A Joint Particle Filter for Audio-visual Speaker Tracking

Kai Nickel, Tobias Gehrig, Rainer Stiefelhagen, John McDonough
Interactive Systems Labs
Universität Karlsruhe (TH)
Germany
{nickel,tgehrig,stiefel,jmcd}@ira.uka.de

ABSTRACT

In this paper, we present a novel approach for tracking a lecturer during the course of his speech. We use features from multiple cameras and microphones, and process them in a joint particle filter framework. The filter performs sampled projections of 3D location hypotheses and scores them using features from both audio and video. On the video side, the features are based on foreground segmentation, multi-view face detection and upper body detection. On the audio side, the time delays of arrival between pairs of microphones are estimated with a generalized cross correlation function. Computationally expensive features are evaluated only at the particles' projected positions in the respective camera images, thus the complexity of the proposed algorithm is low. We evaluated the system on data that was recorded during actual lectures. The results of our experiments were 36 cm average error for video only tracking, 46 cm for audio only, and 31 cm for the combined audio-video system.

Categories and Subject Descriptors

I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis

General Terms

Algorithms, Experimentation, Human Factors

Keywords

Speaker Tracking, Particle Filters, Multimodal Systems

1. INTRODUCTION

Person tracking is a basic technology for realizing context-aware human-computer interaction applications. The scenario addressed in this work is a smart lecture room, where information about the lecturer's location could help to automatically create an audio-visual log of the presentation. Here, the location of the lecturer could be used as a target for active camera control [14], acoustic beamforming [25] and audio-visual speech recognition [18] in order to supply high-resolution images of the speaker and to enhance the quality of automatic speech recognition.

The task of lecturer tracking poses two basic problems: localizing the lecturer (in terms of 3D head coordinates) and disambiguating the lecturer from other people in the room. In the proposed approach, we jointly process images from multiple cameras and the signal from multiple microphones in order to track the lecturer both visually and acoustically. The algorithm is based on the assumption, that the lecturer - among all other people in the room - is the one that is speaking and moving most of the time, i.e. exhibiting the highest visual and acoustical activity.

The central issue in audio-visual tracking is the question of how to combine different sensor streams in a beneficial way. In our approach, we integrate audio and video features such that the system does not rely on a single sensor or a certain combination of sensors to work properly. In fact, each single camera and each microphone pair alone can contribute to the track. The main contribution of this paper is a particle filter framework for the computationally efficient integration of acoustic source localization, multi-camera person detection (frontal face, profile face, upper body) and foreground segmentation. The 3D position of the lecturer is robustly being determined by means of sampled projection instead of triangulation.

1.1 Related Work

Person tracking is one of the primary challenges in human-computer interaction. Already, numerous approaches exist that deal with this problem, employing a variety of features, tracking schemes and sensor setups.

For acoustic source localization, several authors have proposed solving this optimization problem with standard gradient based iterative techniques. While such techniques typically yield accurate location estimates, they are typically computationally intensive and thus ill-suited for real-time implementation [1, 2]. For any pair of microphones, the surface on which the TDOA is constant is a hyperboloid of two sheets. A second class of algorithms seeks to exploit this fact by grouping all microphones into pairs, estimating the TDOA of each pair, then finding the point where all associated hyperboloids most nearly intersect. Several closed-form position estimates based on this approach have appeared in the literature; see Chan and Ho [3] and the literature review found there. Unfortunately, the point of intersection of two hyperboloids can change significantly based on a slight change in the eccentricity of one of the hyper-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
ICMI'05, October 4–6, 2005, Trento, Italy.
Copyright 2005 ACM 1-59593-028-0/05/0010 ...$5.00.

boloids. Hence, a third class of algorithms was developed wherein the position estimate is obtained from the intersection of several spheres. The first algorithm in this class was proposed by Schau and Robinson [19], and later came to be known as *spherical intersection*. Perhaps the best known algorithm from this class is the *spherical interpolation* method of Smith and Abel [20]. Both methods provide closed-form estimates suitable for real-time implementation.

Brandstein *et al* [2] proposed yet another closed-form approximation known as *linear intersection*. Their algorithm proceeds by first calculating a bearing line to the source for each pair of sensors. Thereafter, the point of nearest approach is calculated for each pair of bearing lines, yielding a potential source location. The final position estimate is obtained from a weighted average of these potential source locations. Other recent work on acoustic source localization includes that by Huang *et al* [9], who developed an iterative technique based on a spherical least square error criterion, that is nonetheless suitable for real-time implementation, as well as the work by Ward *et al* [24], who proposed using partice filter together with both time delay of arrival estimation and steered beamformers. In other work by the same authors [11], a variant of the extended Kalman filter was used for acoustic speaker tracking. This approach was extended in [8] to add video features.

