Experiments in Unconstrained Offline Handwritten Text Recognition

Markus Wienecke, Gernot A. Fink, Gerhard Sagerer*

Abstract

A system for off-line handwritten text recognition is presented. It is characterized by a segmentation-free approach, i.e. whole lines of text are processed by the recognition module. The methods used for pre-processing, feature extraction, and statistical modelling are described, and several experiments on writer-independent, multiple writer, and single writer handwriting recognition tasks are conducted. Particularly, the incorporation of linear discriminant analysis, allograph character models, and statistical language knowledge is investigated. The evaluation results for lexicon-free handwriting recognition demonstrate the effectiveness of the proposed methods.

1 Introduction

A challenging area of pattern recognition is the recognition of handwritten texts. So far, most of the off-line handwriting recognition systems are applied for reading postal addresses or processing forms like e.g. bank cheques (cf. e.g. [14]). Whereas these systems are limited to isolated characters or isolated words, only few systems exist for the recognition of unconstrained handwritten texts. This is due to the increased complexity of such a task, which is characterized by the absence of character or word boundary information in addition to a large or even unlimited vocabulary. Nevertheless, it is worthwhile to further investigate techniques for recognizing handwritten texts, because the increase of computational power facilitates the use of more complex processing steps.

In this paper a system for lexicon-free off-line handwriting recognition based on Hidden-Markov-Models is presented and several experiments on full English sentence databases are conducted in writer-independent mode and for comparison also in multiple and single writer mode. The methods for pre-processing and feature extraction will be described, and, additionally, more sophisticated techniques like linear discriminant analysis, the use of allograph character models, and statistical language models will be investigated.

In the following section we will give a short review of related work in off-line handwriting recognition. The databases that we use will be introduced in section 3. Afterwards we describe in subsequent sections the pre-processing steps, the methods for feature extraction, and the techniques used for statistical modelling and recognition. Evaluation results will be presented in section 7 in order to demonstrate the effectiveness of the proposed methods.

2 Related Work

Substantial progress could be obtained during the last years in the field of off-line handwriting recognition. In particular, systems for isolated word recognition using a small lexicon employed for postal address or legal amount reading achieve high recognition rates and there is hardly any room for improvement concerning processing speed as well as recognition accuracy [2, 8].

In contrast to these tasks the recognition of unconstrained handwritten texts using a large or even unlimited vocabulary is much more difficult. This is mainly caused by the absence of context knowledge and word segmentation information that is inherently present in systems for isolated word processing. Despite of these difficulties, some systems for unconstrained handwritten text recognition have been developed [1, 9, 11, 18, 15, 17]. These systems mainly differ in the type of features extracted and whether the text-line is segmented into words before the recognition phase or not. Examples of segmentation-based methods using Hidden Markov Models (HMMs) and hybrids of recurrent neural networks and HMMs can be found in [1, 18] and [15], respectively. Whereas in [15] experiments are conducted on a database produced by a single writer, the systems presented in [1, 18] are tested using the data of multiple writers. For an extensive survey of off-line cursive script word recognition see [16].

In order to avoid errors introduced by segmenting the text-line at an early stage, in [9] a segmentation-free method

*Bielefeld University, Faculty of Technology, 33594 Bielefeld, Germany. E-mail: {mwieneck, gernot, sagerer}@techfak.uni-bielefeld.de

In Proc. 8th Int. Workshop on Frontiers in Handwriting Recognition, Ontario, Canada, August 2002. IEEE.

