Multi-Person Tracking in Meetings: A Comparative Study

Kevin Smith[1], Sascha Schreiber[2], Igor Potúcek[3], Vítzslav Beran[3], Gerhard Rigoll[2] and Daniel Gatica-Perez[1]

[1] IDIAP Research Institute, Switzerland
[2] Technische Universität München, Germany
[3] Brno University of Technology, Czech Republic

Abstract. In this paper, we present the findings of the Augmented Multiparty Interaction (AMI) investigation on the localization and tracking of head positions in meetings. The focus of the study was to test and evaluate various multi-person tracking methods using a standardized data set and evaluation methodology.

1 Introduction

One of the fundamental goals of the AMI project is to formally and consistently evaluate tracking methods developed by AMI members using a standardized data set and evaluation methodology. In a meeting room context, these tracking methods must be robust to real-world conditions such as variation in object appearance and pose, unrestricted motion, changing lighting conditions, and the presence of multiple self-occluding objects. In this paper, we present an evaluation methodology for gauging the effectiveness of various 2D multi-person head tracking methods and provide an evaluation of the four tracking methods developed under the AMI framework in the context of a meeting room scenario.

The rest of this paper is organized as follows. In section 2, we describe the method of evaluation. In Section 3 we briefly describe each of the tracking methods. In Section 4 we present and discuss the results of the evaluation, and finally, in Section 5 we provide some concluding remarks.

2 Evaluation Methodology

To objectively compare the tracking models, we must first define a common evaluation procedure and agree upon a common data set. To this end, we have adopted an evaluation procedure and set of performance measures as defined in [1], and collected meeting-room video data (the AV16.7.ami data corpus).

2.1 Measures and Procedure

In [1], the task of evaluating tracker performance was broken into evaluating two tasks: predicting the correct number and placement of objects in the scene (referred to as *configuration*), and checking the consistency with which each tracking result (or estimate, $\mathcal{E}$) assigns identities a ground truth object ($\mathcal{GT}$)

False negative (**FN**) False positive (**FP**) Multiple trackers (**MT**) Multiple objects (**MO**)

Fig. 1. The four types of configuration errors.

over its lifetime (referred to as *identification*). Several metrics are defined below to evaluate these tasks. Each of these measures depends on information derived from the fundamental *coverage test*.

2.1.1 Coverage Test. The coverage test determines if a $\mathcal{GT}$ is being tracked by an $\mathcal{E}$, if a $\mathcal{E}$ is tracking a $\mathcal{GT}$, and reports the quality of the tracking result. For a given tracking estimate $\mathcal{E}_i$ and ground truth $\mathcal{GT}_j$, the coverage test measures the overlap between the two areas using the F-measure $F_{i,j}$ [2]

$$F_{i,j} = \frac{2\alpha_{i,j}\beta_{i,j}}{\alpha_{i,j} + \beta_{i,j}} \qquad \alpha_{i,j} = \frac{|\mathcal{E}_i \cap \mathcal{GT}_j|}{|\mathcal{GT}_j|} \qquad \beta_{i,j} = \frac{|\mathcal{E}_i \cap \mathcal{GT}_j|}{|\mathcal{E}_i|} \tag{1}$$

where recall (α) and precision (β), are well-known information retrieval measures. If the overlap passes a fixed coverage threshold ($F_{i,j} \geq t_c$, $t_c = 0.33$), then it is determined that $\mathcal{E}_i$ *is tracking* $\mathcal{GT}_j$.

2.1.2 Configuration. In this context, configuration means the number, the location, and the size of all objects in a frame of the scenario. The result of a tracking approach is considered to be *correctly configured* if and only if exactly one $\mathcal{E}_i$ is tracking each $\mathcal{GT}_j$. To identify all types of errors that may occur, four configuration measures are defined:

- **FN** - False negative. A $\mathcal{GT}$ is not tracked by an $\mathcal{E}$.
- **FP** - False positive. An $\mathcal{E}$ exists which is not tracking a $\mathcal{GT}$.
- **MT** - Multiple trackers. More than one $\mathcal{E}$ is tracking a single $\mathcal{GT}$. Each excess $\mathcal{E}$ is counted as an MT error.
- **MO** - Multiple objects. An $\mathcal{E}$ is tracking multiple $\mathcal{GT}$s. An MO error is assigned for each excess $\mathcal{GT}$.

An example of each error type is depicted in Fig. 1, where the $\mathcal{GT}$s are marked with green colored boxes, the $\mathcal{E}$s with red and blue. To assess the overall configuration, one can measure the difference between the number of $\mathcal{GT}$s and the number of $\mathcal{E}$s.

