Writer Identification for Smart Meeting Room Systems

Marcus Liwicki[1], Andreas Schlapbach[1], Horst Bunke[1], Samy Bengio[2], Johnny Mariéthoz[2], and Jonas Richiardi[3]

[1] Department of Computer Science, University of Bern,
Neubrückstr. 10, CH-3012 Bern, Switzerland
{liwicki, schlpbch, bunke}@iam.unibe.ch

[2] IDIAP,
Rue du Simplon 4, Case Postale 592, CH-1920 Martigny, Switzerland
{bengio, mariethoz}@idiap.ch

[3] Perceptual Artificial Intelligence Laboratory, Signal Processing Institute,
Swiss Federal Institute of Technology Lausanne
FSTI-ITS-LIAP, Station 11, ELD 243, CH-1015 Lausanne, Switzerland
jonas.richiardi@epfl.ch

Abstract. In this paper we present a text independent on-line writer identification system based on Gaussian Mixture Models (GMMs). This system has been developed in the context of research on Smart Meeting Rooms. The GMMs in our system are trained using two sets of features extracted from a text line. The first feature set is similar to feature sets used in signature verification systems before. It consists of information gathered for each recorded point of the handwriting, while the second feature set contains features extracted from each stroke. While both feature sets perform very favorably, the stroke-based feature set outperforms the point-based feature set in our experiments. We achieve a writer identification rate of 100% for writer sets with up to 100 writers. Increasing the number of writers to 200, the identification rate decreases to 94.75%.

1 Introduction

The aim of a Smart Meeting Room is to automate standard tasks usually performed by humans in a meeting [12, 13, 15, 22]. These tasks include, for instance, note taking and extracting the important issues of a meeting. To accomplish these tasks, a Smart Meeting Room is equipped with synchronized recording interfaces for audio, video and handwritten notes.

The challenges posed in Smart Meeting Room research are manifold. In order to allow indexing and browsing of the recorded data [23], speech [14], handwriting [9] and video recognition systems [4] need to be developed. Another task is the segmentation of the meeting into meeting events. This task can be addressed by using single specialized recognizers for the individual input modalities [15] or by using the primitive features extracted from the data streams [12]. Further tasks deal with the extraction of 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 have to be developed. They are based on speech [11] and video interfaces [5, 18] or on a combination of both [2].

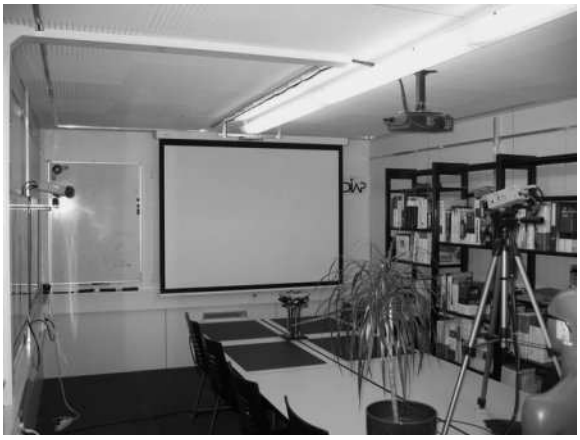

Fig. 1. Picture of the IDIAP Smart Meeting Room with the whiteboard to the left of the presentation screen

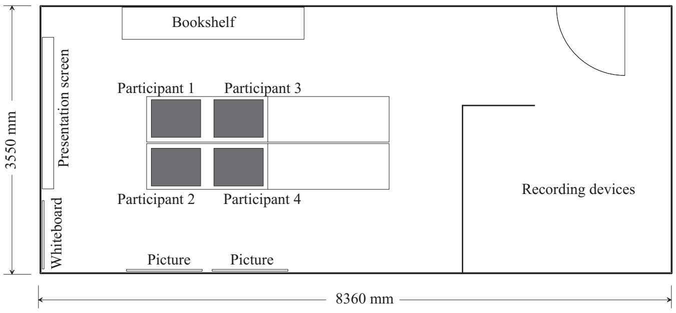

Fig. 2. Schematic overview of the IDIAP Smart Meeting Room (top view)

The writer identification system described in this paper has been developed for the IDIAP Smart Meeting Room [13]. This meeting room is able to record meetings with up to four participants. It is equipped with multiple cameras, microphones, electronic pens for note-taking, a projector, and an electronic whiteboard. Figure 1 shows a picture of this room, and a schematic overview is presented in Fig 2.