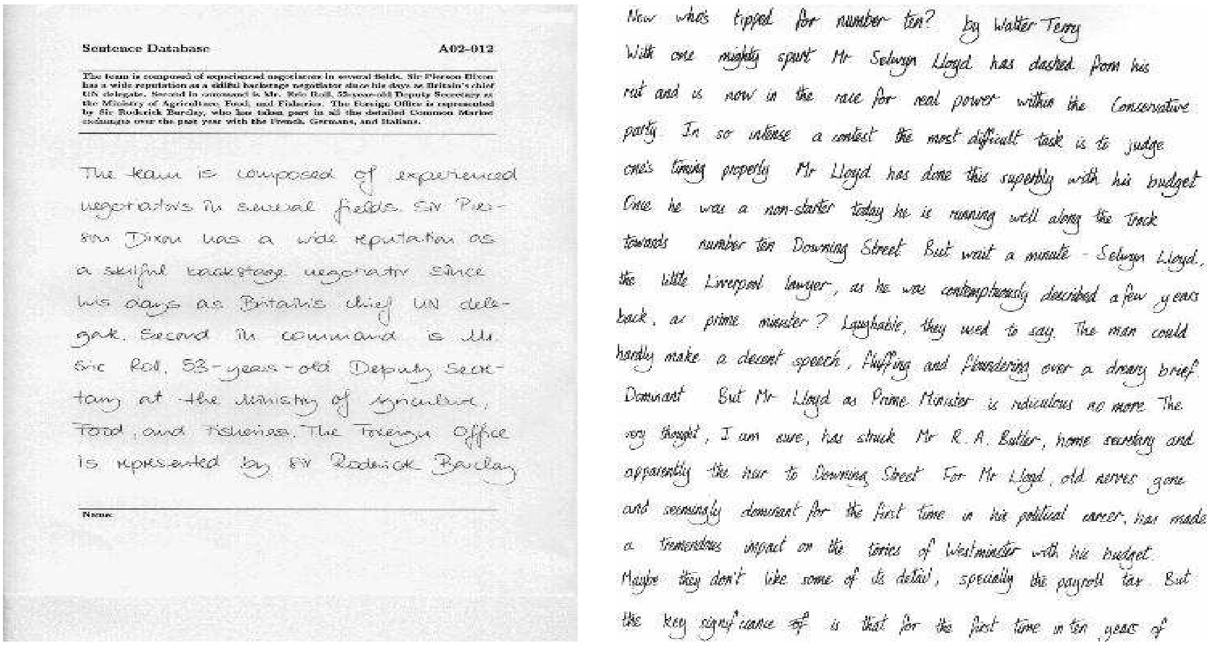


Figure 1. Images of handwritten texts. On the left: Image example of IAM database. On the right: Image example of database collected by Senior.

is proposed, i.e. a whole line of text is fed into the recognition module. This system is tested in single writer mode and achieves promising recognition results by incorporating statistical language knowledge. An advanced system for writer-independent unconstrained text recognition which is also tested on a larger database [10] produced by several hundreds of writers is described in [11]. The system described in the following sections uses a similar approach with slightly different methods for pre-processing and feature extraction. Additionally, the effects of applying linear discriminant analysis (LDA) of the feature vectors are investigated as well as the influence of allograph character models, i.e. using a set of HMMs for each character class, and statistical language models.

3 Corpora

The input data for training and recognition is provided by full English sentence databases, each based on the Lancaster-Oslo/Bergen corpus [7]. The experiments in multiple writer as well as writer independent mode were conducted using the database of handwritten forms [10] collected at the Institute of Informatics and Applied Mathematics (IAM), University of Bern. The entire database covers different text categories (press texts, religion, popular lore, fiction,...) and consists of more than 1200 handwritten forms produced by more than 500 different writers. We use the forms of the categories `[a..f]` produced by about 250 writers for experiments in writer independent mode, and the forms of the subset `c03`, produced by six writers are applied in multiple writer mode.

In order to evaluate our system also in single writer mode we conducted experiments using the handwritten forms provided by the database collected by Senior [15]. This database consists of 25 pages of handwritten text produced by a single writer and is publicly available on the web [1].

The handwritten forms of both databases were scanned with a resolution of 300dpi using 256 grey-levels. Fig. 1 shows an example image for each database.

4 Pre-Processing

Given the images of the handwritten texts first the skew of the whole image is corrected in order to compensate for errors introduced by an unprecise alignment of the form during scanning or a constant "drift" of the writing. Therefore, the image is rotated until the minimum entropy of the horizontal density histogram of the binarized image is reached [4]. Note that this pre-processing step is not applied for formulars of the IAM database because the writers were asked to use rulers on a second sheet put below the form and the formulars itself are aligned precisely during scanning.

