A Writer Identification System for On-line Whiteboard Data

Andreas Schlapbach [a,*], Marcus Liwicki [a], Horst Bunke [a]

[a] *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 99.25% on the paragraph and of 90.30% on the text line level is achieved.

Key words: writer identification, on-line handwriting, Gaussian mixture models, smart meeting rooms

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 [1,2,3,4,5]. To record a meeting, Smart Meeting Rooms are equipped with synchronized recording interfaces to capture audio, video, and handwritten notes.

* Corresponding author.
Email addresses: schlpbch@iam.unibe.ch (Andreas Schlapbach), liwicki@iam.unibe.ch (Marcus Liwicki), bunke@iam.unibe.ch (Horst Bunke).

Preprint submitted to Pattern Recognition *9 August 2006*

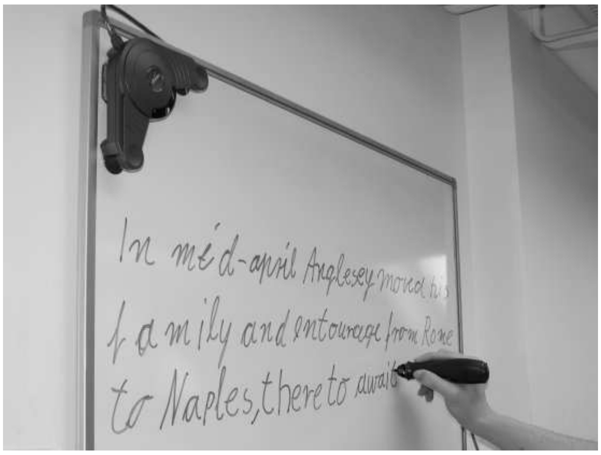

Fig. 1. Recording session with the data acquisition device positioned in the upper left corner of the whiteboard.

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. 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. An illustration of the data acquisition device is shown in Fig. 1.

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

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

of uni-modal Gaussian densities. While GMMs have first been used in speech recognition [14,15], 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 Gaussian Mixture Models (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 work has been done in related fields, such as writer identification and verification of on-line and off-line pen data and signature verification. Surveys covering work in automatic writer identification and signature verification until 1993 are given in [16,17].

Writer identification and verification can be performed on-line, where temporal and spatial information about the writing is available, or off-line, where only a scanned image of the handwriting is available. Recently, different approaches to off-line writer identification and verification have been proposed.

Said et al. [18] 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.

Srihari et al. [19,20] 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. [21] 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 are then applied to the feature vectors.

In Hertel et al. [22] 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. [23] 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 [24] 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 [25], using fragmented connected-component contours (FCO^3).

In a number of papers [26,27,28,29] 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 [28,29] these graphemes are used to address the writer verification task based on text blocks as well as on handwritten words.

Leedham et al. [30] present a set of eleven features which can be extracted easily and used for the identification and 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 [31,32,33] and Gaussian Mixture Models (GMMs) [34,35] 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 recognition rate 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.

A special case of on-line writer identification is signature verification. While signatures differ from normal handwritten texts in the sense that they are more graphical than text, and that letters are often deformed or missing, the underlying methods to identify handwritten text and signature are identical to a great extent.

Early work on on-line signature verification is described in [16,36,37]. In recent works, various approaches based on Dynamic Time Warping, Neural Networks, Hidden Markov Models and Gaussian Mixture Models have been elaborated.

A dynamic time warping approach is presented in [38]. 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 [39].

In [40] three different neural network based approaches for on-line human signature verification are studied. An on-line signature verification system that uses multi-layer perceptrons trained with cepstral coefficients derived from linear predictor coefficients of the writing trajectories is presented in [41]. In [42] a signature verification system that uses wavelets and back-propagation neural networks is proposed.

Kholmatov and Yanikoglu consider the signature verification problem as a two-class pattern recognition problem [43]. A test signature's authenticity is established by first aligning it with each reference signature for the claimed user, using dynamic time warping. 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 [44].

HMMs have been applied to signature verification for a long time [45,46,47,48]. In a number of papers, Muramatsu and Matsumoto [49,50,51] propose a new on-line signature verification algorithm by mapping the trajectory angles to HMM states.

