov models is ignored. As known from different application fields of MM-based recognizers, most notably automatic speech recognition, the basic advantage of Markov models is to perform pattern classification in a segmentation-free manner. Respective procedures are also referred to as implicit segmentation approaches.

For segmentation-free recognition, all base models, i.e., HMMs modeling the respective characters, are integrated into one large recognition model. Technically, this corresponds to a parallel connection of base models by integrating their respective states into a global state-space (see Fig. 4) and adding connections between the HMMs. Viterbi decoding of this global state-space for handwriting data results in the most probable path through all base models. Transitions between modeling units along the path through the global state space correspond to the desired segmentation which is, thus, performed implicitly while classifying.

The standard approach for MM-based segmentation-free HWR is based on some variant of the sliding window technique as described in Sect. 2.3. Features, that are consecutively calculated while moving the analysis window along the image of the particular handwritten word, are fed into the recognition system, and Viterbi decoding is performed for simultaneous classification and implicit segmentation (cf., e.g., [16, 42, 94, 100, 103, 117, 125]). Segmentation-free recognition based on the sliding-window principle has recently even been applied to offline Chinese handwriting recognition [109], which emphasizes the importance of implicit segmentation-based HWR using Markov models.

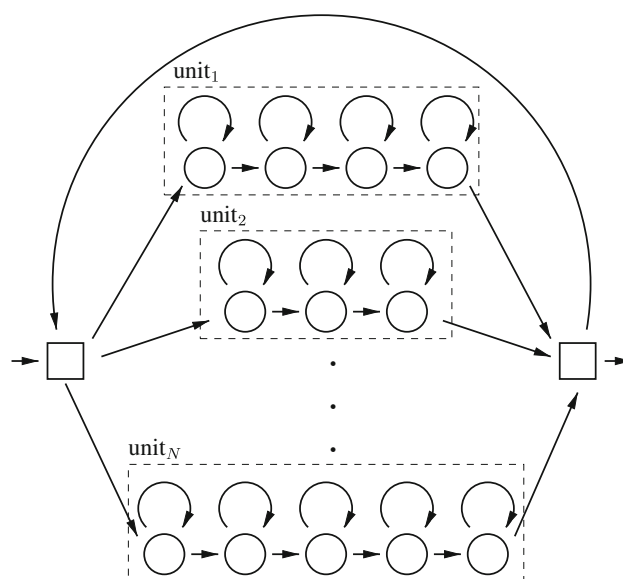


Fig. 4 Parallel connection of HMMs corresponding to certain basic modeling units, e.g., characters or words, for implicit segmentation during Viterbi decoding. Transitions between modeling units represent segment boundaries

4.2 Serialization and feature extraction

Basically, there is a discrepancy between MM-based modeling and the “nature” of the images of handwritten data to be analyzed. Markov models are most suitable for sequential data. However, handwriting data as analyzed by offline recognition approaches corresponds to two-dimensional images. In order to overcome this mismatch, input data need to be serialized appropriately. The most commonly used approach is the application of some variant of a sliding-window technique. In its standard version, a small analysis window is moved along the handwritten text (in the writing direction), and features are calculated serving as sequential representation of the handwriting. Differences exist in the particular configurations of the method, for example, regarding the width and height of the analysis window, or the overlap between adjacent positions. Further variations comprise, e.g., the integrated analysis of multiple analysis windows for automatic slant correction [43]. The basic principle of local analysis, however, remains the same and most major recognition systems are based on it [16, 41, 87, 94, 100, 103, 117].

In virtually all applications, HWR using Markov models is based on certain feature representation of the input data. The motivation for this is twofold. First, some data-reduction is required since otherwise the number of parameters to be estimated during training is far too large for robust modeling based on usual sample sets. Second, despite the application of sophisticated preprocessing techniques that aim at some normalized version of the image data to be analyzed,

handwriting data usually still exhibits substantial variance, which is due to, e.g., the capturing process. In order to focus on the essentials of handwriting, deriving reasonable features is mandatory.

Features for Markov-model-based handwriting recognition can broadly be classified into the following categories: *analytic features* computed directly on the raw pixel intensities or on intensity distributions; *heuristic features* usually based on the analysis of certain geometrical properties. Combinations of these types of features—so-called hybrid approaches—aim at maximally benefiting from both kinds. Feature extraction methods not fitting into this classification scheme can hardly be found in the literature. The majority of current feature extraction techniques, however, can be classified into one of these types.

According to the literature, the calculation of certain statistical features directly on the particular pixel intensities, usually preprocessed, i.e., normalized, in some reasonable way, appears to be rather attractive (also) for Markov-model-based handwriting recognition. Often (local) pixel intensities or pixel density distributions—optionally their average or median values—are considered as some sort of basic features (as, e.g., in [3,40,45,87,117]). Quite frequently, starting from some raw feature set optimized representations are computed by some standard analytic transforms like PCA [28,40,58,95,97,117] LDA [16,58] or function transforms (DCT, FFT, Wavelet, Radon, etc.) [16,30,45]. Based on raw pixel intensities, appearance-based features like Eigen projections can be derived as well [48].

In addition to analytic approaches, various features have been proposed that are based on certain heuristic considerations. Many of them describe structural properties of handwriting like loops, ascenders, descenders [44], slopes [36], directional information [112], or concavity features [41]. In [125], a combination of several geometrical properties of the analysis windows are used as features.

Independently of the basic principle used for feature extraction, feature sets are frequently complemented by adding discrete approximations of the time-derivative of the individual vector components (cf., e.g., [94,125]). This technique, which is also widely used in automatic speech recognition, improves the ability of the final statistical sequence model to capture dynamic aspects of the data representation.

The process of feature extraction is critical for automatic handwriting recognition. This is especially the case for Markov-model based approaches. If the feature representation of handwritten data misses important properties, the recognition itself is likely to fail. In contrast, if too much redundancy is included by some feature representation, robust modeling becomes complicated when only limited sample sets are available for classifier training—as it is usually the case. According to the literature, analytical and heuristic—most notably structural—features are currently

used in virtually equal shares. Since successful recognition systems have been developed based on both types of features, it appears to be some sort of matter of taste which features to use.

4.3 Building the writing model

In MM-based recognition systems, the appearance of handwriting is analyzed using HMMs serving as writing models. Thereby, the regular case is the use of character models, which are concatenated to word models. For example, recognizers for Roman script contain models for upper and lower case letters, numerals, and those for punctuation symbols, which results in 70+ base models in total (c.f., e.g., [16,94,125]). For Arabic recognizers, substantially more models have to be used since Arabic uses 28 basic character shapes, which can appear realized quite differently in four types of contexts (cf. [76]).

The use of character models can limit the recognition performance of an HWR system. Therefore, some recognizers are based on an alternative modeling approach using graphemes as basic modeling units. Graphemes represent more fundamental units in handwriting ranging from single strokes to actual characters [33, p. 3f]. They are used for both Roman and Arabic script recognizers (cf., e.g., [58,89,112,126]).

In some approaches, different variants of graphemes—so-called allographs—are combined in multipath letter models with parallel state paths (cf., e.g., [102]). The advantage of this type of modeling units lies in the increased robustness of the resulting HMMs regarding writing style variations. As Arabic characters have different representations, depending on their particular contexts (isolated, initial, medial, and final), a similar type of modeling can be used for Arabic recognizers [104].

4.3.1 Modeling output behavior

A fundamental question to be answered for HMM-based models is how the data is represented with respect to the statistical outputs generated by the HMM. The simplest way of specifying output distributions is by defining discrete probability distributions over some finite set of symbols. It is, however, quite cumbersome to define a coding of the inherently numeric feature representations of offline handwriting data into a symbol set. Therefore, today, only very few approaches still make use of discrete HMMs operating on either discretely modeled distributions in feature space [70] or a hand-crafted symbolic coding of the data [3]. Very rarely, discrete symbols are combined with continuous attributes for output modeling [127].

As the HMM-internal modeling is greatly simplified for discrete models, quite a number of approaches have been proposed that combine discrete HMMs with a vector

quantization (VQ) step in order to be able to handle continuous feature representations (cf. [44, 45]). However, most methods of this type that are still in use today try to compensate the inherent limitations that arise from splitting up the model into two separate components. In [36], a fuzzy VQ is used to reduce the negative effects of quantization errors. Other approaches combine discrete HMMs and neuronal networks (NN). In [20] and related publications, the NN-based VQ offers the advantage that it can be optimized jointly with the parameters of the HMM. In [89], the NN is used to directly compute the output probabilities of the HMM.

The majority of current offline handwriting recognition systems is based on continuous HMMs that describe output probability density distributions by mixtures of Gaussians [41, 57, 61, 87, 109, 111, 117]. An important parameter of continuous density HMMs is the number of Gaussians used per state. As the sets of Gaussians are state specific and are not shared across the overall model, this number is usually quite small. With the exception of [111] where 64 diagonal covariance Gaussians are used, the number of Gaussians per state ranges between 3 and 12.

In order to construct more compact HWR systems and to use limited training data more effectively, instead of continuous HMMs with state-specific mixtures, tied-mixture HMMs are also frequently used [7, 15, 97, 100, 125]. In these systems, the shared codebooks contain between several hundred component densities (e.g., 300 full-covariance Gaussians in [15]) and some thousand distributions (e.g., 1.5k diagonal-covariance Gaussians in [100]). Several more specific variants of mixture-tying with large numbers of overall densities used (up to approximately 150k) are explored in [94].

Even more parameter tying in a complex configuration of mixture density models can be exploited by using shared codebooks for multiple lower-dimensional sub-spaces of the original feature space [12]. This approach was used successfully in [57] to compress the storage requirements of a large-vocabulary Chinese handwritten character recognition system by a factor of ten without sacrificing recognition accuracy.

4.3.2 Model architecture

According to the sequential structure of handwriting data, the most obvious topology for hidden Markov models, as applied in HWR, is the linear architecture. Here, every state is connected to itself and to its immediate successor state to the right. Note that for Arabic HWR the same holds but, as Arabic is written from right to left, here linear topologies with reversed directionality represent the methodology of choice [41]. Since larger contexts are typically not relevant for modeling handwriting, the number of neighbors that are

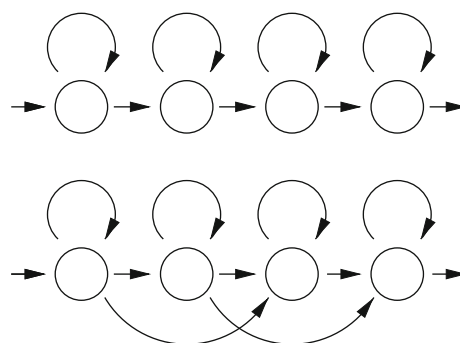


Fig. 5 Most common model topologies used for elementary units in HMM-based writing models (*upper*: linear; *lower*: Bakis)

directly connected to some particular HMM state is usually restricted to more or less directly adjacent states. In order to allow for more variability in the length of the segments described by some basic model, the skipping of the immediate successor state is frequently allowed. Consequently, in this so-called Bakis topology every state has three potential successors (see Fig. 5). Most major recognition systems are based on either the linear or the Bakis topology (cf., e.g., [16, 41, 87, 94, 100, 102, 125]).

The number of states per character model is usually fixed according to certain heuristics. Often the average lengths of characters to be modeled (optionally determined on outlier-reduced sample sets) in combination with the chosen model architecture determine the number of states used (see Sect. 5). There are, however, also approaches for model optimization, e.g., with respect to the number of states (individually) used per character model (cf., e.g., [60, 130]), that allow for minor improvements in classification accuracy. For example, Bakis models with n states could cover characters with a minimum length of $n/2$ frames, thereby—due to self-transitions—not implying an upper limit for the sequence length.⁷

According to the literature, linear (or Bakis) modeling represents the basis for the majority of current HWR systems. However, in addition to this, certain specialized architectures for character models have been developed. In [44], a rather complicated topology is described, which is based on eight internal states. However, the necessity of some specialized model architecture is in this case certainly justified by the segmentation-based nature of the overall approach (see Sect. 4.1). The same holds for the “handmade” model architecture in [91]. For signature verification in [30], a ring-topology of HMMs is successfully applied.