The whiteboard shown in Figs. 1 and 2 is equipped with the eBeam[4] system, which acquires the text written on the whiteboard in electronic format. A normal pen in a special casing is used to write on the board. The casing sends infrared signals to a triangular receiver mounted in one of the corners of the whiteboard. The acquisition system 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 process is shown in Fig. 3.

[4] eBeam System by Luidia, Inc. – www.e-Beam.com

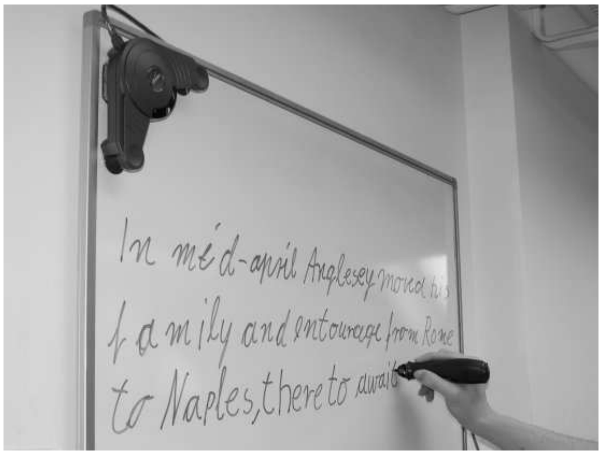

Fig. 3. Recording session with the data acquisition device positioned in the upper left corner of the whiteboard

In this paper we describe a system for writer identification using the on-line data acquired by the eBeam interface. Our system uses Gaussian Mixture Models (GMMs) as classifiers which are often used in state-of-the-art speaker verification systems [11]. Our system is text-independent, i.e., any text can be used to identify the writer. In [20] a text-independent system for writer identification is presented. This system uses off-line data, i.e., only an image of the handwriting, with no time information, is available and HMM-based recognizers are used as classifiers. There exist other on-line writer identification and verification systems in the literature [7]. These systems are mainly based on signature, which makes them text dependent compared to our approach which is text independent. An approach to writer verification for texts different from signature has been proposed in [24], but there the transcription has to be made available to the system. To compare the results of our proposed system with other work, we use a modified version of the on-line signature verification system described in [17] as a reference in our experiments. A modification of the system described in [17] has to be made because not all features, i.e., pen pressure, can be extracted from the electronic whiteboard data.

The rest of the paper is structured as follows. In Sect. 2 we present two sets of on-line features for our writer identification system. The Gaussian Mixture Model classifiers are described in Sect. 3. The results of our experiments are presented in Sect. 4. Finally, Sect. 5 concludes the paper and proposes future work.

2 Features

The text written on the whiteboard is encoded as a sequence of time-stamped (x, y)-coordinates. From this sequence, we extract a sequence of feature vectors and use them to train the classifier. Before feature extraction, some simple preprocessing steps are applied to remove spurious points and to fill gaps within strokes [9]. In order to preserve

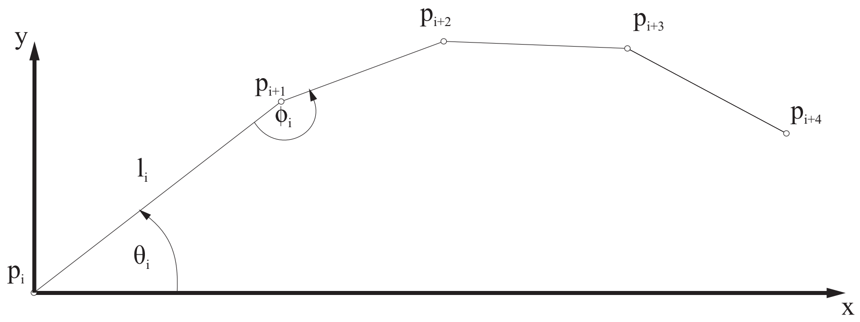

Fig. 4. Point-based features

writer specific information, no other normalization operations, such as slant or skew correction, are applied and no resampling of the points is performed. Furthermore, we do not interpolate missing points if the distance between two successive points of a stroke exceeds a predefined threshold [6] as this would remove information about the writing speed of a person.

In this paper we investigate two different approaches for the extraction of the features. In the first approach, we extract features directly from the (x, y)-coordinates of the handwriting (denoted as *point-based* features). In the second approach, we use strokes for the calculation of the features (denoted as *stroke-based* features). A stroke starts with a pen-down movement of the pen and ends with the next pen-up movement. Thus a stroke is a sequence of points during a certain time interval when the pen-tip touches the whiteboard.