[1] ftp://svr-ftp.eng.cam.ac.uk/pub/data

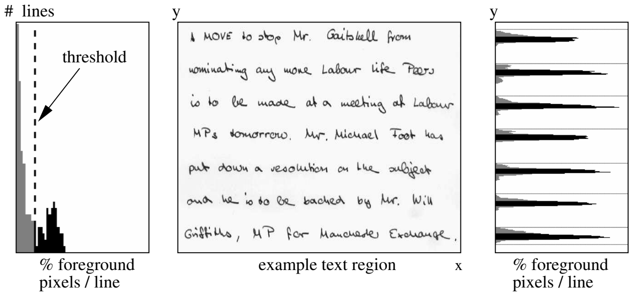


Figure 2. Left: The horizontal density histogram (The threshold is found using the Otsu method). Middle: The text area. Right: The horizontal projection histogram (Black lines denote core-regions, dotted lines mark the splitting positions).

In order to further process the text, the individual lines of handwriting have to be extracted . This is achieved by splitting the image between the core regions of the handwritten lines. The core region, i.e. the area between upper and lower baseline of a text line, can be found by applying a threshold, which denotes the minimum number of foreground pixels necessary for lines belonging to a core region. This threshold is determined automatically by applying the Otsu method [12] using the horizontal density histogram of the binarized handwriting-area. The number of black pixels is then accumulated for each line in a horizontal projection histogram and the image is split between the core regions along the minima of this projection histogram.

Because of the variability of the writing-styles of different writers the handwriting has to be normalized to simplify the recognition task. Particularly, the correction of the vertical position, skew, and slant of the writing have turned out as important pre-processing steps. Further normalization concerns the size of the handwriting and the intensity of the gray-levels.

Motivated by the observation that the writing-style sometimes changes significantly within one line, we correct the vertical position, skew, and slant of each handwritten line locally. Thus each row is split by searching for white-spaces between the segments of handwriting. A threshold is used to avoid segments that are too short for calculating reliable normalization factors.

Whereas the vertical position and skew is corrected by applying a baseline estimation method using linear regression similar to the approach described in [15], the calculation of the slant angle is based on edge orientation. Therefore the image is binarized and the horizontal black-to-white and white-to-black transitions are extracted, considering that only vertical strokes are decisive for slant estimation. A Canny edge detector is applied in order to obtain the edge orientation data which is accumulated in an angle histogram. The mean of the histogram is then used as slant angle.

In order to normalize the size of the handwriting, we count the number of local extrema of each handwritten line and put this number in relation to the width of the line. The scaling factor is based linearly on this relation, because the larger this relation the narrower the writing-style.

The last pre-processing step consists of normalizing the gray-levels in order to compensate for variations concerning the pen and background color, caused by different kinds of pen and paper. Therefore the gray-level interval of the image is scaled so that the darkest intensity becomes zero and the brightest 255. The effect of these pre-processing steps is shown in Fig. 3 using a typical text line from the corpus.

5 Feature Extraction

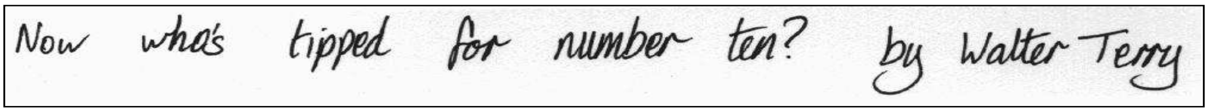
The pre-processed images of the handwritten lines are used as input data for the feature extraction step. A sliding window technique is applied similar to the approach described in [11]. In our case, a window of the image's height and four columns width is moved with an overlap of two columns from left-to-right over the image and a couple of geometrical features is extracted.

For each column of the sliding window the following seven features are calculated: (1) Number of black-to-white transitions (after a binarization of the windowed text image), (2) position of the mean value of the intensity distribution with respect to the baseline (3) distance from the uppermost text pixel to the baseline, (4) distance from the lowermost text pixel to the baseline, (5) distance of uppermost and lowermost text pixel, (6) average intensity between uppermost and lowermost text pixel, (7) average intensity of the column. The features (2)-(5) are normalized by the core-size, i.e. the distance of lower and upper baseline (computed by line fitting through local maxima), in order to increase the robustness against variations in writing-size. Afterwards, all features are averaged over the four columns of the window.