For standard HMMs, the duration that a model can spend in a particular state is implicitly modeled by a geometric

⁷ A potentially differing number of states does usually not affect the parallel and competitive evaluation of HMMs (see again Fig. 4) since HMM scores are normalized over the particular sequence’s length.

probability distribution. However, in some (rare) cases, it can be beneficial to explicitly model the probability of consecutively observing some number of states in a particular state. In [7], experiments with explicit state duration modeling for MM-based HWR are described. It is shown that explicit state duration modeling can best be achieved using Gamma distributions.

4.3.3 2D-extensions of the writing model

The predominant approach for making HMMs cope with handwriting data in the form of offline captured images is the aforementioned serialization of the images by means of the sliding window technique. Despite its popularity and its doubtless effectiveness, alternative approaches for directly dealing with two-dimensional handwriting data have also been developed. Basically, these techniques address the direct treatment of the data by the HMM without the need for explicit “conversion.”

In [112], the use of planar HMMs, i.e., writing models whose emissions are also modeled using HMMs, for Arabic handwriting recognition is described. The five (horizontal) writing zones are directly covered by 1D-HMMs whose combined evaluation allows for directly analyzing image data without the need for explicit serialization. Alternatively, by means of the integration of Markov random fields (MRF) into MM-based handwriting recognition systems, handwriting images can be processed (cf., e.g., [29, 114]). Although 2D-extensions of the writing model are reasonable for certain applications, explicit serialization of handwriting images is, apparently, more promising.

4.3.4 Adaptation

The ultimate goal of automatic handwriting recognition systems is their independence of any constraints regarding the handwriting analyzed. Among others this includes writer independence as well as robustness with respect to different writing styles, or lexicon changes. In order to reach this level of independence for practical applications, certain techniques for model adaptation have been proposed.

In [19], transformation-based model adaptation using EM, MAP, MLLR, and SLLR is described. Similar experiments have also been reported in [49, 73, 116]. It has been shown that model adaptation w.r.t. writer changes or regarding lexicon restrictions (relevant for mail sorting applications) can be realized very effectively.

4.4 The role of language models

In the statistical recognition paradigm, the writing model represented by the HMM needs to be complemented by the language model component for the representation of long-term

sequencing constraints. As such long term constraints are not essential for all applications of HWR, the use of a language model is not as widespread as in the field of automatic speech recognition. However, in recent years, the application of language models for handwritten text recognition and the so-called lexicon-free recognition of virtually unlimited vocabularies have become quite popular.

4.4.1 Word-based language models

In applications of handwritten text recognition where sequences of words are fed into a statistical recognizer without the prior attempt to perform a segmentation, word-based language models are used in much the same way as in automatic speech recognition. Probably, the first report involving the use of statistical language models for handwritten text recognition is given in [85]. The authors investigated the relation between uni- and bi-gram language models of different perplexities in detail, and their impact on recognition quality in writer-independent experiments. In [8], a bi-gram language of similar origin is combined in different parameterizations with a writing model to create alternative recognizers. The perplexity of the model, however, is not given. The considerable performance gains to be achieved by using a bi-gram language model were confirmed by [125] for a quite similar task. Recently, a tri-gram model was used in experiments on the same database [94].

A slightly different use of language models is reported in [111]. There the primary goal is not the recognition of the written text but the classification of spontaneous handwriting obtained from survey forms into a small number of semantic classes. Preliminary results for uni- and bi-gram models are given, which, due to the unique nature of the task, can, however, not be put into perspective.

Unquestionably, the most detailed account of the impact of word-based language modeling on HWR performance is given in [117]. The authors create uni-, bi-, and tri-gram language models on different text corpora and use them in recognition experiments on three different corpora of handwritten text (including a writer independent task). As expected, the use of a language model considerably improves recognition accuracy. However, it does not become completely clear why the perplexities obtained for bi-, and tri-gram models—and consequently the recognition accuracies achieved—remain relatively close in all configurations investigated.

4.4.2 Character-based language models

A quite interesting method for using n -gram models in HWR is to apply them at the character level. Thus, the explicit use of a lexicon can—to some degree—be avoided and recognition is performed in so-called lexicon-free mode.

Character-based language models were first proposed for the recognition of degraded machine-printed documents in [20] and later applied to handwriting recognition in [14]. Experiments on a small (four writers) word-segmented database are reported for 3-, 5-, and 7-gram models including perplexities. In [17], the same methodology is applied to the problem of address recognition. Quite promising results have been obtained for back-off smoothed n -gram models up to length 7.

The lexicon-free approach has also been applied successfully to text recognition tasks, i.e., without relying on word presegmentation. Character tri-gram models were used in [94], whereas n -gram model lengths from 2 to 5 were investigated in [123] and later in [125].

4.4.3 Integration

Especially, for the use of long-span statistical language models as presented earlier, the correct integration of the writing and the language model during decoding is of fundamental importance. Unfortunately, details about the solutions used can rarely be found in the literature.

In both [111] and [117], the combination of HMM and n -gram model is compiled into a combined finite state automaton usually referred to as a word network.

Research groups that entered the field of HWR from the area of automatic speech recognition usually apply the same decoder as previously developed for automatic speech recognition, which is then referred to via bibliographic reference only. Thus, in [14], a stack decoder and in [125], a time-synchronous decoder applying beam-search and time-based copies of search trees is used, whereas the system described in [94] applies a two-pass decoding strategy [4].

4.5 Multi-classifier combination

Probably due to the growing popularity of multi-classifier systems and methods for combining classifiers, several researchers also explored such techniques in the context of MM-based offline HWR. In [15], different classifiers are derived from different feature representations. The combination of the different hypotheses obtained is then achieved by a slight variant of the ROVER framework [52], which was originally proposed in the context of automatic speech recognition. On a small (six writers) proprietary datasets, significant performance improvements are reported.

A rather unusual combination method is proposed in [59]. HMMs for word models are assumed to result from a concatenation of character models. Different such models are built by different training methods, namely, boosting and bagging. Finally, combined models are formed by allowing the character sequences within a word model to switch freely between

model variants. Results reported on a small task (only six writers) show a minor improvement which might, however, not be significant.

The multi-classifier system reported in [8] is constructed by varying the parameters for combining HMMs and n -gram models used. The combined result is then obtained by simply applying ROVER. Significant improvements are reported on a large vocabulary recognition task for a combination of 18 individual configurations.

An interesting application of a multi-classifier approach is reported in [43]. The three baseline classifiers for recognition of Arabic handwriting are constructed by compensating for different slant angles during feature extraction. As only the recognition of isolated words is considered, the combination of the results can be achieved by a second classification stage which is most successfully realized as an MLP.

At a more abstract level, multi-pass recognition techniques are also related to multi-classifier approaches. As, for example, described in [21, 121, 122], MM-based HWR systems have been developed that combine the results of at least two consecutive recognition stages.

The combined use of HMMs and neuronal networks for HWR reported in [68] and refined in [67] can be considered a mixture between a multi-pass recognition and parallel classifier combination. First, HMM-based classifiers are used for character segmentation. Then NN-based character classifiers produce local recognition scores that are finally combined by a NN classifier with the scores obtained from the HMMs.

5 Markov-model-based offline handwriting recognition systems for practical applications

The theoretical foundations of Markov models are, basically, independent of their specific application domain. However, when aiming at fully functional MM-based recognition systems that can actually be used for practical tasks, domain-specific know-how is the key prerequisite for their successful application. Consequently, the majority of this kind of HWR research effort performed in the last 20+ years has been devoted to the development of techniques for the adoption of Markov models to offline handwriting recognition.

Aiming at a comprehensive overview of Markov-model-based HWR as it is actually performed in current practical applications, in the following, the focus of this article is shifted toward the description of recognition *systems*. After the theoretical aspects and key developments in the field have been surveyed, integration aspects and concrete evaluations of recognition capabilities are now discussed. Reviewing the literature, seven major recognition systems were identified, thereby concentrating on those systems that, according to recent publications and to the authors' knowledge, are still being maintained and developed by the particular authors.

For most of the systems, detailed system descriptions exist. Furthermore, numerous refinements are often described in the particular follow-up publications.

5.1 Datasets

The evaluation of HWR systems is usually performed by means of practical experiments. Therefore, parameter estimation for the particular statistical models (HMMs and n -gram language models, respectively) is performed on (annotated) training data and the recognition capabilities are measured on more or less well-defined test-sets.

The description of the HWR systems, which is given in this section, also includes summaries of their most important recognition results. In the following, the databases that are most frequently used are briefly described. The documents contained by the particular databases are usually scanned with 300 dpi at a grey level resolution of 8 bit (exceptions will be denoted).

5.1.1 IAM-DB

The IAM dataset represents a handwritten English sentences database for offline HWR [88]. It is based on the LOB corpus, a collection of texts that comprise about one million word instances. In its version 3.0, the database includes images of 1,539 forms that were produced by 657 writers, which results in a total of more than 115k word instances. Overall, a total number of 10,841 word tokens is included in the database.

5.1.2 IAM-OnDB^{Cam}

This set of images corresponds to a side-product of the IAM online database captured for whiteboard reading applications (IAM-OnDB [75]). It consists of color images of whiteboard texts that have been taken with a digital camera with a resolution of $3,264 \times 2,448$ pixels each. The database contains 491 documents written by 62 subjects without any constraints w.r.t. writing style. Similar to IAM-DB, the text written on the whiteboard is based on prompts from the LOB-corpus. In total, the database comprises a dictionary of 11,059 words. Unfortunately, unlike the other IAM databases this database is not yet publicly available.

5.1.3 IFN/ENIT

The IFN/ENIT database represents a standardized set of handwritten Arabic town/village names [96]. It consists of scanned forms of more than 400 writers with about 26,400 city names containing 210k+ characters. In addition to the images and their annotation, further information as for example the correct baseline of the cropped and preprocessed words are also provided.

5.1.4 Cambridge

The Cambridge database contains handwritten documents—353 handwritten text lines are split into training (153 lines), validation (83 lines), and test (117 lines) sets [106]. In total, the database contains images of 2,360 training words, 675 validation words, and 1,016 test words. The overall vocabulary consists of 1,334 words.

5.2 Systems

According to the reviewed literature and to the criteria for selection as defined earlier, seven major recognition systems are considered that focus on offline Markov-model-based HWR. In the following, their key features and recognition results are described. For the reader's convenience and for easier comparability of the particular systems, Table 1 summarizes the main characteristics of these MM-based handwriting recognizers together with key references.

5.2.1 BBN

BBN Technologies Cambridge, USA, can be considered as one of the pioneers in transferring Markov-model-based techniques from the domain of automatic speech recognition to the field of optical character recognition. Since the mid-1990s BBN worked extensively on various aspects of statistical modeling for optical character recognition (cf., e.g., [93]). BBN's recognition framework is based on the BYBLOS engine, which was originally developed for automatic speech recognition purposes (cf., e.g., [32] for a system description). Recently, BBN has successfully applied its OCR framework to the recognition of handwritten texts in different scripts [94]. Although BBN as a company mainly addresses commercial applications, which usually implies certain non-disclosure of technical details, an astonishingly large number of publications exists that very thoroughly describe the recognition system(s).

