RECOGNITION OF PARTLY OCCLUDED PERSON ACTIONS IN MEETING SCENARIOS

Martin Zobl, Andreas Laika, Frank Wallhoff, Gerhard Rigoll

Institute for Human-Machine Communication, Munich University of Technology, D-80290 München
{zobl, laika, wallhoff, rigoll}@mmk.ei.tum.de

ABSTRACT

This proposal describes a novel approach for handling partly occluded gestures in the feature domain with gesture specific Kalman-Filters. An estimation of the Kalman-Filter parameters using artificial neural networks is introduced. The approach is demonstrated and evaluated on data from a meeting scenario and can be generalized to all gesture based problems concerning partial occlusions.

1. INTRODUCTION

The automated analysis of meetings has become focus of interest for some EU funded projects like the MultiModal Meeting Manager (M4) [1]. The meeting recording, automated generation of transcriptions, representation and offline browsing of meetings are goals of such projects [2]. Single tasks out of this area are for example tracking the focus of attention [3] and multimodal recognition of group actions [4]. These group actions can be derived by interpretation of single person actions like nodding, raising a hand or speaking and laughing. One approach to recognize such person actions is to calculate global motion features from difference images in regions of interest around meeting participants. After a temporal segmentation step the feature segments are classified with the help of a Hidden-Markov-Model (HMM) framework [5][6]. A detailed description of this action recognition system for unoccluded actions is given in [7]. Problems in recognition of single person actions arise, when parts of the performed action are partially occluded by objects like a desk, or if actions of one person are superimposed by actions of another person because of overlapping action regions. A method for handling partly occluded body gestures in the feature domain with a hand designed Kalman-Filter was already presented in [8]. In the following sections a novel extended approach with action specific Kalman-Filters is presented and compared to a single Kalman-Filter approach.

2. FEATURE EXTRACTION

Person actions are always coupled with motion, which can be easily spatially segmented by building the differences between subsequent frames. Out of this difference images a feature vector $\mathbf{x_k}$ containing global motion features like center of mass, change of center, variance and intensity of motion is calculated for each time step k and each action region. The result is one feature stream $\mathbf{X} = \mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_k}$ for every person participating the meeting.

3. SYSTEM OVERVIEW

As shown in figure 1 each calculated feature stream $\mathbf{X}$ is fed into a system of N action specialized Kalman-Filters $K_1, K_2, ..., K_N$. The process to determine the parameters of the Kalman-Filters is briefly described in section 5. Each Kalman-Filter modifies the feature stream according to its system model and alters the feature sequence to be more likely to the underlying action. This leads to the modified feature streams $\mathbf{X_1}, \mathbf{X_2}, \ldots, \mathbf{X_N}$, which are scored by their according Hidden-Markov-Model. The model M with the highest production probability is chosen.

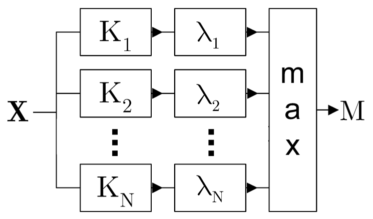

Fig. 1. System Overview.

4. OCCLUSIONS

To simulate artificial occlusions, parts of the action region are camouflaged by black boxes. Examples out of the six different types of tested occlusions are given in figure 2. The inserted occlusions result in a deranged feature stream.

