A writer identification system for on-line whiteboard data

Andreas Schlapbach*, Marcus Liwicki, Horst Bunke

Institute of Computer Science and Applied Mathematics, Universität Bern, Neubrückstrasse 10, CH-3012 Bern, Switzerland

Abstract

In this paper we address the task of writer identification of on-line handwriting captured from a whiteboard. Different sets of features are extracted from the recorded data and used to train a text and language independent on-line writer identification system. The system is based on Gaussian mixture models (GMMs) which provide a powerful yet simple means of representing the distribution of the features extracted from the handwritten text. The training data of all writers are used to train a universal background model (UBM) from which a client specific model is obtained by adaptation. Different sets of features are described and evaluated in this work. The system is tested using text from 200 different writers. A writer identification rate of 98.56% on the paragraph and of 88.96% on the text line level is achieved.
© 2008 Elsevier Ltd. All rights reserved.

Keywords: Writer identification; On-line handwriting; Gaussian mixture models

1. Introduction

The work described in this paper has been conducted in the context of research on smart meeting rooms. The aim of this research is to automate standard tasks usually performed by humans in a meeting [1–5]. To record a meeting, smart meeting rooms are equipped with synchronized recording interfaces to capture audio, video, and handwritten notes.

Smart meeting rooms pose interesting pattern recognition and classification problems. Speech [6], handwriting [7], and video recognition systems [8] have been developed. Other tasks include segmenting a meeting into meeting events [3,4], indexing the recorded data [9] or extracting non-lexical information, such as prosody, voice quality variation, and laughter. To authenticate the meeting participants and to assign utterances and handwritten notes to their authors, identification and verification systems are developed. They are based on speech [10] and video interfaces [11,12] or on a combination of both [13].

An important task in a smart meeting room is to capture the handwriting rendered on a whiteboard during a meeting. In this paper we address the problem of identifying the author of a text written on a whiteboard. Solving this problem enables us to label the handwriting with the writer's identity. Furthermore, it allows us to validate the identification results of a video- or audio-based person identification system within the smart meeting room scenario.

The text written on the whiteboard is recorded by the eBeam interface.[1] A normal pen in a special casing sends infrared signals to a triangular receiver mounted in one of the corners of the whiteboard [14]. The acquisition interface outputs a sequence of (x, y)-coordinates representing the location of the pen-tip together with a time stamp for each location. The sampling resolution varies around 30–70 samples per second with a sampling resolution of 4 points per millimeter. Spurious points and gaps within strokes can occur if the writer's hand is between the pen and the receiver, or if the pen is tilted too much. An illustration of the data acquisition device is shown in Fig. 1.

The input to our system are lines of handwritten text. Typical data acquired from a whiteboard in a meeting may also include sketches, tables or enumerated lists. However, there exists techniques to extract text regions from the data collected [15,16].

---

* Corresponding author. Tel.: +41 31 631 49 02; fax: +41 31 631 86 81.
E-mail addresses: schlpbch@iam.unibe.ch (A. Schlapbach),
liwicki@iam.unibe.ch (M. Liwicki), bunke@iam.unibe.ch (H. Bunke).

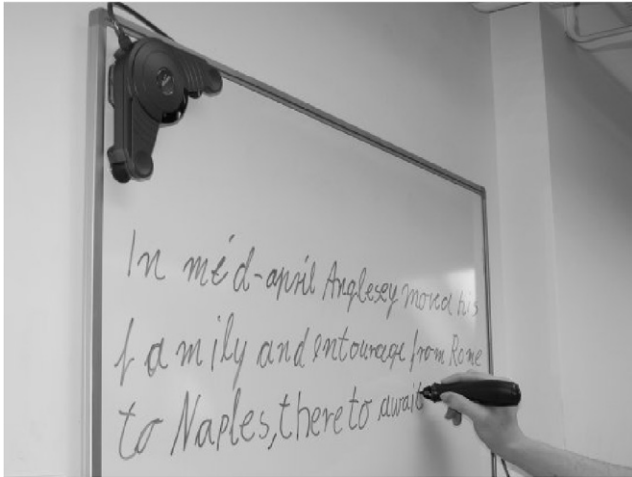[1] eBeam system by Luidia, Inc.—www.e-Beam.com.

Fig. 1. Recording session with the data acquisition device positioned in the upper left corner of the whiteboard.

We use Gaussian mixture models (GMMs) to model a person's handwriting. GMMs provide a powerful yet simple means of representing the distribution of the features extracted from the text written by one person. GMMs have a mathematically simple and well understood structure, and there exist standard algorithms for training and testing. Formally, GMMs consist of a weighted sum of uni-modal Gaussian densities. While GMMs have first been used in speech recognition [17,18], to the best of our knowledge, they have not been applied to on-line writer identification of whiteboard data before.

For each writer in the considered population, an individual GMM is trained using data from that writer only. Thus for n different writers we obtain n different GMMs. Intuitively, each GMM can be understood as an expert specialized in recognizing the handwriting of one particular person. Given an arbitrary text as input, each GMM outputs a recognition score. Assuming that the recognition score of a model is higher on input from the writer the model is trained on than on input from other writers, we can utilize the scores produced by the different GMMs to identify the writer of a text.

The outline of this paper is as follows. In the next section related work is presented. Section 3 gives an overview of our system and describes the normalization operations applied to the acquired data. In Section 4 the feature sets extracted from the normalized data are described. The GMMs used to model a person's handwriting are presented in Section 5. In Section 6 the experimental setup is described, while the results of our experiments are presented and discussed in Section 7. Section 8 concludes the paper and proposes future work.

2. Related work

The topic of writer identification from on-line whiteboard data has not been addressed in the literature to the best of our knowledge. However, much research has been performed in related fields, such as identification and verification of signatures and general handwriting.

Work in these fields can be differentiated according to the available data. If only a scanned image of the handwriting is available then writer classification is performed with *off-line* data. Otherwise, if temporal and spatial information about the writing is available, writer classification is performed with *on-line* data. On-line handwritten data contains more information about the writing style of a person, such as speed, angle or pressure. This information is not available in off-line handwritten data. Thus the on-line classification task is considered to be less difficult than off-line classification [19].

Surveys covering work in automatic writer identification and signature verification until 1993 are given in Refs. [19,20]. Subsequent works up to 2000 are summarized in Ref. [21]. Recently, several additional approaches have been proposed. In Section 2.1 work on off-line writer identification and verification is presented. Section 2.2 summarizes papers on signature verification. Work in the new field of on-line writer identification and verification is presented in Section 2.3.

2.1. Off-line writer identification and verification

Said et al. [22] treat the writer identification task as a texture analysis problem. They use global statistical features extracted from the entire image of a text using multi-channel Gabor filtering and gray-scale co-occurrence matrix techniques. In Ref. [23] this approach is extended to Chinese handwriting. He et al. [24] present a wavelet-based generalized Gaussian density (GGD) method which decomposes the image into subbands of different frequencies and orientations and uses its parameters as features.

Srihari et al. [25–27] address the problem of writer verification, i.e., the problem of determining whether two documents are written by the same person or not. In order to identify the writer of a given document, they model the problem as a classification problem with two classes, *authorship* and *non-authorship*. Given two handwriting samples, one of known and the other of unknown identity, the distance between two documents is computed. Then the distance value is used to classify the data as positive or negative.

Zois et al. [28] base their approach on single words by morphologically processing horizontal projection profiles. The projections are partitioned into a number of segments from which feature vectors are extracted. A Bayesian classifier and a neural network (NN) are then applied to the feature vectors.

In Ref. [29] a system for writer identification is described. The system first segments a given text into individual text lines and then extracts a set of features from each text line. The features are subsequently used in a k-nearest-neighbor classifier that compares the feature vector extracted from a given input text to a number of prototype vectors coming from writers with known identity.

Bulacu et al. [30] use edge-based directional probability distributions as features for the writer identification task. The authors introduce edge-hinge distribution as a new feature. The key idea behind this feature is to consider two edge fragments in the neighborhood of a pixel and compute the joint probability distribution of the orientations of the two fragments.

Additionally, in Ref. [31] as a new feature the histogram of connected-component contours (CO^3) for upper-case handwriting is introduced. This approach is extended to mixed-style handwriting in Ref. [32], using fragmented connected-component contours (FCO^3).

Wang et al. extract directional element features (DEFs) and then reduce the dimensionality of the feature space using principal component analysis (PCA) and linear discriminant analysis (LDA) [33,34]. The Euclidean distances between the unknown script and the mean vector of the known scripts are calculated to identify the writer. Finally, the score is normalized with a measure expressing the similarity between the determined script and script samples written by other writers [35].

In a number of papers [36–39] graphemes are proposed as features for describing the individual properties of handwriting. Furthermore, it is shown that each handwriting can be characterized by a set of invariant features, called the writer's invariants. These invariants are detected using an automatic grapheme clustering procedure. In Refs. [38,39] these graphemes are used to address the writer verification task based on text blocks as well as on handwritten words.

Fractal analysis of handwriting has been proposed in Refs. [40,41]. For each writer a set of invariant features is gained by adapting techniques that have been developed for the compression of fractals [42]. During learning the invariant patterns are extracted and used as a reference base in order to analyze an unknown writing by measuring the similarity between the images modified by the compression and the decompression process. Among other features, fractal features have also been extracted from text lines in Ref. [43].

Leedham et al. [44] present a set of 11 features which can be extracted easily and used for the identification and the verification of documents containing handwritten digits. These features are represented as vectors, and by using the Hamming distance measure and determining a threshold value for the intra-author variation a high degree of accuracy in authorship detection is achieved.

We have proposed to use hidden Markov model (HMM)-based text recognizers [45–47] and GMMs [48,49] for off-line writer identification and verification. For each writer, a model is built and trained on text lines of that writer. This results in a number of models, each of which is an expert on the handwriting of exactly one writer. Assuming that the score of a system is higher on input from the writer the system was trained on than on input from other writers, the scores produced by the models are used to decide who has written the input text line.

2.2. On-line signature verification

From Section 2.1 it can be concluded that there exists quite a body of work on off-line writer identification and verification. By contrast, little has been published on the on-line case (see Section 2.3). However, a number of research articles in the field of on-line signature verification, which is a special case of on-line writer verification, have been reported. They will be reviewed in this section.

Signatures differ from normal handwritten texts in the sense that they are more graphical than text, and letters are often deformed or missing. Early work on on-line signature verification is described in Refs. [19,50,51]. In recent works, various approaches based on dynamic time warping (DTW), NNs, HMMs and GMMs have been proposed.

Lee et al. [52] use a very simple classifier that implements a majority decision rule. If more than half of the features are from the respective distributions describing the features of the claimed person then the signature is classified as genuine; otherwise it is assumed to be forged. A basic set and an advanced set of features are extracted from the on-line data. Different algorithms are studied for selecting and orthogonalizing the feature sets. A common feature set is defined consisting of features that are good for most persons in the considered population.

