Action Recognition in Meeting Scenarios using Global Motion Features

Martin Zobl, Frank Wallhoff and Gerhard Rigoll
Munich University of Technology
Department of Electrical Engineering and Information Technology
Institute for Human-Machine-Communication
Arcisstraße 21, 80290 München, Germany
{zobl,wallhoff,rigoll}@mmk.ei.tum.de

Abstract

An approach for an off-line segmentation- and recognition system for actions in meeting scenarios is presented. The deployed system consists of three major modules, which are a feature extraction module, a segmentation module and a statistical classifier which all have been assembled in conjunction of an EU funded automated meeting transcription system. All these modules and their functionality are briefly introduced. The performance of the system is evaluated on a part of the PETS-ICVS 2003 dataset (scenario B, camera 1 and 2). It is shown, that the presented system is able to detect and recognize single person actions in a meeting with an acceptable performance.

1. Introduction

Automated processing of recorded meetings and the generation of transcriptions is a relatively new and demanding task. Several activities in different fields within meeting scenarios have been launched. These cover for example recording, representation and browsing of meetings [8, 2]. Furthermore the aspects of tracking the focus of attention [6] and multimodal recognition of group actions [4] are covered. The recognition of such actions delivers good results by finding high level events in meetings, like consensus and discussion. Additionaly it may be interesting what single participants are doing throughout the meeting. An approach to recognize some of these actions by video sequence processing is described in more detail in this paper. The presented system is based on a body gesture recognizer by Eickeler [3]. It has been modified for the application in multi person meeting scenarios. Our aim is to test our system with the scenario B, using cameras one and two. A previously existing database of the institute was used for the training of our system. Since some of the PETS-defined actions are not included in out existing database, these actions could not be validated.

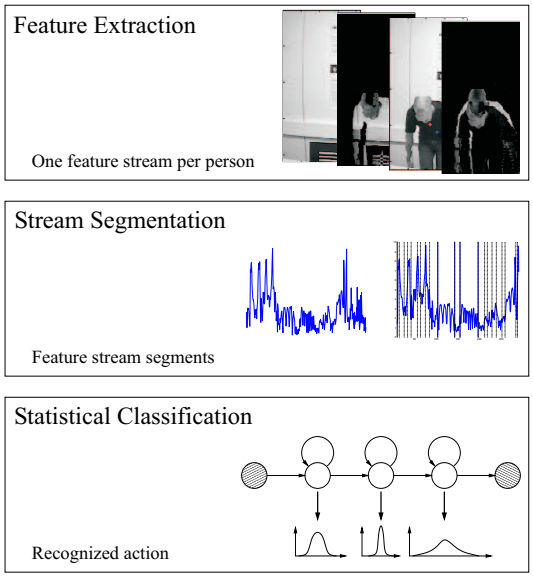

Figure 1. Processings steps for action recognition.

Fig. 1 gives an overview of the action recognition system. As can be seen, it consists of three processing levels, which are:

1. Feature extraction
2. Stream segmentation
3. Statistical classification

Starting with the original image sequence of the meeting scenario, the action region for every person participating the meeting is extracted by background differencing. Each found subregion of the video is then decomposed to an independent single video stream. From this image sequence, the difference image sequence is computed and a feature vector

From: PETS-ICVS 2003, Ed.: J. Ferryman, University of Reading, UK

sequence is derived. To find the start and stop boundaries of actions, the feature vector stream is segmented. A subsequent HMM based recognition module classifies the actions, represented by the pre-segmented feature vector sequences. The output of the system is the recognized action, given an unknown segment.

2. Feature Extraction

2.1. Finding of Action Regions

In a first step, people taking part at the meeting, have to be found. In a meeting room, there are typical locations, where the participants are expected, such as chairs, or a whiteboard. These *hot spots* R_i are monitored for changes relativ to a pre-recorded background image without the presence of a person I_b. This is done by the computation of the difference image I'_{db} between the background image and the actual image I_a in the videostream. A background subtracted image is defined by:

$$I'_{db}(x,y,t) = I_b(x,y,t) - I_a(x,y,t). \tag{1}$$

Then the intermittend noise is reduced by a threshold operation. Every pixel with an absolute value smaller than a threshold T is set to zero.

$$I_{db}(x,y,t) = \begin{cases} 0 & |I'_{db}(x,y,t)| < T \\ I'_{db}(x,y,t) & |I'_{db}(x,y,t)| \geq T \end{cases} \tag{2}$$

Further interferences are removed with morphological cleaning operations. Finding, that changes inside a hotspot R_i exceed a given threshold T

$$\sum_{(x,y)\in R_i} |I_{db}(x,y,t)| > T, \tag{3}$$

an image subregion, consisting of a fixed size rectangle, is defined as action region around the persons center of mass $[p_x(t), p_y(t)] = \mathcal{F}(|I_{db}(x,y,t)|)$ (calculation as described in section 2.3). For every action region the following steps are performed to extract the appropriate features.