The BBN handwriting recognition system follows the classical architecture of Markov-model-based recognizers for general sequential data. It integrates both HMMs and statistical n -gram language models. Apparently, the BBN systems, which includes both OCR and handwriting recognition utilizing Markov models, aim at universal applicability without language or script dependent restrictions. As an example, the OCR system has already been used for the recognition of numerous script types even including such "exotic" languages as Pashto (main language in Afghanistan) [35]. In the same way, multi-linguality is addressed by BBN's research activities in Markov-model-based handwriting recognition.

The offline HWR system includes modules for complete document layout analysis that segments each input image

Table 1 Overview of the state-of-the-art in Markov-model-based handwriting recognition systems

System / maintainer	Primary domain	Results (WER) / database	Serialization	Features used	HMM topology	Emission modeling	# HMM states	Language model	Specials	Key references
BEN BBN Technologies Cambridge, USA ^a	Text recognition	40.1% IAM-DB 10.6% IFN/ENIT 17% (CER) ETL9B	Sliding window width: 1/15 text-height height: text-height shift: 1/3 width	Blackness percentiles, angle & correl. features (81)+ LDA to 15-D	Bakis (1D)	Semicontinuous 150k densities (est.) / variants of tying	14 (empiric)	Tri-gram word-based	-Context-dep. char. models -Multi-lingual	[93, 94]
CENPARMI Concordia Univ. Montreal, Canada ^{b,c}	Bank cheque reading/ address signature verification	0.8–12% Proprietary (city names) ≈10% (Brazilian cheques)	Sequence of segmentation points (SP) defines sequence of context windows	Geometric+histo- gram based + segmentation features (opt.: LDA)	Special; separate models for characters and spaces, resp.	Discrete, string encoding	8 Incl. 3 diff. paths for characters; 2 for linear space models	-	-Segmentation- based -Graphemes -Rejection of poor hypotheses	[44, 58, 67, 68]
IAM Univ. Bern, Switzerland ^d	Text recognition	37.3–53% IAM-DB ≈9% Cambridge	Sliding window width: 1px height: text-height	Geometric (9)/ grid-based px. counts	Linear (1D)	Continuous	14 (empiric)	Bi-gram word-based	-	[9, 87, 117]
SIEMENS Siemens Konstanz, Germany ^e	Postal automation (address reading)	6–12% Proprietary (city names, 1k imgs.) 12% IFN/ENIT	Sliding window	Geometric (20)	Linear (1D) left-to-right (Roman) right-to-left (Arabic)	Semicontinuous (Roman) right-to-left (Arabic)	Character- dependent	Likelihood trellis for words, breadth-first search+ pruning	-Allographs as modeling units -Model adaptation (state path changes) -Also for Arabic	[24, 65, 103, 104]

Table 1 continued

System / maintainer	Primary domain	Results (WER) / database	Serialization	Features used	HMM topology	Emission modeling	# HMM states	Language model	Specials	Key references
TPar/UoB Telecom ParisTech Paris, France Univ. Balamond ^{f,g}	Arabic handwriting recognition	13% IFN/ENIT	Sliding window with vertical zoning	Distributions + derivative & concavity (24)	Bakis (1D) right-to-left	Continuous 3 Gaussians per state	4	–	–Viterbi train –Language independent features, so far only used for Arabic	[41, 42]
TUdo TU Dortmund, Germany ^{h,i}	Text recognition (scanned documents/whiteboard images)	28.9% IAM-DB 39.8% IAM-OnDB ^{Cam}	Sliding window width: 4px height: text-height shift: 2px	Geometric (9) + 1st derivatives	Bakis (1D)	Semicontinuous 1.5k Gaussians (diag. cov.)	Character-dependent, empiric (avg. 30)	Bi-gram word-based	–	[100, 125]
TUM TU Munich, Germany ^{j,k}	Postal automation (address reading)	14/36.1% Proprietary (handwritten addresses streets/cities)	Sliding window	Geometric (20) + LDA on 3 adj. frames to 30-D	Linear (1D)	Semicontinuous 300 Gaussians (full cov.)	Mostly 3 dependent on special char. width (empiric)	3–7 gram character based	–HMM adaptation (SLLR) –Confidence measures –Opt: multi-pass	[16–19]

^a BBN Technologies, Cambridge, MA, USA

^b Center for Pattern Recognition and Machine Intelligence, Concordia University, Montréal, Canada

^c Also: Pontificia Universidade Católica do Paraná, Curitiba, Brazil

^d Institute of Informatics and Applied Mathematics, University of Bern, Bern, Switzerland

^e Siemens AG, Logistics and Assembly Systems, Postal Automation, Konstanz, Germany

^f Telecom ParisTech (former Ecole Nationale Supérieure des Télécommunications / TSI), Paris, France

^g University of Balamand, Faculty of Engineering, Tripoli, Lebanon

^h Intelligent Systems Group, TU Dortmund University, Dortmund, Germany

ⁱ Previously: Applied Computer Science Group, Faculty of Technology, Bielefeld University, Bielefeld, Germany

^j Inst. for Human-Machine Communication, Technical University of Munich

^k Previously: Dept. of Computer Science, Faculty of Electrical Engineering, Gerhard-Mercator-University Duisburg, Duisburg, Germany

into single column text zones. After normalization (de-skewing and line finding, plus some preliminary—according to [94]—slant correction) the latter represents the input for the recognition system. Serialization is performed using the sliding window technique where overlapping frames are extracted for every line of text. Thereby, the height of the analysis windows equals to the height of the particular line, the width is 1/15 of the height and adjacent frames overlap by 2/3 of window width. In these frames, “percentile features” (patent pending) are calculated on binarized pixels. The blackness of a frame is integrated from top to bottom. After normalizing by the sum of black pixels, a monotonically increasing function encodes the amount of blackness up to any particular position within the frame. By sampling the function equidistantly, 20 features are calculated. In addition to this, horizontal and vertical derivatives, respectively, complement the feature vectors. Together with angle and correlation features (ten each) that are calculated from scatter plots of the text pixels, 80-dimensional vectors are extracted for every frame. In the description of their OCR system [93], which represents the origin of the BBN HWR system, the authors describe 80 components. However, for the HWR system in [94], 81 features are mentioned. Unfortunately, it remains unclear how the missing component is calculated. The resulting 81-dimensional feature vectors are then reduced to 15 dimensions by applying LDA.

The BBN recognizer uses tied-mixture character HMMs as elementary writing models with 14-state Bakis topology. Different variants of mixture tying (classical global tied mixtures over all states, character-tied mixtures where Gaussians are shared by states of single characters, and some mixture form of both types—state tied mixtures with a total of approximately 150k Gaussians) are integrated for most effective exploitation of sample data for model estimation. The latter is performed using classical Baum–Welch training, whereas the recognition itself is a two-pass beam-search process combining a fast forward match with a more detailed backward search [4]. Optionally, a lexicon is used during recognition, and statistical n -gram language models (word-based tri-grams) are integrated.

Apparently, it is especially the large number of mixtures, which are used for emission modeling, that enables truly multi-lingual handwriting recognition. Experimental evaluations have been described that address the recognition of English, Arabic, and Chinese script [94]. It needs to be mentioned that, generally, very detailed descriptions of the BBN recognizer’s configuration for the particular experimental evaluations are given. For the first set of experiments the IAM-DB is used. By means of a tri-gram language model word error rates of approximately 40% have been achieved. Experiments for Chinese HWR have been conducted, for example, on the ETL9B dataset (200 instances each of 71 Hiragana

and 2,965 Kanji characters—in total 3,036 unique characters). Here, character-based error rates of approximately 17% were reported. Finally, the system has been evaluated on the IFN/ENIT database. Word error rates of 10.6% indicate the suitability of the BBN recognition system also for Arabic.

5.2.2 CENPARMI

Document analysis in general represents one of the major working fields of the research group of Ching Suen at the Centre for Pattern Recognition and Machine Intelligence (CENPARMI) at Concordia University, Montréal, Canada including associated scientists from other institutions. Among others their activities are focused on HMM-based offline handwriting recognition. Over the years a recognition system has been developed that is successfully being used for various applications including signature verification, postal address reading, recognition of (Brazilian) bank cheques, and so forth. A general system description has been published in [44]. Additionally, numerous papers addressing applications and enhancements of the basic system exist (cf., e.g., [58,67,68]). The base system was also used for hybrid recognition approaches or for classifier Ensembles [66,68].

The CENPARMI system for Markov-model-based offline HWR differs from most of the systems described in this survey in certain aspects. First, recognition relies on explicit segmentation of extracted words into (pseudo-) characters. Second, compared to other approaches, a radically differing strategy regarding feature extraction is pursued. According to [44] “lexicon-driven word recognition approaches do not require features to be very discriminative at the character or pseudo character level because other information, such as context [...], word length, etc., are available and permit high discrimination of words. Thus, [they] consider features at the segment level with the aim of clustering letters into classes.” Furthermore, the CENPARMI system is based on discrete hidden Markov models using emissions calculated on string encoding of different feature sets.

After preprocessing word images by applying skew compensation, lower case letter area (upper-baseline) normalization, character slant correction, and smoothing, an explicit segmentation of the input data is computed. The procedure explicitly performs over-segmentation by generating a high number of possible segmentation points (SP) that sub-divide words into units that not necessarily correspond to actual characters but to some smaller portions. In this way, several segmentation options are offered, the best ones to be validated during recognition. At every segmentation point, the neighborhood is divided into writing zones that represent small analysis windows centered at the particular SP. Based on this, three different sets of features are calculated.

The first set comprises global features (loops, ascenders, and descenders). For the second set of features, bi-dimensional contour transition histograms of each segment in the horizontal and vertical directions are analyzed and certain statistics are derived that serve as discrete feature values (from a set of 14 symbols). Furthermore, segmentation features that reflect the way segments are linked together are considered. Optionally, an LDA transformation is applied to the resulting feature vectors aiming at the introduction of “class information during feature extraction” [58].

CENPARMI’s writing models (HMMs) are based on graphemes as modeling units. A special model topology has been developed that consists of eight segment specific states and a rather complicated transition scheme. The rationale is to explicitly map the over-, under-, or correct segmentation of a letter to the model architecture. Additionally, the recognizer uses separate space models with modified linear left-to-right architecture without self-transitions. Parameter estimation is performed using a slightly modified version of Baum–Welch training. Recognition itself is performed using a rather complex decoding procedure that apparently allows an implicit detection of the writing style (no further details given).

Since the CENPARMI recognizer together with variants and enhancements of the base system has been used for various application domains, the results reported for experimental evaluations of its recognition capabilities are rather diverse. Addressing postal address reading experiments in recognizing unconstrainedly handwritten French city names were performed on proprietary data. Depending on the lexicon size (varying from 10 to 1,000) results between almost perfect recognition and approximately 12% word error rate have been reported for test sets containing between 4k and 11k images [44, 58]. For the analysis of bank cheques, handwritten month words, i.e., a lexicon of 12 entries, are recognized. Here, recognition results of about 10% word error rate have been achieved for a set of 402 test images [44].

5.2.3 IAM

The group of Horst Bunke at the Institute of Informatics and Applied Mathematics (IAM) at the University of Bern, Bern, Switzerland can without doubt be called one of the most productive teams with substantial influence on general research in handwriting recognition. In fact, one of the earliest papers on Markov-model-based recognition of unconstrained cursive script, i.e., handwriting, was published by them in 1995 [23]. Over the years, numerous scientists affiliated or associated to this group (including the group at the swiss IDIAP institute) have contributed to the research field of (Markov-model-based) handwriting recognition. Consequently, today IAM maintains a mature handwriting recognition system, which has successfully been applied to both online and offline recognition tasks.

The IAM system for offline reading of unconstrained handwritten pages has, for example, been described in detail in [87, 117]. Additionally, a multitude of refinements, enhancements, and specialties have been described in numerous papers published in the last few years. Basically, the system follows the classical architecture of a Markov-model-based recognition system as summarized in Sect. 2. Handwritten documents are presegmented regarding single text lines, which are then fed into the recognition system that proceeds in a segmentation-free manner, i.e., not relying on further segmentation. The IAM system integrates continuous HMMs as writing models and statistical n -grams as language models. The first step in the processing chain comprises suitable preprocessing that addresses skew and slant correction, text line normalization and horizontal scaling.