Nalwa [53] claims that the temporal characteristics of a signature are not as consistent as its shape information. Five characteristic functions are derived, each describing a local feature of the signature. The characteristic functions of a signature with a claimed identity are simultaneously warped against their prototypes and the overall alignment cost is then considered as one global feature. The final dissimilarity measure is defined as the weighted harmonic mean of the global features.

A DTW approach is presented in Ref. [54]. Global and local features are extracted from the slope of the signature and stored in a string representation. The similarity between an input signature and the reference set is then computed by string matching. A new warping technique called extreme point warping, which only warps selective points, is proposed in Ref. [55].

In Ref. [56] three different NN-based approaches for on-line human signature verification are studied. An on-line signature verification system that uses multi-layer perceptrons (MLPs) trained with cepstral coefficients derived from linear predictor coefficients of the writing trajectories is presented in Ref. [57]. In Ref. [58] a signature verification system that uses wavelets and back-propagation NNs is proposed.

Kholmatov and Yanikoglu consider the signature verification problem as a two-class pattern recognition problem [59]. A test signature's authenticity is established by first aligning it with each reference signature for the claimed user, using DTW. The distances of the test signature to reference signatures are normalized to form a feature vector which is then classified into the genuine or the forgery class. After PCA, a linear classifier is used to classify a signature. This system performed best at the first international signature verification competition [60].

Various HMM-based approaches have been applied to signature verification [61,62]. Yang et al. [63] incorporate dynamic sequence information by extracting normalized angles as features and model the generation of these sequences by HMMs. The best performance is achieved using a left-to-right topology for the HMMs. Kashi et al. [64] combine global features that capture spatial and temporal characteristics of the signature with a local feature based on the signature likelihood obtained from HMMs. Yoon et al. transform the signature data into the polar space, which makes the features size and angle invariant [65]. Muramatsu et al. [66–68] propose an on-line

signature verification algorithm by mapping the trajectory angles to HMM states in a left-to-right topology.

In recent work, Ortega-Garcia et al. [69] compute eight time sequences and their first and second order time derivatives from the on-line signature data. The features are normalized and modeled by a left-to-right HMM with continuous output density functions. In the verification phase, scores are considered as relative values with respect to a reference population and are normalized by best-reference score normalization. The system performs very favorable on the signature corpus of the MCYT bimodal biometric database [70].

Richiardi et al. introduce GMMs for on-line signature verification [71]. They use horizontal and vertical position, pressure, trajectory tangent angle, and velocity as basic features to train the models. A minimum description length (MDL) cost function is introduced that balances modeling errors and model complexity. Using the MDL criterion for each user the best performing model is selected. In an extension to this work, they propose a signature feature selection algorithm that combines a modified Fisher ratio cost function and a sub-optimal but fast floating search algorithm to obtain an initial set of local [72] and global features [73].

In Ref. [74] the authentic and the forgery samples are represented by two separate GMMs. Dissimilarity vectors are obtained after the initial vectors have been aligned by DTW. During training, the normalized dissimilarity vectors along with the labels are used in a discriminative training procedure to train the two GMMs. The two classifiers are optimized on a discriminative objective function derived from the minimum classification error (MCE) criterion. A two-stage statistical system composed of a simplified GMM for global signature features and a discrete HMM for local signature features is presented in Ref. [75].

2.3. On-line writer identification and verification

Only recently, work on on-line writer identification and verification has started and only few papers exist. Chapran [76]

extracts static and dynamic features especially suited for embedded writer identification systems with restricted memory and processing power. The feature set is reduced to a feature subset using a feature selection approach based on likeness coefficients. An optimal number of features is determined by means of discriminant analysis and then passed through three different classifiers, namely minimum distance classifier, Bayes classifier and their serial combination. In other works, Chapran et al. propose a method for dynamic writer identification which uses the relation between static and dynamic information in a handwritten text [77]. The correlation between the length and the direction of the segments of handwriting between two sample points as well as pressure, altitude, and azimuth are used to identify the writer.

Some of the features used in this work are inspired by work on on-line handwriting recognition [78,79] and signature verification [54,73]. The features were chosen because they have shown to adequately capture the form of a person's handwriting. To model the distribution of the features, we have applied GMMs, which have shown good performance in signature verification [73] and off-line writer identification [48].

We first proposed to use GMMs for writer identification of on-line whiteboard data in Ref. [80]. The present paper is a substantially extended version of Ref. [80]. First, three new feature sets, leading to significantly improved writer identification rates, are introduced. Second, different training methods to obtain the client models are evaluated. Third, the influence of having fewer data available for training as well as testing are systematically studied.

3. Writer identification system for on-line whiteboard data

3.1. System overview

The distribution of the features extracted from the handwriting of a person is modeled by one GMM for each writer. The models are obtained by the following two-step training procedure (a detailed description of the training procedure is
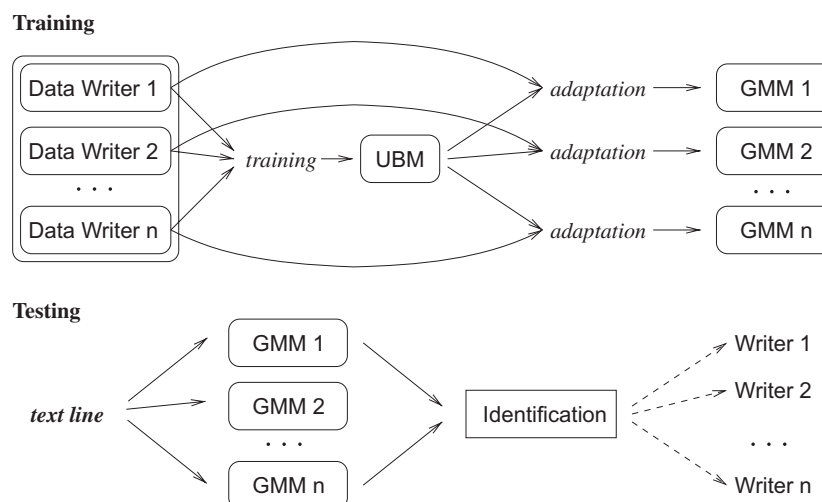

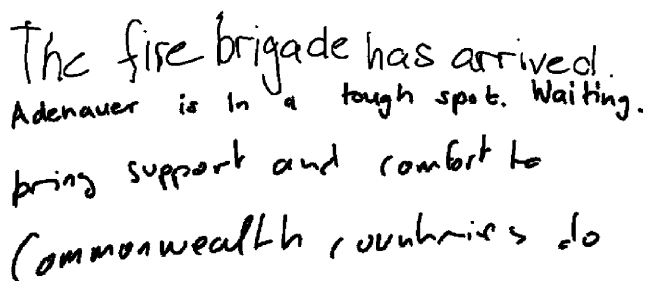Fig. 2. Schematic overview of the training and the testing phase.

given in Section 5). In the first step, all training data from all writers are used to train a single, writer independent universal background model (UBM). In the second step, for each writer a writer specific model is obtained by adaptation using the UBM and training data from that writer. As a result of the training procedure, we get a model for each writer. In the testing phase, a text of unknown identity is presented to each model. Each model returns a log-likelihood score, and these scores are sorted in descending order. Based on the resulting ranking, the text is assigned to the person whose model produces the highest log-likelihood score. A schematic overview of the training and the testing phase is shown in Fig. 2.

To train the models, different feature sets are extracted from the text which are described in Section 4. Before feature extraction, a series of normalization operations are applied. The operations are designed to improve the quality of the features extracted without removing writer specific information. For this reason, e.g., no resampling of the data points is performed.

3.2. Preprocessing

On-line handwriting captured from a whiteboard differs from handwriting acquired by other devices such as digitizing tablets or Tablet PCs. While some of these devices register the pressure and the angle of the pen during writing, this information is not available from whiteboard data due to the way the data are acquired. Furthermore, whiteboard data often have a wave-like baseline and the size of the letters varies (Fig. 3 shows some examples of handwritten texts). This writing style stems from the fact that during writing people stand rather than sit and that their arm does not rest on a table. In this case, approximating the base-line of a text line by one straight line would not yield satisfactory results.

The recorded on-line data contains noisy points and gaps within strokes which are caused by loss of sampling data during

acquisition. In Fig. 4 a paragraph before and after preprocessing is shown. As can been seen in the figure, after preprocessing the strokes constituting a single letter are connected (e.g., the "I" of "In" or the "y" of "Anglesey") and the spurious stroke between the third and fourth line is removed. To recover from noisy points and gaps within strokes, two preprocessing steps are applied to the data. Let $p_1, \ldots, p_n$ be the points of a given stroke and q_1 be the first point of the succeeding stroke. Note that a stroke starts when the pen tip touches the whiteboard and ends when the pen tip looses contact with the whiteboard. To identify noisy points, we check whether the distance between two consecutive points p_i and p_{i+1} is larger than a fixed threshold. Finding an appropriate threshold value is not critical because in case of a noisy point the distance between the two considered points is significantly larger than between two consecutive points. In case the distance exceeds the threshold, one of the points is deleted. To decide which point has to be deleted, the number of points within a small neighborhood of p_i and p_{i+1} is determined, and the point with a smaller number of neighbors is deleted. To recover from the second type of artifacts, i.e., gaps within strokes, we check if the distance between the timestamps of p_n and q_1 is below a fixed threshold. Again it is not critical to define this threshold. If the condition holds the strokes are merged into one single stroke.

Next, the cleaned paragraph of text is automatically divided into lines using a simple heuristic. If there is a pen-movement to the left longer than $\frac{1}{3}$ of the document's width and downwards greater than a predefined threshold the start of a new line is assumed. This condition prevents divisions to be caused by t-bar crossings or by i/j dotting.

The next step is to divide each text line into sub-parts which then can be normalized independently of each other. A text line is split at a gap if the gap is larger than the mean gap size and if the size of both sub-parts is greater than a predefined threshold (see Fig. 5). The mean gap size is defined as the mean of the x-distances between succeeding strokes of a line.

