Tracking of Multiple Humans in Meetings

Bo Wu and Ram Nevatia
University of Southern California
Institute for Robotics and Intelligent Systems
Los Angeles, CA 90089-0273
$\{bowu|nevatia\}@usc.edu$

Abstract

Tracking conferees in meeting rooms is important for many applications. In this paper, we present an approach based on single-frame head-shoulder detection to track multiple humans in meetings. The responses of a multiple view head-shoulder detection system are taken as the observation of the human hypotheses. Trajectory initialization and termination are fully automatic and rely on the evidence collected from the detection responses. An object is tracked by data association if its corresponding detection response can be found; otherwise it is tracked by a meanshift style tracker. Finally the tracked hypotheses are verified by evidence collected from body part movements. The system is evaluated on two meeting video corpora.

1 Introduction

Recently, meeting video analysis has attracted much attention. For many applications, such as an automatic meeting recorder, tracking the conferees is necessary. This is made difficult due to a number of factors, such as the variety of poses; different view points; partial occlusion by other objects; and illumination changes. In this paper, we propose a method to automatically track multiple humans in meetings from a single camera, based on head-shoulder detection. For an automatic multiple object tracking system, three main problems need to be addressed: 1) when to initialize a trajectory? 2) how to track an object? and 3) when to terminate a trajectory? Our approach relies on the detection responses to answer all the three questions. After trajectories are obtained, we use cues from body part movements to verify the hypotheses.

1.1 Related Work

To track the humans we first need to detect them. The observations of the human hypotheses may come from different cues. Some previous efforts use skin-color segmentation [1, 2, 3, 4, 5] or gray-level face detection [2, 6] to find the humans. Some others detect humans by background subtraction [3, 4, 7]. However the application of these methods are limited, since the underlying assumptions are not always valid. For example, faces are not visible from rear view point.

In meetings, usually the conferees sit around a table with the lower body occluded. This suggests that an upper-body or a head-shoulder detector is a good choice to find the humans. Recently, Zhao and Davis [8] proposed an upper-body detection and segmentation method to find conferees in meeting. They use template matching as a shape constraint on color-based segmentation. The two steps process in an iterative manner. In our previous work [9], we proposed a human detection method by combining body part detection responses. For each part a cascade detector is learned by boosting *edgelet* features based weak classifiers. In this work, we use the learning method in [9] to train *head-shoulder* detectors to detect humans in meetings. We believe local features are more robust to partial occlusion than the global features.

For multiple object tracking, because the hypotheses space is usually of high dimension and the observations are relatively weak, many previous methods have used sampling, *e.g.* a particle filter [1, 2, 4], to track the objects. In our approach, the observations of the human hypotheses obtained from the head-shoulder detector are more informative than those from skin-color detection and background subtraction. Hence we develop a data association style algorithm for tracking.

1.2 Outline of Our Approach

We track humans by detecting head-shoulder part of the body frame by frame. This avoids the necessity of computing reliable motion blobs and enable us to find static humans as well as the moving ones. The responses from the detector are used as the inputs for the tracker. In [9] only the frontal/rear view is considered. We build our detection system by extending this method to include left and right profile. Since our method does not rely on face or

0-7695-2646-2/06 $20.00 (c) 2006 IEEE

skin-color, it can deal with rear view. We initialize a trajectory when enough evidences are collected from the detection responses. To track the human, we first look for the correspondent response of the hypothesis from the detection result, if the response is found, the human is tracked by data association; otherwise, a color based meanshift tracker [10] is used to follow the human. Most of the time, humans are tracked successfully by data association, while the meanshift tracker gets utilized occasionally and for short periods. If a hypothesis is not "seen" by the detector for a certain period, the trajectory is terminated. The initialization/termination strategy can be improved by the knowledge of the entrance/existance of the room. However in this work we do not make use of such information.

After tracking we use body part movements to verify the tracked hypotheses. This is based on the observation that although most of the time the conferees tend to sit there without big movements, the body parts, *e.g.* head, and hands, move sometime; while the false alarms from scene objects are always stationary. In practice, we calculate the *accumulated motion* as the criterion.