- **CD** - Configuration distance. For a given frame,the difference between the number of $\mathcal{E}$s ($N^t_{\mathcal{E}}$) and $\mathcal{GT}$s ($N^t_{\mathcal{GT}}$) normalized by the number of $\mathcal{GT}$s ($N^t_{\mathcal{GT}}$). Specifically,

$$\mathbf{CD} = \frac{N^t_{\mathcal{E}} - N^t_{\mathcal{GT}}}{max(N^t_{\mathcal{GT}}, 1)} \tag{2}$$

2.1.3 Identification. In the field of tracking, identification implies the persistent tracking of an $\mathcal{GT}$ by a particular $\mathcal{E}$ over time. Though several methods to associate identities exist, we adopt an approach based on a "majority rule" [1]. A $\mathcal{GT}_j$ is said to be identified by the $\mathcal{E}_i$ which passes the coverage test for the majority of its lifetime, and similarly $\mathcal{E}_i$ is said to identify the $\mathcal{GT}_j$ which it passes

the coverage test for the majority of its lifetime (this implies that associations between $\mathcal{GT}$s and $\mathcal{E}$s will not necessarily match).

In this approach there arise two types of identification failures. The first type (FIT) occurs when $\mathcal{E}_i$ suddenly stops tracking $\mathcal{GT}_j$ and another $\mathcal{E}_k$ continues tracking this ground truth. The second error type (FIO) results from swapping the ground truth paths, i.e. $\mathcal{E}_i$ initially tracks $\mathcal{GT}_j$ and subsequently changes to track $\mathcal{GT}_k$.

- **FIT** - Falsely identified tracker. Occurs when a $\mathcal{E}_k$ which passed the coverage test for $\mathcal{GT}_j$ is not the identifying tracker, $\mathcal{E}_i$.
- **FIO** - Falsely identified object. Occurs when a $\mathcal{GT}_k$ which passed the coverage test for $\mathcal{E}_i$ is not the identifying object, $\mathcal{GT}_j$.

Additionally, two purity measures are introduced to evaluate the degree of consistency to associations between $\mathcal{E}$s and $\mathcal{GT}$s.

- **OP** - Object purity. If $\mathcal{GT}_j$ is identified by $\mathcal{E}_i$, then OP is the ratio of frames in which $\mathcal{GT}_j$ and $\mathcal{E}_i$ passed the coverage test ($n_{i,j}$) to the overall number of frames $\mathcal{GT}_j$ exists (n_j).
- **TP** - Tracker purity. If $\mathcal{E}_i$ identifies $\mathcal{GT}_j$, then TP is the ratio of frames in which $\mathcal{GT}_j$ and $\mathcal{E}_i$ passed the coverage test ($n_{j,i}$) to the overall number of frames $\mathcal{E}_i$ exists (n_i).

2.1.4 Procedure. To evaluate the ability of each tracking model to correctly predict the configuration and identification over diverse data sets, the above measures are normalized by the instantaneous number of ground truth objects and the total number of frames, T as shown.

Evaluation procedure.

- for each frame
  - perform the coverage test over all pairs of $\mathcal{E}$s and $\mathcal{GT}$s.
  - compute configuration errors (FN,FP,MT,MO) and F-measure.
- associate $\mathcal{E}$ and $\mathcal{GT}$ pairs for identification.
- for each frame
  - compute identification errors (FIT,FIO)
- compute and normalize TP, OP, CD, configuration and identification errors

$$\overline{FP} = \frac{1}{T}\sum_{t=1}^{T}\frac{FP_t}{max(N_{\mathcal{GT}}^t,1)}\ ,\ \overline{FN} = \frac{1}{T}\sum_{t=1}^{T}\frac{FN_t}{max(N_{\mathcal{GT}}^t,1)}$$

$$\overline{MT} = \frac{1}{T}\sum_{t=1}^{T}\frac{MT_t}{max(N_{\mathcal{GT}}^t,1)}\ ,\ \overline{MO} = \frac{1}{T}\sum_{t=1}^{T}\frac{MO_t}{max(N_{\mathcal{GT}}^t,1)}$$

$$\overline{FIT} = \frac{1}{T}\sum_{t=1}^{T}\frac{FIT_t}{max(N_{\mathcal{GT}}^t,1)}\ ,\ \overline{FIO} = \frac{1}{T}\sum_{t=1}^{T}\frac{FIO_t}{max(N_{\mathcal{GT}}^t,1)}$$

$$\overline{OP} = \frac{1}{N_{\mathcal{GT}}}\sum_{j=1}^{N_{\mathcal{GT}}}\frac{n_{i,j}}{n_j} \qquad \overline{TP} = \frac{1}{N_{\mathcal{E}}}\sum_{i=1}^{N_{\mathcal{E}}}\frac{n_{j,i}}{n_i} \qquad \overline{CD} = \frac{1}{T}\sum_{t=1}^{T}|\mathbf{CD}|$$