For each sub-part the skew angle is calculated and corrected. Linear regression through all points of a sub-part is performed to estimate the base and the corpus lines. For this purpose, the minima and maxima of the y-coordinates of the strokes are calculated. Then two linear regressions through the minima and maxima are computed with the constraint that the two resulting lines have to have the same slope. After regression the least fitting points are removed and another linear regression is performed. This correction step is repeated twice, which produces an estimated base line (minima) and a corpus line (maxima). Fig. 6 illustrates the estimated base line and the corpus line of
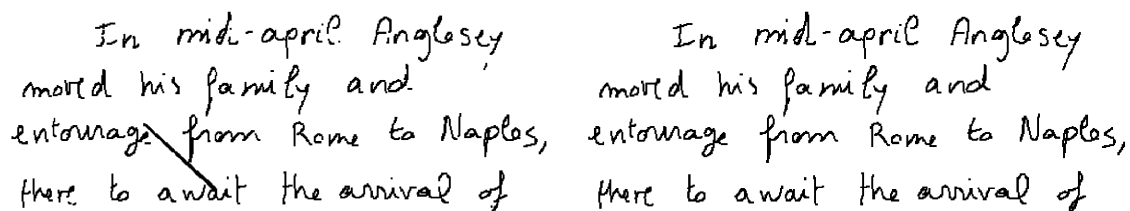

Fig. 3. Examples of handwritten texts acquired by the electronic acquisition device from the whiteboard.
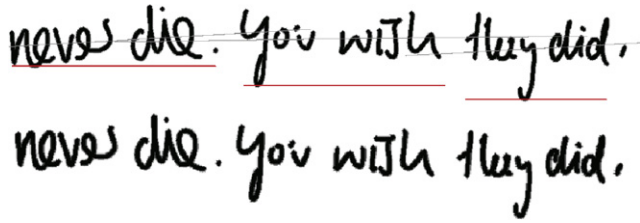

Fig. 4. A text paragraph before and after preprocessing.
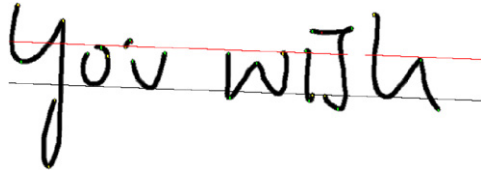
Fig. 5. Splitting a text line into its sub-parts.

Fig. 6. Baseline and corpus line of an example part of a text line.

Fig. 7. Illustration of point-based features.

an example word sequence. The base line is subtracted from all y-coordinates to make it equal to the x-axis. The two lines divide the text into three areas: the upper area, which mainly contains the ascenders of the letters; the median area, where the corpus of the letters is present; and the lower area with the descenders of the letters. These three areas are normalized to predefined heights. This means that the height of the upper area is set to be equal to the height of the median area. The lower area is scaled to this height if it is larger than the predefined height. This condition avoids unnecessary scaling of the lower area if no ascenders exist.

Finally, the width of each sub-part is normalized. First, the number of characters is estimated as a fraction of the number of strokes crossing the horizontal line between the base line and the corpus line. The text is then horizontally scaled according to this value. This preprocessing step is needed because we use the relative x-coordinate. The relative x-coordinate is calculated by subtracting the x-coordinate of a point from a moving average coordinate. The moving average is the x-coordinate of a point which moves with constant velocity from left to right.

4. Feature sets for whiteboard writer identification

Five feature sets for whiteboard writer identification are presented in this section. The first two feature sets, denoted as *point-based feature set* and *stroke-based feature set*, have been described previously [80]. The third set of features (*extended point-based feature set*) describes an extended set of features extracted from the on-line data. The fourth feature set (*off-line point-based feature set*) is obtained by first transforming the on-line data into an off-line representation from which the features are extracted. The fifth feature set (*all point-based feature set*) is the union of the *extended point-based feature set* and the *off-line point-based feature set*. In the remainder of this section, the number in round brackets behind the name of a feature indicates the number of individual feature values.
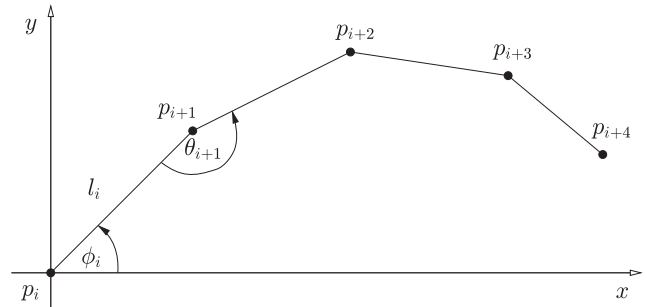
4.1. Point-based feature set

The features of this feature set are similar to the ones used in on-line handwriting recognition systems [78,79] and signature verification systems [54,73]. For a given stroke s consisting of points p_1 to p_n, the following features for each consecutive pair of points (p_i, p_{i+1}) are computed. In our notation, angle ϕ_i denotes the angle between the horizontal line and the line (p_i, p_{i+1}), and angle θ_i represents the angle between the lines (p_{i-1}, p_i) and (p_i, p_{i+1}) (see Fig. 7 for an illustration).

The following features are calculated for each point p_i:

- *Speed* (1): The speed v_i of the segment

$$v_i = \frac{\Delta(p_i, p_{i+1})}{t},$$

where t equals the sampling rate of the acquisition device.
- *Writing direction* (2): The writing direction at p_i, i.e., the cosine and sine of θ_i:

$$\cos(\theta_i) = \frac{\Delta x(p_i, p_{i+1})}{l_i},$$

$$\sin(\theta_i) = \frac{\Delta y(p_i, p_{i+1})}{l_i}.$$

- *Curvature* (2): The curvature, i.e., the cosine and the sine of the angle ϕ_i. These angles are derived by the following trigonometric formulas:

$$\cos(\phi_i) = \cos(\theta_i) * \cos(\theta_{i+1}) + \sin(\theta_i) * \sin(\theta_{i+1}),$$

$$\sin(\phi_i) = \cos(\theta_i) * \sin(\theta_{i+1}) - \sin(\theta_i) * \cos(\theta_{i+1}).$$

The *point-based feature set* thus contains five feature values.

4.2. Stroke-based feature set

In this set, the individual features are based on strokes. For each stroke $s = p_1, \ldots, p_n$ we calculate the following features (for an illustration see Fig. 8):

- *Accumulated length* (1): The accumulated length l_{acc} of all lines l_i:
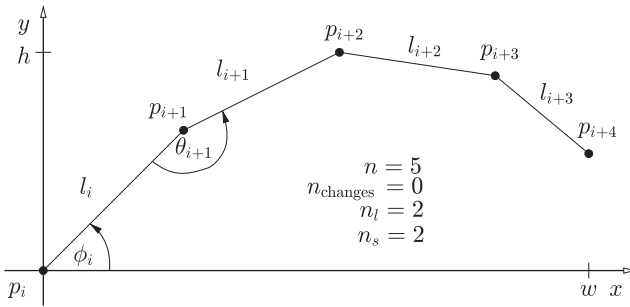
$$l_{acc} = \sum_{i=1}^{n-1} l_i.$$
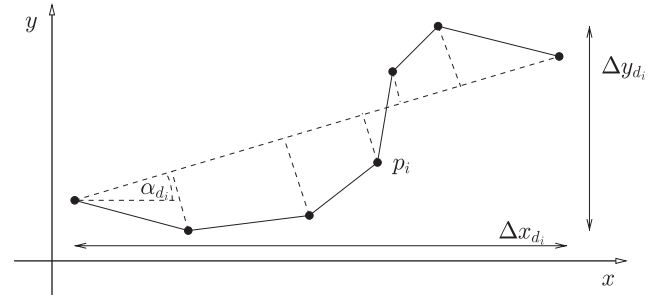
Fig. 8. Illustration of stroke-based features.

Fig. 9. Illustration of vicinity features.

- *Accumulated angle* (1): The accumulated angle θ_{acc} of the absolute values of the angles of the writing directions of all lines:

$$\theta_{acc} = \sum_{i=1}^{n-1} |\theta_i|.$$

- *Width and height* (2): The width $w = x_{max} - x_{min}$ and the height $h = y_{max} - y_{min}$ of the stroke.
- *Duration* (1): The duration t of the stroke.
- *Time to previous stroke* (1): The time difference Δt_{prev} to the previous stroke.
- *Time to next stroke* (1): The time difference Δt_{next} to the next stroke.
- *Number of points* (1): The total number of points n.
- *Number of curvature changes* (1): The number of changes $n_{changes}$ in the curvature.
- *Number of up strokes* (1): The number of angles n_l of the writing direction larger than zero.
- *Number of down strokes* (1): The number of angles n_s of the writing direction smaller than zero.

The *stroke-based feature set* thus contains 11 feature values.

4.3. Extended point-based feature set

This feature set describes a set of point-based features found in work on handwriting recognition [78,79] and signature verification [54,73]. It is an extended set of the feature set described in Section 4.1.

For each point p_i, we calculate the following features:

- *x/y-coordinate* (2): The relative x/y-position of the point p_i. The relative x-coordinate is calculated by subtracting the x-coordinate of a point from a moving average coordinate.
- *Speed* (1): The speed v_i of the segment.
- *Speed in x/y-direction* (2): The speed v_{i_x}/v_{i_y} in x/y-direction.
- *Acceleration* (1): The overall acceleration a_i.
- *Acceleration in x/y-direction* (2): The acceleration a_{i_x}/a_{i_y} in x/y-direction.
- *Log curvature radius* (1): The curvature radius is the length of the circle which best approximates the curvature at the

point p_i. It is derived from the local velocities and the local accelerations as follows:

$$r = \frac{(v_{i_x} * a_{i_y} - a_{i_x} * v_{i_y})}{\sqrt{(v_{i_x}^2 + v_{i_y}^2)^3}}.$$

- *Writing direction* (2): The cosine and the sine of the angle between the line segment of the starting point and the x-axis.
- *Curvature* (2): The cosine and the sine of the angle between the lines to the previous and to the next point.
- *Vicinity aspect* (1): The aspect of the trajectory in the vicinity $d_i = \{p_{i-n}, \ldots, p_i, \ldots, p_{i+n}\}$ of the point p_i:

$$va = \frac{\Delta y_{d_i} - \Delta x_{d_i}}{\Delta y_{d_i} + \Delta x_{d_i}}.$$

The vicinity aspect characterizes the ratio of height to width of the bounding box containing the preceding and the succeeding points [78]. Fig. 9 illustrates the computation of this feature. The vicinity of a point is also used to define the following three features: vicinity curliness, vicinity linearity, and vicinity slope.

- *Vicinity curliness* (1): This feature describes the deviation from a straight line in the vicinity d_i (see Fig. 9). It is computed from the length of the trajectory in the vicinity divided by $\max(\Delta x_{d_i}, \Delta y_{d_i})$ [78].
- *Vicinity linearity* (1): The average square distance between every point in the vicinity and the straight line linking the first and the last point in the vicinity [78].
- *Vicinity slope* (2): The cosine and the sine of the angle α_{d_i} of the straight line from the first to the last point in the vicinity (see Fig. 9) [78].

The *extended point-based feature set* thus contains 18 feature values.