For feature extraction, text line images are serialized by means of a sliding window technique. Thereby the analysis window, which is moved from left to right along the text line, is one column wide, i.e., there is no overlap between consecutive frames. The height of the sliding window is identical to the text line’s height. For every frame, nine local geometrical features are computed. On the one hand, these features cover the characteristics of the analysis window from a global point of view (weight of the window, center of gravity and so forth). On the other hand, the features also describe details about the writing itself by considering positions and orientations of contours and certain pixel statistics. Note that variants of the feature extraction process have been described where only local grid-based pixel counts were used [117, 118]. The IAM system is based on character models with linear topology that each consist of a fixed number of 14 states (empirically found) with continuous emissions. Identical models are used for capital and small letters and word models are obtained by simple concatenation of character models. Parameter estimation is performed using Baum–Welch training, and for recognition, Viterbi decoding is applied. The IAM system also integrates statistical n -gram language models (up to tri-grams), which are estimated using discounting and backing-off.

The effectiveness of the overall IAM system as well as of its recent enhancements is usually documented by the results of experimental evaluations based on the IAM database. Depending on the actual configuration of the particular experiments and the recognizer used, word error rates between 37.3% [9] and approximately 53% [117] have been achieved. Furthermore, in [117] results for evaluations on the Cambridge database ($\approx 9\%$ WER) were reported.

5.2.4 SIEMENS

The primary field of activity of the department of logistics and assembly systems—postal automation—at Siemens AG in Konstanz, Germany is address reading. Most notably,

automatic recognition systems for mail sorting as applied by major postal companies are being developed. The beginnings of the recognition system lie in developments that have been pursued already in the early 1990s—at this time at Daimler-Benz research center in Ulm, Germany [24,65]. It evolved from a framework that had originally been developed for automatic speech recognition. Recently, the system, which is created to analyze Roman script, has even been used for Arabic handwriting recognition—requiring minor modifications only [104]. Since Siemens mainly addresses commercial applications, not all details regarding the recognizer have been published, some developments might even have been patented. However, in various papers, the key components of the system are described, which provides an overview of the contributions.

The Markov-model-based address reading system of Siemens is applied to those automatically extracted portions of some scanned image that are relevant for mail routing (zip code, city, street, and so on). Prior to actual recognition, shear angle, rotation, and stroke width and size, respectively, are normalized using (undocumented) standard procedures. Furthermore, skeletonization is applied. Handwriting is modeled by means of semicontinuous (tied mixture) hidden Markov models with full covariance matrices. The script model of the Siemens system is defined by a set of graphemes, namely, letters, numbers and special characters. Variants of these graphemes—allographs—then represent the final modeling units for character HMMs that are composed to word-HMMs. The character models consists of various “paths” that can also be dynamically adapted during training. All paths are jointly followed by an optional pause state. Apparently, the number of states per character model is somehow determined automatically (no details given). The models exhibit the classical linear left-to-right topology for Roman script, and right-to-left model architecture for Arabic, respectively.

The Siemens recognizer uses the standard sliding window approach for serialization of the two-dimensional image data. The concrete parametrization of the procedure has, however, only been documented for the most recent developments at Siemens toward offline HWR for Arabic scripts [104], and it can be assumed that the parameters for the analysis of Roman script are comparable. A window width of 11 pixels together with an overlap of two thirds between adjacent frames seems to be optimal. Thereby, the height of the analyzed text lines determines the vertical size of the sliding window. Feature extraction is based on the analysis of binary connected components (BCC)—black or white contour-polygons, the surrounding rectangle, and a reference to inner context areas—that are extracted on normalized and approximated word skeletons. Frames are divided into five horizontal zones where geometric features are calculated that mainly describe upstrokes and cross-lines. Additionally, curves, strokes, and cusps are described by the 20-dimensional

features used [24]. In some publications, an LDA-transformation is applied but, unfortunately, no further details are given.

In postal address reading, the variability of putative addresses is extremely large. For practical applications of such an automatic recognition system very large vocabularies that typically contain more than 20k words need to be considered. Efficient recognition, therefore, requires sophisticated model decoding techniques. Among others, a two-stage decoding strategy was developed for the Siemens system [103]. In the first phase, lexicon-free recognition is performed that produces sequences of characters. Following this, a breadth-first search on a lexicon tree including pruning is performed. It operates on intermediate results of the first step to estimate likelihoods for character paths. The latter corresponds to an effective restriction of character sequences apart from the popular use of statistical n -gram language models.

Because of its commercial background, most of the experiments performed to evaluate the recognition capabilities of the Siemens system are based on proprietary data from the company. Here, for example, word error rates for city name recognition between 6 and 12% (depending on the lexicon size of either 1,000 or 100 city names) on address reading tasks with more than 1,000 images have been achieved [103]. Since these data-sets are not publically available, direct comparisons to other recognition systems are difficult to perform. However, since the Siemens system recently won the ICDAR 2007 Arabic HWR competition [82], the recognition capabilities of the system can also be compared more objectively. On the IFN/ENIT database of Tunisian city names, the system achieved approximately 12% word error rate [104].

5.2.5 TPar/UoB

Another Markov-model-based offline HWR system is being developed and maintained by a group of researchers around Laurence Likforman-Sulem at Telecom ParisTech (former GET-Ecole Nationale Supérieure des Télécommunications / TSI), Paris, France, and the University of Balamand, Faculty of Engineering, Tripoli, Lebanon, respectively. So far, the TPar/UoB system has been applied exclusively to Arabic handwriting recognition tasks. The superior recognition capabilities of the system—in its state at that time [41]—for the analysis of Arabic script were impressively demonstrated when it has won the ICDAR 2005 Arabic HWR competition [81]. The handwriting recognition system is derived from a general purpose HMM toolkit that was originally developed for speech recognition applications. A most recent description of the TPar/UoB system can be found in [42]. The basic system has been enhanced toward a hybrid recognition approach, i.e., utilizing multiple classifier combination techniques (cf. also [43]).

The recognition system uses character-based HMMs that are concatenated to word models, which are evaluated in parallel. No further restrictions (like, for example, language models) are applied to the hypotheses generated. According to the literature, there is hardly any preprocessing applied to binarized input images, which stands in contrast to most alternative recognizers. The authors argue that normalization may introduce image distortions which apparently has negative influence on the overall recognition process. The only preprocessing, as it is documented in the literature, is restricted to proper baseline detection.

Input images are serialized using a standard sliding window technique. Further details regarding the concrete parametrization are not given. Since Arabic script is addressed, the analysis window is shifted along the word image from right to left. At every position, vertical zones are created using the lower and upper baselines as previously extracted. Thereby, the middle zone does not contain ascenders and descenders, whereas in the remaining zones, ascenders and descenders can be found, respectively [41]. For every frame, 24-dimensional feature vectors are extracted. They consist of distribution features that are based on foreground pixel densities, one derivative feature, and concavity features. The authors claim (but do not prove) that the features can universally be used for any script, which can be decomposed into the aforementioned three zones.

The TPar/UoB system uses continuous HMMs for modeling Arabic characters in their particular context variations (159 models in total). Every model contains four states and a mixture of three Gaussians (presumably with full covariances) for emission modeling. Although it is not explicitly denoted as such, the model topology of the HMMs corresponds to a (1D) Bakis right-to-left architecture. Parameter estimation is performed using a segmental version of the standard EM algorithm. In this so-called Viterbi training procedure, the most probable state sequence is integrated into the estimation (for details cf., e.g., [47, p. 80f]). Recently, the base system has been enhanced with the explicit goal of increased robustness against inclination, overlap, and shifted positions of diacritical marks. Therefore, a combination of three homogeneous HMM-based classifiers is proposed that all have the same topology as described here and differ only in the orientation of the sliding window [42].

For the evaluation of the recognition capabilities of the TPar/UoB systems, results of recognition experiments on the IFN/ENIT database have been reported. By means of the recognition framework, word error rates of approximately 13% have been achieved.

5.2.6 TUDo

At TU Dortmund University (TUDo) in Dortmund, Germany, research in automatic offline handwriting recognition is pur-

sued with special emphasis on camera-based approaches (cf. [100]). At the maintainers' former affiliation, Bielefeld University, Bielefeld, Germany, considerable effort was already devoted to MM-based handwriting recognition (cf., e.g., [125]) addressing for the first time the task of automatic whiteboard reading using a camera-based approach [124]. The HWR system is based on a general Markov model toolkit (ESMERALDA⁸ [51]), which has originally been developed for automatic speech recognition purposes.

The general task considered by the TUDo system is writer-independent offline recognition of handwritten texts. Therefore, an integrated recognition framework is being developed, which quite closely follows the general architecture for MM-based handwriting recognizers. It uses semi-continuous HMMs as writing models and statistical n -gram models as language models. In order to allow for its application to "real-world" scenarios like the aforementioned camera-based whiteboard reading task, the TUDo system includes a text-detection module that extracts lines of text from image data. Prior to the actual recognition stage, standard preprocessing operations are applied (skew, slant, and size normalization). For serialization of the two-dimensional handwriting data, a sliding window approach is applied to normalized text lines. They are subdivided into a sequence of overlapping stripes of 8 pixels width (overlapping each other by 75%) and the height of the line. For each of these frames, a set of nine geometric features that describe the coarse shape of the writing within the local analysis window plus their first derivatives are computed. The resulting feature vectors are fed into the recognition system, which consists of word models that are built by concatenation of separate models for upper and lower case letters plus numerals and punctuation symbols (75 in total). All models have Bakis topology and share a codebook of 1.5k Gaussians with diagonal covariance matrices. The number of model states is automatically determined depending on the length of the respective unit in the training material (30 on average). Parameter estimation is performed using standard Baum-Welch training. Plausible word sequences, determined by the HMM decoding process (Viterbi), are restricted by word based statistical n -gram models that are estimated by applying absolute discounting and backing-off.

For the judgment of the overall recognition capabilities of the TUDo system, word error rates obtained in experimental evaluations have been reported in various publications (cf., e.g., [100, 124, 125]). The basis for these experiments are either images of the IAM-DB or those from the IAM-OnDB^{Cam}. For the first task, word error rates of 28.9% were reported (perplexity of bi-gram used is 645) and for the whiteboard images of IAM-OnDB, 39.8% error rate are achieved (perplexity of the bi-gram used is 310).

⁸ <http://sourceforge.net/projects/esmeralda>.

5.2.7 TUM

Another system for offline Markov-model-based handwriting recognition is being developed and maintained at the Institute for Human-Machine Communication at the Technische Universität München (TUM) in Munich, Germany. Already back in the mid-1990s, Gerhard Rigoll and his group (at this time at Gerhard-Mercator-University, Duisburg, Germany) successfully worked on aspects related to Markov-model-based offline HWR (cf., e.g., [101]). After more than a decade of intensive research, a mature recognition system has evolved that, among others,⁹ has widely been used for offline postal automation tasks (cf. [16] for the most recent description of the overall system). Comparable to other Markov-model-based HWR systems, the origins of the TUM framework lie in speech recognition research.

Automatic address reading requires robust, writer-independent recognition of unconstrained handwriting with a virtually unlimited vocabulary. In order to tackle this challenging problem, the core of the TUM system consists of semicontinuous hidden Markov models serving as the writing models combined with character-based n -gram language models where n ranges from 3 to 7. Preprocessing of scanned documents to be recognized includes localization and segmentation of the address words plus the usual stages of image enhancement (denoising) and normalization (w.r.t. skew, slant, and height).