The features extracted in the first approach are similar to the ones used in on-line handwriting recognition systems [19] and signature verification systems [7]. For a given stroke s consisting of points p_1 to p_n, we compute the following five features for each consecutive pair of points (p_i, p_{i+1}); for an illustration see Fig. 4:

- the length l_i of the line
$$l_i = d(p_i, p_{i+1})$$
- the writing direction at p_i, i.e., the cosine and sine of θ_i
$$\cos(\theta_i) = \Delta x(p_i, p_{i+1})$$
$$\sin(\theta_i) = \Delta y(p_i, p_{i+1})$$
- the curvature, i.e., the cosine and sine of the angle ϕ_i. These angles can be derived by the following trigonometric formulas:
$$\cos(\phi_i) = \cos(\theta_i) * \cos(\theta_{i+1}) + \sin(\theta_i) * \sin(\theta_{i+1})$$
$$\sin(\phi_i) = \cos(\theta_i) * \sin(\theta_{i+1}) - \sin(\theta_i) * \cos(\theta_{i+1})$$
where $\phi_i = \theta_{i+1} - \theta_i$ (see Fig. 4).

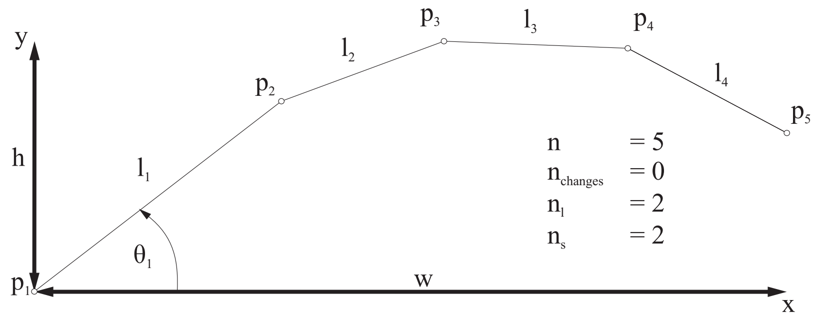

Fig. 5. Stroke-based features

These five features are computed for all the points of each stroke of a text line. We thus get a sequence of five-dimensional feature vectors which can be used for classification. The lengths of the lines l_i implicitly encode the writing speed as the sampling rate of the acquisition hardware is approximately constant.

In the second approach, the extracted feature set is based on strokes. These *stroke-based* features have been designed in the context of this work. For each stroke $s = p_1, \ldots, p_n$ we calculate the following eleven features; for an illustration see Fig. 5:

- the accumulated length l_{acc} of all lines l_i

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

- the cosine and the sine of the accumulated angle θ_{acc} of the writing directions of all lines

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

- the width $w = x_{\max} - x_{\min}$ and height $h = y_{\max} - y_{\min}$ of the stroke
- the duration t of the stroke
- the time difference Δt_{prev} to the previous stroke
- the time difference Δt_{next} to the next stroke
- the total number of points n
- the number of changes n_{changes} in the curvature
- the number of angles n_l of upward writing direction (where $\theta_i > 0$)
- the number of angles n_s of downward writing direction (where $\theta_i < 0$)

The two sets of features presented above provide different information about a person's handwriting. The point-based feature set contains local information about each point of the writing. By contrast, strokes consist of sequences of points and provide rather global information about a handwriting. For example, it is possible to determine whether a person's handwriting is cursive or not from the number of points and changes in the curvature of a stroke. This information is not available from the point-based features.

3 Gaussian Mixture Models

In text-independent speaker recognition, Gaussian Mixture Models (GMMs) have become a dominant approach [11, 16]. In this paper we use GMMs to model the handwriting of each person of the underlying population. More specifically, the distribution of feature vectors extracted from a person's on-line handwriting is modeled by a Gaussian mixture density. For a D-dimensional feature vector denoted as $\mathbf{x}$, the mixture density for a given writer is defined as

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

The density is a weighted linear combination of M uni-modal Gaussian densities, $p_i(\mathbf{x})$, each parameterized by a $D \times 1$ mean vector, μ_i, and $D \times D$ covariance matrix, C_i.

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

The mixture weights, w_i, furthermore satisfy the constraint $\sum_{i=1}^{m} w_i = 1$. Collectively, the parameters of a writer's density model are denoted as $\lambda = \{w_i, \mu_i, C_i\}$, $i = 1, \ldots, M$. While the general model supports full covariance matrices, only diagonal covariance matrices are used in this paper as they perform better than full matrices in experiments [16].