Fig. 2. *Examples from seq14 of the AV16.7.avi data corpus.* Left: Typical meeting room data with four participants (free to stand, sit, walk). Center: Participant heads near the camera are not fully visible and often move in and out of the scene. Right: The data set also contained challenging situations (large variations in head size, occlusions, and blocked camera views). This frame was annotated with four head locations.

Table 1. Challenges in the AV16.7.ami data corpus test set.

	seq01		seq02		seq03		seq08		seq09		seq12		seq13		seq14		seq16	
	L	R	L	R	L	R	L	R	L	R	L	R	L	R	L	R	L	R
frames	1571		1196		5196		2483		1738		2584		2346		2938		2221	
total # heads	1	1	1	1	1	1	2	2	2	2	3	3	3	3	4	4	4	4
frontal heads	1	1	1	1	1	1	2	0	2	0	3	0	3	0	2	2	4	2
rear heads	1	1	1	1	1	1	0	2	0	2	0	3	0	3	2	2	4	4
event: occlusion	n	n	n	n	n	n	y	n	y	y	y	y	y	y	y	y	y	n
event: camera blocked	y	y	y	y	n	n	y	y	n	y	n	y	n	y	y	y	y	y
event: sit down	n	n	n	n	y	y	y	y	n	n	y	y	y	y	y	y	n	n

2.2 Data Set

Testing was done using the AV16.7.ami corpus, which was specifically collected to evaluate localization and tracking algorithms[4]. The corpus consists of 16 sequences of duration 1-4 minutes recorded from two camera angles. Half of the corpus was designated as the training set, and half for testing. The sequences depict up to four people in a meeting-room scenario performing common actions such as sitting down, discussing around a table, etc (see Figure 2). Participants acted according to a predefined agenda (they were told the order in which to enter the room, sit, or pass each other), but the behavior of the subjects was otherwise natural. The sequences contain many challenging phenomena for tracking methods including person occlusion, cameras blocked by passing people, partial views of backs of heads, and large variations in head size (see Table 1). The corpus was annotated for head location for use in training and evaluation.

3 Tracking Models

Four head tracking models built within the AMI framework were applied to the data corpus and evaluated as described in Section 2. These models include: a trans-dimensional MCMC tracker developed at IDIAP, a probabilistic active shape tracker developed at TUM, a KLT tracker developed at BRNO, and a

[4] We are thankful to Bastien Crettol for his support with the collection, annotation, and distribution of the AV16.7ami corpus, and to the participants for their time.

Table 2. Properties of the various head tracking approaches.

	Trans-MCMC	Active Shape	KLT	Face Detector
Learned Models	binary, color, head shape	skin color, shape	skin color	face/nonface weak classifiers
Initialization	automatic	automatic	automatic	automatic
Features	background sub, silhouette, color	motion detection, skin color, head/shoulder shape	background sub, skin color, local charact.	skin color, gabor wavelets
Mild Occ.	robust	robust	robust	robust
Severe Occ.	semi-robust	semi-robust	sensitive	sensitive
Identity Recovery	yes swap, rebirth	yes swap,rebirth	yes rebirth	no
Comp. Exp.	∼1 frame/sec	∼3 frame/sec	∼20 frame/sec	∼0.2 frame/sec

face detector also developed at BRNO. Each model approached the problem of head tracking differently, and it is noteworthy to list some of the qualitative differences (see Table 2).

3.1 Method A: Trans-Dimensional MCMC Tracking

This model uses a multi-person tracking approach based on a hybrid Dynamic Bayesian Network that simultaneously infers the number of people in the scene and their body and head locations in a joint state-space formulation that is amenable for person interaction modeling [3]. The state contains a varying number of interacting person models. Each person model moves and interacts according to a dynamical model and a Markov Random Field (MRF) based interaction model (which helps prevent multiple trackers from following the same person). A person is modeled by two bounding boxes, one corresponding to the body and one to the head. The bounding boxes are defined by image coordinates, eccentricity, height, and rotation (head only).