2.2. Image Processing

The region of interest inside an action region has to be extracted. Since we want to recognize actions, and actions are always coupled with motion, we extract regions of motion. As already shown in previous approaches, building the difference image is an efficient and effective method to extract motion or moving objects in a static or slow changing environment. The difference image sequence $I'_d(x,y)$ is built by subtracting the pixel values at equal positions (x,y) of every second frame of the original image sequence, according to the predescribed method. $I'_d(x,y)$ may consist of positive and negative values. The thresholded absolute gray values $I_d(x,y)$ represent the intensity of motion for each spatial position (x,y) of the difference image.

2.3. Feature Calculation

The difference image can then be interpreted as a distribution of the motion over the image space in x and y-direction with the weights $I'_d(x,y)$. Each distribution is characteristic for a specific motion, and so describes an action. Characterizing this distribution with certain features results in a good representation for the current motion in the difference image. Calculating the center of mass $\vec{m}'(t) = [m'_x(t), m'_y(t)]^T$ delivers the *center of motion*.

$$m'_x(t) = \frac{\sum_{(x,y)\in R_i} x|I_d(x,y,t)|}{\sum_{(x,y)\in R_i} |I_d(x,y,t)|} \qquad m'_y(t) = \frac{\sum_{(x,y)\in R_i} y|I_d(x,y,t)|}{\sum_{(x,y)\in R_i} |I_d(x,y,t)|} \tag{4}$$

Since the features should be independent regarding the location of a participant, the center of motion relative to the person's center of mass $\vec{m}(t) = [m_x(t), m_y(t)] = [m'_x(t) - p_x(t), m'_y(t)]^T$ is used for further processing steps. A relativization of $m'_y(t)$ is not necessary, because participants are located at the same "height" in the image.

To consider the changes in the direction of a movement, the relative change of the center of mass $\Delta m_x(t) = m_x(t) - m_x(t-1)$ and $\Delta m_y(t) = m_y(t) - m_y(t-1)$ is added to the feature vector, too.

Additionally, the mean absolute deviation of a pixel (x,y) relative to the center of motion $\vec{\sigma(t)} = [\sigma_x(t), \sigma_y(t)]^T$ is used for the motion description.

$$\sigma_x(t) = \frac{\sum_{(x,y)\in R_i} |I_d(x,y,t)|(x - m_x(t))}{\sum_{(x,y)\in R_i} |I_d(x,y,t)|}$$
$$\sigma_y(t) = \frac{\sum_{(x,y)\in R_i} |I_d(x,y,t)|(y - m_y(t))}{\sum_{(x,y)\in R_i} |I_d(x,y,t)|} \tag{5}$$

With this feature we can distinguish between an action where large parts of the body are in motion (e.g. "get-up") and an action concentrated in a smaller area, where only small parts of the body move (e.g. "nodding"). This feature can be also considered as *wideness of motion*.

Another important feature describing motion, is the *intensity of motion* $i(t)$, which is simply the average absolute

height of the motion distribution, which can be expressed as

$$i(t) = \frac{\sum_{(x,y)\in R_i} |I_d(x,y,t)|}{\sum_{(x,y)\in R_i} 1}. \quad (6)$$

A large value of $i(t)$ represents a very intensive motion of parts of the body, and a small value characterizes an almost stationary image.

The complexity and dimension of the high dimensional action region is heavily reduced by scaling down the region into a 7 dimensional vector

$$\vec{x}_t = [m_x, m_y, \Delta m_x, \Delta m_y, \sigma_x, \sigma_y, i]^T, \quad (7)$$

while preserving the characteristics of the currently observed motion. This motion vector is derived for every second frame so that a vector sequence $\vec{X}_n = [\vec{x}_1, ..., \vec{x}_n]^T$ arises, where each vector carries important information about the current motion, and thus the entire sequence contains the information about the performed actions.