Richiardi et. al introduce GMMs for on-line signature verification [52]. Furthermore, they propose a signature feature selection algorithm that combines a modified Fisher ratio cost function and a sub-optimal but fast floating search

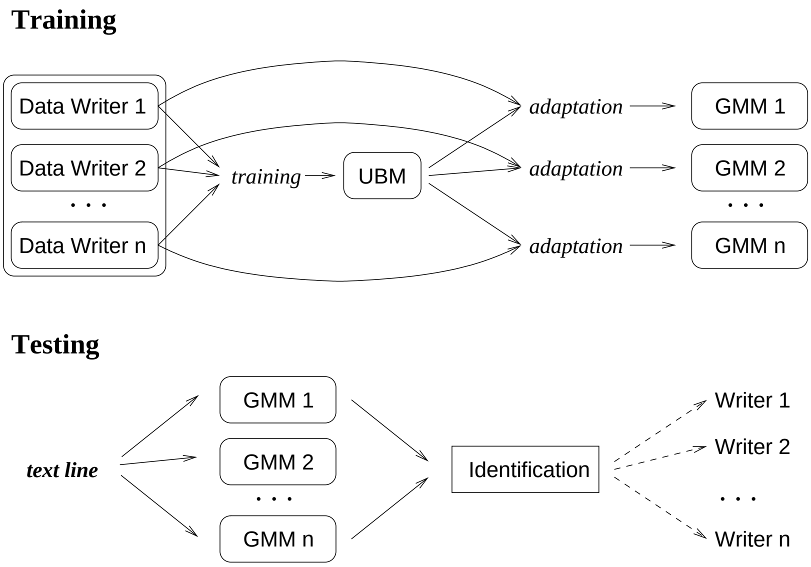

Fig. 2. Schematic overview of the training and the testing phase.

algorithm to obtain an initial feature set of local [53] and global features [54].

In [55] the authentic and the forgery samples are represented by two separate GMMs. Dissimilarity vectors are obtained after the initial vectors have been aligned by Dynamic Time Warping. 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 [56].

We first proposed to use GMMs for writer identification of on-line whiteboard data in [57]. The current paper is a substantially extended version of [57]. Three new features sets leading to significantly improved writer identification rates are introduced and the influence of having fewer data available for training as well as testing are systematically studied.

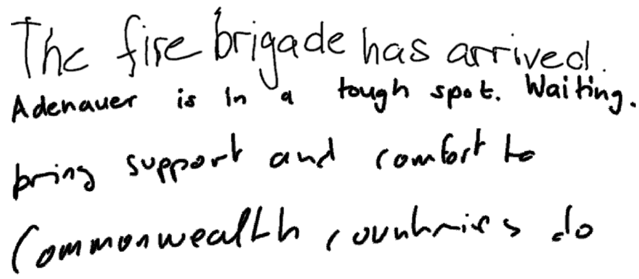

Fig. 3. Examples of handwritten texts acquired by the electronic acquisition device from the whiteboard.

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 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 is acquired. Furthermore, whiteboard data often have a wave

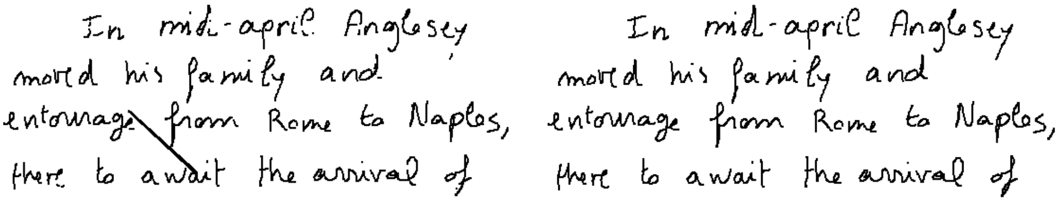

Fig. 4. A text paragraph before and after preprocessing.

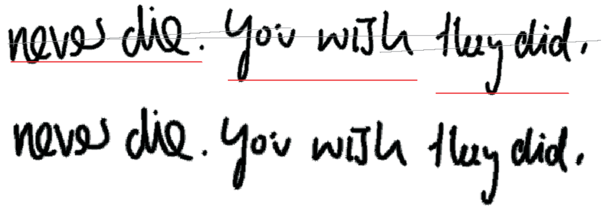

Fig. 5. Splitting a text line into its sub-parts.

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 people stand, rather than sit, during writing and 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). To recover from artifacts of this kind, 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. 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. In this case 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., from gaps within strokes, we check if the distance between the timestamps of p_n and q_1 is below a fixed threshold. If the condition holds the strokes are merged into one single stroke.

The cleaned paragraph of text is then automatically divided into lines using a simple heuristics. If there is a pen-movement to the left and down greater than a predefined threshold the start of a new line is assumed.