The following two-step training procedure is used. In the first step, all training data from all writers is used to train a single, writer independent *universal background model (UBM)*. Maximum likelihood writer model parameters are estimated using the iterative Expectation-Maximization (EM) algorithm [3]. The EM algorithm iteratively refines the GMM parameters to monotonically increase the likelihood of the estimated model for the observed feature vectors.

In the second step, for each writer a writer dependent *writer model* is built by updating the trained parameters in the UBM via adaptation using all the training data from this writer. We derive the hypothesized writer model by adapting the parameters of the UBM using the writer's training data and a form of Bayesian adaptation called *Maximum A Posteriori (MAP)* estimation [16]. The basic idea of MAP is to derive the writer's model by updating the well-trained parameters in the UBM via adaptation. The adaptation is a two-step process. The first step is identical to the expectation step of the EM algorithm, where estimates of the sufficient statistics of the writer's training data are computed for each mixture in the UBM. Unlike the second step of the EM algorithm, for adaptation these new statistical estimates are then combined with the old statistics from the UBM mixture parameters using a data-dependent mixture coefficient [16].

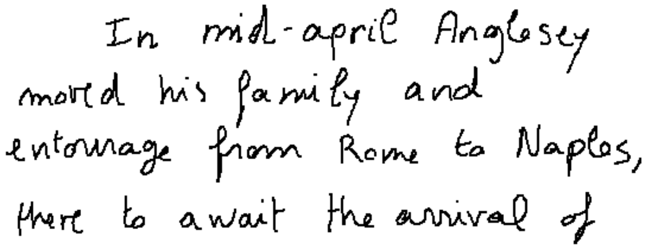

Fig. 6. Example of a paragraph of recorded text

The system was implemented using the Torch library [1]. In this implementation, only the means are adapted during MAP adaptation. Variances and weights are unchanged, as experimental results tend to show that there are no effects when they are adapted [16].

4 Experiments and Results

Our experiments are based on the IAM-OnDB database [10], which contains more than 1,700 handwritten texts in on-line format from over 220 writers. During writing on the whiteboard, the data is acquired using the eBeam system which is also used in the IDIAP Smart Meeting Room [13]. Each writer writes eight paragraphs of text compiled from the Lancaster-Oslo/Bergen corpus (LOB) [8]. The acquired data is stored in XML-format, including the writer's identity, the transcription and the setting of the recording.

One paragraph of text contains 40 words on average. In Figure 6 an example of a paragraph of recorded text is shown. Four paragraphs are used for training, two paragraphs are used to validate the global parameters of the GMMs (see Sect. 3) and the remaining two paragraphs form the independent test set.

The baseline system [17] uses 32 Gaussian mixture components with diagonal covariance matrices. No adaptation is performed and each user model is initialised on its own data set. The nine point-based features used are (x, y) position, writing path tangent angle ϕ, total velocity v, x and y components of velocity v_x, v_y, total acceleration a, and x and y components of velocity a_x, a_y. Note that the pen pressure feature which is used in [17] is not available from the whiteboard data. The data is preprocessed by subtracting the initial point from all samples, so all paragraphs start at $(0, 0)$. Each feature is then normalized in respect to its mean and its variance.

In our system, all training data from each writer is used to train the UBM. The background model is then adapted for each writer using all writer-specific training data. We have increased the numbers of Gaussian from 50 to 400 by steps of 50. In this initial experiment the adaptation factor was set to 0.0, i.e., full adaptation was performed. For the other meta parameters we used standard values [1]. The optimal number of Gaussians was determined on the validation set and this number is then used to compute the identification rate on the test set. The identification rate is determined by dividing the number of correctly assigned text paragraphs by the total number of text paragraphs.

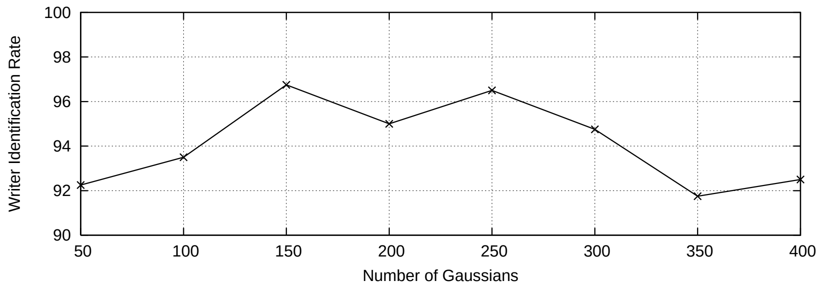

Fig. 7. Identification rate as a function of the number of Gaussians on the validation set