Concerning video features, it has often be proposed to use color models for the task of tracking articulated objects like the human body. Wren et al. [26], e.g., use a region-based model of color and shape to obtain a 2D-blob representation of the human body in front of a static scene in their well-known system "Pfinder". Perez et al. [17] apply a color-based particle filter to the person tracking problem and score the particles by comparing a histogram-based model of the target object with histograms of candidate regions. Unfortunately, the appearance of color in real-world scenarios is fragile because of different light sources, shadowing, and – specific for our lecture scenario – by the bright and colorful beam of the video projector that often overlays the lecturer. Furthermore, neither the skin color nor the color of clothing is generally specific for the lecturer when compared to people from the audience.

Mikic et al. [15] rely on background subtraction as their primary feature, thus their approach does not suffer from changing color appearance. Their sensor setup is similar to ours: four fixed calibrated cameras with highly overlapping fields of view. The main difference is the way in which the 3D location is acquired: Mikic et al. combine foreground region centroids from different views using triangulation, and exclude mismatches with high residuals. However, this approach suffers from over- or under-segmentation as an effect of noisy foreground classification. Instead of triangulating connected foreground segments, our algorithm performs sampled projections of 3D hypotheses, as proposed by Zotkin et al. [27], and gathers support for the respective sample in the resulting image region in each view. It is thus less dependent on the quality of the segmentation.

Face-detection cascades as proposed by Viola and Jones [23] are known to be both robust and fast, which makes them a good feature to support a person tracker. However, searching high-resolution camera images exhaustively for faces in multiple scales still goes beyond the possibilities of real-time operation. This is why we propose to apply the detectors only at the particles' locations.

We use a particle filter framework [10] to guide the evaluation of the sensor streams and the generation of the tracking hypothesis. Particle filters have been shown to be applicable successfully to audio-visual tracking for example by [22] for a video telephony application, by [4] for multi-person tracking or by [7] for multi-party conversation in a meeting situation.

2. AN AUDIO-VISUAL PARTICLE FILTER

Particle filters [10] represent a generally unknown probability density function by a set of m random samples $s_{1..m}$. Each of these particles is a vector in state space and is associated with an individual weight. The evolution of the particle set is a two-stage process which is guided by the observation and the motion model:

1. The prediction step: From the set of particles from the previous time instance, an equal number of new particles is generated. In order to generate a new particle, a particle of the old set is selected randomly in consideration of its weight, and then propagated by applying the motion model. In the simplest case, this can be additive Gaussian noise, but higher order motion models can also be used.

2. The measurement step: In this step, the weights of the new particles are adjusted with respect to the current observation z_t: $\pi_i = p(z_t|s_i)$. This means computing the probability of the observation given that the state of particle s_i is the true state of the system.

As we want to track the lecturer's head centroid, each particle $s_i = (x, y, z)$ represents a coordinate in space. The ground plane is spanned by x and y, the height is represented by z. The particles are propagated by simple Gaussian diffusion, thus representing a coarse motion model:

$$s_{i,t} = s_{i,t-1} \cdot (N_{\sigma=0.2m}, N_{\sigma=0.2m}, N_{\sigma=0.1m}) \qquad (1)$$

Using the features described in Sections 4 and 5, we can calculate a weight for each particle by combining the normalized probabilities of the visual observation V_t and the acoustical observation A_t using an adaptive weighting factor α:

$$\pi_i = \alpha \cdot p(A_t|s_i) + (1 - \alpha) \cdot p(V_t|s_i) \qquad (2)$$

The weighting factor α was adjusted dynamically according to an acoustic confidence measure that is described in section 5. The average value of α was approximately 0.4. Therefore, more weight was given to the video features. A particle's weight is set to 0 if the particle leaves the lecture room[1] or if its z-coordinate leaves the valid range for a standing person ($1.2m < z < 2.1m$). The weighted mean of the particle set is the final hypothesis about the lecturer's location.

3. SAMPLED PROJECTION INSTEAD OF TRIANGULATION

A common way [15, 6] to obtain the 3D position of an object from multiple views is to locate the object in each

[1]We restrict the particles to be within the full width of the room's ground plane ($0 < y < 7.1m$) and half of the depth ($0 < x < 3m$).

of the views and then to calculate the 3D position by using triangulation, i.e. by searching for the intersection of the lines-of-view (LOVs) from the calibrated cameras to the object. However, this approach has several weak points: first of all, the object has to be detected in at least two different views at the same time. Second, the quality of triangulation depends on the points of the object's images that are chosen as starting points for the LOVs: if they do not represent the same point of the physical object, there will be a high triangulation error. Furthermore, searching for the object in each of the views separately – without incorporating geometry information – results in an unnecessarily large search space. It would be more efficient to only evaluate those image coordinates in each of the views that make up a valid 3D location when combined with their respective partners in the other views.