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).

For each sub-part the skew angle is calculated and corrected. Linear regression

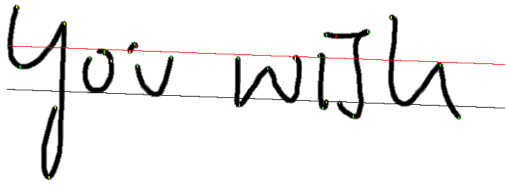

Fig. 6. Baseline and corpus line of an example part of a text line.

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 corpus line of 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.

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.

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 [57]. 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.

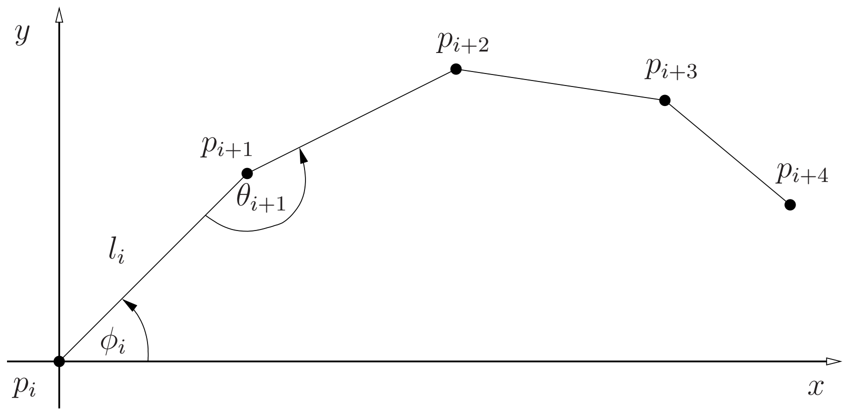

Fig. 7. Illustration of point-based features.

4.1 Point-based Feature Set

The features of this feature set are similar to the ones used in on-line handwriting recognition systems [58,59] and signature verification systems [38,54]. 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 (for an illustration see Fig. 7):

- *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 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 5 feature values.

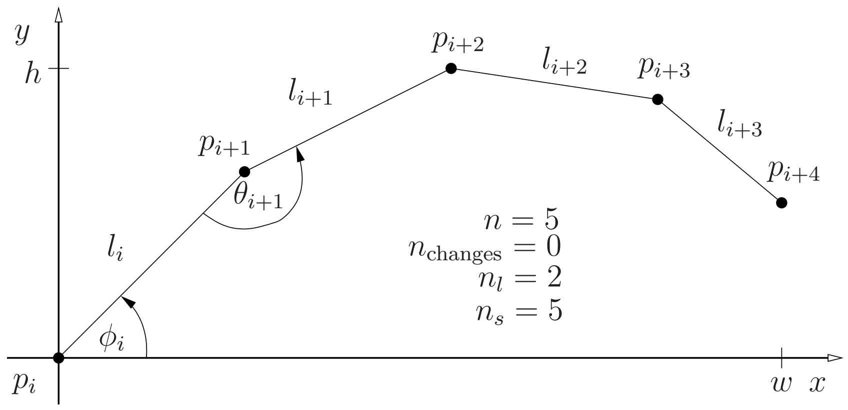

Fig. 8. Illustration of stroke-based features.

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:

$$l_{acc} = \sum_{i=1}^{n-1} l_i$$

- *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.

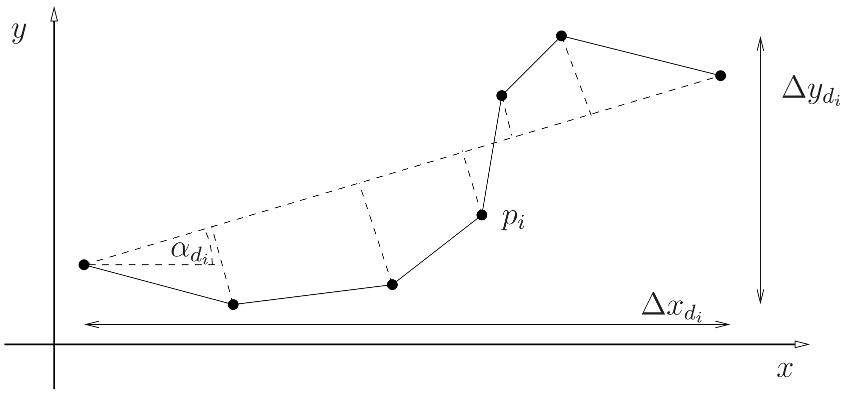

Fig. 9. Illustration of vicinity features.