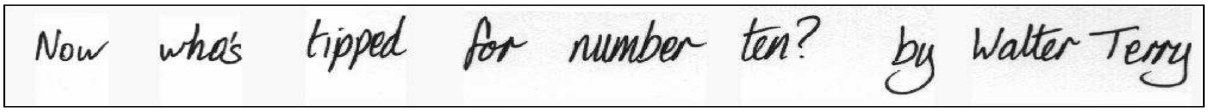
In order to consider the direction of the lower and upper contour as well as the gradient of the mean value of the intensity distribution, we additionally calculate three directional features. Therefore, we estimate lines through the four lower contour points, upper contour points, and mean values within the sliding window, and use the line orientations as features (8), (9), and (10), respectively.

For considering a wider temporal context, we additionally compute an approximate horizontal derivative for each component of the feature vector, so that a 20 dimensional feature vector is obtained (10 features per window + 10 derivatives).

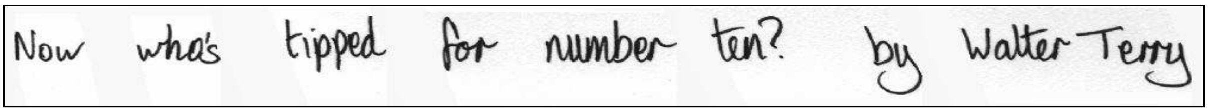
In order to decorrelate the feature vectors and to improve



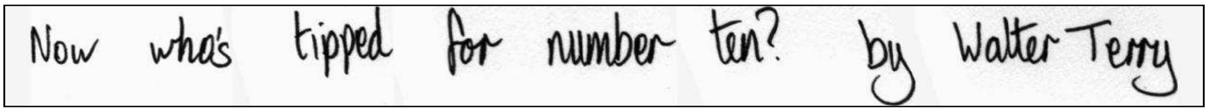
a) extracted text line



b) local skew and displace correction



c) local slant correction



d) scaling and intensity normalization

Figure 3. Pre-processing steps.

the class separability we integrate linear discriminant analysis (LDA) in the training and recognition phase (cf. [6]). The original feature representation is optimized by applying a linear transformation and simultaneously reducing the dimensionality of the feature space. The linear transformation A is obtained by solving an eigenvalue problem using the within class scatter matrix S_w and the between class scatter matrix S_b of the training data. As for computing these scatter matrices each feature vector has to be labelled with an HMM state, we at first carry out an ordinary training followed by a state-based alignment of the training data. When the scatter matrices are known the LDA transformation is computed by solving the following eigenvalue problem

$$\mu_i (A^T \Psi)_i = S_w^{-1} S_b (A^T \Psi)_i \qquad (i = 1, 2, \ldots, m)$$

where μ_i and $(A^T \Psi)_i$ are the eigenvalues and eigenvectors of $S_w^{-1} S_b$. A reduction of the dimensionality is obtained by taking into account only m eigenvectors belonging to the m largest eigenvalues. After LDA transforming all feature vectors a completely new HMM training is carried out.

6 Statistical Modelling and Recognition

Configuration, training, and decoding of Hidden-Markov Models (HMMs) for the handwritten text recognition task is carried out in the framework of the methods and tools provided by the ESMERALDA development environment [5]. As general setup of the HMMs we use semi-continuous systems with a shared codebook containing 512 (2048 for the large writer-independent system) Gaussian mixtures with diagonal covariance matrices. For the base system models for 52 letters, ten numbers, twelve punctuation marks and brackets, and one white space character are trained using standard Baum-Welch reestimation. A combined model consisting of a loop over the character models is then used for recognizing a whole line of text. The most likely character sequence is calculated using standard Viterbi beam-search.

In order to consider the variances among the different writing styles that can not be compensated during pre-processing we apply character allograph models for recognition in multiple writer and writer independent mode similar to the approach described in [13]. Allographs denote character subclasses, i.e. different realizations of a specific character. This means, that, opposed to the baseline system, HMMs are now used for modelling the different character *subclasses*. The number of subclasses per character and thus the number of allograph HMMs is determined heuristically, for the system applied in multiple writer mode for example, we assume that the number of allographs is given by the number of writers. For initialization, the training data is labelled by randomly choosing allograph HMMs. During training for every character sample the parameter reestimation is applied to all corresponding allograph models in parallel. The probability of match determines how strongly the parameters of a specific model are influenced by the current sample. As allograph labels are thus not uniquely determined and only chosen probabilistically the procedure is similar to soft vector quantization.