Input images of successfully localized handwriting are first binarized, and binary connected components (BCC) that are comparable to those used by the Siemens system are analyzed for the automatic extraction of ruler lines. These lines (baseline and the line above lowercase letters) are used for rotation normalization and shearing, detection of ascenders and descenders, and height normalization. For feature extraction, a small local analysis window is shifted from left to right over the normalized BCCs of some extracted word. Consecutive analysis windows overlap to some proportion. Unfortunately, no further details about the concrete parameterization of the sliding window approach have been reported. However, it can be assumed that the window width is substantially smaller than some typical character whereas the height of the sliding window corresponds to the height of the extracted words. For the majority of applications considered, geometrical features are calculated within the sliding window. Therefore, frames are divided horizontally by the ruler lines in five overlapping areas, in which dashes, dots, cusps, upstrokes, curves, and horizontal or vertical lines are detected to determine 20 features. Finally, a linear discriminant analysis (LDA) is performed on three adjacent frames.

The resulting feature vectors are then reduced from 60 to 30 dimensions.

The TUM system is based on character models (77 in total) with linear topology, “mostly three states (except for the special characters depending on their lengths)” [17] and tied-mixture modeling with 300 Gaussians (full covariances). Parameter estimation is performed using Baum–Welch training and recognition utilizes Viterbi decoding as usual. In addition to the classical procedure, certain specialties have been integrated into the TUM-system. First, it also utilizes model adaptation techniques for specialization of the writing models. According to [19], the application of scaled likelihood linear regression (SLLR) corresponds to the most effective adaptation method. Furthermore, confidence measures are used for hypotheses rejection if the recognition accuracy becomes too low [18]. More exotically, the TUM system has also been enhanced by optionally integrating a multi-pass HMM approach utilizing extended feature sets for a combination of horizontal and vertical HMMs [121, 122]. TUM integrates statistical n -gram language models for restricting character sequence hypotheses as provided by the open-vocabulary HMM recognition subsystem. Compared to competing systems, rather large contexts of up to $n = 7$ are here considered in n -grams that are estimated using discounting and backoff techniques [14].

In earlier publications, the capabilities of the TUM system have been judged by means of experimental evaluations on the SEDAL database (cf., e.g., [14]). In most recent papers, however, results of experiments on proprietary databases of handwritten addresses are reported. These datasets consist of scanned images of address fields from letters (envelopes) as they have been scanned in real German post offices. For a test set that consists about 2,000 words, error rates of 14% for the recognition of city names using a 20k dictionary, and 36.1% for the recognition of street names (also using a 20k dictionary) are reported.

6 Discussion

In the last few years, Markov models have been applied very successfully to the research field of (offline) handwriting recognition. In this article, the research field of MM-based offline HWR in its current state has been surveyed.

In order to draw conclusions in the following, we will first summarize the state of the field, followed by the description of methodological trends and future challenges that have been identified while analyzing the literature. Since the particular approaches as they were described in the literature are still difficult to compare *objectively* some general remarks on reporting results will be given additionally.

The practical outcome of this section is a set of guidelines and hints that, at least to the authors’ minds, should be considered for future research and development in the field.

⁹ The Duisburg handwriting recognizer (now TUM) also has a fairly long history in online recognition applications.

6.1 General state of the field

Tackling the problem of offline handwriting recognition does not necessarily require the application of Markovian models. In fact, over the years, virtually all major techniques from the wealth of pattern recognition methods have been applied to the task (including Neural Networks, Support Vector Machines, Graphical Models, etc.). Markov-model-based recognizers, however, gained special importance since they are apparently extremely suitable for the analysis of handwriting data (images)—once they have been transformed into a proper sequential representation. Today the field of Markov-model-based HWR can be considered being mature according to the large number of related publications and the existence of several competing recognition systems. In this article, the state of the art of Markov-model-based offline handwriting recognition has been surveyed with special focus on the most widely used hidden Markov models and statistical n -gram models. Although being highly desirable, a comparison of the general MM-based approach with other offline HWR techniques at that level of detail, which would be necessary for a truly objective judgment, is far beyond the scope of this survey. In fact, this would be the subject of another major article (good starting points for this are, e.g., [22,54]). The clear focus of this survey is on Markov models—more precisely on related theoretical and practical aspects for their use in offline HWR applications.

Certainly, the most critical aspect of MM-based offline HWR approaches is the serialization of the two-dimensional input images. HMMs and n -gram models are eminently suited for sequential data—but strictly speaking, images originally do not correspond to this kind of data. Applying the sliding window technique is, basically, sort of a work around that converts image data into sequences. However, there is no theoretical justification for this since humans do usually not perform something similar while reading (cf. [10]), and there is no physical image-formation mechanism behind. Nevertheless, it works very well in practice. By means of the “short-time” analysis underlying the sliding window procedure actual sequences of features are derived. This allows the use of Markovian models—the de facto standard for the analysis of sequential data.

The superiority of hidden Markov models for the analysis of sequential data lies in the fact that segmentation and classification are performed simultaneously in an integrated procedure. HMMs are able to cope with input data that varies substantially in length. On the other hand, there is not much motivation for applying HMMs to more or less static data with only little length variation. More precisely and being a bit provocative, there are better suited classification approaches in the field than HMMs for, e.g., isolated character recognition. As long as segmentation is not the most critical issue, HMMs might not necessarily outperform alterna-

tive classification approaches (cf. [11] for a general treatment of what “HMMs can/cannot do”).

Similar to the majority of statistical pattern recognition approaches, the theory of Markov models alone is not sufficient for setting up recognizers that can successfully be applied to practical tasks. There is always some sort of expertise within the particular application domain required to properly make necessary design decisions, e.g., regarding the choice of the basic modeling units, model topologies, model combination, etc. (cf. [47, part II—Practice]).

6.2 Methodological trends

When analyzing recent publications related to the field of Markov-model-based offline HWR, certain methodological trends can be identified. In the following, we will briefly describe the most important ones and discuss their impact on future research in the area.

6.2.1 Segmentation-free recognition

Basically, offline handwriting recognition analyzes two-dimensional data, namely, images of cursive script that are recorded either using scanners or using cameras. Although there are exceptions, today the standard approach for processing this kind of data is based on a transformation of the images into a sequential representation by the sliding window approach. Currently, practically all major MM-based HWR systems integrate modules that explicitly transform (word) images into one-dimensional sequential data prior to recognition. Subsequently, the majority of these systems performs segmentation free recognition. To sum up, the segmentation-free paradigm for building the writing model can be considered the most successful approach to date for offline handwritten text recognition.

6.2.2 “Simple structure – lots of parameters”

Similar to alternative application domains of hidden Markov models (cf. speech recognition tasks or bioinformatics applications) model topologies of HMMs that are successfully applied for robust handwriting recognition tend to have simple structures. The majority of writing models is based on classical linear left–right (Roman) / right–left (Arabic) HMMs where every state is connected to itself (self-transition) and to its immediate neighbor in writing direction only. A slightly modified version—the Bakis architecture—also introduces skips of adjacent states. Apparently, complicated model architectures are used only very rarely. Instead of focusing on complex model architectures, researchers rather concentrate on the estimation of lots of parameters for emission modeling. Most notably, the multi-lingual HWR system by BBN consists of HMMs with simple Bakis topology but

with (roughly estimated) 150k mixtures. The latter allows for very robust recognition of unconstrained handwriting.

6.2.3 Integration of language models

The use of language models is more and more becoming the standard for general handwriting recognition. This survey shows that this is especially the case for offline HWR. In fact, five of the seven recognition systems reviewed integrate language models for effectively restricting hypotheses of recognized sequences of characters or words that are generated by the particular writing models. Furthermore, it can be concluded that n -gram models represent the state of the art for statistical language modeling. However, details about the language models actually used (method for smoothing probability estimates, perplexity achieved, integration with the writing model and so forth) are frequently not reported in the literature, which sometimes complicates comparability and potential adoption.

6.2.4 Use of classifier ensembles

The use of hybrid classification techniques for handwriting recognition has a rather long history. Various approaches have been proposed to integrate, for example, artificial neural networks and hidden Markov models into handwriting recognition frameworks. Recently, this concept of combining multiple classifiers has been studied more extensively and it has been generalized with rather encouraging results. The integration of Markov models (both HMMs and n -gram models) into Ensemble classification approaches thus represents another recent methodological trend. Especially, for the analysis of unconstrained handwriting with potentially numerous different writing styles or for huge vocabularies, parallelization to multiple diverse classifiers is promising.

6.2.5 Multi-linguality script-independency

Another recent trend clearly indicates the maturity that the research field of offline handwriting recognition meanwhile has reached. After more or less explicitly focusing on fundamental recognition problems (like, for example, how to treat two-dimensional data using statistical models that are, without substantial modification, suitable for 1-D data only), the community now has also turned toward other practical application problems like multi-linguality and script-independency. In fact, some of the reviewed systems have actually been used for the recognition of multiple different scripts and languages. The Siemens recognizer—originally developed for the recognition of Roman scripts—has been adapted to process Arabic handwriting, apparently requiring only minor

modifications. The BBN recognizer has even been designed explicitly to cope with multiple languages and script types. Especially, for a successful commercial application, script-independency and multi-linguality can be considered to be important properties of HWR systems.

6.2.6 Camera-based HWR

Offline handwriting recognition is usually performed on images of handwritten data that have been recorded by means of scanners. As one prominent example, for postal automation applications large, sophisticated scanning appliances have been installed in major logistics centers of postal companies aiming at optimal image quality. Recently, the general field of document analysis has been extended toward camera based input. The reason for this is the almost ubiquitous availability of cameras integrated into the latest generations of cell phone technology that allows spontaneous image capturing. Moreover, the emergence of new application domains like automatic whiteboard reading for smart conference rooms requires more flexible input devices than bulky scanners. In the last few years, certain approaches for both online and offline HWR based on camera images have been proposed and have already been applied successfully. Although camera based input cannot yet be considered an actual trend for offline handwriting recognition it corresponds, however, to a very promising and at the same time challenging application field.

6.2.7 Universal toolkits

Reconsidering the survey of handwriting recognition systems, it can be concluded that most of those systems that are applied to practical tasks in both industrial and academic context are based on more general Markov model frameworks. Often these toolkits originally were developed for their use in alternative application domains—most prominently, e.g., for automatic speech recognition. When analyzing the general architecture of the systems, it becomes clear that the recognizers from BBN, IAM, TUM, and TUD are principally comparable. They follow—each more or less strictly—the classic approach of Markov-model-based recognition for sequential data.

Unfortunately, so far most Markov-model-based HWR systems are not publicly available including all the parameters and configurations necessary to set up handwriting recognizers from scratch. At least in some cases (Siemens, BBN), this might be reasoned by commercial interests. There are, however, publicly available HMM toolkits (HTK, ghmm, ESMERALDA, etc.) that can freely be (and in fact have been) used at least for non-commercial research.

6.3 Some remarks on reporting results

Ideally, other researchers should be able to reproduce results achieved by a proposed method. Therefore, when reporting results, besides the understandable desire to show the advantages of one's own method over others, it should be a primary goal to be as precise as necessary in documenting the parameters of the experiments. Such documentation comprises the datasets used as well as the parametrization and configuration of the recognizer. In the following, we will discuss the problems related to these aspects and give some recommendations for producing "valuable" results.

6.3.1 Use well-defined benchmarks!

Comparability of results is only possible when working on datasets that are rather widely used in the research community. This almost immediately implies that this data either needs to be publicly available or at least available for reasonable costs. Unfortunately, the field of handwriting recognition is extremely diverse with respect to tasks considered—e.g., touching numerals versus handwritten text, Roman versus Chinese script. Therefore, there probably will never be the *universal* handwriting recognition benchmark. Within the different sub-disciplines one can, however, observe a tendency of researchers working on well-defined and well-known datasets—as, e.g., the results published for the recognition systems described in the previous section. Fortunately, fewer and fewer groups today still publish results on crudely defined or proprietary data.