3.1.1 Features. The overall observation model consists of a set of global observations combined with individual observations (used to localize the head of each person). The global observation model automatically adjusts the number of people in the scene by adding and removing them from the state. The global features consists of binary and color measurements defined pixel-wise over the entire image. The binary observations predict the multi-object configuration using an adaptive background subtraction scheme which separates the image into foreground and background pixels. Training is done using switching Gaussian Mixture Models (GMM)s on features that measure the overlap of the predicted body locations with foreground and background pixels. These features are defined in the precision-recall space of the foreground and background. When observed values match these features well a high response is given. A global color model is used to maintain object identity. The individual head observations also make use of the background subtraction. A head silhouette model is constructed from the training set by averaging the binary patches taken from known head locations.

3.1.2 Inference Inference on the filtering distribution in our model is done by trans-dimensional Markov Chain Monte Carlo (MCMC) sampling. Trans-MCMC brings the following advantages: 1) because it can change the dimension of the state, it can easily add or remove people from the scene, 2) it can efficiently search high dimensional state spaces, and 3) it can help solve the problem of normalizing the likelihoods of various objects by decomposing move types. A

proposed configuration is generated by first selecting a move type: *birth* of a new object or *rebirth* of a dead object, *death* of an existing object, *swap* of two object identities, update of the *body* parameters for one person, or update of the *head* parameters for one person. After the state has been modified according to the chosen move type, the likelihood of the proposed configuration is computed from the observation model. The proposal is then accepted with a probability proportional to the ratio of its likelihood to the likelihood of the previous state. If the proposal is accepted, it is added to the Markov Chain, otherwise the previous state is added. After the chain reaches sufficient length, the MAP estimate is computed. For further details, see [3].

3.2 Method B: Probabilistic Active Shape Tracking

The core of this algorithm is a double-layered particle filter. The control layer is responsible for the detection of new objects and for the allocation of hypotheses / particle sets on different supposed objects. For this task, skin colored blobs together with a simple motion detector serve as a basic indicator for new possible objects and for the validation of existing tracks. In the basic layer, the real measurement is executed to build a local probability distribution for the existence of a head in the observed image region. A deformable model was chosen to represent the human head allowing for nearly unrestricted movement.

3.2.1 Skincolor In order to extract skin colored regions in the image, the RGB-color intensities are transformed into the normalized rg-chromatic color space. Observing the rg-chroma space for the training material, the mean vector and the covariance matrix have been computed, modeling a 2-dimensional Gaussian. After a threshold operation, a binary mask indicates areas with skin colored pixels. To avoid initialization of hypotheses on skin colored areas which obviously do not indicate a head, the aspect ratios of all skin colored blobs are analyzed. Blobs differing from the aspect ratio of a fitted bounding box will be rejected.

3.2.2 Particle filter The framework is based on an algorithm that uses factored sampling ([4], [5]), which provides simultaneous alternative hypotheses $\mathbf{s}_t$ modeling the probability distribution $\mathbf{w}_t$ at each time step t. Based on the observations $\mathbf{z}_t$, representing the image features, the aim is to track the position of the persons throughout the posterior probability $p(\mathbf{w}_t|\mathbf{z}_{1:t})$. In most cases, there is no functional representation available for this conditional probability, but it can be derived by using Bayes' rule, i.e. all hypotheses have to be evaluated on the image data, described in Section 3.2.3.

3.2.3 Active shape model The shape of the head is exploited as an alternative key feature for our tracking algorithm introduced in the previous section. Due to the variations of the shape (caused by turning the head, for example), a flexible head-shoulder model consisting of 20 landmark points based on the work of Cootes et al. ([6], [7]) was chosen to represent the sihlouette of the head.

3.2.4 Control layer A hyper layer is introduced to control the allocation of the different hypotheses sets, where each of these sets consists of a fixed number of hypotheses to represent exactly one object. The hyper layer is organized similar to the basic particle filter described in Section 3.2.2, but here hypotheses comprise complete hypotheses sets. After sampling and predicting, for each

hypothesis of the sets, the active shape model is run to obtain a weight. Due to this measurement a weight for the hyper layer hypotheses can be computed by summing up the quality scores of all basic layer hypotheses belonging to one hypotheses set, normalized by the number of hypotheses in this set. Additionally a skin color validation is used to verify the number of objects being tracked. For the validation of the sets, the ratio between the area covered by the mean shape and the corresponding skin blob, i.e. the mean blob with the smallest distance to the mean shape, is computed. To allow some tolerance, especially for situations like sitting down or occlusion where no skin color is available, the hyper layer hypothesis weight is only updated if the ratio is less than a given threshold.