In the proposed method, we followed an approach also taken by [27] and avoid the problems mentioned above by not using triangulation at all. Instead, we make use of the particle filter's property to predict the object's location as a well-distributed set of hypotheses: many particles cluster around likely object locations, and less particles populate the space in between. As the particle set represent a probability distribution of the predicted object's location, we can use it to narrow down the search space. So instead of searching a neighborhood exhaustively, we only look for the object at the particles' positions.

In order to evaluate a particle, we both use video and audio features. For video, we project the represented 3D object location to the image planes and look for evidence of the object at that point, using the foreground and detection features described in Section 4. For audio, we calculate the theoretical time delay of arrival for the hypothesized location and score it using the generalized cross correlation function as described in Section 5. The magnitude of the joint evidence from all views and microphone pairs is then used to adjust the respective particle's weight in the measurement step.

When comparing the proposed method to Kalman filter-based tracking, the following advantage becomes apparent: As a particle filter is capable of modeling multi-modal distributions, the true distribution of measurement likelihoods is considered at each step. This way, no information is lost by suppressing all but the strongest measurement, as it is the case for Kalman filter's update step. Furthermore, no data-association problem occurs as it would be the case when trying to match object candidates from different views in order to do explicit triangulation. The latter problem, however, could also be avoided for Kalman filter tracking using incremental updates and a linearization of the camera projection function as shown in [8].

4. VIDEO FEATURES

For the task of person tracking in video sequences, there is a variety of features to choose from. In our lecture scenario, the problem comprises both locating the lecturer and disambiguating the lecturer from the people in the audience. As lecturer and audience cannot be separated reliably by means of fixed spatial constraints as, e.g., a dedicated speaker area, we have to look for features that are more specific for the lecturer than for the audience.

Intuitively, the lecturer is the person that is standing and moving (walking, gesticulating) most, while people from the audience are generally sitting and moving less. In order to exploit this specific behavior, we use dynamic foreground segmentation based on adaptive background modeling as primary feature, as described in Section 4.1. In order to support the track indicated by foreground segments, we use detectors for face and upper body (see Section 4.2). Both features – foreground F and detectors D – are linearly combined[2] using a mixing weight β. So the probability of the visual information V_t^j in view j, given that the true state of the system is characterized by s_i, is set to be

$$\overline{p}(V_t^j|s_i) = \beta \cdot p(D_t^j|s_i) + (1-\beta) \cdot p(F_t^j|s_i) \quad (3)$$

By means of the sum rule, we integrate the weights from the v different views in order to obtain the total probability of the visual observation:

$$\overline{p}(V_t|s_i) = \frac{1}{v} \sum_{j=1..v} \overline{p}(V_t^j|s_i) \quad (4)$$

To obtain the desired (pseudo) probability value which tells us how likely this particle corresponds to the visual observation we have to normalize over all particles:

$$p(V_t|s_i) = \frac{\overline{p}(V_t|s_i)}{\sum_i \overline{p}(V_t|s_i)} \quad (5)$$

Figure 1: Foreground segmentation is performed by means of an adaptive background model.

4.1 Foreground Segmentation

In order to segment the lecturer from the background, we build an adaptive background model using the algorithm proposed by [21]. For each pixel, a Gaussian mixture model represents the pixel's appearance. If the observed value differs significantly from the learned distribution, the respective pixel is assigned to the foreground. The background model is continuously updated, so that a permanent change in a pixel's appearance is anticipated by the background model after a while. This ensures that people from the audience as well as moved objects become part of the back-

[2]Note that $p(D_t^j|s_i)$ and $p(F_t^j|s_i)$ respectively have to be normalized before combination so that they sum up to 1.

ground, whereas the frequently moving speaker remains in the foreground.

However, as Fig. 1 shows, the resulting segmentation of a crowded lecture room is far from perfect. Morphological filtering of the foreground map is generally not sufficient to remove the noise and to create a single connected component for the lecturer's silhouette. Nonetheless, the combination of the foreground maps from different views contains enough information to locate the speaker. Thus, our approach gathers support from all the views' maps without making any "hard" decisions like selecting a connected component or a cluster centroid.

