Fusing Asynchronous Feature Streams for On-line Writer Identification

Andreas Schlapbach and Horst Bunke
Institute of Computer Science and Applied Mathematics
Neubrückstrasse 10, CH-3012 Bern, Switzerland
{schlpbch, bunke}@iam.unibe.ch

Abstract

In this paper, we present a new approach to improving the performance of a writer identification system by fusing asynchronous feature streams. Different feature streams are extracted from on-line handwritten text acquired from a whiteboard. The feature streams are used to train a text and language independent writer identification system based on Gaussian Mixture Models (GMMs). From a stroke consisting of n points, n point-based feature vectors and one stroke-based feature vector are extracted. The resulting feature streams thus have an unequal number of feature vectors. We evaluate different methods to directly fuse the feature streams and show that, by means of feature fusion, we can improve the performance of the writer identification system on a data set produced by 200 different writers.

Keywords: writer identification, feature fusion, on-line handwriting

1 Introduction

The work described in this paper has been conducted in the context of research on Smart Meeting Rooms [12]. The aim of this research is to automate standard tasks usually performed by humans in a meeting [1, 11]. To record a meeting, Smart Meeting Rooms are equipped with synchronized recording interfaces to capture audio, video, and handwritten notes.

An important task in a Smart Meeting Room is to capture the handwriting rendered on a whiteboard during a meeting and to identify the author of the text. Having identified the author enables us to label the handwriting with the writer's identity. Furthermore, it allows us to validate the identification results of video- or audio-based person identification systems within the Smart Meeting Room scenario.

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 lines written by a person. The models have a mathematically simple, well understood structure and there exist standard algorithms for training and testing.

GMMs have originally been used in speech recognition [14]. Richiardi et. al introduce GMMs for on-line signature verification [15]. For off-line writer identification of text lines GMMs are used in [19].

In [9] we have introduced a language and text independent GMM-based system to identify the writer of on-line handwriting captured from a whiteboard. The system is trained using point- and stroke-based feature sets extracted from the on-line data. The current paper presents an extension of the system described in [9] that is based on the fusion of point- and stroke-based features.

Feature fusion has been used, for example, in the automatic recognition of audio-visual speech where a single classifier is trained on the concatenated vector of audio and visual features [13]. Feature fusion is complementary to decision fusion, where individual classifiers are first trained on the different feature sets and then the outputs of the classifiers are combined [7, 17].

If two feature streams contain the same number of feature vectors then the fused feature stream can be easily obtained by concatenating all pairs of corresponding feature vectors. However, in our case an unequal number of point-based and stroke-based feature vectors are extracted from the data. Therefore, the fused feature stream can not be derived by simple concatenation. In this work we propose a new approach to directly fusing the asynchronous feature streams and show that it improves the performance of the writer identification system.

The rest of this paper is structured as follows. In the next section we present our writer identification system based on GMMs. In Section 3 we describe the feature sets extracted from the handwritten whiteboard data, and Section 4 introduces our approach to fusing the asynchronous feature streams. Section 5 describes the experimental setup. The results are presented and discussed in Section 6. Finally, Section 7 concludes the paper and proposes future work.

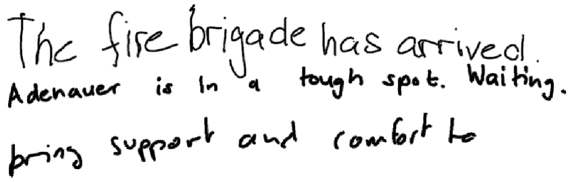

Figure 1. Examples of handwritten texts acquired from the whiteboard.

2 System Overview

On-line handwriting captured from a whiteboard differs from handwriting acquired by other devices such as digitizing tablets or Tablet PCs. While some devices register the pressure and the angle of the pen during writing, this information is not recorded by the eBeam acquisition device[1]. Furthermore, whiteboard data often have a wave like baseline and the size of the letters varies (Fig. 1 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.

Different feature sets are extracted from the text to train the models. The feature sets are described in detail in Section 3. Before feature extraction, a series of normalization operations are applied. The operations are designed to improve the quality of the features without removing writer specific information. The preprocessing and feature extraction steps are described in detail in [21].

We use Gaussian Mixture Models (GMMs) to model a person's handwriting. GMMs are defined by a weighted linear combination of M uni-modal Gaussian densities, p_i, parametrized by a $D \times 1$ mean vector, μ_i, a $D \times D$ covariance matrix, C_i, and a mixture weight, w_i:

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

where the mixture weights w_i sum up to one [14]. The parameters of a writer's density model are denoted as $\lambda = \{w_i, \mu_i, C_i\}$ with $i = 1, \ldots, M$. While the general model supports full covariance matrices, diagonal covariance matrices are used in this paper as experiments have shown that they perform better than full covariance matrices [14].