Table 1. Identification rates on the test set (in %)

no. of writers	50	100	150	200
baseline system	94.4	91.4	90.5	85.3
point-based features	98.0	92.5	87.0	85.0
stroke-based features	100.0	100.0	96.7	94.75

To examine the scalability of the system, we performed the experiments on four sets. First, we randomly choose 50 writers that form the set S_1. Than we added 50 randomly chosen writers to get the second set S_2 ($S_1 \subset S_2$). We continued adding 50 writers to get set S_3 and set S_4, respectively ($S_2 \subset S_3 \subset S_4$).

In Figure 7 the identification rate as a function of the number of Gaussians mixture components on the validation set for the 200 writers experiment with the stroke-based features is shown. On this set, the best identification rate of 96.75% is obtained when using 150 Gaussians. With this number of Gaussians, an identification rate of 93.5% is achieved on the test set.

We repeated the experiments with different number of Gaussians and different adaptation factors and optimized their values on the validation set. The number of Gaussians was varied between 50 and 400 by steps of 50. The MAP factor was increased from 0.0 to 0.5 in steps of 0.1. The other meta parameters were again set to standard values. This optimization further increases the identification rate. Table 1 shows the results on the test set. The performance of the baseline system [17] is comparable to our system when point-based features are used. The stroke-based features perform superior to the point-based features for every number of writers tested. They achieve a perfect identification rate of 100% for 50 and 100 writers. For 200 writers the identification rate is 94.75%.

To investigate how our system performs if fewer data is used for training, we have reduced the number of paragraphs from each of the 200 writers from four paragraphs to one paragraph. The stroked-based features are used in this experiment. The number of Gaussians was varied between 50 to 150 by steps of 50 and the MAP factor was increased from 0.0 to 0.5 in steps of 0.1. Both parameters were optimized on the validation set. The results of our experiments on the test set are given in Table 2. If we use two instead of four paragraphs of text, the writer identification rate of our system using

Table 2. Identification rates on the test set using different number of paragraphs (in %)

no. of paragraphs	4	3	2	1
stroke-based features	94.75	91.75	86.25	71.25

stroke-based features is still better compared to our system using point-based features and the baseline system both trained on all four paragraphs of text (see Table 1).

5 Conclusions and Future Work

In this paper we introduced an on-line writer identification system for Smart Meeting Rooms. A person's writing on an electronic whiteboard is the input to a Gaussian mixture model based classifier, which returns the identity of the writer. This identity can then be used for indexing and browsing the recorded data of the meeting.

In our experiments we achieve perfect identification rates of 100% on data sets produced by 50 and 100 writers. Doubling the number of writers to 200, the identification rate decreases to 94.75%. This results implies that our approach scales well with a larger number of writers. Furthermore, we argue that even in large organizations, there will rarely be more than 200 potential participants to a meeting held in a smart meeting room.

We have introduced two sets of new features extracted from the recorded on-line data. The first set consists of feature vectors from each recorded point, while the second set consists of vectors extracted from strokes. In our experiments the stroke-based features perform consistently better than the point-based features. This indicates that strokes contain more information to characterize a person's handwriting than single points.

In future work we plan to test our writer identification system on a refined scenario. For real world applications it is too time consuming and cumbersome to ask a person to copy large amounts of text before the system can be adapted with the writer's data. Therefore, we intend to further reduce the amount of data which is needed for adapting the GMMs as well as the amount of data needed to test the system. In the current scenario, we use the same data from each writer to train the UBM and the client model. In our future work, we plan to train the UBM with a training set consisting of a disjoint set of persons.

The point-based and the stroke-based feature sets describe different aspects of a person's handwriting. It is reasonable to combine the two sets to get a better performance. Initial experiments show promising results. Another approach to increase the system's performance is to generate multiple classifier systems by varying the system's parameters, e.g., the number of Gaussian components or the adaptation factor.

While our system has been developed for handwriting data acquired by the eBeam whiteboard system, our approach can potentially also be applied to other on-line handwriting data, e.g., data acquired by an electronic pen used on a Tablet PC [21].

Acknowledgments

This work was supported by the Swiss National Science Foundation program "Interactive Multimodal Information Management (IM)2" in the Individual Project "Access and Content Protection", as part of NCCR. The authors would like to thank Dr. Darren Moore for helping us with technical issues of the IDIAP Smart Meeting Room. Furthermore, we thank Christoph Hofer for conducting part of the experiments.