As described in Section 2, the particle filter framework merely requires us to assign scores to a number of hypothesized head positions. In order to evaluate a hypothesis $s_i = (x, y, z)$, we project a person-sized cuboid $(x \pm 0.3m, y \pm 0.3m, 0..z+0.15m)$ centered around the head position to the image plane of each camera view, and count the number of foreground pixels inside the projected polygon. The fraction of foreground pixels within the polygon is then used as the particle's score.

As this calculation has to be done for each of the particles in all views, we use the following simplification in order to speed up the procedure: all cameras are set upright with respect to the ground plane, so the projection of the cuboid can be approximated by a rectangle orthogonal to the image plane, i.e. the bounding box of the projected polygon (see Fig. 2).

The sum of pixels inside the bounding box can be computed efficiently using the integral image introduced by [23]. Given the foreground map $m(x, y)$, the integral image $ii(x, y)$ contains the sum of the pixels above and to the left of (x, y):

$$ii(x, y) = \sum_{y'=0}^{y} \sum_{x'=0}^{x} m(x', y') \tag{6}$$

Thus, the sum of the rectangle (x_1, y_1, x_2, y_2) can be determined by four lookups in the integral image. So the particle score for the foreground feature is defined by the sum of pixels inside the bounding box normalized by the size of the bounding box:

$$p(F_t^j | s_i) = \frac{ii(x_2, y_2) - ii(x_1, y_2) - ii(x_2, y_1) + ii(x_1, y_1)}{(x_2 - x_1 + 1)(y_2 - y_1 + 1)} \tag{7}$$

Using the recurrent formulation from [23], the generation of the integral image only takes one pass over the foreground map, so the complexity of the foreground feature preparation is linear to the image size. The evaluation of one particle can then be done in constant time, and is thus independent of the image resolution and the projected size of the person tracked. Note that this is not limited to the foreground segmentation: in general, any kind of support map (e.g. histogram backprojection) could be used to contribute to the tracking at this stage.

4.2 Face and Upper Body Detection

As we aim at tracking the coordinates of the lecturer's head – serving as model point for the full body –, we need a feature that gives evidence for the head position. The face detection algorithm proposed by Viola and Jones [23] is known to be both robust and fast: it uses Haar-like features that can be efficiently computed by means of the integral image, thus being invariant to scale variations. The features

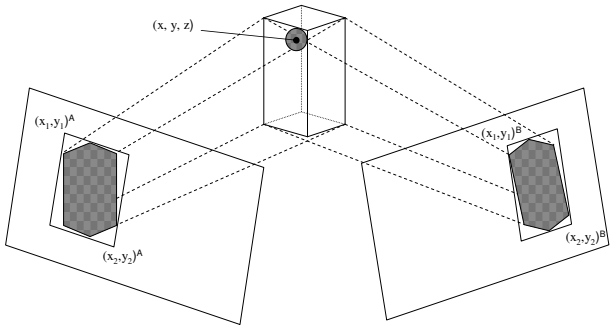

Figure 2: For each particle, a cuboid centered around the hypothesized head position (x, y, z) is projected into the views A and B. The resulting polygon is approximated by a bounding box $(x_1, y_1, x_2, y_2)^{A/B}$.

are organized in a cascade of weak classifiers, that is used to classify the content of a search window as being face or not.

Typically, a variable-size search window is repeatedly shifted over the image, and overlapping detections are combined to a single detection. Exhaustively searching a $W \times W$ image region for a $F \times F$ sized face while incrementing the face size n times by the scale factor s requires the following number of cascade runs (not yet taking into account post-filtering of overlapping detections):

$$\#cascade\ runs = \sum_{i=0}^{n-1} \left(W - F \cdot s^i\right)^2 \tag{8}$$

In case of for example a 100x100 pixel image region, and a face size in between 20 and 42 ($n = 8, s = 1.1$), this results in 44368 cascade runs.

In the proposed particle filter framework however, it is not necessary to scan the image exhaustively: the places to search are directly given by the particle set. For each particle, a head-sized cuboid (30cm edge length) centered around the hypothesized head position is projected to the image plane, and the bounding box of the projection defines the search window that is to be classified. Thus, the evaluation of a particle takes only one run of the cascade:

$$\#cascade\ runs = \#particles \tag{9}$$

The face detector is able to locate the vertical and horizontal position of the face precisely with respect to the image plane. However, the distance to the camera, i.e. the scaling, cannot be estimated accurately from a single view. In order to achieve tolerance against scale variation and to smooth the scores of nearby particles, we set the i-th particle's score to the (average) overlap[3] between the particle's head rectangle $r_i = (x_1, y_1, x_2, y_2)$ and all the positively classified head rectangles $r'_{0..N}$ by any of the other particles:

$$p(D_t^j | s_i) = \frac{1}{N} \sum_{n=0}^{N} overlap(r_i, r'_n) \tag{10}$$

A detector that is trained on frontal faces only is unlikely to produce many hits in our multi-view scenario. In order

[3]The auxiliary function $overlap(a, b)$ calculates the ratio of the shared area of two rectangles a and b to the sum of the areas of a and b.

