MULTIMODAL 3-D TRACKING AND EVENT DETECTION VIA THE PARTICLE FILTER

Dmitry Zotkin, Ramani Duraiswami, Larry S. Davis

Perceptual Interfaces and Reality Laboratory, UMIACS
University of Maryland, College Park, MD 20742
{dz,ramani,lsd}@cs.umd.edu

ABSTRACT

Determining the occurrence of an event is fundamental to developing systems that can observe and react to them. Often, this determination is based on collecting video and/or audio data and determining the state or location of a tracked object. We use Bayesian inference and the particle filter for tracking moving objects, using both video data obtained from multiple cameras and audio data obtained using arrays of microphones. The algorithms developed are applied to determining events arising in two fields of application. In the first, the behavior of a flying echolocating bat as it approaches a moving prey is studied, and the events of search, approach and capture are detected. In a second application we describe detection of turn-taking in a conversation between possibly moving participants recorded using a smart video conferencing setup.

1. INTRODUCTION

An event is characterized by some typical change in the state of some object. Robust detection of events thus requires robust tracking of an object's state. Typically this state includes the object's position, either in an absolute frame, or relative to some other object. Further, to detect an event change the detecting system must focus its attention on the object location (e.g., the position of a human) at a given time. Systems that seek to recognize events in applications such as surveillance, creating perceptually immersive realities, or HCI, must thus be able to focus on particular object locations in order to obtain a better view of the actions taking place. This focusing can involve zoom and focus of an active camera, enhanced audio from the spot obtained via a microphone array beamforming procedure, or some other attention focusing mechanism. All these require robust tracking of the position of an object. The tracking algorithm might require *a priori* knowledge of the nature of the actions that are of interest, and it would be desirable to be able to incorporate data from any available active sensors.

We develop a multimodal sensor fusion framework based on particle filters and apply it to tracking and event detection using audio and video modalities. We show that the performance of the multimodal tracker is superior to that of unimodal tracking, and that availability of information from a complementary modality simplifies the event detection task.

The developed algorithm is an application of sequential Monte-Carlo methods (also known as particle filters) to 3-D tracking using two calibrated cameras and a microphone array. Particle filters were introduced to the vision community in the form of the CONDENSATION algorithm [1]. Improvements of a technical nature to the condensation algorithm were provided by Isard and Blake [2] (importance sampling), MacCormick and Blake [5], Li and Chellappa [11], and Philomin et al [10]. The algorithm has seen application to tracking people in video, and face tracking.

The reason these algorithms have attracted much interest is that they offer a framework for dynamic state estimation where the underlying probability density functions (pdfs) need not be Gaussian, and state and measurement equations can be nonlinear – situations that are commonly encountered in vision. The method is relatively robust to noise, and recovers from tracking misses in intermediate frames. In addition, they are relatively simple to implement, and allow one to conveniently combine multiple feature types in the same tracker.

This paper is arranged as follows. In section 2 the notation and the basic equations for the video tracker, the audio tracker, and the particle filter are introduced. In section 3 we introduce two event detection problems for which the multimodal action recording setup is available (a flying bat in a dark room and multiple speakers in an office environment). In section 4, we study the performance of our tracking algorithm on Monte Carlo simulations and present results of tracking and event detection for real and simulated data in both bat and videoconferencing experiments. Section 5 concludes the paper with an assessment of the algorithm and a discussion of future work needed to achieve better performance on the tracking problem.

2. FORMULATION

We describe the particle filter in general terms first. Then the motion model and the posterior probability distributions

that are used in particle filter are described.

2.1. The particle filter

For an accessible discussion of the details of the particle filter and its deficiencies see Forsyth & Ponce [3]. The particle filter represents the underlying pdf that describes the state of the object by a set of random samples from the space on which the pdf is defined. Every sample is commonly referred to as a *particle*. Associated with each particle is a *weight*. The particle locations and weights are used to achieve "Monte-Carlo" approximations to integrals involving the unknown pdf that is being determined.