4.3 Extended Point-based Feature Set

This feature set describes a set of point based features found in work on handwriting recognition [58,59] and signature verification [38,54]. It is an extended set of the feature set described in Sec. 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 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:

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

It characterizes the ratio of height to width of the bounding box con-

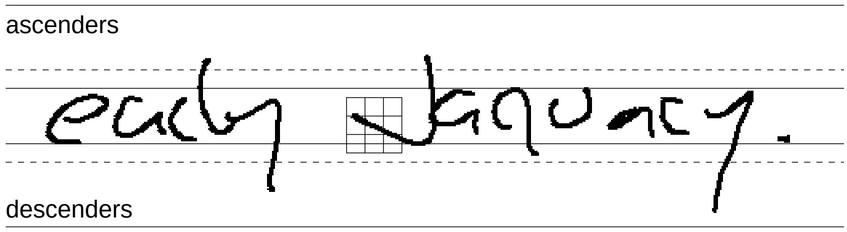

Fig. 10. Illustration of off-line features.

taining the preceding and the succeeding points [58]. 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 $\max(\Delta x_{d_i}, \Delta y_{d_i})$ [58].
- *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 [58].
- *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) [58].

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 [58]. The matrix is obtained by projecting the on-line strokes on the two-dimensional plane (see Fig. 10 for an illustration). The following features are used:

- *ascenders/descenders (2):* the number of points above/below the corpus 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 Gaussian Mixture Models (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 $\mathbf{x}$ 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:

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

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 completely describes the model and enables us 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 of 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 outperformed full matrix GMMs in various experiments [15].

Instead of training a writer model from scratch for every writer, we obtain the models of the writers 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 is used to train a single, writer independent UBM. In the second step, for each writer a writer

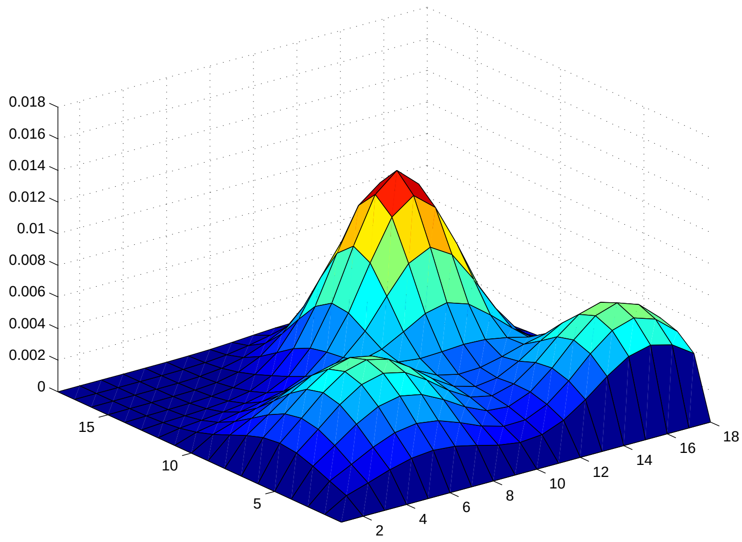

Fig. 11. A two-dimensional GMM consisting of a weighted sum of three uni-modal Gaussian densities.

dependent *writer model* is build 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 [60]. 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 N feature vectors $\mathbf{x}_n$, an initial set of M uni-modal Gaussian densities, $N_i \hat{=} N(\mu_i, C_i)$, and M mixture weights w_i. Then, in the first step, for each training data point x_n the responsibility $P(i|x_n)$ of each component N_i is determined. In the second step, the component densities, i.e., the mean vector μ_i and the variance matrix C_i for each component, and the weights w_i are re-estimated based on the training data. These 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 [61]. Furthermore, variance flooring is employed to avoid an overfitting of the variance parameter [62]. 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 [62]. The minimal variance value is defined by

$$\min \sigma^2 = \alpha * \sigma^2_{global} \tag{3}$$

where α denotes the *variance flooring factor* and the global variance σ^2_{global} 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 [63].

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 (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 [15].

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

$$\log p(X|\lambda) = \sum_{t=1}^{T} \log p(\mathbf{x}_t|\lambda), \qquad (4)$$

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 [11].

6 Experimental Setup

In our experiments we use data from the IAM On-line English Handwritten Text Database (IAM-OnDB) [2] [64]. 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 [65].

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

The texts are of diverse nature, ranging from press and popular literature to scientific and religious writing. The resulting database consists of more than 1,700 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.