However, a dataset does not make a benchmark. For example, the quite common practice to just subdivide the data "randomly" into training and test sets cannot be reproduced by anybody and makes the largely optimistic assumption that the complexity and variability in the data is homogeneous. Therefore, fixed subdivisions into training, validation, and test data should be used¹⁰ as they are sometimes already pre-defined for certain databases (cf., e.g., IAM-DB).

For a complete handwriting recognition benchmark, now only the inventory of recognition units is missing. On the character level, this might seem obviously defined. However, punctuation symbols and other special characters—just to give two examples—can augment a character set considerably and make a big difference in recognition accuracy. On the word level, the problem becomes even more severe as it is not clearly defined what a "word" is supposed to be. For example, "Mr." seems to rightfully be a word including the period. However, at the end of a sentence, the final word and the punctuation symbol would clearly be considered separate units. This situation becomes more complicated if numerals

and other special characters come into play. Obviously, these problems can be completely avoided on presegmented data—on IFN/ENIT, for example, it is sufficient to map the word image to the zip code of the corresponding Tunisian town. As, however, the real challenges lie beyond such tasks, it will be important for the definition of future benchmarks to address these problems properly.

6.3.2 Give all necessary technical details!

Though the HMM technology constitutes a rather well-defined modeling and recognition paradigm, the devil is still in the details. Every reasonable HMM for HWR that is worth reporting on will contain quite a number of free parameters and configuration options that usually decide about either success or failure on a certain task. Therefore, it is absolutely necessary to report what basic type of HMM is used (discrete, semicontinuous/tied-mixture, or continuous mixture). For HMMs based on mixture densities, it is furthermore important to tell whether the Gaussians used have full or only diagonal covariances (or use some even more sophisticated method of parameter tying). Additionally, the basic model topology (linear, Bakis, or hand-crafted), the type of elementary units (strokes, characters, or words), and the number of states used per unit are of fundamental importance. Though the algorithms for creating and decoding HMMs are pretty much standard, it is better to also state explicitly how model parameters are estimated and how the model is decoded. The latter information is crucial as soon as a language model is incorporated into the overall recognition model.

For the language model part, which will most probably be an n -gram model, the type of smoothing applied (e.g., absolute discounting in combination with either backing-off or interpolation) and the perplexity achieved on the test set considered need to be given. Especially, the model's perplexity is a crucial figure as without it there is absolutely no chance to either judge the achieved model quality or compare results reported on otherwise identical tasks but using different language models.

6.3.3 Use hard tasks!

It is a quite common misconception that high accuracy figures or close to zero error rates are good *per se* without considering the underlying task. There are two main problems with such figures. First, when operating in regions of ninety-nine-point-something accuracies an improvement in the second decimal is hardly statistically significant given the limited size of current databases. Therefore, improvements reported should always be checked for being significant at a level of at least 95%. All other minor changes in the results achieved are just noise and not worth reporting. The second and more

¹⁰ Though fixed subdivisions are good, there is no reason to prove one's creativity by defining yet another one without giving clear reasons for it.

severe problem of close-to-zero error rates is that any performance figure approaching its theoretical limit—i.e., 100% accuracy or 0% error rate—renders the underlying task useless. In such situations, it is completely clear that the task considered has become too easy for the recognition methodology used. Therefore, as soon as improvements become marginal or even potentially impossible it is about time to move on to a more challenging experimental setup. The best thing to do to convince the reader that reporting on error rates ten times as high as in your previous publication is to make clear with a rigorous account of all details of the task addressed that you are really working on a hard and, therefore, interesting problem.

6.4 Future challenges

For the automatic recognition of handwriting, the application of Markovian models has become a standard procedure. In our review of the field that covered both theoretical aspects as well as practical and integration issues, we identified several mature recognition systems that are being used for non-trivial recognition tasks in both industrial and academic contexts. According to the literature, quite diverse research directions are still being explored and standard procedures for building Markov-model-based offline handwriting recognizers could not be established so far. However, some trends toward unified approaches can be identified as, for example, the quite widely used sliding-window approach for obtaining sequential representations from images of handwriting.

Although substantial progress has already been made toward the ultimate goal of automatic reading systems for handwritten script, challenging problems still need to be tackled. The most prominent one, which can be considered a universal problem of any area of statistical pattern recognition, is the problem of limited data. Though some notable data collection efforts exist and some quite substantial datasets have also been made publicly available already, these sample sets are still far too small—and probably will be for the foreseeable future—for training a statistical recognizer that might be able to show close to human performance in automatically reading handwritten script. Consequently, robust parameter estimation on limited sample sets remains an open research issue for MM-based handwriting recognizers. Extensions of classical model adaptation techniques or methods for discriminative training might provide the ingredients for solutions to this fundamental problem.

Major challenges can also be identified for the feature extraction process. Although the sliding-window technique has become a quasi-standard, it has serious drawbacks, too. Most prominently, the dynamics of the process of handwriting is captured to a quite limited extent only. More importantly, however, there is no real biological justification for a small analysis window that is moved along the text-line as it

is, e.g., for acoustic signal analysis in the speech recognition domain. From a theoretical point of view, holistic recognition approaches match the reading process performed by humans more appropriately. Current approaches are, however, not yet as effective as the classical sliding-window-based techniques. Hence, further research on the convergence of methods is necessary.

Generally, the features as they are currently used for handwriting recognition applications are purely heuristic. In contrast to other domains, there is no clear theory behind them, which justifies the feature representation used based on some underlying domain-specific knowledge about the signal data (the script images) and its origin (handwriting performed by a human). Especially, for more challenging recognition tasks aiming at the analysis of truly unconstrained handwriting with virtually no lexicon restrictions, further research needs to be devoted to alternative feature representations.

Finally, there is still only a limited overlap between methods applied to OCR and those used for handwriting recognition, which is understandable as OCR (as the easier problem) can be solved without putting the same effort into normalization, feature extraction, and modeling as is currently done in the handwriting recognition community. However, as demonstrated, for example, by the BBN recognizer a convergence of methods is quite promising. HWR systems could, for example, in the future also be trained on machine printed text and later only be adapted to handwritten data. The problem of limited datasets for handwritten script would then be largely alleviated.

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

1. Arica, N., Yarman-Vural, F.T.: One-dimensional representation of two-dimensional information for HMM based handwriting recognition. *Pattern Recogn. Lett.* **21**, 583–592 (2000)
2. Arica, N., Yarman-Vural, F.T.: An overview of character recognition focused on off-line handwriting. *IEEE Trans. Syst. Man Cybern. C Appl.* **31**(2), 216–232 (2001)
3. Arica, N., Yarman-Vural, F.T.: Optical character recognition for cursive handwriting. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(6), 801–813 (2002)
4. Austin, S., Schwartz, R., Placeway, P.: The forward-backward search algorithm. In: *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 697–700. Toronto (1991)
5. Baum, L., Petrie, T.: Statistical inference for probabilistic functions of finite state markov chains. *Ann. Math. Stat.* **37**, 1554–1563 (1966)
6. Baum, L., Petrie, T., Soules, G., Weiss, N.: A maximization technique occurring in the statistical analysis of probabilistic

- functions of Markov chains. *Ann. Math. Stat.* **41**, 164–171 (1970)
7. Benouareth, A., Ennaji, A., Sellami, M.: Semicontinuous HMMs with explicit state duration applied to Arabic handwritten word recognition. In: *Proceedings of the International Workshop on Frontiers in Handwriting Recognition*, pp. 97–102. La Baule, France (2006)
 8. Bertolami, R., Bunke, H.: Multiple handwritten text line recognition systems derived from specific integration of a language model. In: *Proceedings of the International Conference on Document Analysis and Recognition*, vol. 1, pp. 521–525. Seoul, Korea (2005)
 9. Bertolami, R., Uchida, S., Zimmermann, M., Bunke, H.: Non-uniform slant correction for handwritten text line recognition. In: *Proceedings of the International Conference on Document Analysis and Recognition*, vol. 1, pp. 18–22. Curitiba, Brazil (2007)
 10. Besner, D., Humphreys, G.W. (eds.): *Basic Processes in Reading: Visual Word Recognition*. Lawrence Erlbaum Associates, Hillsdale (1991)
 11. Bilmes, J.: What HMMs can't do: a graphical model perspective. In: *Beyond HMM: Workshop on Statistical Modeling Approach for Speech Recognition*. Kyoto, Japan (2004). ATR Invited Paper and Lecture
 12. Bocchieri, E., Mak, B.K.W.: Subspace distribution clustering hidden Markov model. *IEEE Trans. Speech Audio Process.* **9**(2), 264–275 (2001)
 13. Bozinovic, R.M., Srihari, S.N.: Off-line cursive script word recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **11**(1), 69–83 (1989)
 14. Brakensiek, A., Rigoll, G.: A comparison of character n-grams and dictionaries used for script recognition. In: *Proceedings of the International Conference on Document Analysis and Recognition*, pp. 241–245. Seattle (2001)
 15. Brakensiek, A., Rigoll, G.: Combination of multiple classifiers for handwritten word recognition. In: *Proceedings of the International Workshop on Frontiers in Handwriting Recognition*, pp. 117–122. Niagara on the Lake, Canada (2002)
 16. Brakensiek, A., Rigoll, G.: Handwritten address recognition using hidden Markov models. In: Dengel, A., Junker, M., Weisbecker, A. (eds.) *Reading and Learning—Adaptive Content Recognition*, Lecture Notes in Computer Science, vol. 2956, pp. 103–122. Springer, Berlin (2004)
 17. Brakensiek, A., Rottland, J., Rigoll, G.: Handwritten address recognition with open vocabulary using character n-grams. In: *Proceedings of the International Workshop on Frontiers in Handwriting Recognition*, pp. 357–362. Niagara on the Lake, Canada (2002)
 18. Brakensiek, A., Rottland, J., Rigoll, G.: Confidence measures for an address reading system. In: *Proceedings of the International Conference on Document Analysis and Recognition*, vol. 1, pp. 294–298. Edinburgh (2003)
 19. Brakensiek, A., Rottland, J., Wallhoff, F., Rigoll, G.: Adaptation of an address reading system to local mail streams. In: *Proceedings of the International Conference on Document Analysis and Recognition*, pp. 872–876. Seattle (2001)
 20. Brakensiek, A., Willett, D., Rigoll, G.: Improved degraded document recognition with hybrid modeling techniques and character n-grams. In: *Proceedings of the International Conference on Pattern Recognition*, vol. 4, pp. 438–441. Barcelona (2000)
 21. Britto, A.D.S., Sabourin, R., Bortolozzi, F., Suen, C.Y.: A two-stage HMM-based system for recognizing handwritten numeral strings. In: *Proceedings of the International Conference on Document Analysis and Recognition*, pp. 396–400. Seattle (2001)
 22. Bunke, H.: Recognition of cursive Roman handwriting—Past, present and future. In: *Proceedings of the International Conference on Document Analysis and Recognition*, vol. 1, pp. 448–459 (2003)
 23. Bunke, H., Roth, M., Schukat-Talamazzini, E.G.: Off-line cursive handwriting recognition using hidden Markov models. *Pattern Recogn.* **9**(9), 1399–1413 (1995)
 24. Caesar, T., Gloger, J.M., Mandler, E.: Preprocessing and feature extraction for a handwriting recognition system. In: *Proceedings of the International Conference on Document Analysis and Recognition*, pp. 408–411. Tsukuba Science City, Japan (1993)
 25. Cai, J., Liu, Z.Q.: Markov random field models for handwritten word recognition. In: *Proceedings of the International Conference Intelligent Processing Systems (ICIPS)*, vol. 2, pp. 1400–1404. IEEE, Beijing (1997)
 26. Cai, J., Liu, Z.Q.: Off-line unconstrained handwritten word recognition. *Int. J. Pattern Recogn. Artif. Intell.* **14**(3), 259–280 (2000)
 27. Chen, S.F., Goodman, J.: An empirical study of smoothing techniques for language modeling. *Comput. Speech Lang.* **13**, 359–394 (1999)
 28. Cho, W., Lee, S.W., Kim, J.H.: Modeling and recognition of cursive words with hidden Markov models. *Pattern Recogn.* **28**(12), 1941–1953 (1995)
 29. Choisy, C.: Dynamic handwritten keyword spotting based on the NSHP-HMM. In: *Proceedings of the International Conference on Document Analysis and Recognition*, vol. 1, pp. 242–246. Curitiba, Brazil (2007)
 30. Coetzer, J., Herbst, B.M., du Preez, J.A.: Offline signature verification using the discrete Radon transform and a hidden Markov models. *EURASIP J. Appl. Signal Process.* **4**, 559–571 (2004)
 31. Coetzer, J., Herbst, B.M., du Preez, J.A.: Off-line signature verification: a comparison between human and machine performance. In: *Proceedings of the International Workshop on Frontiers in Handwriting Recognition*, pp. 481–486. La Baule, France (2006)
 32. Colthurst, T., Kimball, O., Richardson, F., Shu, H., Wooters, C., Iyer, R., Gish, H.: The 2000 BBN Byblos LVCSR system. In: *2000 Speech Transcription Workshop*. Maryland (2000)
 33. Daniels, P.T., Bright, W. (eds.): *The World's Writing Systems*. Oxford University Press, New York (1996)
 34. Davis, R.: Magic paper: sketch-understanding research. *IEEE Comput.* **40**(9), 34–41 (2007)
 35. Decerbo, M., MacRostie, E., Natarajan, P.: The BBN Byblos Pashto OCR system. In: *Proceedings of the 1st ACM Workshop on Hardcopy Document Processing*, pp. 29–32. ACM New York, NY, USA, Washington, DC, USA (2004)
 36. Dehghan, M., Faez, K., Ahmadi, M., Shridhar, M.: Off-line unconstrained Farsi handwritten word recognition using fuzzy vector quantization and hidden Markov word models. In: *Proceedings of the International Conference on Pattern Recognition*, vol. 2, pp. 351–354. Barcelona (2000)
 37. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Stat. Soc. Ser. B* **39**(1), 1–22 (1977)
 38. Ding, Y., Kimura, F., Miyake, Y., Shridhar, M.: Accuracy improvement of slant estimation for handwritten words. In: *Proceedings of the International Conference on Pattern Recognition*, vol. 4, pp. 527–530. Barcelona (2000)
 39. Duda, R.O., Hart, P.E., Stork, D.G.: *Pattern Classification*. 2nd edn. Wiley Interscience, New York (2000)
 40. El Abed, H., Märgner, V.: Comparison of different preprocessing and feature extraction methods for offline recognition of handwritten Arabic words. In: *Proceedings of the International Conference on Document Analysis and Recognition*, vol. 2, pp. 974–978. Curitiba, Brazil (2007)