The sketch of a particle filter update algorithm is as follows. Initially, all particles have equivalent weight attached to them. To progress to the next time instance, two steps are performed in sequence. First, at the prediction step, the state of every particle is updated according to the motion model. An accurate dynamical model is essential for robust tracking and for achieving real-time performance. Next, during the measurement step, new information that became available about the system is used to adjust the particle weights for every particle. The weight is set to be the likelihood of this particle state describing the true current state of the object, which can be computed via Bayesian inference to be proportional to the probability of the observed measurements given the particle state (assuming all object states are equiprobable). The sample points are then redistributed to obtain uniform weighting for the next algorithm iteration by resampling them from the computed posterior probability distribution. No explicit integral characteristics of the process is kept by the algorithm; however, at any time such characteristics (position, speed etc.) can be directly computed, if desired, by using the particle set and weights as an approximation to the true pdf.

The algorithm works well in many cases where the Kalman filter (or the extended Kalman filter) would fail due to poor approximation of the process pdf by the Gaussian (for example, when the pdf is multimodal). An advantage of the formulation is that it can be easily applied even when the state update model and the measurement model are non-linear since they are only evaluated in the forward direction, and need not be inverted. This allows the use of error functions that make "sense" for the problem, including nonlinear ones [8].

Since it is not necessary to invert measurement equations to perform tracking, it is possible to perform seamless integration of multiple data streams and multiple modalities. To exploit this possibility, our tracker uses video data from multiple cameras to obtain 3D coordinates of the object without explicit triangulation. In addition, another modality (audio) is used to perform joint tracking, with the audio data being provided by the microphone array. The formulation we are developing thus accepts multiple streams of infor-

mation with entirely different noise probability densities. While the assumption of Gaussian noise in the video data and cross-correlation peaks does not cause significant difficulty, the fact that both the audio and video data are subject to the presence of substantial outliers does. These outliers arise due to spurious cross correlation peaks in audio and due to missed or incorrect correspondences in video. In addition, the times at which the data are available from each modality may be different. Our multi-modal particle filter addresses these problems. The filter formulation including the motion model, measurement equation and the posterior probability update equation are given below.

2.2. State vector and observation vector

We use a first-order motion model to effectively learn the motion of the tracked object. The six-dimensional state vector of a particle is composed of its coordinates and velocities and will be denoted as $[x_t \ \dot{x}_t]^T$. The observation vector consists of available measurements, which include audio and video data. For N cameras in the system, the 2D image coordinates of P corresponding points on the object in all these views contribute $2NP$ elements to the observation vector. The microphone array consisting of M microphones provide additional $C(M, 2)$ elements which are the peak positions in all possible cross-correlations between pairs of microphones. Thus, the observation vector has total of $2NP + C(M, 2)$ elements. In our setup, $N = 2$, $P = 1$, $M = 7$ and $2NP + C(M, 2) = 25$.

2.3. Motion Model

We use a simple first-order rigid-body motion model with a random excitation force applied to the particles. If $x(t)$ is a 3-D vector of source coordinates at time t and $\dot{x}(t)$ is a vector of corresponding velocities, then the motion model can be written as

$$x(t + \delta t) = x(t) + \dot{x}(t)\delta t, \quad \dot{x}(t + \delta t) = \dot{x}(t) + F\delta t \tag{1}$$

where F is the random partition excitation acceleration – a normally distributed random variable with zero mean and standard deviation $\sigma = 10^2 \text{m/s}^2$. The initial distribution of particles is chosen to be Gaussian with $[x_d \ \mathbf{0}]^T$ mean and $\sigma = 0.2$, where x_d is the initial object coordinate vector obtained by a suitable detection process. In the absence of a detection process a uniform distribution of particles is used.