We have a paper accepted by the main conference of CVPR'06 [11], which proposes a method to track multiple walking/standing humans by detecting the body parts. The approach described in this paper is a modified version of that in [11], and is applied on a different problem, meeting video conferee tracking. The rest of the paper is organized as follows: Section 2 describes our multi-view head-shoulder detection system; Section 3 gives the details of our detection based tracking method; Section 4 describes the motion based verification method; Section 5 shows the experiment results; and conclusions and future work are given in the last section.

2 Detection of Head-Shoulder

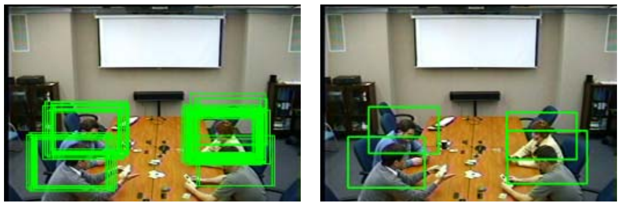
In [9], four part detectors are learned for full-body, head-shoulder, torso, and legs respectively. We only use the one for head-shoulder here, since in a typical meeting room scenario, only the head-shoulder is visible consistently. For head-shoulder part, two detectors are learnt: one for the left profile view, and one for the frontal/rear view (the detector for right profile view is generated by flipping the left profile view horizontally). Nested cascade detectors are learned by boosting edgelet feature based weak classifiers, as in [9]. The training set contains 2,542 positive samples for frontal/rear views, 3,011 for left profile view, and 655 negative images. The positive samples are from the NIST meeting video corpus [12] and the negative images are collected from the Internet. The positive samples are normalized to 36×24 pixel. All the negative images are of indoor scenes. See Fig.1 for some examples of the training set.

For detection, the input image is scanned by all three detectors and the union of their responses is taken as the

(a) Positive samples (b) Negative images

Figure 1. Training samples for head-shoulder detector.

multi-view detection result. The responses of the detection system have two levels. The first level is of the *original responses* of the detectors, see Fig.2(a). The second level is of the *merged responses*, which is the result of applying a clustering algorithm to the original responses, see Fig.2(b). The second level is considered to be a one-to-one mapping, while the first is not. Note, the detection response is a rectangle around the head-shoulder region, not an accurate segmentation.



(a) Original responses (b) Merged responses

Figure 2. Static detection responses.

3 Detection based Tracking

We now describe our method for trajectory initialization, object tracking, and trajectory termination. First we introduce the data association method which is common to the three modules.

3.1 Data Association

The task of data association is to match the detection responses with the human hypotheses. We use a greedy algorithm to do this. Suppose at time t, we have n hypotheses $H_1, \ldots, H_n$, whose predictions at time $t+1$ are $\widehat{\mathbf{r}}_{t+1,1}, \ldots, \widehat{\mathbf{r}}_{t+1,n}$, and at time $t+1$ we have m responses $\mathbf{r}_{t+1,1}, \ldots, \mathbf{r}_{t+1,m}$. First we compute an $m \times n$ affinity matrix $\mathbf{A}$ of all $(\widehat{\mathbf{r}}_{t+1,i}, \mathbf{r}_{t+1,j})$ pairs, *i.e.* $\mathbf{A}(i,j)$ is an affinity score between $\widehat{\mathbf{r}}_{t+1,i}$ and $\mathbf{r}_{t+1,j}$. Then in each step, the pair, denoted by $(i^\star, j^\star)$, with the largest affinity is taken as a match and the $i^\star$-th row and the $j^\star$-th column of $\mathbf{A}$ are deleted. This procedure is repeated until no more valid pairs are available.

We represent a detection response by a 4-tuple, $\mathbf{r} = \{\mathbf{p}, s, f, \mathbf{c}\}$, where $\mathbf{p}$ is the image position (x, y); s is the size; f is a real-valued detection confidence; and $\mathbf{c}$ is an appearance model. The first three elements $\mathbf{p}$, s, and f are obtained from the detection process directly. The appearance model $\mathbf{c}$ is implemented as a color histogram; computation and update of $\mathbf{c}$ is described, in detail, later in Section 3.3. Detection is done frame by frame. The affinity between two responses, $\mathbf{r}_1$ and $\mathbf{r}_2$, is defined by

$$A(\mathbf{r}_1, \mathbf{r}_2) = A_{pos}(\mathbf{p}_1, \mathbf{p}_2) A_{size}(s_1, s_2) A_{appr}(\mathbf{c}_1, \mathbf{c}_2) \quad (1)$$