In our experiments, we use writing data from 200 different writers. The task is to identify which person out of these 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.

The first experimental setup is based on paragraphs of text. For each writer, we have eight paragraphs of text. Four paragraphs of text are used for training, two paragraphs are used to validate the meta parameters of the GMMs and the remaining two paragraphs form the independent test set.

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. 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 [66]. The optimal number of Gaussian mixture components and the optimal variance factor are determined on the validation set and these values are then used to compute the final identification rate on the test set.

A paragraph of text consists of eight text lines in average. To measure the performance of the system if fewer data is available during recognition, we split each paragraph into its individual text lines for the second set of experiments. The training set, the validation set, and the test set thus consist no longer of full paragraphs, but of the 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 set and the final writer identification rate is calculated on the test set.

In the third experimental setup, we measure 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

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

Feature Set	Validation Set	Test Set
point-based feature set (5)	93.00% (50, 0.001)	87.50%
stroke-based feature set (11)	96.00% (150, 0.029)	91.75%
extended point-based feature set (18)	99.00% (250, 0.001)	97.25%
off-line point-based feature set (11)	97.50% (250, 0.025)	95.50%
all point-based feature set (29)	99.75% (250, 0.023)	99.25%

steps of one. The meta parameters considered in this setup are the number of Gaussian mixture components (from 50 to 300 by steps of 50) and the MAP adaptation factor (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. The rest of the experimental setup is identical to the previous two experimental setups.

7 Results and Discussion

In Tables 1 and 2 the writer identification rates of the different feature sets are given on the paragraph and the text line level, respectively. The number in brackets in the feature set column denotes the number of features in a feature vector. The number in brackets in the validation set column denotes the number of Gaussian mixture components and the variance flooring factor that achieved the highest writer identification rate on the validation set. The last column describes the results achieved on the test set.

Table 1 gives the results of the experiments on the paragraph level. Using the *point-based feature set* and the *stroke-based feature set* writer identification rates below 92.00% are obtained. The *extended point-based feature set* performs slightly better than the *off-line point-based feature set*. The highest writer identification rate of 99.25% is obtained using the *all point-based feature set*. In this case, only three text paragraphs are classified incorrectly. If not only the first but the first nine out of a total of 200 ranks are considered then all paragraphs are identified correctly.

The writer identification rates shown in Table 2 are calculated on the text line level. 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 better result, with a writer identification rate above 73%. The *off-line point-based feature set* performs slightly worse

Table 2
Writer identification rates for different feature sets on the line level.

Feature Set	**Validation Set**	**Test Set**
point-based feature set (5)	52.64% (50, 0.001)	47.50%
stroke-based feature set (11)	66.10% (150, 0.001)	62.61%
extended point-based feature set (18)	76.05% (300, 0.019)	73.01%
off-line point-based feature set (11)	75.67% (300, 0.021)	71.27%
all point-based feature set (29)	91.86% (300, 0.029)	90.30%

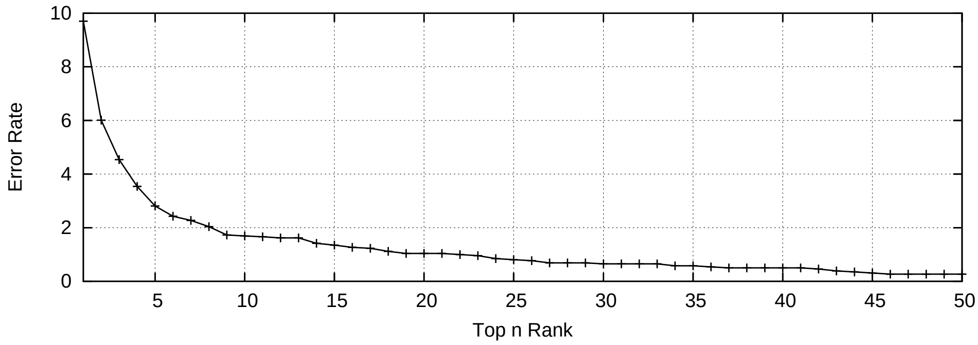

Fig. 12. n-best list for the *all point-based feature set* on the text line level.

than the *extended point-based feature set*. The highest writer identification rate of 90.30% is again obtained by 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. In Fig. 12, for the *all point-based feature set* the n-best list is shown. The error rate drops from 9.70% to below 2% considering the first nine ranks, and below 1% if the first 23 ranks are taken into account.