3.3 Method C: KLT Tracking

The method proposed in [11] is based on the public domain KLT feature tracker [8], which uses an image pyramid in combination with Newton-Raphson style minimization to efficiently find the most likely position of features in a new image. Flocking behavior and color cues are modeled in this method. The color cue is an RG color model which can be either predefined or trained when an $\mathcal{E}$ is placed on an object. The color cue discards features whose color does not match the expected object color. This color cue in combination with the flock compactness criterion almost eliminates feature drift of background and non-stationary objects in the scene. This method is also resistant to partial occlusions. For head detection, it is assumed that faces correspond to ellipse-like shapes with distinctive axis aspect ratios. The skin color analysis, background subtraction, and con-nected component analysis are used to extract suitable objects for head detection. We designed methods based on progressive background model improvement. The model improvement is done through accumulation of RGB pixel values of current frame in model buffer. Only those pixels evaluated as background are updated. The spatial component analysis by statistical moment calculation is used to distinguish between the heads and other skin colored human parts. Results of the background subtraction are used for tracker initialization, not tracking itself, though it may be suitable even to do tracking.

3.4 Method D: Face Detector

The face detection and tracking method suggested in [10] is based on skin color segmentation, face detection, and tracking which uses movement prediction. Skin color blobs are obtained by connected component analysis and morphological operations. A generalized skin color model is used to avoid hand initialization. The Gaussian color model was trained from manually extracted skin color areas for image segmentation. The face detection algorithm is then applied only on the detected skin colored areas in order to increase the alogirthm speed. The detected skin colored objects, which are recognized as faces are then tracked by using movement prediction. The face detector is based on the weak classifier compound of a Gabor wavelet and a decision tree. Its output determines "probability" that the input image is a face. We constructed a strong classifier as a linear combination of several weak classifiers issued from AdaBoost algorithm (Viola&Jones) [9] and Gabor wavelets. The face detector is trained on normalized face images (24x24 pixels) from MIT (the CBCL data

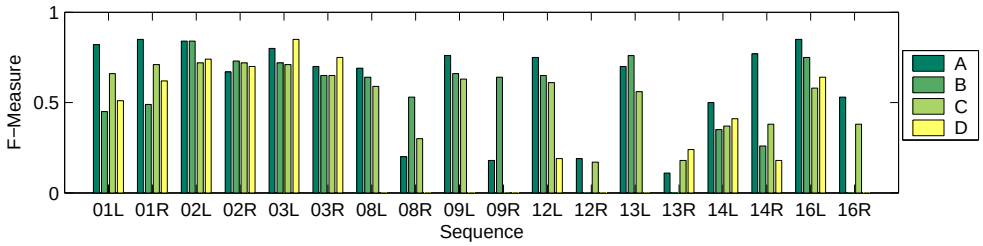

Fig. 3. The F-Measure shows how tightly the estimated bounding boxes fit the ground truth (when passing the coverage test, A = Trans-Dimensional MCMC, B = Probabilistic Active Shape, C = KLT, D = Face Detection).

set containing 1500 face and 14000 non-face images), where simple rectangle image/facial features are replaced by more complex Gabor wavelets. This method is able to detect faces rotated along the depth axis and partially rotated along the vertical/horizontal axis. This is possible because the resulting face detection is performed above sub-sampled and rotated input image.

4 Evaluation

4.1 Method A: Trans-Dimensional MCMC Tracking

This method performed best when tracking frontal heads in the far field of view (such as in sequences 01L,01R,02L, 02R,03L,08L,09L,13L,and 16R). Large variations in head size proved to be problematic for the strong learned head-size prior used in the body/head model. Method A exhibited a high quality of tracking (see Figure 3) and produced a low $\overline{FP}$ rate, which can be attributed to the body-head modeling. However, it reacted poorly to situations in which the back of a participants head was close to the camera, as in the right pane of Figure 2. Such situations attracted MT, MO, FP and FN errors as size-constrained tracking estimates often wandered within the space occupied by the back of the participants head.

Using the $\overline{CD}$ as a measure of overall configuration performance, Method A was outperformed for cases with few participants, but performed better for multiple participants. Typical causes of configuration errors included: MT and MO errors caused by meeting participants in close proximity for long durations, FN errors caused by heads near to the camera (the tracker often missed heads partially in the scene and assigned several small $\mathcal{E}$s to large heads directly in front of the camera).