3. Segmentation using the Bayesian Information Criterion

For the automatic temporal segmentation of a feature vector sequence, we decided to use an efficient variant of the BIC approach presented by Tritschler and Gopinath [7] which we will introduce briefly. In the following, we use the term *features* in the meaning of *features for BIC segmentation*. We tested two kinds of features, which are normalized feature vectors derived from the feature extraction introduced above, and the energy of the feature vectors alternatively.

A sliding window, beginning at feature s with the length n of the feature stream, is scanned for an action boundary. If no boundary is found, the length of the window is enlarged and the process is repeated until a boundary is found. Hereafter the start of the window is moved to the found segment boundary. This process is repeated until the end of the given stream is reached.

Inside a window with n features $x_s, \ldots, x_{s+n}$, a boundary at position $i \in \{s+4, \ldots, s+n-4\}$ is arbitrarily placed first, so that two segments arise (called the first and the second). In a second step, it is tested whether it is more likely that one process Φ_1 has produced the output $x_s, \ldots, x_{s+n}$, or that two different processes Φ_{21} and Φ_{22} have generated the two segments' output $x_s, \ldots, x_{s+i}$ and $x_{s+i+1}, \ldots, x_{s+n}$ respectively. To represent these processes, we use the covariance matrices Σ of the features. The decision rule to test a feature segment at a given discrete time i is:

$$\Delta BIC_i \overset{!}{<} 0 \quad \text{where } \Delta BIC_i \text{ is the minimum} \quad (8)$$

$$\Delta BIC_i = -\frac{n}{2}\log\|\Sigma_w\| + \frac{i}{2}\log\|\Sigma_f\| + \frac{n-i}{2}\log\|\Sigma_s\| + \frac{1}{2}\lambda\left(d + \frac{d(d+1)}{2}\right)\log n \quad (9)$$

In this equation the variable Σ_w denotes the covariance matrix of all feature vectors $x_s, \ldots, x_{s+n}$, where Σ_f and Σ_s are the covariance matrices of the features of the first and the second segment respectively. The dimension of the feature vector is given by d, which is 7 for motion features and 1 for the energy. According to the theory, the penalty weight λ should be 1. But in praxis it has turned out, that this weight heavily influences the sensitivity of the segment finder. Several segmentation results with different parameter constellations are depicted in figure 2.

The experiments have shown, that best results can be achieved by using the energy of the feature vectors instead of the features themselves. As can be seen in figure 2(a) (oversegmentation) and 2(b) (most segments lost), no suitable λ can be found for a good segmentation results. Furthermore it has turned out, that a initial window length of the ΔBIC with $n = 15$ gives optimal results. This value seems to be a good trade-off between the duration of actions and the time between them.

4. Training Material

The training material was collected for the use within the *MultiModal Meeting Manager* project (M4 [1]). Twelve actions were performed by six persons at several times with the same camera setup. Five of the collected actions were used for the training of the HMMs (see section 5 below). For two PETS-defined actions ("yawning", "laughing") no training material was available in our database. Fig. 3 shows typical images from our training setup. A comparison to images from the PETS database shows the difference in the size of persons, the room setup, pan and tilt of the camera, and the illumination condition.

3a 3b 3c

Figure 3. Examples of Images in the trainings corpus (a) pointing (b) raising-hand (c) get-up