The distribution of the features extracted from the handwriting of a person is modeled by one GMM for each writer. Instead of training a writer model from scratch for every writer, we obtain the writer models from an *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 a second step, for each writer a *writer model* is build by updating the parameters of the UBM via adaptation using all training data from this writer.

The UBM is trained by means of the Expectation-Maximization (EM) algorithm [3]. 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 writer models are obtained from the UBM by a modified version of the EM algorithm based on the *Maximum a Posteriori (MAP)* principle [4]. The MAP approach provides a way of incorporating prior information in the training process which is particularly useful to deal with problems arising from sparse training data, for which the ML approach gives inaccurate estimates. An 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 [14].

As a result of the training procedure, we get a model for each writer that is well adapted to the handwriting of that particular writer. In the testing phase, a text of unknown identity is presented to each model. During decoding, the feature vectors extracted from a text line are assumed to be independent. Each model returns a log-likelihood score and based on these scores, the text is assigned to the person whose model produces the highest log-likelihood score. The GMMs are implemented using the Torch library [10].

3 Feature Sets

Three feature sets extracted from the on-line whiteboard data are presented in this section. The first two feature sets, denoted as *point-based feature set* and *stroke-based feature set*, respectively, have been used in previous work [9]. The third set of features (*all point-based feature set*) contains an extended set of on-line features as well as features extracted from an off-line representation of the on-line data. In the remainder of this section, the number in round brackets behind the name of a feature indicates the number of individual features.

The first feature set denoted as *point-based feature set* is similar to feature sets used in on-line handwriting recognition [5, 18] and signature verification [6, 16]. For a given stroke s consisting of points p_1 to p_n, the following five features for each consecutive pair of points (p_i, p_{i+1}) are computed (for an illustration see Fig. 2): *speed (1)*, *writing direction (2)*, i.e., the cosine and the sine of the angle ϕ_i between the line segment (p_i, p_{i+1}) and the x-axis, and *curvature (2)*, i.e., the cosine and the sine of the angle θ_i between the line segments (p_{i-1}, p_i) and (p_i, p_{i+1}).

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

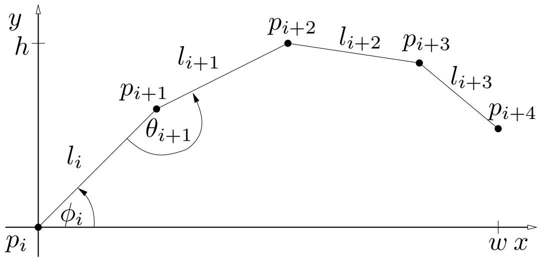

Figure 2. Point- and stroke-based features.

The second feature set (*stroke-based feature set*) is based on strokes. A stroke starts with a pen-down movement of the pen and ends with the next pen-up movement. For each stroke $s = p_1, \ldots, p_n$ we calculate the following eleven features (for an illustration see Fig. 2): *width and height of the stroke (2)*, *duration of the stroke (1)*, *time difference to previous and next stroke (2)*, *number of points (1)*, *number of curvature changes (1)*, *accumulated length (1)*, i.e., the accumulated length l_{acc} of all lines, *accumulated angle (1)*, i.e., the accumulated angle θ_{acc} of the absolute values of the angles of the writing directions of all lines, and *number of up/down strokes (2)*, i.e., the number of angles of the writing direction larger/smaller than zero.

The third feature set (*all point-based feature set*) contains an extended set of the *point-based feature set* as well as features extracted from an off-line representation of the features. The following 18 on-line features are extracted: *speed (1)*, *writing direction (2)*, *curvature (2)*, *x- and y-coordinate (2)*, *speed in x- and y-direction (2)*, *overall acceleration (1)*, *acceleration in x- and y-direction (2)*, *log curvature radius (1)*, i.e., the curvature radius is the length of the circle which best approximates the curvature at the point p_i [16], *vicinity aspect (1)*, i.e., the aspect of the trajectory in the vicinity of the point p_i [5], *vicinity curliness (1)*, i.e., the deviation from a straight line in the vicinity of the point p_i [5], *vicinity linearity (1)*, i.e., the average square distance between every point in the vicinity and the straight line linking the first and the last point in the vicinity [5], and *vicinity slope (2)*, i.e., the cosine and the sine of the angle of the straight line from the first to the last vicinity point [5].

The off-line features of the *all point-based feature set* are computed using a two-dimensional matrix representing an off-line version of the data [5]. The matrix is obtained by projecting the on-line strokes on the two-dimensional plane. The following features are used: *ascenders/descenders (2)*, i.e., 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 and *context map (9)*, i.e., the two-dimensional vicinity of the point is divided into three regions for each dimension. The number of black points in each region is taken as a feature value. The *all point-based feature set* consists of 29 features.