With regard to maintaining object identity, the model generally performed well, with respectable rates of $\overline{FIT}$ and $\overline{FIO}$. A general trend of $\overline{TP} > \overline{OP}$ indicates that $\mathcal{GT}$s were often tracked by multiple $\mathcal{E}$s, suggesting that tracking estimates periodically died prematurely. This can be partially attributed to difficulty adapting the body model when a person sits down. Typical causes of identification errors included: FIT and FIO errors from improper swapping/occlusion handling, FIT and FIO errors caused by people entering and exiting the scene or standing in close proximity to each other. The computational cost of this model is high as implemented unoptimized in Matlab. However, MCMC particle

Table 3. Configuration Results. (A = Trans-Dimensional MCMC, B = Probabilistic Active Shape, C = KLT, D = Face Detection)

	$\overline{\text{FN}}$				$\overline{\text{FP}}$				$\overline{\text{MT}}$				$\overline{\text{MO}}$				$\overline{\text{CD}}$			
seq	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
01L	.06	.28	.08	.08	.03	.23	0	0	0	0	0	0	0	0	0	0	.06	.08	.08	.07
01R	.05	.39	.12	.06	.03	.27	.03	.02	.03	0	0	0	0	0	0	0	.23	.13	.13	.05
02L	.02	.02	.02	0	.02	0	0	0	0	0	0	0	0	0	0	0	0	.02	.02	0
02R	.14	.20	.14	.10	.44	.16	.04	.02	0	0	0	0	0	0	0	0	.31	.16	.14	.08
03L	.22	.20	.11	.11	.09	.11	.06	0	0	0	0	0	0	0	0	0	.29	.14	.15	.11
03R	.37	.25	.16	.10	.07	.13	.08	.23	0	0	0	0	0	0	0	0	.41	.12	.22	.27
08L	.08	.27	.01	.46	.08	.01	.33	.04	.06	.01	0	0	0	0	0	0	.09	.14	.32	.42
08R	.71	.72	.52	.69	.03	.10	.05	.23	0	0	0	0	0	0	0	0	.74	.62	.47	.52
09L	.04	.18	.11	.16	.11	.25	.13	.19	.02	.03	0	0	0	0	0	.01	.08	.21	.09	.16
09R	.27	.48	.49	-	.08	.33	.31	-	0	0	0	-	0	0	0	-	.21	.54	.27	-
12L	.38	.51	.05	.28	.07	.35	.41	.01	0	.02	.01	0	0	0	.01	0	.43	.40	.37	.27
12R	.70	-	.75	.75	0	-	.65	.02	0	-	0	0	0	-	0	0	.70	-	.65	.73
13L	.38	.71	.06	.72	.09	.28	.46	.16	.17	0	.01	0	0	0	.01	.01	.17	.54	.43	.20
13R	.44	-	.82	.46	.06	-	.36	.08	.23	-	0	0	0	-	0	.01	.36	-	.58	.66
14L	.48	.67	.40	.46	.11	.30	.33	.22	.03	.01	0	0	0	0	0	0	.41	.43	.32	.31
14R	.32	.71	.55	.54	.06	.31	.27	.30	.01	.04	0	0	0	0	0	0	.32	.41	.33	.27
16L	.22	.67	.10	.07	.08	.22	.07	.24	.01	0	0	.01	0	0	0	.01	.22	.46	.08	.24
16R	.08	-	.21	-	0	-	.14	-	0	-	0	-	0	-	0	-	.08	-	.20	-

Table 4. Identification Results. (A = Trans-Dimensional MCMC, B = Probabilistic Active Shape, C = KLT, D = Face Detection)