Experiment		No LM	Bi-gram
Writer indep.	Baseline	31.3	24.3
	LDA	29.1	22.2
	Allographs	31.1	24.0
Multiple Writer	Baseline	16.9	12.5
	LDA	14.2	11.2
	Allographs	15.1	11.8
	LDA + Allographs	13.3	11.1
Single Writer	Baseline	13.3	12.1
	LDA	13.5	12.5

Table 1. Character error rates.

Additionally, statistical language models are integrated in the recognition process for providing estimates about character sequences likely to occur. The goal of the recognition process is then to find the character sequence $\hat{\mathbf{w}}$ that maximizes the probability of the combined statistical model given the observed data $\mathbf{x}$ according to:

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmax}} P(\mathbf{w}) P(\mathbf{x}|\mathbf{w})$$

In the above equation $P(\mathbf{w})$ denotes the language model probability for the character sequence $\mathbf{w}$ and $P(\mathbf{x}|\mathbf{w})$ the probability of observing this character sequence as input data $\mathbf{x}$ according to the character model. In our case a bi-gram language model using absolute discounting and backing-off for smoothing of probability distributions is applied (cf. e.g. [3]).

7 Results

We carried out three kinds of experiments in order to evaluate our handwriting recognition system: recognition in single writer mode, multiple writer mode, and writer independent recognition. The character error rates of these experiments are shown in table 1, where the type of experiment is described in the first two columns, the error rates without applying a language model are shown in the third column, and the results using a bi-gram language model at the character level can be found in the fourth column. The language model was generated using all texts of categories `[a..d]` of the IAM database and it remains the same for all experiments. The results for lexicon-free and lexicon-based word recognition are shown in table 2.

The experiments in single writer mode are conducted using 282 text lines of the Senior database for training and 141 text lines for testing, where the test set reveals a bi-gram perplexity at the character level of 15.3. A character error rate of 13.3% for the baseline system is achieved that decreases to 12.1% by employing the bi-gram language model. By using LDA transformed features the error rate does not increase significantly, even though the dimensionality of the LDA transformed feature space is reduced to 12. The word error rate (table 2) of the single-writer system is 28.5% without lexicon, and 10.5% with a 1.5k lexicon, respectively. These results compare well to the error rates we found in the literature using the same database, even though it is difficult to compare because the training and test set sizes differ. In [17], the error rates measured are 28.3% for characters, 84.1% for words without lexicon, and 16.5% for words with 1.3k lexicon, respectively. [15] reports word error rates of 6.6% with lexicon and 41.1% lexicon-free. The best lexicon-based word error rate reported in [9] is 15.0%.

Experiment	No Lexicon	Lexicon (Size)
Writer indep.	60.6	42.2 (7.5k)
Multiple Writer	39.0	13.9 (421)
Single Writer	28.5	10.5 (1.5k)

Table 2. Word error rates.

For the multiple writer handwriting recognition task 440 text lines of the subset `c03` of the IAM-database are used for training, and 109 lines for testing. These text-lines were produced by six different writers with quite different writing-styles. For this task a character error rate of 14.2% using LDA (reduced dimensionality 12) is obtained that decreases significantly to 13.3% by additionally using allograph models (6 allographs for each lowercase character). The application of the bi-gram language model further improves the error rate to 11.1% (test set perplexity 12.0). The word error rate without using a lexicon is 39.0%, by applying a lexicon containing 421 words (incl. punctuation marks) the error rate decreases to 13.9%, that compares well with the word error rate of 20.5% published in [11].

These results encouraged us to carry out experiments for the much harder task of writer independent recognition. The subset `[a-f]` (approx. 250 writers) of the IAM-database is used as input data, where 4321 lines of text (forms `[a-d]`) are used for training and 1097 lines of text (forms `[e-f]`) are used for testing. The baseline system achieves a character error rate of 31.3%. In the writer independent case the application of allograph models yields no significant improvement in contrast to the multiple writer experiment. An error rate of 31.1% is achieved using three allographs per character and both error rate (34.8%) and recognition speed got worse when we conducted experiments using ten allographs per character. But the error rate decreases significantly to 29.1% when using LDA transformed features. By additionally incorporating the language model the character error rate further improves to 22.2% (test set perplexity 12.0). This corresponds to a word error rate of 60.6% without using a lexicon.