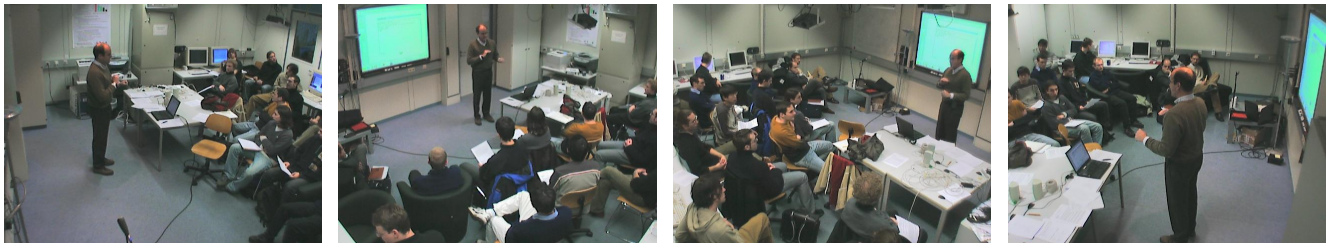

Figure 3: Snapshot from a lecture showing all 4 camera views. The native resolution is 640x480.

to improve the performance, we used two cascades for face detection: one for frontal faces in the range of $\pm 45°$ and one for profile faces $(45° - 90°)$[4]. Our implementation of the face detector is based on the OpenCV library, that implements an extended set of Haar-like features as proposed by [13]. This library also includes a pre-trained classifier cascade for upper body detection [12]. We used this detector in addition to face detection, and incorporated it's results using the same methods as described for face detection.

5. AUDIO FEATURES

The lecturer is the person that is normally speaking, therefore we can use audio features using multiple microphones to detect the speaker position. Consider the j-th pair of microphones, and let $\mathbf{m}_{j1}$ and $\mathbf{m}_{j2}$ respectively be the positions of the first and second microphones in the pair. Let $\mathbf{x}$ denote the position of the speaker in a three dimensional space. Then the *time delay of arrival* (TDOA) between the two microphones of the pair can be expressed as

$$T_j(\mathbf{x}) = T(\mathbf{m}_{j1}, \mathbf{m}_{j2}, \mathbf{x}) = \frac{\|\mathbf{x} - \mathbf{m}_{j1}\| - \|\mathbf{x} - \mathbf{m}_{j2}\|}{c} \quad (11)$$

where c is the speed of sound. To estimate the TDOAs a variety of well-known techniques [16, 5] exist. Perhaps the most popular method is the *phase transform* (PHAT), which can be expressed as

$$R_{12}(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{X_1(e^{j\omega\tau}) X_2^*(e^{j\omega\tau})}{|X_1(e^{j\omega\tau}) X_2^*(e^{j\omega\tau})|} e^{j\omega\tau} \, d\omega \quad (12)$$

where $X_1(\omega)$ and $X_2(\omega)$ are the Fourier transforms of the signals of a microphone pair in a microphone array. Normally one would search for the highest peak in the resulting cross correlation to estimate the position. But since we are using a particle filter, as described in Section 2, we can simply set the PHAT value at the time delay position $T_j(\mathbf{x} = \mathbf{s}_i)$ of the MA pair j of a particular particle s_i as

$$\overline{p}(A_t^j | \mathbf{s}_i) = max(0, R_j(T_j(\mathbf{x} = \mathbf{s}_i))) \quad (13)$$

As the values returned by the PHAT can be negative, but probability density functions must be strictly nonnegative, we found that setting all negative values of the PHAT to zero yielded the best results.

To get a better estimate we repeat this over all m pair of microphones (in our case 12), sum their values and normalize by m:

$$\overline{p}(A_t | \mathbf{s}_i) = \frac{1}{m} \sum_{j=1}^{m} \overline{p}(A_t^j | \mathbf{s}_i) \quad (14)$$

[4]The profile face cascade has to be applied twice: to the original image and to a horizontally flipped image.