where A_{pos}, A_{size}, and A_{appr} are affinity measures based on position, size, and appearance respectively. Their definitions are

$$\begin{aligned} A_{pos}(\mathbf{p}_1, \mathbf{p}_2) &= \gamma_{pos} \exp\left[-\frac{(x_1-x_2)^2}{\sigma_x^2}\right] \exp\left[-\frac{(y_1-y_2)^2}{\sigma_y^2}\right] \\ A_{size}(s_1, s_2) &= \gamma_{size} \exp\left[-\frac{(s_1-s_2)^2}{\sigma_s^2}\right] \\ A_{appr}(\mathbf{c}_1, \mathbf{c}_2) &= B(\mathbf{c}_1, \mathbf{c}_2) \end{aligned} \quad (2)$$

where $B(\mathbf{c}_1, \mathbf{c}_2)$ is the Bhattachayya distance between two color histograms and γ_{pos} and γ_{pos} are normalizing factors. The underlying assumption for this affinity measure is that within a short period the position and the appearance of a human do not change much.

3.2 Trajectory Initialization

The basic idea of our initialization strategy is to start a trajectory when enough evidence is collected from the detection responses. Due to the correlation between neighboring frames, if the detector outputs a false alarm at certain position in one frame, the probability is high that a false alarm will appear around the same position in the next frame. This is called *persistent false alarm* problem. However, suppose we have found T consecutive responses, $\{\mathbf{r}_1, \ldots, \mathbf{r}_T\}$ corresponding to one object hypothesis H, still the probability of H being a false alarm should be an exponentially decreasing function of T. We model it as $e^{-\lambda_{init}\sqrt{T}}$. The confidence of initializing a trajectory for H is defined by

$$\begin{aligned} &InitConf(H; \mathbf{r}_{1..T}) \\ &= \underbrace{\frac{1}{T-1}\sum_{t=1}^{T-1} A(\hat{\mathbf{r}}_{t+1}, \mathbf{r}_{t+1})}_{(1)} \cdot \underbrace{\left(1 - e^{-\lambda_{init}\sqrt{T}}\right)}_{(2)} \end{aligned} \quad (3)$$

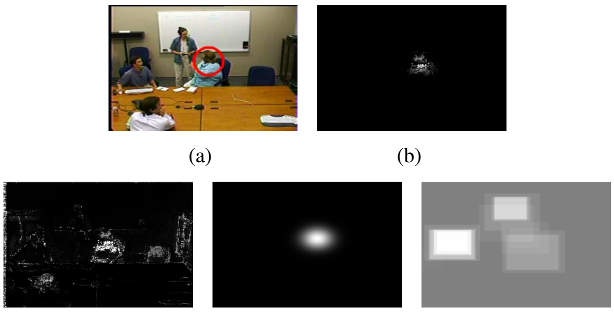
The first term in the left side of Eq.3 is the average affinity of the T responses, and the second term is based on the detector's property. Our trajectory initialization strategy is: if $InitConf(H)$ is larger than a threshold, θ_{init}, a trajectory is started from H, and H is called a *confident trajectory*. Otherwise, H is called a *potential trajectory*. A trajectory hypothesis H is represented as a 3-tuple, $\{\{\mathbf{r}_t\}_{t=1,\ldots,T}, \mathbf{D}, \mathbf{C}\}$, where $\{\mathbf{r}_t\}$ is a series of responses, $\mathbf{C}$ is the appearance model, and $\mathbf{D}$ is a dynamic model. In practice, $\mathbf{C}$ is the average of the appearance models of all detection responses, and $\mathbf{D}$ is modeled by a Kalman filter for constant speed motion.