41. El-Hajj, R., Likforman-Sulem, L., Mokbel, C.: Arabic handwriting recognition using baseline dependant features and hidden Markov modeling. In: *Proceedings of the International Conference on Document Analysis and Recognition*, vol. 2, pp. 893–897. Seoul, Korea (2005)
42. El-Hajj, R., Likforman-Sulem, L., Mokbel, C.: Combining slanted-frame classifiers for improved HMM-based Arabic handwriting recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **31**(7) (2009)
43. El-Hajj, R., Mokbel, C., Likforman-Sulem, L.: Combination of HMM-based classifiers for recognition of Arabic handwritten words. In: *Proceedings of the International Conference on Document Analysis and Recognition*, vol. 2, pp. 959–963. Curitiba, Brazil (2007)
44. El-Yacoubi, A., Gilloux, M., Sabourin, R., Suen, C.Y.: An HMM-based approach for off-line unconstrained handwritten word modeling and recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **21**(8), 752–760 (1999)
45. Feng, B., Ding, X., Wu, Y.: Chinese handwriting recognition using hidden Markov models. In: *Proceedings of the International Conference on Pattern Recognition*, vol. 3, pp. 212–215. Québec (2002)
46. Feng, S., Manmatha, R., McCallum, A.: Exploring the use of conditional random field models and HMMs for historical handwritten document recognition. In: *2nd International Conference on Document Image Analysis for Libraries (DIAL)*, pp. 8–37 (2006)
47. Fink, G.A.: *Markov Models for Pattern Recognition—From Theory to Applications*. Springer, Heidelberg (2008)
48. Fink, G.A., Plötz, T.: On appearance-based feature extraction methods for writer-independent handwritten text recognition. In: *Proceedings of the International Conference on Document Analysis and Recognition*, vol. 2, pp. 1070–1074. IEEE, Seoul, Korea (2005)
49. Fink, G.A., Plötz, T.: Unsupervised estimation of writing style models for improved unconstrained off-line handwriting recognition. In: *Proceedings of the 10th International Workshop on Frontiers in Handwriting Recognition*. IEEE, La Baule, France (2006)
50. Fink, G.A., Plötz, T.: Tutorial on Markov models for handwriting recognition. In: *Proceedings of the International Conference on Document Analysis and Recognition*. Curitiba, Brazil (2007)
51. Fink, G.A., Plötz, T.: Developing pattern recognition systems based on Markov models: the ESMEALDA framework. *Pattern Recogn. Image Anal.* **18**(2), 207–215 (2008)
52. Fiscus, J.: A post-processing system to yield reduced word error rates: recognizer output voting error reduction. In: Furui, S., Huang, B.H., Chu, W. (eds.) *Proceedings of the Workshop on Automatic Speech Recognition and Understanding*, pp. 347–352. Santa Barbara (1997)
53. Fujisawa, H.: Robustness design of industrial strength recognition systems. In: Chaudhuri, B. (ed.) *Digital Document Processing: Major Directions and Recent Advances*, pp. 185–212. Springer, London (2007)
54. Fujisawa, H.: Forty years of research in character and document recognition—an industrial perspective. *Pattern Recogn.* **41**, 2435–2446 (2008)
55. Gader, P.D., Keller, J.M., Krishnapuram, R., Chiang, J.H., Mohamed, M.A.: Neuronal and fuzzy methods in handwriting recognition. *Computer* **2**, 79–86 (1997)
56. Gauthier, N., Artières, T., Dorizzi, B., Ballinari, P.: Strategies for combining on-line and off-line information in an on-line handwriting recognition system. In: *Proceedings of the International Conference on Document Analysis and Recognition*, pp. 412–416. Seattle (2001)
57. Ge, Y., Huo, Q.: A study on the use of CDHMM for large vocabulary offline recognition of handwritten Chinese characters. In: *Proceedings of the International Workshop on Frontiers in Handwriting Recognition*, pp. 334–338. Niagara on the Lake, Canada (2002)
58. Grandidier, F., Sabourin, R., Suen, C.Y.: Integration of contextual information in handwriting recognition systems. In: *Proceedings of the International Conference on Document Analysis and Recognition*, vol. 2, pp. 1252–1256. Edinburgh (2003)
59. Günter, S., Bunke, H.: A new combination scheme for HMM-based classifiers and its application to handwriting recognition. In: *Proceedings of the International Conference on Pattern Recognition*, vol. 2, pp. 332–337. Québec (2002)
60. Günter, S., Bunke, H.: Optimizing the number of states, training iterations and Gaussians in an HMM-based handwritten word recognizer. In: *Proceedings of the International Conference on Document Analysis and Recognition*, vol. 1, pp. 472–476. Edinburgh (2003)
61. Günter, S., Bunke, H.: HMM-based handwritten word recognition: on the optimization of the number of states, training iterations and Gaussian components. *Pattern Recogn.* **37**, 2069–2079 (2004)
62. Huang, X.D., Ariki, Y., Jack, M.A.: *Hidden Markov Models for Speech Recognition*. No. 7 in Information Technology Series. Edinburgh University Press, Edinburgh (1990)
63. Huang, X.D., Jack, M.A.: Semicontinuous hidden Markov models for speech signals. *Comput. Speech Lang.* **3**(3), 239–251 (1989)
64. Justino, E.J.R., El Yacoubi, A., Bortolozzi, F., Sabourin, R.: An off-line signature verification system using hidden Markov model and cross-validation. In: *Proceedings of the XIII Brazilian Symposium on Computer Graphics and Image Processing*, pp. 105–112 (2000)
65. Kaltenmeier, A., Caesar, T., Gloger, J.M., Mandler, E.: Sophisticated topology of hidden Markov models for cursive script recognition. In: *Proceedings of the International Conference on Document Analysis and Recognition*, pp. 139–142 (1993)
66. Ko, A.H.R., Sabourin, R., de Souza Britto, A. Jr.: Ensemble of HMM classifiers based on the clustering validity index for a handwritten numeral recognizer. *Pattern Anal. Appl. J.* **12**(1), 21–35 (2009)
67. Koerich, A.L., Britto, A.S., de Oliveira, L.E.S., Sabourin, R.: Fusing high- and low-level features for handwritten word recognition. In: *Proceedings of the International Workshop on Frontiers in Handwriting Recognition*, pp. 151–156. La Baule, France (2006)
68. Koerich, A.L., Leydier, Y., Sabourin, R., Suen, C.Y.: A hybrid large vocabulary handwritten word recognition system using neuronal networks with hidden Markov models. In: *Proceedings of the International Workshop on Frontiers in Handwriting Recognition*, pp. 99–104. Niagara on the Lake, Canada (2002)
69. Kundu, A., He, Y., Bahl, P.: Recognition of handwritten words: first and second order hidden Markov model based approach. *Pattern Recogn.* **22**(3), 283–297 (1989)
70. Kundu, A., Hines, T., Phillips, J., Huyck, B.D., Van Guilder, L.C.: Arabic handwriting recognition using variable duration HMM. In: *Proceedings of the International Conference on Document Analysis and Recognition*, vol. 2, pp. 644–648. Curitiba, Brazil (2007)
71. Li, Y., Zheng, Y., Doermann, D., Jaeger, S.: Script-independent text line segmentation in freestyle handwritten documents. *IEEE Trans. Pattern Anal. Mach. Intell.* **30**(8), 1313–1329 (2008)
72. Likforman-Sulem, L., Zahour, A., Taconet, B.: Text line segmentation of historical documents: a survey. *Int. J. Doc. Anal. Recogn.* **9**(2), 123–138 (2007)