4.4. Point-based offline feature set

These features are computed using a two-dimensional matrix representing an off-line version of the data [78]. The matrix is obtained by projecting the on-line strokes on the two-dimensional plane. Therefore all consecutive points within the same stroke are connected. This results in one connected line per stroke. See Fig. 10 for an illustration where the strokes are widened for ease of visualisation.
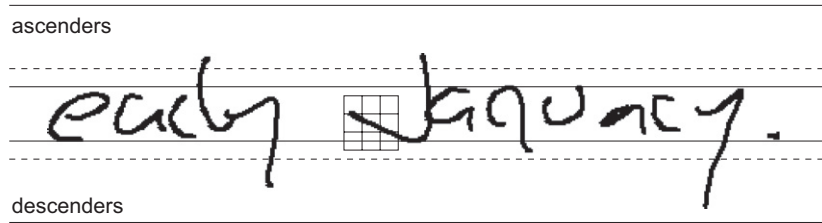
Fig. 10. Illustration of off-line features.

From the two-dimensional representation, the following features are computed:

- *Ascenders/descenders* (2): The number of points above/below the corpus/base line whose *x*-coordinates are in the vicinity of the point and which have a minimal distance to the corpus/base line (denoted by the two dashed lines in Fig. 10). The distance is set to a predefined fraction of the corpus height.
- *Context map* (9): The two-dimensional vicinity of the point is divided into three regions for each dimension (illustrated by the 3×3 matrix in Fig. 10). The number of black points in each region is taken as a feature value.

The *off-line point-based feature set* thus contains 11 feature values.

4.5. All point-based feature set

This feature set is the union of both the *extended point-based feature set* and the *off-line point-based feature set*. In total 29 feature values are extracted.

5. Gaussian mixture models

We use GMMs to model the handwriting of each person of the underlying population. The distribution of the feature vectors extracted from a person's handwriting is modeled by a Gaussian mixture density. For a *D*-dimensional feature vector **x** the mixture density for a specific writer is defined as

$$p(\mathbf{x}|\lambda) = \sum_{i=1}^{M} w_i \, p_i(\mathbf{x}), \tag{1}$$

where the mixture weights w_i sum up to one. The mixture density is a weighted linear combination of *M* uni-modal Gaussian densities $p_i(\mathbf{x})$, each parametrized by a $D \times 1$ mean vector μ_i and a $D \times D$ covariance matrix C_i:

$$p_i(\mathbf{x}) = \frac{1}{(2\pi)^{D/2}|C_i|^{1/2}} \times \exp\left\{-\frac{1}{2}(\mathbf{x} - \mu_i)'(C_i)^{-1}(\mathbf{x} - \mu_i)\right\}. \tag{2}$$

The parameters of a writer's density model are denoted as $\lambda = \{w_i, \mu_i, C_i\}$ for all $i = 1, \ldots, M$. This set of parameters
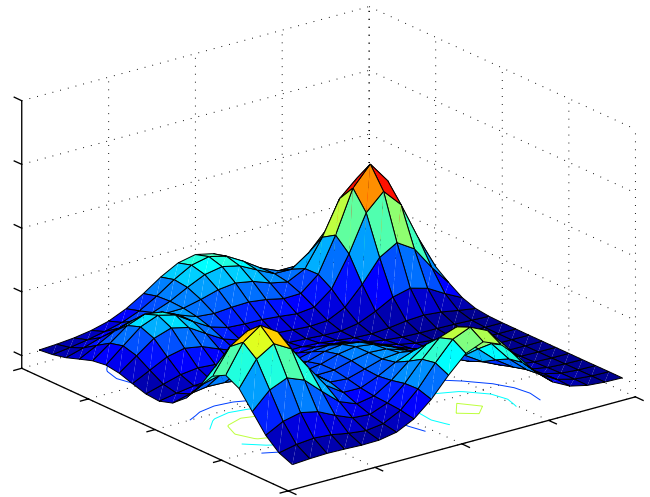

Fig. 11. A two-dimensional GMM consisting of a weighted sum of six uni-modal Gaussian densities.

completely describes the model and enables to concisely model a person's writing on the whiteboard.

While the general model supports full covariance matrices, often only diagonal covariance matrices are used. An example for the two-dimensional case is shown in Fig. 11. This simplification is motivated by the following observations: first, theoretically the density modeling of an *M*-dimensional full covariance matrix can equally well be achieved using a larger order diagonal covariance matrix. Second, diagonal covariance matrices are computationally more efficient than full covariance matrices, and third, diagonal matrix GMMs have outperformed full matrix GMMs in various experiments [18].

The models of the writers are obtained from a *universal background model* (*UBM*). The basic idea is to derive the writer's model by updating the well-trained parameters from the UBM. In a first step, all data from all writers are used to train a single, writer independent UBM. In the second step, for each writer a writer dependent *writer model* is built by updating the parameters in the UBM via adaptation using all training data from this writer.

The UBM is trained using the *expectation–maximization* (*EM*) algorithm [81]. The EM algorithm follows the *maximum likelihood* (*ML*) principle by iteratively refining the parameters of the GMM to monotonically increase the likelihood of the estimated model for the observed feature vectors. The algorithm starts with a data set **X** of *T* feature vectors $\mathbf{x}_t$, an initial set of

M uni-modal Gaussian densities, $N_i \hat{=} N(\boldsymbol{\mu}_i, \mathbf{C}_i)$, and M mixture weights w_i. Then, in the first step, for each training data point $\mathbf{x}_t$ the responsibility $P(i|\mathbf{x}_t)$ of each component N_i is determined. In the second step, the component densities, i.e., the mean vector $\boldsymbol{\mu}_i$ and the variance matrix $\mathbf{C}_i$ for each component, and the weights w_i are re-estimated based on the training data. The model's parameters are updated as follows [82]:

$$\boldsymbol{\mu}_i = \frac{\sum_{t=1}^{T} P(i|\mathbf{x}_t) * \mathbf{x}_t}{\sum_{t=1}^{T} P(i|\mathbf{x}_t)}, \tag{3}$$

$$\boldsymbol{\sigma}_i^2 = \frac{1}{d} \frac{\sum_{t=1}^{T} P(i|\mathbf{x}_t) * \|\mathbf{x}_t - \boldsymbol{\mu}_i\|^2}{\sum_{t=1}^{T} P(i|\mathbf{x}_t)}, \tag{4}$$

$$w_i = \frac{1}{T} \sum_{t=1}^{T} P(i|\mathbf{x}_t), \tag{5}$$

where $\boldsymbol{\sigma}_i$ is the diagonal standard deviation ($\sigma_i(j) = \mathbf{C}_i(j, j)$).

The two steps are repeated until the likelihood score of the entire data set does not change substantially or a limit on the number of iterations is reached.

The Gaussian component densities of the UBM can either be initialized randomly or by using vector quantization techniques such as k-means clustering [83]. Furthermore, variance flooring is employed to avoid an overfitting of the variance parameter [84]. The idea of variance flooring is to impose a lower bound on the variance parameters as a variance estimated from only few data points can be very small and might not be representative of the underlying distribution of the data [84]. The minimal variance value is defined by

$$\min \sigma^2 = \varphi * \sigma_{global}^2, \tag{6}$$

where φ denotes the *variance flooring factor* and the global variance σ_{global}^2 is calculated on the complete training set. The minimal variance, $\min \sigma^2$, is used to initialize the variance parameters of the model. During the EM update step, if a calculated variance parameter is smaller than $\min \sigma^2$, then the variance parameter is set to this value.

The writer models are obtained from the UBM by a modified version of the EM algorithm based on the *maximum a posteriori* (*MAP*) principle. The MAP approach provides a way of incorporating prior information in the training process which is particularly useful for dealing with problems posed by sparse training data for which the ML approach gives inaccurate estimates [85].

Similarly to the EM algorithm, the MAP adaptation algorithm consists of two steps. The first step is identical to the expectation step of the EM algorithm, where estimates of the statistics of the writer's training data are computed for each mixture component in the UBM. Unlike the second step of the EM algorithm, however, for adaptation these new statistical estimates are then combined with the old statistics from the UBM mixture parameters using a data-dependent mixture coefficient. This adaptation coefficient α (called *MAP adaptation factor*) controls the adaptation process by emphasizing either on the well-trained data of the UBM or on the new data when estimating the parameters [18].

To adapt the mean values of the diagonal covariance matrix the new mean $\boldsymbol{\mu}_{i_{\text{Client}}}$ of Gaussian i of the client model is obtained as follows:

$$\boldsymbol{\mu}_{i_{\text{Client}}} = \alpha * \boldsymbol{\mu}_{i_{\text{UBM}}} + (1 - \alpha) * \frac{\sum_{t=1}^{T} P(i|\mathbf{x}_t) * \mathbf{x}_t}{\sum_{t=1}^{T} P(i|\mathbf{x}_t)}, \tag{7}$$

where $\mu_{i_{\text{UBM}}}$ is the corresponding mean in the UBM and $P(i|\mathbf{x}_t)$ is the posterior probability of Gaussian i. The weights $w_{i_{Client}}$ of Gaussian i of the client model are updated from $w_{i_{\text{UBM}}}$ as follows:

$$w_{i_{Client}} = \alpha * w_{i_{\text{UBM}}} + (1 - \alpha) * \frac{\sum_{t=1}^{T} P(i|\mathbf{x}_t)}{\sum_{i=1}^{N} \sum_{t=1}^{T} P(i|\mathbf{x}_t)}. \tag{8}$$

The adaptation of the variance value σ of the diagonal covariance matrix is calculated similarly; for details see Ref. [18].

During decoding, the feature vectors $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$ extracted from a text line are assumed to be independent. The log-likelihood score of a model λ for a sequence of feature vectors $\mathbf{X}$ is defined as

$$\log p(X|\lambda) = \sum_{t=1}^{T} \log p(\mathbf{x}_t|\lambda), \tag{9}$$

where $p(\mathbf{x}_t|\lambda)$ is computed according to Eq. (1).

Diagonal covariance matrices are used and initialized by k-means clustering in our system. The number of clusters equals the number of Gaussian mixture components. The GMMs are implemented using the Torch library [86].

6. Experimental setup

In our experiments we use data from the IAM On-line English Handwritten Text Database (IAM-OnDB)^2 [87]. The IAM-OnDB consists of on-line data acquired from a whiteboard. All texts are taken from the Lancaster–Oslo/Bergen corpus (LOB) which is a large electronic corpus of text [88]. The texts are of diverse nature, ranging from press and popular literature to scientific and religious writing. The resulting database consists of more than 1700 handwritten forms from over 220 writers. It contains over 86,000 word instances with around 11,000 distinct words extracted from more than 13,000 text lines.

The task in our experiments is to identify which person out of 200 individuals has written a given text. It can be argued that even in large organizations there will rarely be more than 200 potential participants to a meeting held in a smart meeting room.