In both experimental setups, the *off-line point-based feature set* consisting of 11 feature values performs only slightly inferior than the *extended point-based feature set* with a higher number of 18 feature values. This result is surprising in the light of the fact that no temporal information is explicitly encoded in the *off-line point-based feature set*.

A stroke consists of 21 points in average. The *stroke-based feature set* thus contains around 21 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.

In Table 3, using the *all point-based feature set*, the results of reducing the number of training data from four paragraphs to one paragraph is shown. The numbers in brackets indicate the number of Gaussian mixture components

Table 3
Influence of the amount of data available for training on the writer identification rate, using the *all point-based feature set*.

Number of Paragraphs	**Validation Set**	**Test Set**
one paragraph	70.50% (150, 0.2)	69.23%
two paragraphs	81.57% (300, 0.2)	80.75%
three paragraphs	87.62% (300, 0.1)	86.18%
four paragraphs	90.36% (250, 0.0)	87.68%

and the MAP factor tested which produced the highest writer identification rates on the validation 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. 13. For each validation experiment with one to four paragraphs of training data, the writer identification rate as a function of the number of Gaussian mixture components and the MAP adaptation factor is plotted. If only one or two paragraphs of text are used for training then the highest writer identification rates are achieved with an 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. A set of features is extracted from the acquired data and used to train Gaussian mixture models (GMM). 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.

We use writing data from 200 different writers. The task is to determine which person out of the 200 individuals has written the text. It can be argued that even in large organizations, there will rarely be more than 200 potential par-

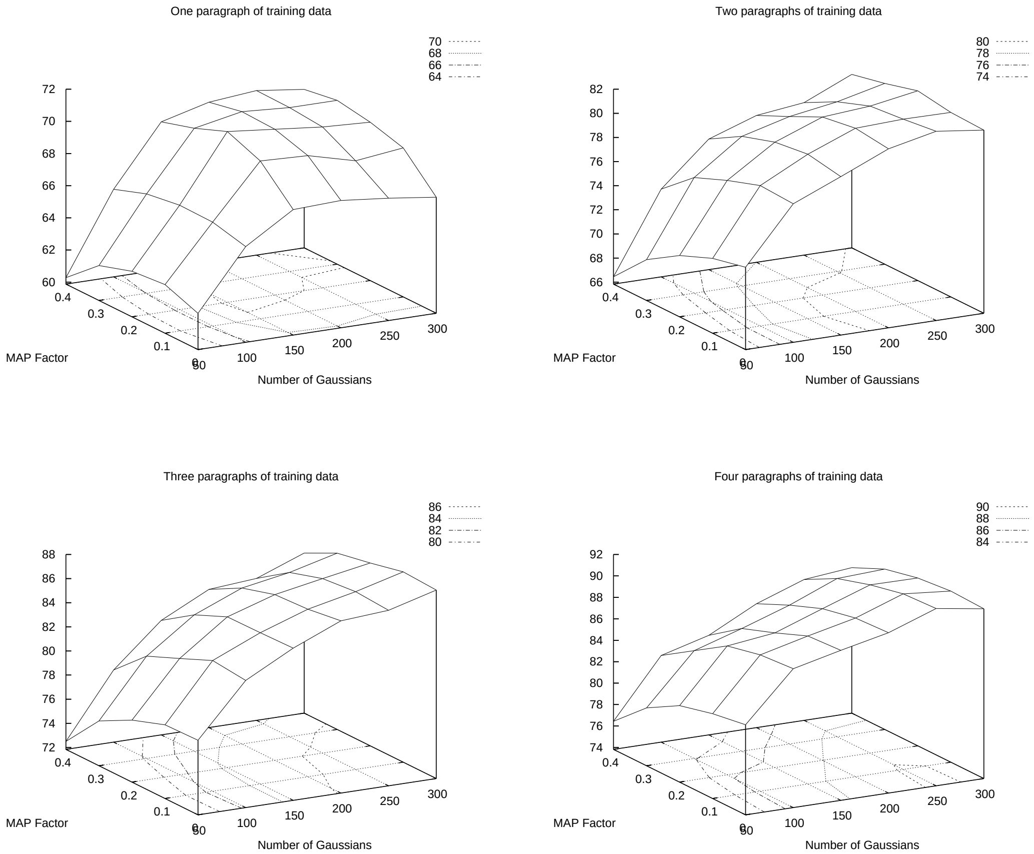

Fig. 13. Writer identification rate as a function of the number of Gaussian mixture components and the MAP adaptation factor for different amounts of training data.

ticipants to a meeting held in a smart meeting room.