3.3 Trajectory Growth

After a trajectory is initialized, the object is tracked by two strategies, data association and meanshift tracking. For a new frame, first, for all existing hypotheses, we look for their corresponding responses. If there is a new response matched with a hypothesis H, then H grows by data association, otherwise a meanshift tracker [10] is applied. The basic idea of meanshift is to track a probability distribution. In our method we combine the appearance model, $\mathbf{C}$, the dynamic model, $\mathbf{D}$, and the detection confidence, f, to build a likelihood map which is then fed into the meanshift tracker. A dynamic probability map, $P_{dyn}(\mathbf{u})$, where $\mathbf{u}$ represents the image coordinates, is calculated from the dynamic model $\mathbf{D}$, see Fig.3(d). Suppose, at one frame the set of the original responses of the head-shoulder detector is $\{\mathbf{r}_j\}$, then the detection probability map $P_{det}(\mathbf{u})$ is defined by

$$P_{det}(\mathbf{u}) = \sum\nolimits_{j:\mathbf{u}\in Reg(\mathbf{r}_j)} f_j + ms \quad (4)$$

where $Reg(\mathbf{r}_j)$ is the image region corresponding to $\mathbf{r}_j$, f_j is a real-valued confidence of $\mathbf{r}_j$, and ms is a constant corresponding to the missing rate. Note, the original response is used here, because of possible errors in the clustering algorithm (see Fig.3(e)).



(a) (b)

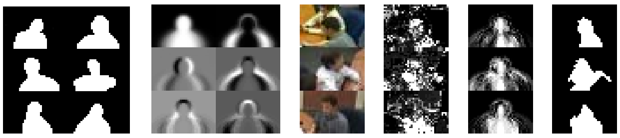
(c) (d) (e)

Figure 3. Probability map for meanshift: a) original frame; b) final probability map; c) appearance probability map; d) dynamic probability map; e) detection probability map. (The human concerned is marked by a red ellipse.)

Let $P_{appr}(\mathbf{u})$ be the appearance probability map. As $\mathbf{C}$ is a color histogram, $P_{appr}(\mathbf{u})$ is the bin value of $\mathbf{C}$ (see Fig.3(c)). To estimate $\mathbf{C}$, the object needs to be segmented so that we know which pixels belong to the object. The rectangle of the detection response is not accurate enough for this purpose. Also as human head can articulate with respect to the shoulder, it's difficult to build a constant segmentation mask. Here we proposed a simple PCA based approach to do this. At the training stage, examples are collected and the object regions are labeled by hand, see Fig.4(a). Then a PCA model is learned from this data, see Fig.4(b). Suppose we have an initial appearance model $\mathbf{C}_0$, which may be calculated from the mean vector of the PCA model. Given a new sample (Fig.4(c)), first its color probability map is calculated from $\mathbf{C}_0$ (Fig.4(d)), then we use the PCA model as a global shape constraint by reconstructing the probability map (Fig.4(e)). The thresholded reconstruction map (Fig.4(f)) is taken as the final object segmentation, which is used to update $\mathbf{C}_0$. This segmentation method is far from perfect, but very fast and adequate to update the appearance model. Combining $P_{appr}(\mathbf{u})$, $P_{dyn}(\mathbf{u})$, and $P_{det}(\mathbf{u})$, we define the image likelihood at pixel $\mathbf{u}$ by

$$L(\mathbf{u}) = P_{appr}(\mathbf{u})P_{dyn}(\mathbf{u})P_{det}(\mathbf{u}) \tag{5}$$

Fig.3 shows an example of probability map computation. Another issue during tracking is of updating the model. Our strategy is to do so only when the object is found by the detection system, as the detected responses are more reliable than the tracked ones.



(a) (b) (c) (d) (e) (f)

Figure 4. PCA based head-shoulder segmentation: a) training samples; b) eigenvectors. The left top one is the mean vector; c) original human samples; d) color probability map; e) PCA reconstruction; f) thresholded segmentation map.

3.4 Trajectory Termination

The strategy of terminating a trajectory is similar to that of initializing it. If in consecutive T time steps, no detection responses are found for an object H, we compute a termination confidence of H by

$$\begin{aligned} &EndConf(H; \mathbf{r}_{1..T}) \\ &= \left(1 - \frac{1}{T-1}\sum_{t=1}^{T-1} A(\hat{\mathbf{r}}_{t+1}, \mathbf{r}_{t+1})\right)\left(1 - e^{-\lambda_{end}\sqrt{T}}\right) \end{aligned} \tag{6}$$

Note here that the responses $\mathbf{r}_t$ are obtained from the meanshift tracker, not from the head-shoulder detector. If $EndConf(H)$ is larger than a threshold, θ_{end}, hypothesis H is terminated; we call it a *dead trajectory*, otherwise we call it an *alive trajectory*.