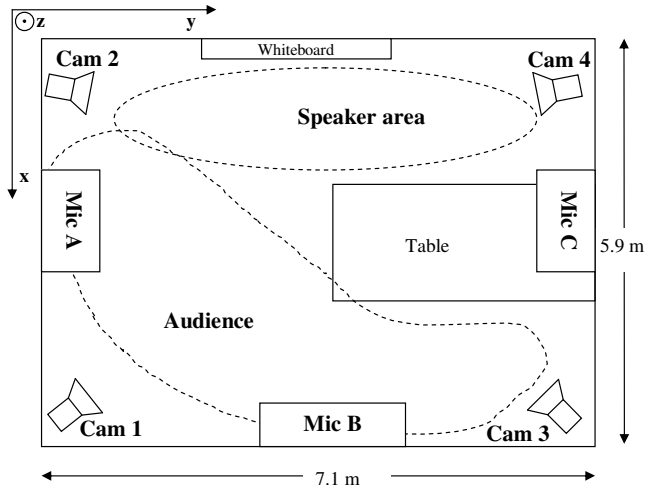

Figure 4: The lecture room is equipped with four fixed cameras and three 4-channel microphone arrays. Lecturer and audience typically reside in the depicted areas, but are not limited to these areas.

Just like for the visual features, we normalize over all particles in order to get the acoustic observation likelihood for each particle:

$$p(A_t | \mathbf{s}_i) = \frac{\overline{p}(A_t | \mathbf{s}_i)}{\sum_i \overline{p}(A_t | \mathbf{s}_i)} \quad (15)$$

The weighting factor α used in equation 2 was set by

$$\alpha = \frac{m_0}{m} \cdot 0.6 \quad (16)$$

where m is the total number of microphone pairs and m_0 the number of PHAT values above 0. The maximum weighting factor of 0.6 for the audio features has been determined experimentally.

6. EXPERIMENTS AND RESULTS

In order to evaluate the performance of the proposed algorithm, we ran experiments on recordings of real lectures that were held at the University of Karlsruhe by students, visitors and members of the lab.

6.1 Lecture Dataset

The dataset for the evaluation consists of a total number of 5 recordings, each featuring a different speaker. The length of a recording is typically 45min. The lectures are complemented by slides which are projected to a whiteboard

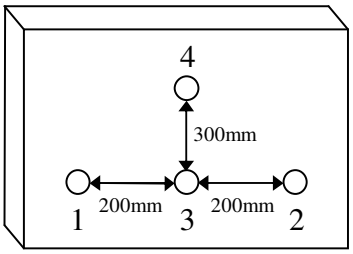

Figure 5: Each array is made up of four microphones in a T-shaped setup.

next to the speaker. Apart from the lecturer, there is a number of about 5-20 people in the audience. As can be seen in Fig. 4, there is no clear separation between speaker area and audience. It must further be noted that every once in a while, auditors cross the speaker area in order to enter or to leave the room.

For our experiments, we used 4 fixed cameras that are placed at a height of 2.7m in the room corners. Their joint field of view covers almost the entire room. The images are captured at a resolution of 640x480 pixels and a framerate of 15 fps, and stored as jpg-files for offline processing. To capture the lecturer's speech, we used 3 T-shaped microphone arrays that are placed on the walls. The wall behind the lecturer does not carry microphones, as they would hardly capture direct sound from the lecturer. As depicted in Figure 5 each array consists of 4 microphones: 3 at the bottom of the box with a inner distance of 20 cm and one 30 cm above the center microphone. The microphones were all recorded synchronously using an RME Hammerfall sound card at a samplerate of 44.1 kHz and a resolution of 24 bit.

From each of the 5 lectures two segments (each approximately 15min) were extracted and used for evaluation. In all of the resulting 10 segments and each of the 4 views, the lecturer's head centroid was labeled manually every 10th frame. By means of the calibration information, a 3D label file was generated and serves as ground truth for the evaluation. A separate training dataset has been used to tune tracking parameters and to train the face detection cascades. This training set consists of different lectures that were collected using the same setup as for the evaluation set.

6.2 Experiments

In a preliminary experiment, the ideal number of particles - being an important parameter both for performance and speed - was to be determined on the training set. As Fig. 6 shows, the tracking performance improves up to a particle number of about 300. Beyond that point, the system appears to be saturated. Thus, a number of 300 particles has been used for the actual evaluation.

The actual evaluation results presented in Table 1 are average values of all 10 lecture segments, i.e. 127min of lecture data and 5 different speakers. The error measure is the average Euclidean distance between the hypothesized head coordinates and the labeled ones.