For each writer, there are eight paragraphs of text. A paragraph of text consists of eight text lines in average. Thus there are in total 1600 paragraphs of text consisting of 11,170 text lines. A text line contains 627 points and 24 strokes in average.

---

^{2} The IAM-OnDB is publicly available at the following address: www.iam.unibe.ch/~fki/iamondb.

6.1. Evaluation of feature sets

In the first set of experiments, the five feature sets presented in Section 4 are evaluated on the paragraph as well as on the text line level. In the former case all text lines of a paragraph are used for writer identification, while in the latter case the writer identification is based on the individual text lines. Four fold cross validation is performed which enables us to use all text lines for training without committing the error of training or of optimizing meta parameters on the test set [89].

The experiments on the paragraph level use four paragraphs of text for training, two paragraphs to validate the meta parameters of the GMMs and the remaining two paragraphs form the independent test set. Training, validation and test set are iteratively rotated. All training data from all writers are used to train the UBM. The model of each writer is then obtained by adapting the UBM with writer-specific training data. In this case the mean values of the diagonal covariance matrix of the client model are adapted.

The following two meta-parameters are systematically varied: The number of Gaussian mixture components is increased from 50 to 300 by steps of 50 and the variance flooring factor is increased from 0.001 to 0.031 in steps of 0.002. As there is a high amount of training data available, full adaptation is performed, i.e., the MAP factor is set to 0.0. The other meta parameters are set to standard values [86]. The optimal number of Gaussian mixture components and the optimal variance flooring factor are determined by averaging the writer identification rates achieved on the four validation sets. The final writer identification rate is the average of the writer identification rates on the four test sets.

In order to measure the performance of the identification system if fewer data are available during recognition, we split each paragraph into its individual text lines for the experiments on the line level. The training set, the validation set, and the test set thus consist no longer of the full paragraphs, but of the individual text lines of the paragraphs. While the amount of data available for training is identical to the first experimental setup, the models are optimized and tested on the text line level. The rest of the experimental setup is equal to the first experimental setup, i.e., the number of Gaussian mixture components is varied from 50 to 300 by steps of 50 and the variance flooring factor from 0.001 to 0.031 in steps of 0.002. Both meta-parameters are optimized on the validation sets and the final writer identification rate is calculated on the test sets.

6.2. Evaluation of training methods

In our approach, the client models are obtained from the UBM by adapting the mean values of the diagonal covariance matrix. The variance as well as the weight values are not adapted. The set of experiments described in this section compares the writer identification rates achieved by this approach to two alternative approaches to obtain the client models.

In the first alternative approach, the client models are obtained from the UBM by adapting the weights of the diagonal

covariance matrix. The mean and the variance values of the covariance matrix are not adapted. The number of Gaussian mixture components is varied from 50 to 300 in steps of 50 and the variance flooring factor from 0.001 to 0.031 in steps of 0.002 which is identical to the first series of experiments.

The second alternative approach does not use any UBM at all. Instead, for every writer the client models are trained from scratch using the EM algorithm (see Section 5). Again the number of Gaussian mixture components is increased from 50 to 300 in steps of 50 and the variance flooring factor is varied from 0.001 to 0.031 in steps of 0.002.

Only the *all point-based feature set* on the text line level is considered in this series of experiments, as it shows the best performance in the first set of experiments. The training, the validation, and the test set are identical to the first set of experiments and four-fold cross validation is performed in the same manner.

6.3. Evaluation of reducing training data

The third set of experiments measures the influence of using less data to train the GMMs. In the previous two sets of experiments text from four paragraphs are used for training. In this experimental setup, we reduce the amount of data available for training from four paragraphs to one paragraph in steps of one. The meta parameters considered in this setup are the number of Gaussian mixture components (varied from 50 to 300 by steps of 50) and the MAP adaptation factor (varied from 0.0 to 0.4 in steps of 0.1). In this setup the MAP parameter is varied in order to measure the influence of the UBM on the system's performance which increases when fewer training data are available. The variance flooring factor is set to 0.001.

In this series of experiments no cross validation is performed to reduce the computational complexity. The initial training set consists of four paragraphs, and the validation and the test set each consists of two paragraphs. The *all point-based feature set* is used in this set of experiments. The experiments are performed on the text line level. Only the mean values are updated during adaptation. The rest of the experimental setup is identical to the ones of the previous two sets of experiments.

7. Results and discussion

7.1. Evaluation of feature set—results and discussion

In Tables 1 and 2 the writer identification rates of the different feature sets on the paragraph level are given on the validation and the test sets, respectively. The results for the validation and the test sets on the text line level are given in Tables 3 and 4. In all four tables, the numbers in brackets in the *feature set* column denotes the number of features in a feature vector. The meta parameters, i.e., the number of Gaussian mixture components and the variance flooring factor that achieved the highest writer identification rate on the validation set are given in the second column. The last two columns describe the average writer identification rate and the standard deviation on the validation and on the test set, respectively.

Table 1
Writer identification rates for different feature sets on the paragraph level measured on the validation sets

Feature set	Parameters	Average (%)	Std. dev. (%)
Point-based feature set (5)	50G, 0.001VF	88.19	4.75
Stroke-based feature set (11)	100G, 0.015VF	92.31	4.87
Extended point-based feature set (18)	200G, 0.011VF	95.81	3.03
Off-line point-based feature set (11)	250G, 0.017VF	95.88	2.95
All point-based feature set (29)	300G, 0.007VF	98.31	1.77

Table 2
Writer identification rates for different feature sets on the paragraph level measured on the test sets

Feature set	Parameters	Average (%)	Std. dev. (%)
Point-based feature set (5)	50G, 0.001VF	88.56	3.90
Stroke-based feature set (11)	100G, 0.015VF	92.56	3.01
Extended point-based feature set (18)	200G, 0.011VF	95.75	2.70
Off-line point-based feature set (11)	250G, 0.017VF	96.44	1.60
All point-based feature set (29)	300G, 0.007VF	98.56	1.13

Table 3
Writer identification rates for different feature sets on the text line level measured on the validation sets

| Feature set	Parameters	Average (%)	Std. dev. (%)
Point-based feature set (5)	50G, 0.001VF	49.24	2.27
Stroke-based feature set (11)	100G, 0.001VF	61.80	4.29
Extended point-based feature set (18)	300G, 0.029VF	69.77	4.71
Off-line point-based feature set (11)	300G, 0.029VF	68.60	4.15
All point-based feature set (29)	300G, 0.027VF	86.45	4.19

Table 4
Writer identification rates for different feature sets on the text line level measured on the test sets

Feature set	Parameters	Average (%)	Std. dev. (%)
Point-based feature set (5)	50G, 0.001VF	48.67	3.88
Stroke-based feature set (11)	100G, 0.001VF	62.55	3.29
Extended point-based feature set (18)	300G, 0.029VF	71.84	2.62
Off-line point-based feature set (11)	300G, 0.029VF	71.64	3.35
All point-based feature set (29)	300G, 0.027VF	88.96	2.64
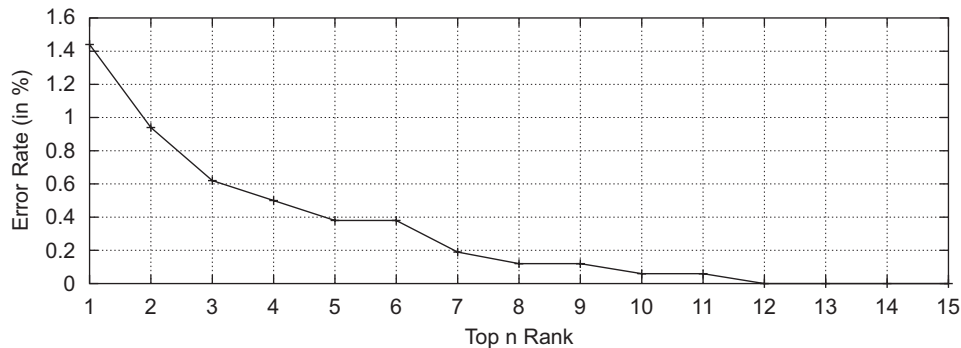
Fig. 12. *n*-Best list for the *all point-based feature set* on the paragraph level.

In Tables 1 and 2 we observe that the results on the validation and on the test set are very similar. Therefore only the results on the test set are discussed in detail. Using the *point-based feature set* and the *stroke-based feature set* writer identification rates below 93.00% are obtained. The *off-line point-based feature set* performs slightly better than the *extended point-based*
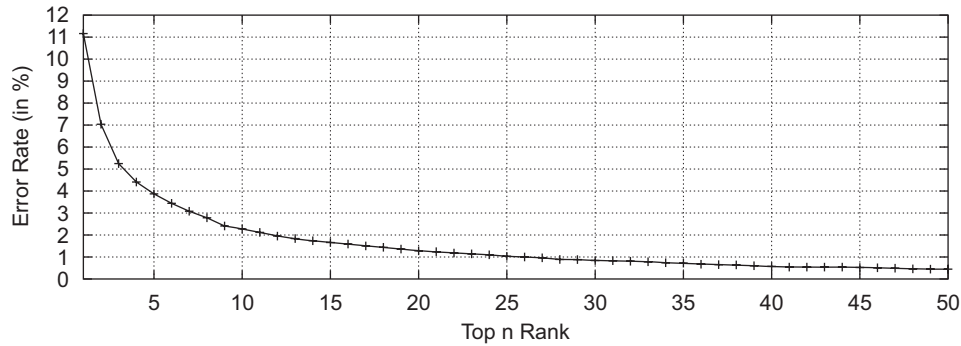
Fig. 13. *n*-Best list for the *all point-based feature set* on the text line level.

Table 5
Evaluation of three different trainings methods to obtain the client models measured on the validation sets

Training method	Parameters	Average (%)	Std. dev. (%)
UBM, learn means	300G, 0.027VF	86.45	4.19
UBM, learn weights	300G, 0.031VF	74.62	4.10
No UBM	300G, 0.001VF	65.75	0.66

Table 6
Evaluation of three different trainings methods to obtain the client models measured on the test sets

Training method	Parameters	Average (%)	Std. dev. (%)
UBM, learn means	300G, 0.027VF	86.96	2.64
UBM, learn weights	300G, 0.031VF	74.43	3.06
No UBM	300G, 0.001VF	66.32	2.41

Table 7
Influence of the amount of data available for training on the writer identification rate, using the *all point-based feature set*

Number of paragraphs	Parameters	Validation set (%)	Test set (%)
One paragraph	150G, 0.2MAP	70.50	69.23
Two paragraphs	300G, 0.2MAP	81.57	80.75
Three paragraphs	300G, 0.1MAP	87.62	86.18
Four paragraphs	250G, 0.0MAP	90.36	87.68

feature set. The highest writer identification rate of 98.56% is obtained using the *all point-based feature set*.