Now let's put the above three modules, trajectory initialization, tracking, and termination, together. Fig.5 gives our tracking algorithm. The algorithm is called *forward* tracking, as it only looks ahead. Because the trajectory initialization may have some delay, we also use a *backward* tracking procedure which is the exact reverse of the forward tracking. After a trajectory is initialized, it may grow in both forward and backward directions.

4 Verification based on Motion

The main issue of the head-shoulder tracker is the persistent false alarm problem mentioned in Section 3.2. Many of these persistent false alarms are from some human like scene object or background clutter, see Fig.6 for some examples. These false alarms are stationary, while the real persons move the body parts at least occasionally. As the humans usually sit in a meeting without global translation, it is infeasible to track the humans based on motion blobs, see Fig.7(b). However after we get the trajectory hypotheses, the body part motion can be used to verify them.

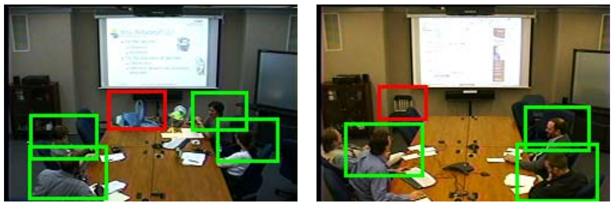
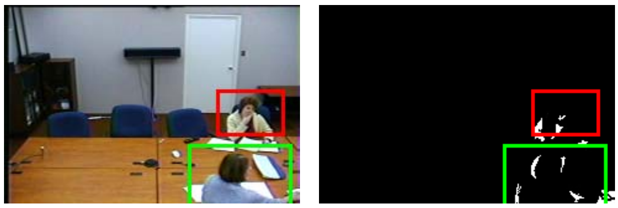


Figure 6. Examples of persistent false alarm (the red rectangles).

Denote by $R_t(H)$ and M_t the region of a human hypothesis H, and the motion detection result at frame t respectively. $R_t(H)$ is obtained by the segmentation method described in Section 3.3. Define the accumulated motion ratio of this human by

$$M_r(H) = \frac{|\bigcup_t (R_t(H) \cap M_t)|}{|\bigcup_t R_t(H)|} \tag{7}$$

If M_r is larger than a threshold, θ_m, the hypothesis is verified; otherwise, it is discarded.



(a) head-shoulder tracking (b) motion detection

Figure 7. Motion detection.

5 Experimental Results

We show results and evaluations on two video sets to demonstrate the effectiveness of our method. The first set is a selection of the IDIAP meeting corpus [13]. The second set is a selection of the NIST meeting corpus [12]. The frame size of these two sets is 720×480 pixel and the sampling rate is 30 FPS. The test sets have no overlap with the training set described in Section 2.

5.1 Performance Evaluation Criteria

To evaluate the performance of our system quantitatively, we defined five criteria for tracking: number of mostly tracked trajectories (more than 80% of the trajectory is tracked), number of mostly lost trajectories (more than 80% of the trajectory is lost), number of fragments of trajectories (a result trajectory which is less than 80%, of a ground truth trajectory), number of false trajectories (a result trajectory corresponding to no object), and the frequency of identity switches (identity exchanging between a pair of result trajectories). These five categories cover most of the typical errors observed in our experiments.

Forward Head-shoulder Tracking

Let the set of hypotheses be F, initially $F = \Phi$.

For each time step t (denote by F_t the set of all alive trajectories in F at time t)

1. Single frame detection: Do head-shoulder detection. Let the result set be R_t.
2. Data association:
   (a) Associate hypotheses in F_t with responses in R_t. Let the set of matched hypotheses be $\tilde{F}_t$.
   (b) Build a new hypothesis H from each unmatched response in R_t, and add H into F and F_t.
3. Pure tracking: For each confident trajectory in $F_t - \tilde{F}_t$, grow it by meanshift tracking.
4. Model update:
   (a) For each hypothesis in $\tilde{F}_t$, update its appearance model and dynamic model.
   (b) For each potential trajectory in $\tilde{F}_t$, update its initialization confidence.
   (c) For each trajectory in $\tilde{F}_t$, reset its termination confidence to 0.
   (d) For each trajectory in $F_t - \tilde{F}_t$, update its termination confidence.