It can be seen that even though the video-only tracker performs considerably better than the audio-only tracker, the performance can still be significantly increased by combining both modalities. This effect is particularly distinctive during one recording in which the lecturer is standing most

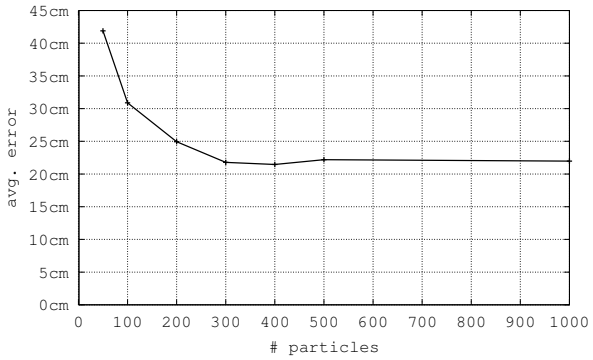

Figure 6: Tracking performance in relation to the number of particles. This experiment has been conducted on a separate training dataset.

Tracking mode	Avg. error
Video only	36.3cm
Audio only	46.1cm
Video + Audio	30.5cm

Table 1: Error in 3D head position averaged over 2 hours of actual lectures.

of the time in one dark corner of the room, thus being hard to find using solely video features (116cm mean error). In combination with audio, the error could well be reduced.

6.3 Implementation and Complexity

On the video side, the proposed algorithm consists of two parts that can be characterized by their relation to the three factors that determine the runtime of the algorithm. The feature preparation part (foreground segmentation, integral image calculation) is related linearly to the image size S and a constant time factor t_{VS}. In contrast, the particle evaluation part is independent from S and related linearly to the number of particles P and a constant t_{VP}. Both parts are likewise related linearly to the number of views V. On the audio side, the runtime is linearly related to the number of microphone pairs M. Like in the video case, this can be further decomposed into a constant preprocessing part t_{AM} and a part t_{AP} that has to be repeated for each particle. Thus, the total processing time per frame is determined by:

$$t_{total} = (t_{VS} \cdot S + t_{VP} \cdot P) \cdot V + (t_{AM} + t_{AP} \cdot P) \cdot M \quad (17)$$

On a standard desktop PC (3.0GHz) we measured the following values for the constants in our implementation:

$$\begin{aligned} t_{VS} &= 4.63 \cdot 10^{-5}\ ms \\ t_{VP} &= 1.01 \cdot 10^{-1}\ ms \\ t_{AM} &= 1.25 \cdot 10^{1}\ ms \\ t_{AP} &= 1.06 \cdot 10^{-1}\ ms \end{aligned}$$

Given the proposed number of particles $P = 300$, the native camera resolution[5] $S = 640 \cdot 480$, and the $V = 4$ cameras from our setup, the processing time for a video frame was 178 ms. Given a number of $M = 12$ microphone pairs, the

[5]The full resolution was only used for the detectors, while the foreground segmentation runs at 4 times lower scale.

time to process the audio signal corresponding to one video frame was 532 ms. It has to be noted, however, that our implementation is straightforward and could certainly well be improved for speed – especially in the audio part.

As Equation 17 indicates, the visual part can be intuitively parallelized for the number of views V. We implemented such a video-only tracker using 4 desktop PCs, each connected to a camera, in a way that the image processing is done locally on each machine. Because only low-bandwidth data (particle positions and weights) are shared over the network, the overhead is negligible, and a speed of 11 fps (including image acquisition) could be realized.

7. CONCLUSION

We presented an algorithm for tracking a person in real-time using multiple cameras and multiple pairs of microphones. The core of the proposed algorithm is a particle filter that works without explicit triangulation. Instead, it estimates the 3D location by sampled projection, thus benefiting from each single view and microphone pair. The video features used for tracking are based on adaptive foreground segmentation and the response of detectors for upper body, frontal face and profile face. The audio features are based on the time delays of arrival between pairs of microphones, and are estimated with a generalized cross correlation function.

The audio-visual tracking algorithm was evaluated on recordings of actual lectures and yielded a performance of $30.5cm$ average error in localizing the lecturer's head. Thus, the combined tracker clearly outperforms both the audio- and video-only tracker. One reason for this is that the video and audio features described in this paper complement one another well: the comparatively coarse foreground feature along with the audio feature guide the way for the face detector, which in turn gives very precise results as long as it searches around the true head position. Another reason for the benefit of the combination is that neither motion and face detection nor acoustic source localization responds exclusively to the lecturer and not to people from the audience – so the combination of both increases the chance of actually tracking the lecturer.

Acknowledgments

Thanks to Hazim K. Ekenel for providing the face detection cascades. This work has partially been funded by the European Commission under Project CHIL (http://chil.server.de, contract #506909).

8. REFERENCES

[1] M. S. Brandstein. *A framework for speech source localization using sensor arrays*. PhD thesis, Brown University, Providence, RI, May 1995.