4 Fusion of Asynchronous Features

If two feature streams contain the same number of feature vectors then a fused feature stream can be easily obtained by concatenating the two corresponding feature vectors. However, in our case the point-based and the stroke-based feature streams contain an unequal number of feature vectors. From one handwritten stroke with n points, n point-based feature vectors but only one stroke-based feature vector are extracted. Let p_{t_i} denote the i-th value of the t-th feature vector of the stream of point-based features p and let s_j denote the j value of the stroke-based feature vector s. Then the question is how to fuse the two feature streams:

$$n \text{ points} \quad \left\{ \begin{array}{c} \langle p_{1_1}, \ldots, p_{1_p} \rangle \\ \ldots \\ \langle p_{n_1}, \ldots, p_{n_p} \rangle \end{array} \right. \quad \left. \langle s_1, \ldots, s_s \rangle \right\} \quad 1 \text{ stroke}$$

To address this question, we have implemented the following feature fusion methods:

complete The same stroke-based feature vector is added to each point-based feature vector:

$$\begin{array}{c} \langle p_{1_1}, \ldots, p_{1_p}, s_1, \ldots, s_s \rangle \\ \ldots \qquad \ldots \\ \langle p_{n_1}, \ldots, p_{n_p}, s_1, \ldots, s_s \rangle \end{array}$$

random Randomly the feature vector of one point of the stroke is selected. The stroke-based feature vector is added to the point-based feature vector of this point:

$$\langle p_{r_1}, \ldots, p_{r_p}, s_1, \ldots, s_s \rangle$$

where $r = \mathsf{random}(1, n)$.

median The median value of every individual feature over the set of all points is calculated. The stroke-based feature vector is added to the median vector obtained:

$$\langle \mathsf{median}_t\, p_{t_1}, \ldots, \mathsf{median}_t\, p_{t_p}, s_1, \ldots, s_s \rangle$$

where $t = \{1, \ldots, n\}$.

mean The mean of the set of all point-based feature vectors is calculated. To the thus obtained mean vector the stroke-based feature vector is added:

$$\langle \tfrac{1}{n} \textstyle\sum_{t=1}^{n} p_{t_1}, \ldots, \tfrac{1}{n} \sum_{t=1}^{n} p_{t_p}, s_1, \ldots, s_s \rangle$$

Feature Fusion Method	Validation Set	Test Set
point-based feature set (n)	52.64% (50)	47.50%
stroke-based feature set (1)	66.10% (150)	62.61%
complete feature fusion (n)	65.37% (250)	62.03%
random feature fusion (1)	60.78% (50)	59.30%
median feature fusion (1)	76.17% (50)	**73.08%**
mean feature fusion (1)	78.12% (100)	**75.90%**

Table 1. Results of fusing the *point-based feature set* **and the** *stroke-based feature set*.

Feature Fusion Method	Validation Set	Test Set
all point-based feature set (n)	89.90% (300)	88.68%
stroke-based feature set (1)	66.10% (150)	62.61%
complete feature fusion (n)	87.04% (300)	84.79%
random feature fusion (1)	46.82% (50)	45.48%
median feature fusion (1)	78.71% (50)	74.93%
mean feature fusion (1)	89.09% (50)	86.98%

Table 2. Results of fusing the *all point-based feature set* **and the** *stroke-based feature set*.

The *complete feature fusion* method produces n feature vectors of length $p + s$ for a stroke consisting of n points. The other three methods (*random feature fusion*, *median feature fusion*, and *mean feature fusion*) produce one feature vector of length $p + s$ and thus reduce the amount of feature vectors by a factor of n.

5 Experimental Setup

Our experiments are based on the IAM On-line English Sentence Database (IAM-OnDB) [8][2]. We use data from 200 different writers. The task is to determine which person out of these 200 individuals has written a text under question. For each writer, we have eight paragraphs of text. A paragraph contains eight text lines in average. The text lines of four paragraphs are used for training, the text lines of two paragraphs are used to validate the meta parameters of the GMMs and the remaining text lines form the independent test set.

All training data from all writers are used to train the UBM. The models of each writer are then obtained by adapting the UBM with writer-specific training data. The number of Gaussian mixture components is increased from 50 to 300 by steps of 50. The variance flooring factor is set to 0.001 and full adaptation is performed, i.e., the MAP adaptation factor is set to 0.0. The other meta parameters are set to standard values [2].

6 Results and Discussion

Tables 1 and 2 show the results of fusing the *point-based feature set* and the *all point-based feature set* with the *stroke-based feature set*, respectively. The first two rows give the results achieved by each of the two basic feature sets before fusion. The number in brackets behind the name of the fusion method indicates the number of feature vectors extracted from one stroke. In the validation set column, the number in brackets represents the number of Gaussian mixture components with the highest writer identification rate on the validation set. The results on the test set are given in the last column.