2.4. Video tracking formulation

We are given a pair of widely-spaced calibrated cameras that can view the area under consideration. For determining the 3-D position of points from approximate correspondences, the simplest algorithm to use is a classical one described in Slama [6], that is also extensively used in the gait-analysis and motion capture communities. An improved version of this algorithm is discussed in Chapter 11.2 of

[8] and has the advantage that its generalization to multiple views is straightforward – something that we intend to do in the future.
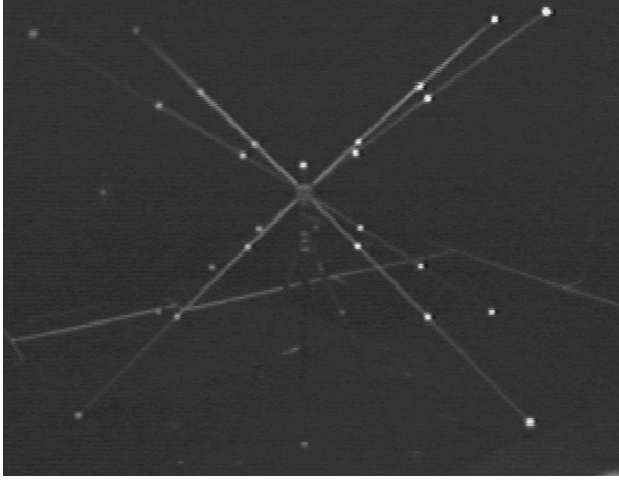

Figure 1: The calibration object.

A large calibration object is used to perform camera calibration; it is shown in Figure 1. The object consists of 25 white balls on black metal sticks. The 3-D coordinates of the balls are known. Using these coordinates (x_n, y_n, z_n), $n = 1 \ldots 25$ and ball locations on the two images (u_{mn}, v_{mn}), $m = 1, 2$, we can calibrate the camera using Direct Linear Transformation equations:

$$u_{mn} = \frac{A_m x_n + B_m y_n + C_m z_n + D_m}{E_m x_n + F_m y_n + G_m z_n + 1} \qquad (2)$$

$$v_{mn} = \frac{H_m x_n + J_m y_n + K_m z_n + L_m}{E_m x_n + F_m y_n + G_m z_n + 1} \qquad (3)$$

This is the expression for the general perspective projection. Using the 50 equations given by the correspondences, one can determine the 11 parameters for each camera $(A_m, ..., L_m)$, via least squares. Knowing the camera parameters, and given a possible coordinate pair of measurements for a (u_1, v_1) and (u_2, v_2), we can write equations (2, 3) in terms of the unknown coordinates $[x\ y\ z]$:

$$\varepsilon \overline{\sigma} = \begin{bmatrix} A_1 - E_1 u_1 & B_1 - F_1 u_1 & C_1 - G_1 u_1 \\ H_1 - E_1 v_1 & J_1 - F_1 v_1 & K_1 - G_1 v_1 \\ A_2 - E_2 u_2 & B_2 - F_2 u_2 & C_2 - G_2 u_2 \\ H_2 - E_2 v_2 & J_2 - F_2 v_2 & K_2 - G_2 v_2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} D_1 - u_1 \\ L_1 - v_1 \\ D_2 - u_2 \\ L_2 - v_2 \end{bmatrix} \qquad (4)$$

This system can, in principle, be solved using least squares to obtain the 3-D position of a point whose correspondences are known. The particle filter, however, never directly solves this inverse problem for the 3-D coordinates of the object. Instead, as described before, the posterior probability for all particles is computed using the forward

problem solution (2, 3). If the state vector for a given particle is $X = [x\ y\ z\ \dot{x}\ \dot{y}\ \dot{z}]$ and the video observation vector is $Z_v = [u_1\ v_1\ u_2\ v_2\ ...\ u_N\ v_N]$, then, first, the projected image coordinates are obtained by

$$\hat{u}_i = \frac{A_i x + B_i y + C_i z + D_i}{E_i x + F_i y + G_i z + 1},\ i = 1...N, \qquad (5)$$

$$\hat{v}_i = \frac{H_i x + J_i y + K_i z + L_i}{E_i x + F_i y + G_i z + 1},\ i = 1...N, \qquad (6)$$

and the projection error is computed as

$$\epsilon_v^2 = \frac{1}{N} \sum_{i=1}^{N} \left[(\hat{u}_i - u_i)^2 + (\hat{v}_i - v_i)^2 \right]. \qquad (7)$$