5. COMPENSATION OF OCCLUSSIONS

To compensate occlusions in the feature domain a model in that domain has to be created. A simple but still suffi-

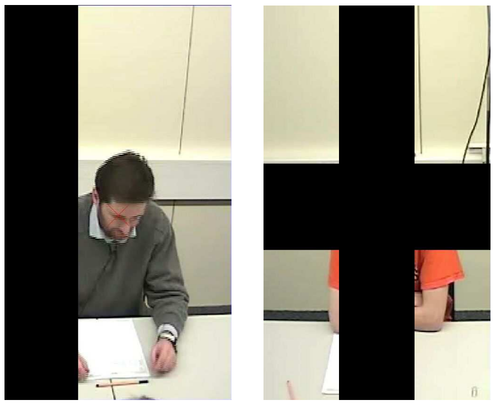

Fig. 2. Examples out of six evaluated types of occlusions. Unsymmetric (a) and symmetric (b) occlusions have been generated to simulate different occlusion scenarios.

cient model for a process generating the feature vectors is the stochastic linear dynamic system. The system is defined by the equations

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{w} \quad (1)$$

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v} \quad (2)$$

with **w** and **v** as zero mean random variables with their covariance matrices

$$\mathbf{Q} = \operatorname{cov}(\mathbf{w}) \quad (3)$$

and

$$\mathbf{R} = \operatorname{cov}(\mathbf{v}). \quad (4)$$

Using a Kalman-Filter, we now compensate occluded data with a model acquired from unoccluded data. The parameters for the Kalman-Filter are the matrices **A**, **C**, **Q** and **R**. For a further reference on the Kalman-Filter see [9].

The Kalman-Filter is applied in two setups:

One global model. A single filter K for all N actions is applied to the feature stream. The filtering step is done before the temporal segmentation to avoid perturbations caused by the initialisation of the Kalman-Filter. The segments are then classified by the HMMs.

N **action specific models.** For each action a specific filter $K_i, i \in \{1, 2, \ldots, N\}$ is applied to the feature stream. As result there are now N sequences each altered by a different filter. Again each action sequence is segmented and fed into the HMM modeling that action.

Note, that in both cases the HMMs are trained with unoccluded data only.

5.1. Parameter Estimation

The most important step in using a Kalman-Filter is the estimation of the filter parameters (i.e. the matrices **A**, **C**, **Q** and **R**). Our approach is to learn them using the training data. The data consist of the segmented sequences of feature vectors $\mathbf{y}_k$ from the unoccluded training set and the corresponding feature vectors from the occluded training set. When training a global model, data from all actions is used to learn one set of filter parameters. In the case of a specific model for each action, only the data corresponding to that action is used.

Before the training, discrete derivatives of each of the seven features are added to the sequences. This leads to the extended sequences

$$\mathbf{x}_k = \begin{pmatrix} y_k^{(1)} \\ y_k^{(2)} \\ \vdots \\ y_k^{(1)} - y_{k-1}^{(1)} \\ y_k^{(2)} - y_{k-1}^{(2)} \\ \vdots \\ \vdots \end{pmatrix} \quad (5)$$

The purpose is to make the models more precise. Tests have been run adding derivatives up to the fifth order. Adding just the first derivative however yielded the best results.

The matrix **A** is being learned from the extended, unoccluded training sequences using an adaptive linear network. As inputs $\mathbf{p}_i$ and targets $\mathbf{t}_i$ consecutive feature vectors are used (i.e. $\mathbf{p}_i = \mathbf{x}_{k-1}^{unocc}$ and $\mathbf{t}_i = \mathbf{x}_k^{unocc}$). **A** is trained in order to minimize the squared error e_i over all learning samples i.

$$e_i = (\mathbf{t}_i - \mathbf{A}\mathbf{p}_i)^T (\mathbf{t}_i - \mathbf{A}\mathbf{p}_i) \quad (6)$$

$$\mathbf{A} = argmin_A \left(\sum_i e_i(\mathbf{A}) \right) \quad (7)$$

Figure 3 shows two examples of trained matrices **A** for two different actions. The diagonal structure accommodates the fact, that the features have been chosen to be independant of each other. Additionally it can be seen that different actions result in different system models in the feature domain.