[2] M. S. Brandstein, J. E. Adcock, and H. F. Silverman. A closed-form location estimator for use with room environment microphone arrays. *IEEE Trans. Speech Audio Proc.*, 5(1):45–50, January 1997.

[3] Y. T. Chan and K. C. Ho. A simple and efficient estimator for hyperbolic location. *IEEE Trans. Signal Proc.*, 42(8):1905–15, August 1994.

[4] N. Checka, K. Wilson, V. Rangarajan, and T. Darrell. A probabilistic framework for multi-modal multi-person tracking. In *IEEE Workshop on Multi-Object Tracking (in conjunction with CVPR)*, 2003.

[5] J. Chen, J. Benesty, and Y. A. Huang. Robust time delay estimation exploiting redundancy among multiple microphones. *IEEE Trans. Speech Audio Proc.*, 11(6):549–57, November 2003.

[6] D. Focken and R. Stiefelhagen. Towards vision-based 3-D people tracking in a smart room. In *IEEE Int. Conf. Multimodal Interfaces*, October 2002.

[7] D. Gatica-Perez, G. Lathoud, I. McCowan, and J.-M. Odobez. A mixed-state i-particle filter for multi-camera speaker tracking. In *Proc. IEEE ICCV Workshop on Multimedia Technologies in E-Learning and Collaboration (ICCV-WOMTEC)*, 2003.

[8] T. Gehrig, K. Nickel, H. K. Ekenel, U. Klee, and J. McDonough. Kalman filters for audio-video source localization. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, to appear Oct. 2005.

[9] Yiteng Huang, Jacob Benesty, Gary W. Elko, and Russell M. Mersereau. Real-time passive source localization: A practical linear-correction least-squares approach. *IEEE Trans. Speech Audio Proc.*, 9(8):943–956, November 2001.

[10] M. Isard and A. Blake. Condensation–conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[11] U. Klee, T. Gehrig, and J. McDonough. Kalman filters for time delay of arrival-based source localization. *EURASIP Special Issue on Multichannel Speech Processing*, submitted for publication.

[12] H. Kruppa, M. Castrillon-Santana, and B. Schiele. Fast and robust face finding via local context. In *IEEE Intl. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, October 2003.

[13] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *ICIP*, volume 1, pages 900–903, September 2002.

[14] Q. Liu, Y. Rui, A. Gupta, , and J.J. Cadiz. Automatic camera management for lecture room environments. In *Proc of SIGCHI'01*, pages 442–449, 2001.

[15] I. Mikic, S. Santini, and R. Jain. Tracking objects in 3d using multiple camera views. In *ACCV*, 2000.

[16] M. Omologo and P. Svaizer. Acoustic event localization using a crosspower-spectrum phase based technique. *Proc. ICASSP*, II:273–6, 1994.

[17] P. Perez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In *Proc. of ECCV 2002*, pages 661–675, 2002.

[18] G. Potamianos, C. Neti, and S. Deligne. Joint audio-visual speech processing for recognition and enhancement. In *Proc. Work. Audio-Visual Speech Processing*, pages 95–104, September 2003.

[19] H. C. Schau and A. Z. Robinson. Passive source localization employing intersecting spherical surfaces from time-of-arrival differences. *IEEE Trans. Acoust. Speech Signal Proc.*, ASSP-35(8):1223–5, August 1987.

[20] J. O. Smith and J. S. Abel. Closed-form least-squares source location estimation from range-difference measurements. *IEEE Trans. Acoust. Speech Signal Proc.*, ASSP-35(12):1661–9, December 1987.

[21] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR*, pages 246–252, 1999.

[22] J. Vermaak, M. Gangnet, A. Blake, and P. Prez. Sequential monte carlo fusion of sound and vision for speaker tracking. In *Proc. IEEE Intl. Conf. on Computer Vision*, volume 1, pages 741–746, 2001.

[23] P. Viola and M. Jones. Robust real-time object detection. In *ICCV Workshop on Statistical and Computation Theories of Vision*, July 2001.

[24] D. B. Ward, E. A. Lehmann, and R. C. Williamson. Particle filtering algorithms for tracking an acoustic source in a reverberant environment. *IEEE Trans. Speech Audio Proc.*, 11(6):826–836, 2003.

[25] M. Wölfel and J. McDonough. Combining multi-source far distance speech recognition strategies: beamforming, blind channel and confusion network combination. In *Interspeech*, to appear Sept. 2005.

[26] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 19(7), 1997.

[27] D. Zotkin, R. Duraiswami, and L. Davis. Joint audio-visual tracking using particle filters. *EURASIP journal on Applied Signal Processing*, 2002(11), 2002.