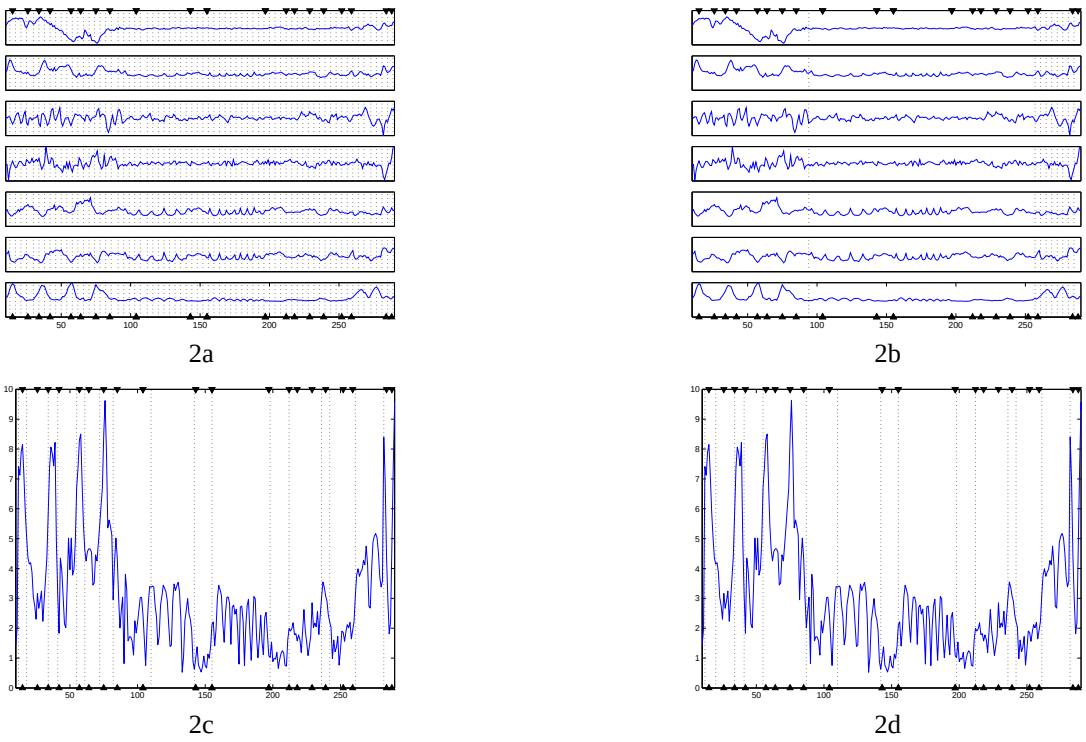

Figure 2. Segmentation boundaries: hand labeled (triangles) and those found by BIC (dashed line) using several parameter variations. (a) BIC applied to whole feature vector sequence (($n = 15, \lambda = 0.9$**), (b) BIC applied to whole feature vector sequence** $(n = 15, \lambda = 1.2)$**, (c) BIC applied to feature vector energy sequence** $(n = 15, \lambda = 6.5)$ **and (d) BIC applied to feature vector energy sequence** $(n = 20, \lambda = 6.5)$

5. Action Modelling using HMMs

In this section we briefly describe how the actions can be recognized using so called Hidden Markov Models. The statistical HMM framework and its theoretical aspects are well covered by Rabiner [5].

HMMs have proven superior results to other classification approaches due to their flexible time warping capabilities. This fact allows the modeling of actions respectively their observation sequences with different lengths. Furthermore these models have the ability to model several variations of the same action performed by a huge group of individuals by using several mixtures or streams. In the following section we give a short overveiew over the use of HMMs and the special properties for our system.

Unknown actions can be classified using the following maximum-likelihood decision:

$$M^* = \underset{M \in \text{all actions}}{\operatorname{argmax}} \quad P(X|M) \tag{10}$$

In this equation, X represents an unknown feature vector sequence of an unknown action and M represents one HMM from the set of all known actions. The classifier recognizes the performed action by finding the model M^* with the highest production probability $P(X|M)$.

Therefore the values of $P(X|M)$ for all models have to be computed. This task can be solved by using a Viterbi decoder. But before that, all model parameters have to be estimated first. These parameters are the state-transition matrices $\overleftrightarrow{A}$ and the production probabilities $P(x|S_i)$ in all states S_i for each HMM among the database.

For our system we use continuous multivariate Gaussian mixtures. In this case the real distributions of the features are approximated by a weighted sum (factor w_k) of several normal distributions $\mathcal{N}$.

$$p(\vec{x}_i|S_i) = \sum_{k=1}^{C_i} \mathcal{N}(\vec{x}_i, \vec{\mu}_k, \Sigma_k) \cdot w_k \tag{11}$$

The normal distribution describes the probability for a J-dimensional observation $\vec{x}_i$ in a certain state S_i. It is given by its mean-vector $\vec{\mu}$ and its covariance matrix Σ:

$$\mathcal{N}(\vec{x}_i, \vec{\mu}, \Sigma) = \frac{1}{\sqrt{(2\pi)^J |\Sigma|}} e^{-\frac{1}{2}(\vec{x}_i - \vec{\mu})^T \Sigma^{-1} (\vec{x}_i - \vec{\mu})} \tag{12}$$

All parameters above can be estimated using the well known Baum-Welch estimation algorithm and prior isolated action feature sequences. For our system we first train a common action model with all available training examples. In a second step the action models are reestimated with their corresponding examples.