An *n*-best list measures the identification rate not only based on the first rank, but based on the first *n* ranks. As can been seen in Fig. 12, for the *all point-based feature set* the error rate drops below 0.2% if the first seven ranks are considered and all paragraphs are identified correctly if the first twelve ranks are considered.

The writer identification rates calculated on the text line level are shown in Tables 3 and 4 on the validation and on the test sets. Again the results on the validation and on the test sets are very similarly, subsequently only the results on the test sets are discussed. The *point-based feature set* and the *stroke-based feature set* produce rather low writer identification rates. The *extended point-based feature set* yields a statistically significantly

higher result, with a writer identification rate above 71%. The *off-line point-based feature set* and the *extended point-based feature set* show almost the same performance. The highest writer identification rate of 88.96% is again obtained by the *all point-based feature set*.

In Fig. 13, the *n*-best list for the *all point-based feature set* on the text line level is shown. The error rate drops below 4% if the first five ranks are considered, and below 1% if the first 27 ranks are taken into account.

In both experimental setups, the *off-line point-based feature set* consisting of 11 feature values performs similar if not better than the *extended point-based feature set* with a higher number of 18 feature values. This result is surprising in light of the facts that no temporal information is explicitly encoded and that the *off-line point-based feature*
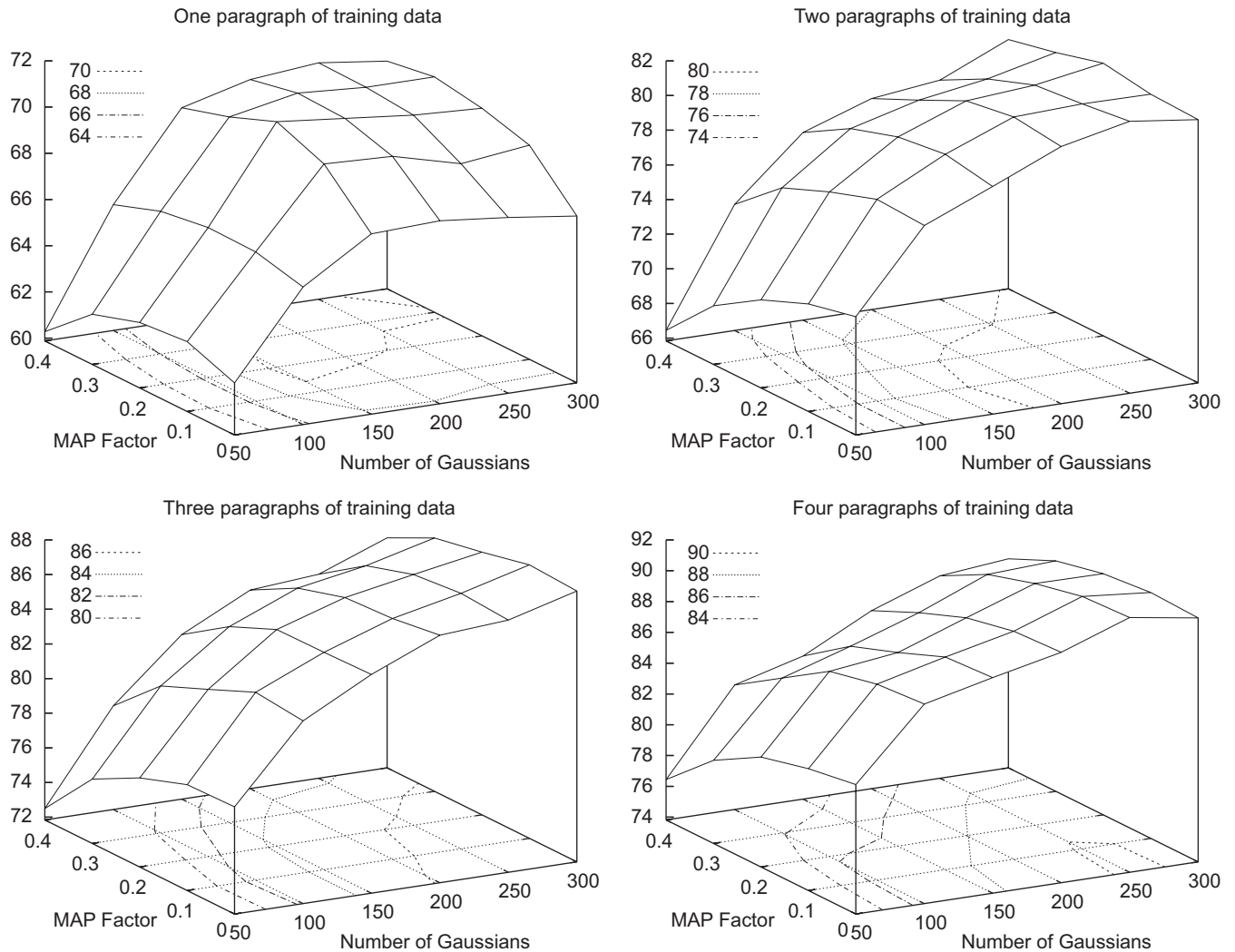
Fig. 14. Writer identification rate as a function of the number of Gaussian mixture components and the MAP adaptation factor for different amounts of training data. The results are reported on the validation set.

set contains fewer feature values than the *extended point-based feature set*.

A stroke consists of 24 points in average. The *stroke-based feature set* thus contains around 24 times fewer feature vectors than the other point-based feature sets. While this leads to fast training and testing times, it only outperforms the *point-based feature set* with a much smaller number of feature values per feature vector.

7.2. Evaluation of training methods—results and discussion

In Tables 5 and 6 the identification rates achieved using three different training methods are shown on the validation sets and the test sets, respectively. As the results on the two sets are very similarly, only the results on the test sets are discussed in the rest of this section.

The first row in Tables 5 and 6 shows the results that are achieved if the client models are obtained by only updating the weights in the adaptation process (denoted by *UBM*, *learn means*). An average identification rate of 88.96% is achieved on the four folds of the training set. This result is achieved by using 300 Gaussian mixture components and a variance flooring factor of 0.027 which was optimized on the validation set. The second row presents the results that are achieved if the client models are obtained by adapting the weights (denoted as *UBM*, *learn weights*). An average identification rate of 74.43% was obtained with 300 Gaussian mixture components and a variance flooring factor of 0.031. In the third row, the identification rate for the case where each client model was trained without the use of an UBM is given (denoted as *No UBM*). An identification rate of 66.32% was obtained on the test set with 300 Gaussian mixture components and a variance flooring factor of 0.001.

The results in Table 6 show that significantly higher writer identification rates are achieved if the client models are obtained by adaptation from an UBM compared to an approach where every client model is trained from scratch. Furthermore, it shows that significantly higher writer identification rates are

achieved if the mean values instead of the weights are updated in the UBM adaptation process. This finding is consistent with the results presented in Ref. [18].

7.3. Evaluation of reducing training data—results and discussion

In Table 7 the results of reducing the number of training data from four paragraphs to one paragraph is shown using the *all point-based feature set*. The parameters in the second row indicate the number of Gaussian mixture components and the MAP adaptation factor which produced the highest writer identification rates on the validation set. The third row presents the results on the validation and the fourth row the results on the test set.

The results show that reducing the number of paragraphs for training from four to three paragraphs does not significantly reduce the writer identification rate. The influence of the UBM on the system's performance is shown in Fig. 14. For each validation experiment, the writer identification rate as a function of the number of Gaussian mixture components and the MAP adaptation factor is plotted with one to four paragraphs of training data. If only one or two paragraphs of text are used for training then the highest writer identification rates are achieved with a MAP adaptation factor of 0.2. If three paragraphs of text are available for training the best result is achieved with a MAP adaptation factor of 0.1. For four paragraphs full adaptation to the writer specific data, i.e., a MAP adaptation factor of 0.0, produces the best writer identification rate.

8. Conclusions and future work

In this paper we present a language and text independent system to identify the writer of on-line handwriting captured from a whiteboard. The task is to determine which person out of a population of 200 individuals has written the text. It can be argued that even in large organizations, there will rarely be more than 200 potential participants to a meeting held in a smart meeting room.

A set of features is extracted from the acquired data and used to train Gaussian mixture models (GMMs). GMMs provide a powerful yet simple means of representing the distribution of the features extracted from handwritten text lines. We use all data to train a universal background model (UBM) and then adapt a specific client model for each writer. During recognition, a text line of unknown origin is presented to each of the models. Each model returns a log-likelihood score for the given input and the text line is assigned to the model which produces the highest score.

Five different feature sets are presented. These sets are either calculated from the single points or the strokes of the writing. The highest writer identification rate is achieved by a point-based feature set consisting of 29 features of which 11 are extracted from an off-line representation of the on-line data. A writer identification rate of 98.56% on the paragraph and of 88.96% on the text line level is achieved.

An experimental evaluation of different training methods to obtain the client models shows that significantly higher writer identification rates are achieved if the client models are obtained by adaptation from an UBM as it is proposed in this paper compared to an approach where every client model is trained from scratch. Furthermore, the results show that significantly higher writer identification rates are achieved if the mean values instead of the weights are updated in the UBM adaptation process.

A third set of experiments measures the influence of having fewer training data available to train the client models. If all training data, i.e., four paragraphs of text, are available, the highest writer identification rates are achieved if full adaptation to the writer specific data is performed, i.e., a MAP adaptation factor of 0.0 is employed. If the amount of training data are reduced, a MAP adaptation factor higher than 0.0 achieves the highest writer identification rates. These results show that the importance of the UBM increases if fewer data are available.

The main motivation of our work is writer identification in a smart meeting room scenario [1–5]. However, there are other potential applications, for example, in text-dependent writer verification [90]. The verification process in this application would work as follows. A writer sends his or her identification to the system. Based on this input, the system generates a random text and presents it to the user. The writer then produces a handwritten version of the transmitted text. The system recognizes the characters of the text and compares the texts. Only if the transcription of the texts is identical writer verification is performed by comparing the submitted characters with the characters of the claimed identification. As different text is presented in every session and recognized before verification the system efficiently prevents copy-based forgery attempts where any text from the claimed writer is used to trick the system.

While our system has been developed for handwriting data acquired by the eBeam whiteboard system, our approach should be easily adaptable to a wide variety of other data by simply modifying or changing the feature set. For example, other on-line handwriting data, e.g., acquired by a digitizing tablet or a Tablet PC can be used [91]. Furthermore, instead of on-line data, off-line data can be applied. The system should also show good performance on signature data, as some of the feature sets are partially inspired by work on signature verification.