Five different feature sets are presented. The feature sets are either calculated from the single points or the strokes of the writing. The highest writer identification rate was achieved by a point-based feature set consisting of 29 feature values of which 11 features are extracted from an off-line representation of the on-line data. A writer identification rate of 99.25% on the paragraph and of 90.30% on the text line level is achieved. We further studied the influence of having fewer data available for training.

While our system has been developed for handwriting data acquired by the eBeam whiteboard system, our approach can easily also be applied to other on-line handwritten data, e.g., data acquired by a digitizing tablet or a Tablet PC [67].

The performance of our system can potentially be improved by using feature selection or extraction methods such as SBFS or FDA [68]. Furthermore, in order to reject a handwritten text in case of an uncertain recognition confidence measures as presented in [35] have to be developed. Another interesting topic is to fuse the different feature sets extracted from the on-line whiteboard data [69]. 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 can not 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

This work is supported by the Swiss National Science Foundation program “Interactive Multimodal Information Management (IM2)” in the Individual Project “Visual/Video Processing”.

References

[1] D. Moore, The IDIAP smart meeting room, Tech. rep., IDIAP-Com (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: Proc. IEEE Int. Conf. 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: Proc. 17th Int. Conf. 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. on Pattern Analysis and Machine Intelligence 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: Proc. 2nd Int. 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, Human Language Technologies Conference (2001) 246–252.

[7] M. Liwicki, H. Bunke, Handwriting recognition of whiteboard notes, in: Proc. 12th Conf. of the Int. Graphonomics Society, 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: Proc. 1st Int. Workshop on Machine Learning for Multimodal Interaction, 2004, pp. 12–21.

[10] Mariéthoz, J., Bengio, S., A comparative study of adaptation methods for speaker verification, in: Proc. 7th Int. Conf. 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 Letters 24 (14) (2003) 2409–2419.

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

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

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

[16] F. Leclerc, R. Plamondon, Automatic signature verification: The state of the art 1989–1993, in: R. Plamondon (Ed.), Progress in Automatic Signature Verification, World Scientific Publ. Co., 1994, pp. 13–19.

[17] R. Plamondon, G. Lorette, Automatic signature verification and writer identification – the state of the art, in: Pattern Recognition, Vol. 22, 1989, pp. 107–131.

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

[19] S.-H. Cha, S. Srihari, Multiple feature integration for writer verification, in: Proc. 7th Int. Workshop on Frontiers in Handwriting Recognition, 2000, pp. 333–342.

[20] B. Zhang, S. N. Srihari, S. Lee, Individuality of handwritten characters, in: Proc. 7th Int. Conf. on Document Analysis and Recognition, Vol. 7, 2003, pp. 1086–1090.

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

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

[23] M. Bulacu, L. Schomaker, L. Vuurpijl, Writer identification using edge-based directional features, in: Proc. 7th Int. Conf. on Document Analysis and Recognition, 2003, pp. 937–941.

[24] L. Schomaker, M. Bulacu, Automatic writer identification using connected-component contours and edge-based features of uppercase western script, IEEE Trans. on Pattern Analysis and Machine Intelligence 26 (2004) 787–798.

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

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

[27] A. Bensefia, T. Paquet, L. Heutte, Information retrieval based writer identification, in: Proc. 7th Int. Conf. on Document Analysis and Recognition, 2003, pp. 946–950.

[28] A. Bensefia, T. Paquet, L. Heutte, Handwriting analysis for writer verification, in: Proc. 9th Int. Workshop on Frontiers in Handwriting Recognition, 2004, pp. 196–201.

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

[30] G. Leedham, S. Chachra, Writer identification using innovative binarised features of handwritten numerals, in: Proc. 7th Int. Conf. on Document Analysis and Recognition, 2003, pp. 413–417.

[31] A. Schlapbach, H. Bunke, Off-line handwriting identification using HMM based recognizers, in: Proc. 17th Int. Conf. on Pattern Recognition, Vol. 2, 2004, pp. 654–658.

[32] A. Schlapbach, H. Bunke, Using HMM based recognizers for writer identification and verification, in: Proc. 9th Int. Workshop on Frontiers in Handwriting Recognition, 2004, pp. 167–172.

[33] A. Schlapbach, H. Bunke, A writer identification and verification system using HMM based recognizers, Pattern Analysis and Applications. Accepted for publication.