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

The results of fusing the *point-based feature set* and the *stroke-based feature set* are shown in Table 1. If the stroke-based and the point-based feature vectors are fused by the *complete feature fusion* method then the resulting fused feature set performs inferior to the single *stroke-based feature set*. Fusing the feature vectors using either the *median feature fusion* or the *mean feature fusion* method significantly increases the writer identification rate by more than 10%. The highest writer identification rate of 75.90% is achieved by the *mean feature fusion* method.

In Table 2 the results of fusing the *extended point-based feature set* with the *stroke-based feature set* is shown. The fused feature set obtained by the *complete feature fusion* method produces a lower writer identification rate than the one resulting from the *all point-based feature set*. Neither the *median feature fusion* nor the *mean feature fusion* method achieve an improved writer identification rate. However, the *mean feature fusion* method produces a result that is statistically not significantly lower then the *all point-based feature set*. Here it is important to point out that the number of features used in the *mean feature fusion* method is much smaller than the number used in the *all point-based feature set*. This data reduction also leads to much faster training and testing times.

In both experiments, the *random feature fusion* method yields rather poor results. This shows that the writer identification rate only increases if distinctive features are added to the stroke-based feature vector. Adding the point-based features of a random point does not improve the performance. Similarly, simply adding the stroke-based features to every point-based feature as in the case of the *complete feature fusion* method does not improve the performance.

The *median feature fusion* and the *mean feature fusion* methods perform best in both experiments. This result can be interpreted as follows: both methods calculate a new set of stroke-based features from the point-based features.

Then the two stroke-based feature sets are fused. It is thus important to condense the point-based feature vectors into distinctive stroke-based feature vectors. Interestingly, in some cases the resulting stroke-based features have an intuitive interpretation, e.g., mean or median speed within a stroke.

7 Conclusion

In this paper we present a language and text independent system to identify the writer of on-line handwriting captured from a whiteboard. Different feature sets are extracted from the acquired data and Gaussian mixture models (GMMs) are used to model the distribution of the features.

From a stroke consisting of n points, n point-based feature vectors and one stroke-based feature vector are extracted. The resulting feature sets thus have an unequal number of feature vectors. We propose a new feature fusion approach by directly fusing the asynchronous feature sets and show that we can improve the performance of the writer identification system on a data set consisting of 200 writers.

As the fused feature sets very likely contain correlated features, the writer identification rate can potentially be improved by applying feature selection and extraction techniques [20]. Furthermore, defining further feature fusion methods and comparing their performance with fusion on other levels, such as the matching score or the decision level, is left for future work.

Acknowledgments

This research is supported by the Swiss National Science Foundation NCCR program "Interactive Multimodal Information Management (IM2)" in the Individual Project "Visual/Video Processing". The authors would like to thank Marcus Liwicki for his support in the extraction of features.

References

[1] S. M. Chu, E. Marcheret, and 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*, pages 332–343, 2005.

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

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

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

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

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

[7] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

[8] M. Liwicki and 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*, volume 2, pages 956–961, 2005.

[9] M. Liwicki, A. Schlapbach, H. Bunke, S. Bengio, J. Mariéthoz, and J. Richiardi. Writer identification for smart meeting room systems. In *Proc. 7th IAPR Workshop on Document Analysis Systems*, volume 3872 of *LNCS*, pages 186–195. Springer, 2006.

[10] Mariéthoz, J. and Bengio, S. A comparative study of adaptation methods for speaker verification. In *Proc. 7th Int. Conf. on Spoken Language Processing*, pages 581–584, 2002.

[11] L. McCowan, D. Gatica-Perez, S. Bengio, G. Lathoud, M. Barnard, and D. Zhang. Automatic analysis of multimodal group actions in meetings. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(3):305–317, 2005.

[12] D. Moore. The IDIAP smart meeting room. Technical report, IDIAP-Com, 2002.

[13] G. Potamianos, C. Neti, G. Gravier, A. Garg, and A. Senior. Recent advances in the automatic recognition of audiovisual speech. In *Proc. of the IEEE*, volume 91 of *9*, pages 1306–1326, 2003.

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

[15] J. Richiardi and A. Drygajlo. Gaussian mixture models for on-line signature verification. In *Proc. ACM SIGMM workshop on Biometrics methods and applications*, pages 115–122, 2003.

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

[17] A. Ross and A. K. Jain. Information fusion in biometrics. *Pattern Recognition Letters*, 24(13):2115–2125, 2003.

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

[19] A. Schlapbach and H. Bunke. Off-line writer identification using Gaussian mixture models. In *Proc. 18th Int. Conf. on Pattern Recognition*, pages 992–995, 2006.

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

[21] A. Schlapbach, M. Liwicki, and H. Bunke. A writer identification system for on-line whiteboard data. *Submitted for publication.*