8 Conclusion

We have presented a system for segmentation-free off-line handwritten text recognition, and conducted several experiments in single writer, multiple writer, and writer independent mode. Promising recognition results have been achieved on the character level as well as word-based without using a lexicon. The recognition accuracy has been considerably improved compared to the baseline system by applying statistical language models on the character level and, for recognition in multiple writer and writer independent mode, by performing linear discriminant analysis of the feature space. Further improvements in recognition accuracy, in particular for recognition in multiple writer mode, have been achieved by using allograph character models.

9 Acknowledgement

This work was supported by the German Research Foundation (DFG) within project **Fi799/1**.

Additionally, we would like to thank the Institute of Informatics and Applied Mathematics, University of Bern, namely Horst Bunke and Urs-Viktor Marti, who allowed us to use the IAM database of handwritten forms [10] for our recognition experiments.

References

[1] H. Bunke, M. Roth, and E. G. Schukat-Talamazzini. Off-line cursive handwriting recognition using Hidden Markov Models. *Pattern Recognition*, 28(9):1399–1413, 1995.

[2] M.-Y. Chen, A. Kundu, and J. Zhou. Off-line handwritten word recognition using a hidden markov model type stochastic network. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(5):481–496, 1994.

[3] S. F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13:359–394, 1999.

[4] M. Côté, E. Lecolinet, M. Cheriet, and C. Suen. Automatic reading of cursive scripts using a reading model and perceptual concepts - the PERCEPTO system. *Int. Journal on Document Analysis and Recognition*, 1:3–17, 1998.

[5] G. A. Fink. Developing HMM-based recognizers with ESMERALDA. In V. Matoušek, P. Mautner, J. Ocelíková, and P. Sojka, editors, *Lecture Notes in Artificial Intelligence*, volume 1692, pages 229–234, Berlin Heidelberg, 1999. Springer.

[6] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, New York, 1972.

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

[8] G. Kim and V. Govindaraju. A lexicon driven approach to handwritten word recognition for real-time applications. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(4):366–379, 1997.

[9] U.-V. Marti and H. Bunke. Erkennung handgeschriebener Wortsequenzen. In P. Levi, R.-J. Ahlers, F. May, and M. Schanz, editors, *Mustererkennung 98, 20. DAGM-Symposium Stuttgart, Informatik aktuell*, pages 263–270, Berlin, 1998. Springer-Verlag.

[10] U.-V. Marti and H. Bunke. A full english sentence database for off-line handwriting recognition. In *Proc. Int. Conf. on Document Analysis and Recognition*, pages 705–708, Bangalore, 1999.

[11] U.-V. Marti and H. Bunke. Handwritten sentence recognition. In *Proc. Int. Conf. on Pattern Recognition*, volume 3, pages 467–470, Barcelona, 2000.

[12] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. on Systems, Man, and Cybernetics*, 9:62–66, 1979.

[13] M. P. Perrone and S. D. Connell. K-means clustering for Hidden Markov Models. In *Proc. 7th Int. Workshop on Frontiers in Handwriting Recognition*, pages 229–238, Amsterdam, 2000.

[14] R. Plamondon and S. N. Srihari. On-line and off-line handwriting recognition: A comprehensive survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(1):63–83, 2000.

[15] A. W. Senior and A. J. Robinson. An off-line cursive handwriting recognition system. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(3):309–321, 1998.

[16] T. Steinherz, E. Rivlin, and N. Intrator. Offline cursive script word recognition – A survey. *Int. Journal on Document Analysis and Recognition*, 2(2):90–110, 1999.

[17] A. Vinciarelli and J. Luettin. Off-line cursive script recognition based on continuous density HMM. In *Proc. 7th Int. Workshop on Frontiers in Handwriting Recognition*, pages 493–498, Amsterdam, 2000.

[18] W. Wang, A. Brakensiek, A. Kosmala, and G. Rigoll. HMM based high accuracy off-line cursive handwriting recognition by a baseline detection error tolerant feature extraction approach. In *Proc. Int. Workshop on Frontiers in Handwriting Recognition*, pages 209–218, Amsterdam, 2000.