References

1. Collobert, R., Bengio, S., Mariéthoz, J.: Torch: a modular machine learning software library. Technical report, IDIAP (2002)
2. Czyz, J., Bengio, S., Marcel, C., Vandendorpe, L.: Scalability analysis of audio-visual person identity verification. In: Audio- and Video-based Biometric Person Authentication. (2003) 752–760
3. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. Journal of Royal Statistical Society **39** (1977) 1–38
4. Fasel, B., Luettin, J.: Automatic facial expression analysis: A survey. Pattern Recognition **36** (2003) 259–275
5. Grudin, M.A.: On internal representations in face recognition systems. Pattern Recognition **33** (2000) 1161–1177
6. Jaeger, S., Manke, S., Reichert, J., Waibel, A.: Online handwriting recognition: the NPen++ recognizer. Int. Journal on Document Analysis and Recognition **3** (2001) 169–180
7. Jain, A., Griess, F., Connell, S.: On-line signature verification. Pattern Recognition **35** (2002) 2663–2972
8. Johansson, S.: The tagged LOB Corpus: User's Manual. Norwegian Computing Centre for the Humanities, Norway (1986)
9. Liwicki, M., Bunke, H.: Handwriting recognition of whiteboard notes. In: Proc. 12th Conf. of the Int. Graphonomics Society. (2005) 118–122
10. Liwicki, M., Bunke, H.: IAM-OnDB – an on-line English sentence database acquired from handwritten text on a whiteboard. In: 8th Int. Conf. on Document Analysis and Recognition. (2005) Accepted for publication.
11. Mariéthoz, J., Bengio, S.: A comparative study of adaptation methods for speaker verification. In: Int. Conf. on Spoken Language Processing, Denver, CO, USA (2002) 581–584
12. McCowan, L., Gatica-Perez, D., Bengio, S., Lathoud, G., Barnard, M., Zhang, D.: Automatic analysis of multimodal group actions in meetings. IEEE Trans. on Pattern Analysis and Machine Intelligence **27** (2005) 305–317
13. Moore, D.: The IDIAP smart meeting room. Technical report, IDIAP-Com (2002)
14. Morgan, N., Baron, D., Edwards, J., Ellis, D., Gelbart, D., Janin, A., Pfau, T., Shriberg, E., Stolcke, A.: The meeting project at ICSI. In: Proc. Human Language Technologies Conf. (2001) 246–252
15. Reiter, S., Rigoll, G.: Segmentation and classification of meeting events using multiple classifier fusion and dynamic programming. In: Proc. 17th Int. Conf. on Pattern Recognition. (2004) 434–437
16. Reynolds, D.A., Quatieri, T.F., Dunn, R.B.: Speaker verification using adapted Gaussian mixture models. Digital Signal Processing **10** (2000) 19–41
17. Richiardi, J., Drygajlo, A.: Gaussian Mixture Models for on-line signature verification. In: Proc. 2003 ACM SIGMM workshop on Biometrics methods and applications. (2003) 115–122

18. Sanderson, C., Paliwal, K.K.: Fast features for face authentication under illumination direction changes. Pattern Recognition Letters **24** (2003) 2409–2419
19. Schenkel, M., Guyon, I., Henderson, D.: On-line cursive script recognition using time delay neural networks and hidden Markov models. Machine Vision and Applications **8** (1995) 215–223
20. Schlapbach, A., Bunke, H.: Off-line handwriting identification using HMM based recognizers. In: Proc. 17th Int. Conf. on Pattern Recognition. Volume 2. (2004) 654–658
21. Schomaker, L.: From handwriting analysis to pen-computer applications. IEE Electronics & Communication Engineering Journal **10** (1998) 93–102
22. Waibel, A., Schultz, T., Bett, M., Malkin, R., Rogina, I., Stiefelhagen, R., Yang, J.: SMaRT: The Smart Meeting Room Task at ISL. In: Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing. Volume 4. (2003) 752–755
23. Wellner, P., Flynn, M., Guillemot, M.: Browsing recorded meetings with Ferret. In: Machine Learning for Multimodal Interaction. (2004) 12–21
24. Yamazaki, Y., Nagao, T., Komatsu, N.: Text-indicated writer verification using hidden Markov models. In: Proc. 7th Int. Conf. on Document Analysis and Recognition. (2003) 329–332