[34] A. Schlapbach, H. Bunke, Off-line writer identification using Gaussian mixture models, in: Proc. 18th Int. Conf. on Pattern Recognition, 2006.

[35] 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: Proc. 10th Int. Workshop on Frontiers in Handwriting Recognition, 2006.

[36] J. Gupta, A. McCabe, A review of dynamic handwritten signature verification, Tech. rep., Departement of Computer Science, James Cook University, Australia (1997).

[37] G. Lorette, R. Plamondon, Dynamic approaches to handwritten signature verification, Computer Processing of Handwriting (1990) 65–85.

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

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

[40] L. L. Lee, Neural approaches for human signature verification, in: Proc. 3rd Int. Conf. on Document Analysis and Recognition, Vol. 2, 1995, pp. 1055–1058.

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

[42] D. Z. Lejtman, S. E. George, On-line handwritten signature verification using wavelets and back-propagation neural networks, in: Proc. 6th Int. Conf. on Document Analysis and Recognition, 2001, pp. 992–996.

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

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

[45] J. Dolfing, E. Aarts, J. van Oosterhout, On-line signature verification with hidden Markov models, in: Proc. 14th Int. Conf. on Pattern Recognition, Vol. 2, 1998, pp. 1309–1312.

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

[47] G. Rigoll, A. Kosmala, A systematic comparison between on-line and off-line methods for signature verification with hidden Markov models, in: Proc. 14th Int. Conf. on Pattern Recognition, Vol. 2, 1998, pp. 1755–1757.

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

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

[50] D. Muramatsu, T. Matsumoto, An HMM on-line signature verification with pen position trajectories, in: Proc. 4th Int. Conf. on Artificial Intelligence, 2003, pp. 299–303.

[51] D. Muramatsu, T. Matsumoto, An HMM signature verifier incorporating signature trajectories, in: Proc. 7th Int. Conf. on Document Analysis and Recognition, 2003, pp. 438–442.

[52] J. Richiardi, A. Drygajlo, Gaussian mixture models for on-line signature verification, in: Proc. ACM SIGMM workshop on Biometrics methods and applications, 2003, pp. 115–122.

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

[54] J. Richiardi, H. Ketabdar, A. Drygajlo, Local and global feature selection for on-line signature verification, in: Proc. 8th Int. Conf. on Document Analysis and Recognition, 2005, pp. 625–629.

[55] G. F. Russell, J. Hu, A. Biem, A. Heilper, D. Markman, Dynamic signature verification using discriminative training, in: Proc. 8th Int. Conf. on Document Analysis and Recognition, 2005, pp. 1260–1264.

[56] L. Wan, B. Wan, Z.-C. Lin, On-line signature verification with two-stage statistical models, in: Proc. 8th Int. Conf. on Document Analysis and Recognition, 2005, pp. 282–286.

[57] M. Liwicki, A. Schlapbach, H. Bunke, S. Bengio, J. Mariéthoz, J. Richiardi, Writer identification for smart meeting room systems, in: Proc. 7th IAPR Workshop on Document Analysis Systems, Vol. 3872 of LNCS, Springer, 2006, pp. 186–195.

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

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

[60] A. P. Dempster, N. M. Laird, D. B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of Royal Statistical Society 39 (1977) 1–38.

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

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

[63] J. Gauvain, C. Lee, Maximum a-posteriori estimation for multivariate Gaussian mixture observation of Markov chains, IEEE Trans. on Specch & Audio Processing 2 (1994) 291–298.

[64] M. Liwicki, H. Bunke, IAM-OnDB – an on-line English sentence database acquired from handwritten text on a whiteboard, in: Proc. 8th Int. Conf. on Document Analysis and Recognition, Vol. 2, 2005, pp. 956–961.

[65] S. Johansson, The tagged LOB Corpus: User's Manual, Norwegian Computing Centre for the Humanities, Norway, 1986.

[66] R. Collobert, S. Bengio, J. Mariéthoz, Torch: a modular machine learning software library, Tech. rep., IDIAP (2002).

[67] L. Schomaker, From handwriting analysis to pen-computer applications, IEE Electronics & Communication Engineering Journal 10 (2) (1998) 93–102.

[68] A. Schlapbach, V. Kilchherr, H. Bunke, Improving writer identification by means of feature selection and extraction, in: Proc. 8th Int. Conf. on Document Analysis and Recognition, 2005, pp. 131–135.

[69] N. Oza, R. Polikar, J. Kittler, F. Roli (Eds.), Multiple Classifier Systems, 6th Int. Workshop, Springer LNCS 3541, 2005.