The performance of our system can potentially be improved by using feature selection or extraction methods such as SBFS or FDA [92]. Furthermore, in order to reject a handwritten text in case of an uncertain recognition, confidence measures as presented in Ref. [49] can be used. Another interesting topic is to fuse the different feature sets extracted from the on-line whiteboard data [93]. The point-based and the stroke-based feature set contain an unequal number of vectors extracted from the same text. This means that the vectors of the different sets cannot be fused by simply concatenating them to form one vector. Development of suitable feature fusion methods and comparison of their performance with fusion on other levels, such as score or decision level fusion, is left for future work.

Acknowledgments

References

[1] D. Moore, The IDIAP smart meeting room, IDIAP-COM 7, IDIAP, 2002.

[2] A. Waibel, T. Schultz, M. Bett, R. Malkin, I. Rogina, R. Stiefelhagen, J. Yang, SMaRT: the smart meeting room task at ISL, in: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 4, 2003, pp. 752–755.

[3] S. Reiter, G. Rigoll, Segmentation and classification of meeting events using multiple classifier fusion and dynamic programming, in: Proceedings of the 17th International Conference on Pattern Recognition, 2004, pp. 434–437.

[4] L. McCowan, D. Gatica-Perez, S. Bengio, G. Lathoud, M. Barnard, D. Zhang, Automatic analysis of multimodal group actions in meetings, IEEE Trans. Pattern Anal. Mach. Intell. 27 (3) (2005) 305–317.

[5] S.M. Chu, E. Marcheret, G. Potamianos, Automatic speech recognition and speech activity detection in the CHIL smart room, in: Proceedings of the 2nd International Workshop on Machine Learning for Multimodal Interaction, 2005, pp. 332–343.

[6] N. Morgan, D. Baron, J. Edwards, D. Ellis, D. Gelbart, A. Janin, T. Pfau, E. Shriberg, A. Stolcke, The meeting project at ICSI, in: Human Language Technologies Conference, 2001, pp. 246–252.

[7] M. Liwicki, H. Bunke, Handwriting recognition of whiteboard notes, in: Proceedings of the 12th Conference of the International Graphonomics Society, Springer, Berlin, 2005, pp. 118–122.

[8] B. Fasel, J. Luettin, Automatic facial expression analysis: a survey, Pattern Recognition 36 (1) (2003) 259–275.

[9] P. Wellner, M. Flynn, M. Guillemot, Browsing recorded meetings with Ferret, in: Proceedings of the 1st International Workshop on Machine Learning for Multimodal Interaction, 2004, pp. 12–21.

[10] J. Mariéthoz, S. Bengio, A comparative study of adaptation methods for speaker verification, in: Proceedings of the 7th International Conference on Spoken Language Processing, 2002, pp. 581–584.

[11] M.A. Grudin, On internal representations in face recognition systems, Pattern Recognition 33 (7) (2000) 1161–1177.

[12] C. Sanderson, K.K. Paliwal, Fast features for face authentication under illumination direction changes, Pattern Recognition Lett. 24 (14) (2003) 2409–2419.

[13] J. Czyz, S. Bengio, C. Marcel, L. Vandendorpe, Scalability analysis of audio-visual person identity verification, Audio- and Video-Based Biometric Person Authentication (2003) 752–760.

[14] M. Liwicki, H. Bunke, Handwriting recognition of whiteboard notes—studying the influence of training set size and type, Int. J. Pattern Recognition Artif. Intell. 21 (1) (2007) 83–98.

[15] M. Shilman, Z. Wei, S. Raghupathy, P. Simard, D. Jones, Discerning structure from freeform handwritten notes, in: Proceedings of the 7th International Conference on Document Analysis and Recognition, 2003, pp. 60–65.

[16] C.M. Bishop, M. Svensén, G.E. Hinton, Distinguishing text from graphics in on-line handwritten ink, in: Proceedings of the 9th International Workshop on Frontiers in Handwriting Recognition, 2004, pp. 142–147.

[17] D.A. Reynolds, Speaker identification and verification using Gaussian mixture speaker models, Speech Commun. 17 (1995) 91–108.

[18] D.A. Reynolds, T.F. Quatieri, R.B. Dunn, Speaker verification using adapted Gaussian mixture models, Digital Signal Process. 10 (2000) 19–41.

[19] F. Leclerc, R. Plamondon, Automatic signature verification: the state of the art 1989–1993, Int. J. Pattern Recognition Artif. Intell. 8 (3) (1994) 643–660.

[20] R. Plamondon, G. Lorette, Automatic signature verification and writer identification—the state of the art, Pattern Recognition 22 (1989) 107–131.

[21] R. Plamondon, S.N. Srihari, On-line and off-line handwriting recognition: a comprehensive survey, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000) 63–84.

[22] H.E.S. Said, T.N. Tan, K.D. Baker, Personal identification based on handwriting, Pattern Recognition 33 (2000) 149–160.

[23] Y. Zhu, T. Tan, Y. Wang, Biometric personal identification based on handwriting, in: Proceedings of the 15th International Conference on Pattern Recognition, 2000, pp. 797–800.

[24] Z. He, X. You, Y.Y. Tang, B. Fang, J. Du, Handwriting-based personal identification, Int. J. Pattern Recognition Artif. Intell. 20 (2) (2006) 209–225.

[25] S.-H. Cha, S.N. Srihari, Multiple feature integration for writer verification, in: Proceedings of the 7th International Workshop on Frontiers in Handwriting Recognition, 2000, pp. 333–342.

[26] B. Zhang, S.N. Srihari, Binary vector dissimilarity measures for handwriting identification, Document Recognition and Retrieval X 5010 (2003) 28–38.

[27] S.N. Srihari, M.J. Beal, K. Bandi, V. Shah, P. Krishnamurthy, A statistical model for writer verification, in: Proceedings of the 8th International Conference on Document Analysis and Recognition, 2005, pp. 1105–1109.

[28] E.N. Zois, V. Anastassopoulos, Morphological waveform coding for writer identification, Pattern Recognition 33 (2000) 385–398.

[29] C. Hertel, H. Bunke, A set of novel features for writer identification, Audio- and Video-Based Biometric Person Authentication (2003) 679–687.

[30] M. Bulacu, L. Schomaker, L. Vuurpijl, Writer identification using edge-based directional features, in: Proceedings of the 7th International Conference on Document Analysis and Recognition, 2003, pp. 937–941.

[31] L. Schomaker, M. Bulacu, Automatic writer identification using connected-component contours and edge-based features of uppercase western script, IEEE Trans. Pattern Anal. Mach. Intell. 26 (2004) 787–798.

[32] L. Schomaker, M. Bulacu, K. Franke, Automatic writer identification using fragmented connected-component contours, in: Proceedings of the 9th International Workshop on Frontiers in Handwriting Recognition, 2004, pp. 185–190.

[33] X. Wang, X. Ding, H. Liu, Writer identification using directional element features and linear transform, in: Proceedings of the 7th International Conference on Document Analysis and Recognition, 2003, pp. 942–945.

[34] X. Wang, X. Ding, H. Liu, An off-line Chinese writer retrieval system based on text-sensitive writer identification, in: Proceedings of the 18th International Conference on Pattern Recognition, 2006, pp. 517–520.

[35] X. Wang, X. Ding, An effective writer verification algorithm using negative samples, in: Proceedings of the 9th International Workshop on Frontiers in Handwriting Recognition, 2004, pp. 509–513.

[36] A. Bensefia, A. Nosary, T. Paquet, L. Heutte, Writer identification by writer's invariants, in: Proceedings of the 8th International Workshop on Frontiers in Handwriting Recognition, 2002, pp. 274–279.

[37] A. Bensefia, T. Paquet, L. Heutte, Information retrieval based writer identification, in: Proceedings of the 7th International Conference on Document Analysis and Recognition, 2003, pp. 946–950.

[38] A. Bensefia, T. Paquet, L. Heutte, Handwriting analysis for writer verification, in: Proceedings of the 9th International Workshop on Frontiers in Handwriting Recognition, 2004, pp. 196–201.

[39] A. Bensefia, T. Paquet, L. Heutte, A writer identification and verification system, Pattern Recognition Lett. 26 (13) (2005) 2080–2092.

[40] A. Seropian, M. Grimaldi, N. Vincent, Writer identification based on the fractal construction of a reference base, in: Proceedings of the 7th International Conference on Document Analysis and Recognition, 2003, pp. 1163–1167.

[41] A. Seropian, N. Vincent, Writer authentication and fractal compression, in: Proceedings of the 8th International Workshop on Frontiers in Handwriting Recognition, 2002, pp. 434–439.

[42] N. Vincent, V. Boulétreau, R. Sabourin, H. Emptoz, How to use fractal dimensions to qualify writings and writers, Fractals 8 (1) (2000) 85–97.

[43] U.-V. Marti, R. Messerli, H. Bunke, Writer identification using text line based features, in: Proceedings of the 6th International Conference on Document Analysis and Recognition, 2001, pp. 101–105.

[44] G. Leedham, S. Chachra, Writer identification using innovative binarised features of handwritten numerals, in: Proceedings of the 7th International Conference on Document Analysis and Recognition, 2003, pp. 413–417.

[45] A. Schlapbach, H. Bunke, Off-line handwriting identification using HMM based recognizers, in: Proceedings of the 17th International Conference on Pattern Recognition, vol. 2, 2004, pp. 654–658.

[46] A. Schlapbach, H. Bunke, Using HMM based recognizers for writer identification and verification, in: Proceedings of the 9th International Workshop on Frontiers in Handwriting Recognition, 2004, pp. 167–172.

[47] A. Schlapbach, H. Bunke, A writer identification and verification system using HMM based recognizers, Pattern Anal. Appl. 10 (1) (2007) 33–43.

[48] A. Schlapbach, H. Bunke, Off-line writer identification using Gaussian mixture models, in: Proceedings of the 18th International Conference on Pattern Recognition, 2006, pp. 992–995.

[49] A. Schlapbach, H. Bunke, Off-line writer verification: a comparison of a hidden Markov model (HMM) and a Gaussian mixture model (GMM) based system, in: Proceedings of the 10th International Workshop on Frontiers in Handwriting Recognition, 2006, pp. 275–280.

[50] J. Gupta, A. McCabe, A review of dynamic handwritten signature verification, Technical Report, James Cook University, 1997.

[51] G. Lorette, R. Plamondon, Dynamic approaches to handwritten signature verification, Comput. Process. Handwriting (1990) 65–85.

[52] L.L. Lee, T. Berger, E. Aviczer, Reliable on-line human signature verification systems, IEEE Trans. Pattern Anal. Mach. Intell. 18 (6) (1996) 643–647.

[53] V.S. Nalwa, Automatic on-line signature verification, Proc. IEEE 85 (2) (1997) 215–239.

[54] A.K. Jain, F.D. Griess, S.D. Connell, Online signature verification, Pattern Recognition 35 (1) (2002) 2963–2972.