Output all confident trajectories in F as the final results.

Figure 5. Forward head-shoulder tracking algorithm.

5.2 Results on IDIAP Set

In IDIAP corpus [13], three static cameras are used to record the meetings. The first two cameras face the humans sitting behind a table. The third camera faces the screen and the presenter if there is one. We select from the corpus, 36 sequences, 12 for each camera, 59,204 frames overall, to form our first test set. In each sequence, the persons enter or exit the scene once or twice. This set is relatively easy, as there are at most two persons in one frame and the scene is quite clear. (If the person leaves and reenters the scene, we count him/her as two separate people.) Table 1 lists the tracking results on this set, without motion verification. It can be seen that the method performs very well and outputs only one false alarm so that the verification is unnecessary. Some sample frames and results are shown in Fig.8.

GT	MT	ML	Fgmt	FAT	IDS
65	60	0	9	1	0

Table 1. Tracking performance on IDIAP set [13], 36 sequences. (GT: ground truth; MT: mostly tracked; ML: mostly lost; Fgmt: fragment of trajectory; FAT: false alarm trajectory; IDS: ID switch)

5.3 Results on NIST Set

In the NIST corpus [12], five cameras are used to record the meetings. Two of the cameras are from the frontal/rear view point; the other two are from the profile view. These four cameras are static. The last camera is moving and/or zooming, usually following the speaker. We select from this corpus, 50 sequences, 10 from each camera, 236,790 frames overall, to form our second test set. This set is difficult compared to the IDIAP set. There are a lot of scenes objects, such as chairs, keyboards, reading lamps, which make the background cluttered. Partial occlusions are common as the persons interact and sit close to each other. Table 2 lists the tracking performance on this set. It can be seen that our method achieves good results. The motion verification is applied on the sequences from the static cameras. It gets rid of a large portion of the false alarms, while only remove a few true trajectories. Some sample frames and results are shown in Fig.8. We only do tracking from individual views in this work, although integrating multiple cameras could help to improve the performance.

	GT	MT	ML	Fgmt	FAT	IDS
Without MV	183	165	8	12	18	2
With MV		163	11	10	10	2

Table 2. Tracking performance on NIST set [12], 50 sequences. (MV: motion verification. For the other abbreviations see Table 1.)

On the two test sets, the average detection rate of the head-shoulder detector is about 80% and there are about 0.25 false alarms per frame. For tracking, about 80% of the successful tracking is due to the data association with detection responses, *i.e.* the object is "seen" by the static detector; the remaining 20% is from the meanshift tracker. The speed of the entire system is about 0.5 FPS on a 2.8G Hz Pentium CPU; the program is coded in C++ using OpenCV functions.

6 Conclusion and Future Work

We proposed a fully automatic human tracking method for meeting videos. The system has achieved very good performance on two challenging meeting video corpora. As we use the frame head-shoulder detection responses as the observation, our approach is insensitive to camera motion and can handle rear view. Currently, the system does not