The Matrix **C** is left constant:

$$\mathbf{C} = [\mathbf{1}; \mathbf{0}; \cdots]. \quad (8)$$

Q is acquired by computing the covariance matrix of the prediction error.

$$\mathbf{Q} = \operatorname{cov}(\mathbf{t}_i - \mathbf{A}\mathbf{p}_i) \quad (9)$$

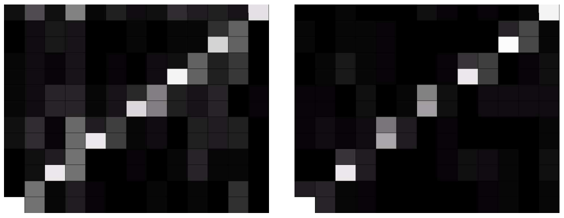

Fig. 3. Visualisation of two trained matrices **A** of two different system models for two different actions. Bright squares mean high absolute values in the matrices **A**.

R is determined by computing the covariance matrix of differences between feature vectors from the occluded and the unoccluded data.

$$\mathbf{R} = \operatorname{cov}\left(\mathbf{t}_i^{occ} - \mathbf{C}\mathbf{A}\mathbf{p}_i\right) \quad (10)$$

with $\mathbf{t}_i^{occ} = \mathbf{y}_k^{occ}$

6. RESULTS

To measure the improvements for different occlusion types we use the relative error reduction (RER) with respect to the unoccluded data. The absolute improvement in recognition rates fails to measure the improvement on different occlusions, because the possible absolute improvement can be much higher for large occlusions than for small occlusions. The RER against unoccluded data is given by

$$RER = \frac{r_{imp} - r_{occ}}{r_{unocc} - r_{occ}} \cdot 100\% \quad (11)$$

with

r_{unocc}	:	recognition rate of original method on unoccluded data set
r_{occ}	:	recognition rate of original method on occluded data set
r_{imp}	:	recognition rate of improved method on occluded data set

In table 1 the recognition rate of the two configurations of the system is compared. The results stem from the cross-occlusion shown in figure 2b, because some of the other occlusions only lead to a relative recognition rate reduction of 1%.

As shown in table 1 both, the global model and the action specific model yield significant results. Here the action specific model beats the global model in terms of performance. However there are constellations where the global model outperformes the action specific models. Further simulations have to be run to analyze this issue. Another fact that should be noticed is the impact on unoccluded data. When adapting the covariance matrix **R** to unoccluded data instead of occluded data, there is little to no influence on the classification results. However, when adapting **R** to occlusions, the filtering of unoccluded actions results in a decreasing recognition rate.

Recognition Method	Recognition Rate	RER
Unoccluded data	81.9	-
Occluded data	51.7	-
global model	62.4	35.5
action specific models	66.4	48.8

Table 1. Relative improvement rates for $N = 6$ different actions deranged with the occlusion shown in figure 2b.

7. CONCLUSIONS AND OUTLOOK

A promising approach for reconstruction of deranged feature streams with Kalman-Filters has been presented. In a meeting scenario first results show an outperforming gain in relative error reduction in case of occlusions of single person actions. The application of artificial neural networks for parameter estimation of the Kalman-Filters allows easy expansion to nonlinear systems which is part of ongoing work.

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].

9. REFERENCES

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

[2] 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.

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

[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*, vol. 77, no. 2, pp. 257–285, Feb. 1989.

[6] S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, "The htk book version," 2000.

[7] M. Zobl; F. Wallhoff; G. Rigoll, "Action recognition in meeting scenarios using global motion features," in *Proceedings Fourth IEEE International Workshop on Preformance Evaluation of Tracking and Surveillance*, J Ferryman, Ed., Graz,Österreich, März 2003, pp. 32–36, University of Reading.

[8] A. Kilinc; I. Yalcin; S. Eickeler; G. Rigoll, "Recognition of partly occluded gestures," in *3. Workshop Dynamische Perzeption*, Ulm, Germany, November 2000.

[9] G. Welch; G. Bishop, "An introduction to kalman filter," *UNC-CH Computer Science Technical Report*, 1995.