	$\overline{\text{FIT}}$				$\overline{\text{FIO}}$				$\overline{\text{TP}}$				$\overline{\text{OP}}$			
	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
seq01L	0	0	0	0	.02	0	0	0	.64	.06	.13	1	.74	.05	.74	.73
seq01R	0	0	0	0	.11	0	.16	.03	.54	.19	.33	.83	.73	.14	.72	.79
seq02L	0	0	0	0	0	0	0	.02	.50	1	.33	1	.88	.88	.89	.88
seq02R	0	0	0	0	0	0	.41	.10	.20	.58	.30	1	.67	.52	.67	.52
seq03L	0	0	0	0	.06	0	.01	0	.29	.70	.50	1	.38	.56	.73	.74
seq03R	0	0	0	0	.04	0	.02	.14	.63	.66	.90	1	.21	.51	.63	.53
seq08L	.03	.20	0	0	.09	.35	.01	.14	.80	.40	.18	1	.56	.34	.98	.64
seq08R	0	0	0	0	.03	0	.01	0	.83	.24	.20	.93	.02	.05	.17	.06
seq09L	.16	.13	0	.03	.40	.35	.06	.41	.65	.51	.38	.84	.49	.38	.54	.29
seq09R	0	0	0	-	.11	0	0	-	.75	.17	0	-	.22	.02	0	-
seq12L	.09	.16	.03	.08	.24	.23	.03	.05	.75	.39	.39	.83	.43	.24	.93	.04
seq12R	0	-	0	0	.01	-	0	.01	1	-	.04	.82	.04	-	.09	.41
seq13L	.37	.17	.04	.07	.28	0	.06	.26	.69	.17	.54	.96	.51	.23	.72	.12
seq13R	.01	-	0	.01	.33	-	.02	.07	.83	-	.05	.91	.31	-	.06	.42
seq14L	.04	.07	0	.01	.05	.12	.04	.17	.76	.35	.28	.86	.40	.13	.53	.29
seq14R	.21	.06	0	0	.27	.13	.04	.22	.65	.41	.15	1	.37	.13	.26	.37
seq16L	.02	.07	0	.03	.04	0	.05	.37	.84	.37	.34	.93	.56	.13	.50	.37
seq16R	.05	-	0	-	.18	-	.02	-	.95	-	.12	-	.46	-	.40	-

filters have proven to be more efficient than SMC methods for searching large spaces, such as the joint state-space of a multi-object configuration. An example of trans-dimensional MCMC tracker output can be seen in Figure 5.

4.2 Method B: Probabilistic Active Shape Tracking

Results for the Active Shape Tracker appear in Tables 3 & 4. Note that for results obtained by the Active Shape Tracker, persons appearing in the test set were not used for training.

Especially for the 1-person scenarios (Sequences 1, 2 and 3) very precise tracking results have been obtained, as demonstrated by very small error measures $\overline{FN}$ and $\overline{FP}$ accompanied by high $\overline{TP}$ and $\overline{OP}$. In scenes with more participants the amount of challenges like occlusion and blocking of a camera raises. The experiments have confirmed one advantage of the proposed ASM-tracking approach: since the gradient image is the basis for the adaption of the shape, par-

tial occlusions only have a very weak influence on the tracking output. In Figure 5, frames showing the robustness of our method are presented. At the beginning of the occlusion the shape fit quality of the occluded person decreases. This is due to the occlusion of a large number of landmark points along the shape. Once enough of the head is visible again, the shape recovers and continues tracking.

While partial occlusions can be handled quite well, challenges like a sudden appearance of the back of a head near a camera (as in the right pane of Fig. 2) pose a problem. Movements next to the camera occur very fast and due to the skin color region corresponding to the neck, the tracker initializes hypotheses at too small a scale. The shape cannot adapt to the real size of the head in the remaining time. For these situations, most of the back heads appearing next to the camera remain untracked and lead to an increase of both $\overline{FN}$ and $\overline{FP}$.

In general, for multi-person meetings, $\overline{FN}$ and $\overline{FP}$ tend to raise with the number of people, indicating that a growing number of $\mathcal{E}$s is not assigned to any $\mathcal{GT}$ object. One step to solve this problem could be to train the variations allowed by the shape model specifically on some of the occurring movements. Nevertheless, the low numbers of $\overline{FIT}$ and $\overline{FIO}$ for all sequences show that there is nearly no problem in following the track of a fixed person. Since our framework is based on statistical techniques, only an averaged frame rate can be declared, which is approximately 3 fps (non-optimized).

4.3 Method C: KLT Tracking Method

The KLT method is the only method of the four which operates, as implemented, at a speed close to real-time. It tracks detected objects with high speed and accuracy. This comes at the expense of a high $\overline{FP}$ rate, as hands are often misinterpreted as heads. $\overline{FP}$ could be reduced using additional topological knowledge about the scene and temporal correspondence, or by using some face detector. The situations in which the back of participants head is visible are not detected properly. $\overline{MT}$ errors occur when the detected area is larger than is specified for a head. Also, $\overline{FN}$ errors are caused by situations where the head lacks a signficatant amount of skin color, one of the important detected features. Some $\overline{FIT}$ errors are caused by tracker re-initialization, which is performed after an object is stationary for more than 10 frames to prevent tracking of the background. The KLT tracking algorithm with head detection runs at approximately 17 frames per second on an Athlon 64 3500+ for a two-person sequence.

4.4 Method D: Face Detector

The face detector is based purely on skin color detection and only gives good results for certain lighting conditions. Skin-colored segments of the background pose a problem for the face detector, and the $\overline{FN}$ increases as the detector struggles with non-frontal faces. Poor FIT rates are caused by $\mathcal{E}$s often terminating, followed by a new detection. The size of training data set showed to be enough, but better a better way to deal with Adaboost overfitting is required.