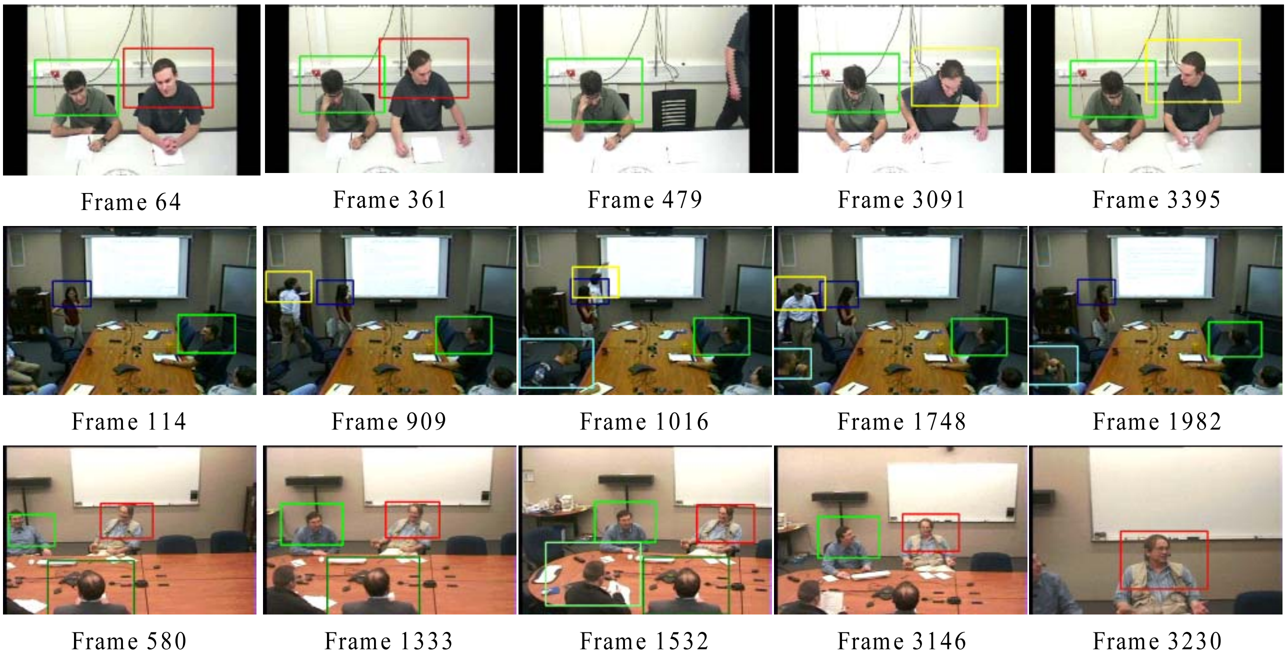


Figure 8. Sample tracking results. The first row is from IDIAP set [13]; the second and third rows are from NIST set [12]. The third row is from a moving/zooming camera.

use multi-camera context and camera calibration information. In our future work, we will try to extend the system to multiple camera settings and extend tracking from 2D to 3D.

Acknowledgements: This research was partially funded by the Advanced Research and Development Activity of the U.S. Government under contract MDA-904-03-C-1786.

References

[1] D. G-Perez, J.-M. Odobez, S. Ba, K. Smith, and G. Lathoud. Tracking People in Meetings with Particles. Proc. Int. Workshop on Image Analysis for Multimedia Interactive Service (WIAMIS), 2005.

[2] F. Wallhoff, M. Zobl, G. Rigoll, and I. Potucek: Face Tracking in Meeting Room Scenarios Using Omnidirectional Views. ICPR'04. Vol IV: 933-936.

[3] I. Potucek, and S. Sumec. Participant Activity Detection by Hands and Face Movement Tracking in the Meeting Room. Computer Graphics International 2004: 632-635.

[4] H. N-Charif and S. J. McKenna. Head Tracking and Action Recognition in a Smart Meeting Room. PETS'03.

[5] R. Stiefelhagen. Tracking Focus of Attention in Meetings. ICMI'02.

[6] X. Song, and R. Nevatia. Combined Face-body Tracking in Indoor Environment. ICPR'04. Vol IV: 159-162

[7] D. Lee, B. Erol, and J. J. Hull. Segmenting People in Meeting Videos Using Mixture Background and Object Models. IEEE Pacific Rim Conference on Multimedia 2002: 791-798

[8] L. Zhao, and L. Davis. Closely Coupled Object Detection and Segmentation. ICCV'05. Vol I: 454-461

[9] B. Wu, and R. Nevatia. Detection of Multiple, Partially Occluded Humans in a Single Image by Bayesian Combination of Edgelet Part Detectors. ICCV'05. Vol I:90-97

[10] D. Comaniciu, V. Ramesh, and P. Meer. The Variable Bandwidth Mean Shift and Data-Driven Scale Selection. ICCV'01. Vol I: 438-445

[11] B. Wu, and R. Nevatia. Tracking of Multiple, Partially Occluded Humans based on Static Body Part Detection. In CVPR 2006.

[12] J. Garofolo, C. Laprum, M. Michel, V. Stanford, and E. Tabassi. The NIST Meeting Room Pilot Corpus. In: Proc. of Language Resource and Evaluation Conference. 2004.

[13] I. McCowan, S. Bengio, D. Gatica-Perez, G. Lathoud, F. Monay, D. Moore, P. Wellner, and H. Bourlard. Modeling Human Interaction in Meetings. in Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), Hong Kong, April 2003.