[55] H. Feng, C.C. Wah, Online signature verification using a new extreme points warping technique, Pattern Recognition Lett. 24 (2003) 2943–2951.

[56] L.L. Lee, Neural approaches for human signature verification, in: Proceedings of the 3rd International Conference on Document Analysis and Recognition, vol. 2, 1995, pp. 1055–1058.

[57] Q.-Z. Wu, I.-C. Jou, S.-Y. Lee, On-line signature verification using LPC cepstrum and neural networks, IEEE Trans. Systems, Man Cybern. (Part B) 27 (1) (1997) 148–153.

[58] D.Z. Lejtman, S.E. George, On-line handwritten signature verification using wavelets and back-propagation neural networks, in: Proceedings of the 6th International Conference on Document Analysis and Recognition, 2001, pp. 992–996.

[59] A. Kholmatov, B.A. Yanikoglu, Identity authentication using improved online signature verification method, Pattern Recognition Lett. 26 (15) (2005) 2400–2408.

[60] D.-Y. Yeung, H. Chang, Y. Xiong, S. George, R. Kashi, T. Matsumoto, G. Rigoll, SVC2004: First International Signature Verification Competition, in: Proceedings of the 1st International Conference on Biometric Authentication, 2004, pp. 16–22.

[61] J. Dolfing, E. Aarts, J. van Oosterhout, On-line signature verification with hidden Markov models, in: Proceedings of the 14th International Conference on Pattern Recognition, vol. 2, 1998, pp. 1309–1312.

[62] G. Rigoll, A. Kosmala, A systematic comparison between on-line and off-line methods for signature verification with hidden Markov models, in: Proceedings of the 14th International Conference on Pattern Recognition, 1998, pp. 1755–1757.

[63] L. Yang, B. Widjaja, R. Prasad, Application of hidden Markov models for signature verification, Pattern Recognition 28 (2) (1995) 161–170.

[64] R. Kashi, J. Lu, W.L. Nelson, W. Turin, A hidden Markov model approach to online signature verification, Int. J. Document Anal. Recognition 1 (2) (1998) 102–109.

[65] H.S. Yoon, J.Y. Lee, H.S. Yang, An on-line signature verification system using hidden Markov model in polar space, in: Proceedings of the 8th International Workshop on Frontiers in Handwriting Recognition, 2002, pp. 329–333.

[66] D. Muramatsu, T. Matsumoto, An HMM on-line signature verification algorithm, Audio- and Video-Based Biometric Person Authentication (2003) 233–241.

[67] D. Muramatsu, T. Matsumoto, An HMM on-line signature verification with pen position trajectories, in: Proceedings of the 4th International Conference on Artificial Intelligence, 2003, pp. 299–303.

[68] D. Muramatsu, T. Matsumoto, An HMM signature verifier incorporating signature trajectories, in: Proceedings of the 7th International Conference on Document Analysis and Recognition, 2003, pp. 438–442.

[69] J. Ortega-Garcia, J. Fierrez-Aguillar, J. Martin-Rello, J. Gonzalez-Rodriguez, Complete signal modeling and score normalization for function-based dynamic signature verification, Audio- and Video-Based Biometric Person Authentication (2003) 658–667.

[70] J. Ortega-Garcia, J. Fierrez-Aguilar, D. Simon, J. Gonzalez, M. Faundez-Zanuy, V. Espinosa, A. Satue, I. Hernaez, J.J. Igarza, C. Vivaracho, D. Escudero, Q.I. Moro, MCYT baseline corpus: a bimodal biometric database, IEEE Trans. Vision, Image Signal Process. 150 (6) (2003) 395–401.

[71] J. Richiardi, A. Drygajlo, Gaussian mixture models for on-line signature verification, in: Proceedings of the ACM SIGMM Workshop on Biometrics Methods and Applications, 2003, pp. 115–122.

[72] H. Ketabdar, J. Richiardi, A. Drygajlo, Global feature selection for on-line signature verification, in: Proceedings of the 12th Conference of the International Graphonomics Society, 2005, pp. 59–63.

[73] J. Richiardi, H. Ketabdar, A. Drygajlo, Local and global feature selection for on-line signature verification, in: Proceedings of the 8th International Conference on Document Analysis and Recognition, 2005, pp. 625–629.

[74] G.F. Russell, J. Hu, A. Biem, A. Heilper, D. Markman, Dynamic signature verification using discriminative training, in: Proceedings of the 8th International Conference on Document Analysis and Recognition, 2005, pp. 1260–1264.

[75] L. Wan, B. Wan, Z.-C. Lin, On-line signature verification with two-stage statistical models, in: Proceedings of the 8th International Conference on Document Analysis and Recognition, 2005, pp. 282–286.

[76] J. Chapran, Biometric writer identification: feature analysis and classification, Int. J. Pattern Recognition Artif. Intell. 20 (4) (2006) 483–503.

[77] J. Chapran, M.C. Fairhurst, Biometric writer identification based on the interdependency between static and dynamic features of handwriting, in: Proceedings of the 10th International Workshop on Frontiers in Handwriting Recognition, 2006, pp. 505–510.

[78] S. Jaeger, S. Manke, J. Reichert, A. Waibel, Online handwriting recognition: the NPen++ recognizer, Int. J. Document Anal. Recognition 3 (3) (2001) 169–180.

[79] M. Schenkel, I. Guyon, D. Henderson, On-line cursive script recognition using time delay neural networks and hidden Markov models, Mach. Vision Appl. 8 (1995) 215–223.

[80] M. Liwicki, A. Schlapbach, H. Bunke, S. Bengio, J. Mariéthoz, J. Richiardi, Writer identification for smart meeting room systems, in: Proceedings of the 7th IAPR Workshop on Document Analysis Systems, 2006, pp. 186–195.

[81] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. R. Statist. Soc. 39 (1977) 1–38.

[82] S. Bengio, J. Mariéthoz, Comparison of client model adaptation schemes, IDIAP-RR 25, IDIAP, 2001.

[83] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, Wiley Interscience, New York, 2001.

[84] H. Melin, J. Koolwaaij, J. Lindberg, F. Bimbot, A comparative evaluation of variance flooring techniques in HMM-based speaker verification, in: Proceedings of the 5th International Conference on Spoken Language Processing, 1998, pp. 2379–2382.

[85] J. Gauvain, C. Lee, Maximum a posteriori estimation for multivariate Gaussian mixture observation of Markov chains, IEEE Trans. Speech Audio Process. 2 (1994) 291–298.

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

[87] M. Liwicki, H. Bunke, IAM-OnDB—an on-line English sentence database acquired from handwritten text on a whiteboard, in: Proceedings

of the 8th International Conference on Document Analysis and Recognition, 2005, pp. 956–961.

[88] S. Johansson, The tagged LOB corpus: user's manual, Norwegian Computing Centre for the Humanities, Norway, 1986.

[89] L.I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, Wiley Interscience, New York, 2004.

[90] Y. Yamazaki, T. Nagao, N. Komatsu, Text-indicated writer verification using hidden Markov models, in: Proceedings of the 7th International Conference on Document Analysis and Recognition, 2003, pp. 329–332.

[91] L. Schomaker, From handwriting analysis to pen-computer applications, IEE Electron. Commun. Eng. J. 10 (2) (1998) 93–102.

[92] A. Schlapbach, V. Kilchherr, H. Bunke, Improving writer identification by means of feature selection and extraction, in: Proceedings of the 8th International Conference on Document Analysis and Recognition, 2005, pp. 131–135.

[93] N. Oza, R. Polikar, J. Kittler, F. Roli (Eds.), Multiple Classifier Systems, Lecture Notes in Computer Science, vol. 3541, Springer, Berlin, 2005.

About the Author—ANDREAS SCHLAPBACH received his M.S. degree in Computer Science from the University of Bern, Switzerland, in 2003. Currently he is a Ph.D. student and lecture assistant at the Institute of Computer Science and Applied Mathematics at the University of Bern, Switzerland. His research interests include writer identification and verification and feature extraction and selection techniques.

About the Author—MARCUS LIWICKI received his M.S. degree in Computer Science from the Free University of Berlin, Germany, in 2004. Currently he is a Ph.D. student and lecture assistant at the Institute of Computer Science and Applied Mathematics at the University of Bern, Switzerland. His research interests include on-line and off-line handwriting recognition and document analysis.

About the Author—HORST BUNKE received his M.S. and Ph.D. degrees in Computer Science from the University of Erlangen, Germany. In 1984, he joined the University of Bern, Switzerland, where he is a professor in the Computer Science Department. He was Department Chairman from 1992 to 1996, Dean of the Faculty of Science from 1997 to 1998, and a member of the Executive Committee of the Faculty of Science from 2001 to 2003.
From 1998 to 2000 Horst Bunke served as 1st Vice-President of the International Association for Pattern Recognition (IAPR). In 2000 he was also Acting President of this organization. Horst Bunke is a Fellow of the IAPR, former Editor-in-Charge of the International Journal of Pattern Recognition and Artificial Intelligence, Editor-in-Chief of the journal Electronic Letters of Computer Vision and Image Analysis, Editor-in-Chief of the book series on Machine Perception and Artificial Intelligence by World Scientific Publ. Co., Advisory Editor of Pattern Recognition, Associate Editor of Acta Cybernetica and Frontiers of Computer Science in China, and Former Associate Editor of the International Journal of Document Analysis and Recognition, and Pattern Analysis and Applications.
Horst Bunke received an honorary doctor degree from the University of Szeged, Hungary, and held visiting positions at the IBM Los Angeles Scientific Center (1989), the University of Szeged, Hungary (1991), the University of South Florida at Tampa (1991, 1996, 1998–2006), the University of Nevada at Las Vegas (1994), Kagawa University, Takamatsu, Japan (1995), Curtin University, Perth, Australia (1999), and Australian National University, Canberra (2005).
He served as a co-chair of the 4th International Conference on Document Analysis and Recognition held in Ulm, Germany, 1997 and as a Track Co-Chair of the 16th and 17th International Conference on Pattern Recognition held in Quebec City, Canada and Cambridge, UK in 2002 and 2004, respectively. Also he was chairman of the IAPR TC2 Workshop on Syntactic and Structural Pattern Recognition held in Bern 1992, a co-chair of the 7th IAPR Workshop on Document Analysis Systems held in Nelson, NZ, 2006, and a co-chair of the 10th International Workshop on Frontiers in Handwriting Recognition, held in La Baule, France, 2006. Horst Bunke was on the program and organization committee of many other conferences and served as a referee for numerous journals and scientific organizations. He is on the Scientific Advisory Board of the German Research Center for Artificial Intelligence (DFKI). Horst Bunke has more than 550 publications, including 36 authored, co-authored, edited or co-edited books and special editions of journals.