The number of states S and mixtures C_i heavily depend on the available training material. For our training material this means, that the maximum number of states is limited to 4, which is the minimal number of observations. Due to the fact that just a little amount of training material was available, the number of mixtures was limited to 3. To model the one-dimesional feature sequences, we a left-to-right state transition topology for all action models.

6. Experiments and Results

Goal of our experiments was to evaluate our present system on part of the PETS-ICVS 2003 datasets. Therefore the above explained feature extraction, segementation and modeling techniques have been applied to scenario B camera 1 and 2. As mentioned above, not all of the performed actions in the test dataset were included in our training material. Therefore our system was just able to identify a subset of five actions, which are *sit down*, *get up*, *raising hand*, *nodding* and *shaking head*. The unknown actions were discarded after the feature extraction.

Under this test conditions, we are able to achieve average recognition scores of up to 66%, which we believe represents an acceptable result for our action recognition system at the present development stage. The results are summarized in tabular 6.

	sit down	get up	raising hand	nodding	shaking head	% Score
sit down	50%	33%	17%	0%	0%	50%
get up	17%	83%	0%	0%	0%	83%
raising hand	21%	4%	63%	0%	15%	63%
nodding	0%	0%	0%	42%	58%	42%
shaking head	0%	0%	0%	8%	92%	92%
Overall						**66%**

Table 1. Confusion matrix of recognized actions.

One of the major reasons for fault detections is based on the huge difference between the test and training datasets, mainly because the available material in the training dataset seems to be too limited. By acquiring more training material with different camera setups, lighting-, and room conditions we will be able to cope with this problem. The applied BIC based segmentation module performs well on the evaluation set used. However some improvements ragarding the finding of an optimal parameter set could further improve the performance.

7. Conclusions and Outlook

This paper reports about a system for the automatic segmentation and recognition of actions in meeting scenarios. The three main parts of the system are the feature extraction based on difference images, a BIC based segmentation module and an action classification module using a statistical HMM approach. The performance of the system was evaluated on a subset of the PETS-ICVS 2003 dataset.

Beside the improvement of all modules, we will implement a fully automated action region finding system in the future, which is based on a neural face detection approach in combination with particle filtering.

8. Acknowledgement

This work was partially funded by the EU IST Programme (project IST-2001-34485). It is part of CPA-2: the Cross Programme Action on Multimodal and Multisensorial Dialogue Modes, and is linked to the activity on Human Language Technologies. For more inforamtion refer to [1].

References

[1] The MultiModal Meeting Manager (M4) Project Homepage. http://www.dcs.shef.ac.uk/spandh/projects/m4/.

[2] R. Cutler, Y. Rui, A. Gupta, J. Cadiz, I. Tashev, L. He, A. Colburn, Z. Zhang, Z. Liu, and S. Silverberg. Distributed meetings: A meeting capture and broadcasting system. In *In Proceedings of ACM Multimedia Conference*, 2002.

[3] S. Eickeler, A. Kosmala, and G. Rigoll. Hidden Markov Model Based Continuous Online Gesture Recognition. In *Int. Conference on Pattern Recognition (ICPR)*, pages 1206–1208, Brisbane, Aug. 1998.

[4] I. McCowan, S. Bengio, D. Gatica-Perez, G. Lathoud, F. Monay, D. Moore, P. Wellner, and H. Bourlard. Modeling human interaction in meetings. In *ICASSP Proceedings (to appear), Hong Kong*, 2003.

[5] L. R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 77(2):257–285, Feb. 1989.

[6] R. Stiefelhagen. Tracking focus of attention in meetings. In *In IEEE International Conference on Multimodal Interfaces, Pittsburgh, PA.*, 2002.

[7] A. Tritschler and R. Gopinath. Improved Speaker Segmentation and Segments Clustering Using the Bayesian Information Criterion. In *Proc. EUROSPEECH*, volume 2, pages 679–682, Paris, France, 1999.

[8] A. Waibel, M. Bett, F. Metze, K. Ries, T. Schaaf, T. Schultz, H. Soltau, H. Yu, and K. Zechner. Advances in automatic meeting record creation and access. In *In Proceedings of ICASSP-2001, Salt Lake City, Utah*, 2001.