4.5 Summary

The trans-dimensional MCMC tracker maintains identities well and boasts a high quality of tracking, but suffers from a high computational cost and lower $\overline{CD}$ for 1 & 2 person scenarios. The Active Shape tracking method is most robust

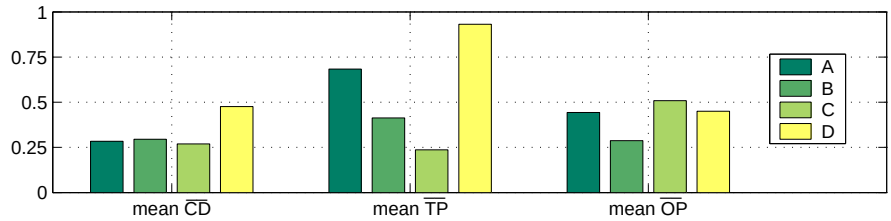

Fig. 4. Mean $\overline{CD}$, $\overline{TP}$, and $\overline{OP}$ computed over the entire test set. (A = Trans-Dimensional MCMC, B = Probabilistic Active Shape, C = KLT, D = Face Detection).

to occlusion and performs well in the 1 & 2 person scenarios, but also suffers from high cost and degraded performance with 3 & 4 people. The KLT tracker provides good results at high processing rates, which makes this algorithm most suitable for real-time applications. On the other hand, the face detection algorithm gives precise positions of the faces at the expense of algorithm speed, sensitivity to lighting conditions, and dependence on frontal faces.

5 Conclusion

The AV16.7.ami corpus contains many difficult real-life scenarios which remain challenging for state-of-the-art tracking models. The results presented are valuable as they represent the first evaluation of methods for multi-person tracking in meetings using a common data set in the context of the AMI project.

Acknowledgements This work was supported by the EC project Augmented Multi-party Interaction (AMI, publication AMI-XX).

References

1. K. Smith, S. Ba, J.M. Odobez, D. Gatica-Perez, "Evaluating Multi-Object Tracking", *CVPR Workshop on Empirical Evaluation Methods in Computer Vision (EEMCV)*, San Diego, CA, June 2005.
2. C.J. Van Rijsbergen, *Information Retrieval*, Butterworth-Heinemann, Newton, MA, USA, 1979.
3. K. Smith, S. Ba, J.M. Odobez, D. Gatica-Perez, "Multi-Person Wander-Visual-Focus-of-Attention Tracking", *IDIAP-RR-05-80*, Nov 2005.
4. M. Isard and A. Blak, "Condensation – conditional density propagation for visual tracking", *International Journal of Computer Vision 29(1)*, pp. 5–28, 1998.
5. Michael Isard and Andrew Blake, "A Mixed-State CONDENSATION Tracker with Automatic Model-Switching", *International Conference on Computer Vision (ICCV)*, 1998.
6. T. Cootes and C. Taylor, *Statistical models of appearance for computer vision*, 2004.
7. T. Cootes, G. Edwards and C. Taylor, "A comparative evaluation of active appearance model algorithms", *British Machine Vison Conference*, Southampton, UK, Sept. 1998.
8. M. Kölsch and M. Turk, "Fast 2D Hand Tracking With Flocks and Multi Cue Integration", Department of Computer Science, University of California, 2005.
9. J. Viola and M. Jones, "Robust Real-time Object Detection", Technical Report 2001/01, Com-paq CRL, February 2001.
10. I. Potucek, S. Sumec, M. Spanel, "Participant activity detection by hands and face move-ment tracking in the meeting room", *Computer Graphics International (CGI)*, Los Alamitos, 2004.
11. M. Hradis, R. Juranek, "Real-time Tracking of Participants in Meeting Video", *Proceedings of CESCG*, Wien, 2006.

Probabilistic Active Shape Trans-Dimensional MCMC Face Detector

Fig. 5. Tracker output on the AV16.7.ami data corpus. Left Column: The Active Shape Tracker fits the head-shoulder contour (black shape) to the image data and thus determines the object. These frames contain a lot of challenges such as partial occlusion and dense clutter in the background. Nevertheless, the tracking output (light gray rectangles) and the number of identified tracks are precise. Center Column: The trans-dimensional MCMC tracker makes use of binary features to track bodies, maintain identity, and localize the head. This sequence shows the ability of the tracker to add and remove objects from the scene. Right Column: Results for the face detection method.