73. Liwicki, M., Bunke, H.: Enhancing training data for handwritten recognition of whiteboard notes with samples from a different database. In: Proceedings of the International Conference on Document Analysis and Recognition, vol. 2, pp. 550–554. Seoul, Korea (2005)
74. Liwicki, M., Bunke, H.: Handwriting recognition of whiteboard notes. In: Proceedings of the 12th Conference of the International Graphonomics Society, pp. 118–122 (2005)
75. Liwicki, M., Bunke, H.: IAM-OnDB—an on-line English sentence database acquired from handwritten text on a whiteboard. In: Proceedings of the International Conference on Document Analysis and Recognition, vol. 2, pp. 956–961. Seoul, Korea (2005)
76. Lorigo, L.M., Govindaraju, V.: Offline Arabic handwriting recognition: a survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(5), 712–724 (2006)
77. Lowerre, B.T.: The HARP speech recognition system. Ph.D. thesis, Carnegie-Mellon University, Department of Computer Science, Pittsburg (1976)
78. Lu, Z., Schwartz, R., Raphael, C.: Script-independent, HMM-based text line finding for OCR. In: Proceedings of the International Conference on Pattern Recognition, pp. 551–554. Barcelona, Spain (2000)
79. Madhvanath, S., Kim, G., Govindaraju, V.: Chaincode contour processing for handwritten word recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **21**(9), 928–932 (1999)
80. Mao, S., Rosenfeld, A., Kanungo, T.: Document structure analysis algorithms: a literature survey. In: Proceedings of the SPIE Electronic Imaging, pp. 197–207 (2003)
81. Märgner, V., El-Abed, H.: ICDAR 2005—Arabic handwriting recognition competition. In: Proceedings of the International Conference on Document Analysis and Recognition. Seoul, Korea (2005)
82. Märgner, V., El-Abed, H.: ICDAR 2007—Arabic handwriting recognition competition. In: Proceedings of the International Conference on Document Analysis and Recognition (2007)
83. Markov, A.A.: Примѣръ статистическаго изслѣдованія надъ текстомъ „Евгенія Онѣгина” иллюстрирующей связиспытанійвъ цѣль (Example of statistical investigations of the text of „Eugen Onegin”, which demonstrates the connection of events in a chain). In: Извѣстія Императорской Академіи Наукъ (Bulletin de l'Académie Impériale des Sciences de St.-Petersbourg), pp. 153–162. Sankt-Petersburg (1913, in Russian)
84. Marti, U.V., Bunke, H.: Handwritten sentence recognition. In: Proceedings of the International Conference on Pattern Recognition, vol. 3, pp. 467–470. Barcelona (2000)
85. Marti, U.V., Bunke, H.: On the influence of vocabulary size and language models in unconstrained handwritten text recognition. In: Proceedings of the International Conference on Document Analysis and Recognition, pp. 260–265. Seattle (2001)
86. Marti, U.V., Bunke, H.: Text line segmentation and word recognition in a system for general writer independent handwriting recognition. In: Proceedings of the International Conference on Document Analysis and Recognition, pp. 159–163. Seattle (2001)
87. Marti, U.V., Bunke, H.: Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition systems. *Int. J. Pattern Recogn. Artif. Intell.* **15**(1), 65–90 (2001)
88. Marti, U.V., Bunke, H.: The IAM-database: an English sentence database for offline handwriting recognition. *Int. J. Doc. Anal. Recogn.* **5**(1), 39–46 (2002)
89. Menasri, F., Vincent, N., Augustin, E., Cheriet, M.: Shape-based alphabet for off-line Arabic handwriting recognition. In: Proceedings of the International Conference on Document Analysis and Recognition, vol. 2, pp. 969–973. Curitiba, Brazil (2007)
90. Miletzki, U., Bayer, T., Schäfer, H.: Continuous learning systems: postal address readers with built-in learning capability. In: Proceedings of the International Conference on Document Analysis and Recognition, pp. 329–332. Bangalore, India (1999)
91. Morita, M., El Yacoubi, A., Sabourin, R., Bortolozzi, F., Suen, C.Y.: Handwritten month word recognition on Brazilian bank cheques. In: Proceedings of the International Conference on Document Analysis and Recognition, pp. 972–976. Seattle (2001)
92. Morita, M., Sabourin, R., Bortolozzi, F., Suen, C.Y.: Segmentation and recognition of handwritten dates. In: Proceedings of the International Workshop on Frontiers in Handwriting Recognition, pp. 105–110. Niagara on the Lake, Canada (2002)
93. Natarajan, P., Lu, Z., Schwartz, R., Bazzi, I., Makhoul, J.: Multilingual machine printed OCR. *Int. J. Pattern Recog. Artif. Intell.* **15**(1), 43–63 (2001)
94. Natarajan, P., Saleem, S., Prasad, R., MacRostie, E., Subramanian, K.: Multi-lingual offline handwriting recognition using hidden Markov models: a script-independent approach. In: Doermann, D.S., Jaeger, S. (eds.) Arabic and Chinese Handwriting Recognition: SACH 2006 Selected Papers, Lecture Notes in Computer Science, vol. 4768, pp. 231–250. Springer, Berlin (2008)
95. Nopsuwanchai, R., Biem, A., Clocksin, W.F.: Maximization of mutual information for offline Thai handwriting recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(8), 1347–1351 (2006)
96. Pechwitz, M., Maddouri, S.S., Märgner, V., Ellouze, N., Amiri, H.: IFN/ENIT-database of handwritten Arabic words. In: Proceedings of the 7th Colloque International Francophone sur l'Écrit et le Document. Hammamet, Tunis (2002)
97. Pechwitz, M., Märgner, V.: HMM based approach for handwritten Arabic word recognition using the IFN/ENIT-database. In: Proceedings of the International Conference on Document Analysis and Recognition, vol. 2, pp. 890–894. Edinburgh (2003)
98. Pittman, J.A.: Handwriting recognition: tablet PC text input. *IEEE Comput.* **40**(9), 49–54 (2007)
99. Plamondon, R., Srihari, S.N.: On-line and off-line handwriting recognition: a comprehensive survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(1), 63–84 (2000)
100. Plötz, T., Thurau, C., Fink, G.A.: Camera-based whiteboard reading: new approaches to a challenging task. In: Proceedings of the 11th International Conference on Frontiers in Handwriting Recognition, pp. 385–390. Montreal, Canada (2008)
101. Rigoll, G., Kosmala, A., Rottland, J., Neukirchen, C.: A comparison between continuous and discrete density hidden Markov models for cursive handwriting recognition. In: Proceedings of the International Conference on Pattern Recognition, vol. 2, pp. 205–209. Vienna (1996)
102. Schambach, M.P.: Determination of the number of writing variants with an HMM based cursive word recognition system. In: Proceedings of the International Conference on Document Analysis and Recognition, vol. 1, pp. 119–123. Edinburgh (2003)
103. Schambach, M.P.: Fast script word recognition with very large vocabulary. In: Proceedings of the International Conference on Document Analysis and Recognition, vol. 1, pp. 9–13. Seoul, Korea (2005)
104. Schambach, M.P., Rottland, J., Alary, T.: How to convert a Latin handwriting recognition system to Arabic. In: Proceedings of the International Conference on Document Analysis and Recognition (2008)
105. Schwartz, R., LaPre, C., Makhoul, J., Raphael, C., Zhao, Y.: Language-independent OCR using a continuous speech recog-

- dition system. In: Proceedings of the International Conference on Pattern Recognition, vol. 3, pp. 99–103. Vienna, Austria (1996)
106. Senior, A.W., Robinson, A.J.: An off-line cursive handwriting recognition system. *IEEE Trans. Pattern Anal. Mach. Intell.* **20**(3), 309–321 (1998)
 107. Starner, T., Makhoul, J., Schwartz, R., Chou, G.: On-line cursive handwriting recognition using speech recognition methods. In: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, vol. 5, pp. 125–128. Adelaide (1994)
 108. Steinherz, T., Rivlin, E., Intrator, N.: Offline cursive script word recognition—a survey. *Int. J. Doc. Anal. Recogn.* **2**, 90–110 (1999)
 109. Su, T.H., Zhang, T.W., Huang, H.J., Zhou, Y.: HMM-based recognizer with segmentation-free strategy for unconstrained Chinese handwriting text. In: Proceedings of the International Conference on Document Analysis and Recognition, vol. 1, pp. 133–137. Curitiba, Brazil (2007)
 110. Tay, Y.H., Pierre-Michel, L., Khalid, M., Knerr, S., Virad-Gaudin, C.: An analytical handwritten word recognition system with word-level discriminant training. In: Proceedings of the International Conference on Document Analysis and Recognition, pp. 726–730. Seattle (2001)
 111. Toselli, A.H., Juan, A., Vidal, E.: Spontaneous handwriting recognition and classification. In: Proceedings of the International Conference on Pattern Recognition, vol. 1, pp. 433–436. Cambridge, UK (2004)
 112. Touj, S.M., Ben Amara, N.E., Amiri, H.: A hybrid approach for off-line Arabic handwriting recognition based on a planar hidden Markov modeling. In: Proceedings of the International Conference on Document Analysis and Recognition, vol. 2, pp. 964–968. Curitiba, Brazil (2007)
 113. Trier, O.D., Taxt, T.: Evaluation of binarization methods for document images. *IEEE Trans. Pattern Anal. Mach. Intell.* **17**(3), 312–315 (1995)
 114. Vajda, S., Belaïd, A.: Structural information implant in a context based segmentation-free HMM handwritten word recognition system for Latin and Bangla scripts. In: Proceedings of the International Conference on Document Analysis and Recognition, vol. 2, pp. 1126–1130. Seoul, Korea (2005)
 115. Vinciarelli, A.: A survey on off-line cursive word recognition. *Pattern Recogn.* **35**, 1433–1446 (2002)
 116. Vinciarelli, A., Bengio, S.: Writer adaptation techniques in off-line cursive word recognition. In: Proceedings of the International Workshop on Frontiers in Handwriting Recognition, pp. 287–291. Niagara on the Lake, Canada (2002)
 117. Vinciarelli, A., Bengio, S., Bunke, H.: Offline recognition of unconstrained handwritten texts using HMMs and statistical language models. *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(6), 709–720 (2004)
 118. Vinciarelli, A., Luettin, J.: Off-line cursive script recognition based on continuous density HMM. In: Proceedings of the International Workshop on Frontiers in Handwriting Recognition, pp. 493–498 (2000)
 119. Vinciarelli, A., Luettin, J.: A new normalization technique for cursive handwritten words. *Pattern Recogn. Lett.* **22**(9), 1043–1050 (2001)
 120. Viterbi, A.: Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inf. Theory* **13**, 260–269 (1967)
 121. Wang, W., Brakensiek, A., Kosmala, A., Rigoll, G.: Multi-branch and two-pass HMM modeling approaches for off-line cursive handwriting recognition. In: Proceedings of the International Conference on Document Analysis and Recognition, pp. 231–235. Seattle (2001)
 122. Wang, W., Brakensiek, A., Rigoll, G.: Combining HMM-based two-pass classifiers for off-line word recognition. In: Proceedings of the International Conference on Pattern Recognition, vol. 3, pp. 151–154. Québec (2002)
 123. Wienecke, M., Fink, G.A., Sagerer, G.: Experiments in unconstrained offline handwritten text recognition. In: Proceedings of the 8th International Workshop on Frontiers in Handwriting Recognition. IEEE, Ontario, Canada (2002)
 124. Wienecke, M., Fink, G.A., Sagerer, G.: Toward automatic video-based whiteboard reading. In: Proceedings of the International Conference on Document Analysis and Recognition, pp. 87–91. IEEE, Edinburgh, Scotland (2003)
 125. Wienecke, M., Fink, G.A., Sagerer, G.: Toward automatic video-based whiteboard reading. *Int. J. Doc. Anal. Recogn.* **7**(2–3), 188–200 (2005)
 126. Xu, Q., Kim, J.H., Lam, L., Suen, C.Y.: Recognition of handwritten month words on bank cheques. In: Proceedings of the International Workshop on Frontiers in Handwriting Recognition, pp. 111–116. Niagara on the Lake, Canada (2002)
 127. Xue, H., Govindaraju, V.: Hidden Markov models combining discrete symbols and continuous attributes in handwriting recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(3), 458–462 (2006)
 128. Young, S.: A review of large-vocabulary continuous-speech recognition. *IEEE Signal Process. Mag.* **13**(9), 45–57 (1996)
 129. Zimmermann, M., Bunke, H.: Automatic segmentation of the IAM off-line database for handwritten English text. In: Proceedings of the International Conference on Pattern Recognition, vol. 4, pp. 35–39. Québec (2002)
 130. Zimmermann, M., Bunke, H.: Hidden Markov model length optimization for handwriting recognition systems. In: Proceedings of the International Workshop on Frontiers in Handwriting Recognition, pp. 369–374. Niagara on the Lake, Canada (2002)