The posterior probability $p_v(Z_v|X)$ of the observation Z_v given the state vector X is

$$p_v(Z_v|X) = \frac{1}{\sqrt{2\pi}\sigma_v} \exp\left(-\frac{1}{2}\frac{\epsilon_v^2}{\sigma_v^2}\right). \qquad (8)$$

2.5. Audio tracking formulation

Triangulation is known to be prone to errors, especially with the wide baseline that is used here, which makes matching features difficult. Further, only a small portion of the space is captured by both cameras. Thus having a complementary modality for tracking would be helpful. Here we use audio tracking, assuming that the object intermittently emits sound as it moves through the space.

Audio tracking is accomplished by using time delays measured at the microphone array . For a given particle state vector, the time delays corresponding to the particle state are computed and compared to the observation vector consisting of the measured time delays to obtain the posterior probability for the particle. Determining the source coordinates from measured time differences at an array is an almost classical problem arising in many fields such as radar, GPS or sonar; however, in the particle filter framework only the forward transformation (from the source coordinates to the time delay space) has to be performed. This forward transformation can be computed easily. Assume that we have N receivers located at known points $\mathbf{m}_i = (x_i, y_i, z_i)$ and a source at $\mathbf{x}_s = (x_x, y_x, z_x)$. Denote the distances between the microphones and the source by χ_i, where

$$\chi_i = \sqrt{(x_i - x_s)^2 + (y_i - y_s)^2 + (z_i - z_s)^2}. \qquad (9)$$

Then, for a given source position $\mathbf{x}_s$, in the ideal case the measured time delay τ_{ij} between microphones i and j would be exactly equal to the computed time delay $\hat{\tau}_{ij}$

$$\hat{\tau}_{ij} = \frac{\chi_j - \chi_i}{c}, \qquad (10)$$

where c is the speed of sound. Thus, we use the difference between the left and the right sides of equation 10 as the error measure. Given the particle characterized by the state vector X as described in the previous section, and given the audio observation vector $Z_a = [\tau_{12} \ \tau_{13} \ ... \ \tau_{1M} \ \tau_{23} \ ... \ \tau_{M-1,M}]$, we can compute the time delays $\hat{\tau}_{ij}$ corresponding to the state $X = [x_p \ \dot{x}_p]$ by letting $\mathbf{x}_s = x_p$ and using equation (10) and then compute the total error as

$$\epsilon_a^2 = \frac{1}{C(M,2)} \sum_{i,j=1...M,i<j} (\hat{\tau}_{ij} - \tau_{ij})^2 \qquad (11)$$

In practice, we use some preliminary processing to reject obvious outliers (a RANSAC-type approach [8]) so that the size of the set over which the sum is taken can be less than $C(M,2)$. In this case the scaling coefficient is adjusted accordingly. For real data, we obtain the time delays using a robust algorithm that uses the noise estimate in the absence of the signal as a weight function in the generalized cross-correlation [7]. (More details on audio processing can be found in [9]). Given an audio error ϵ_a^2, the posterior probability $p_a(Z_a|X)$ of the observation Z_a given the state vector X is

$$p_a(Z_a|X) = \frac{1}{\sqrt{2\pi}\sigma_a} \exp\left(-\frac{1}{2}\frac{\epsilon_a^2}{\sigma_a^2}\right) \qquad (12)$$

2.6. Combined posterior probability estimation

The combined probability $p(Z|X)$ for both audio and video data is obtained by multiplying the corresponding probabilities from audio and video sources, assuming independent estimations by the complementary modalities. If at a given time step either audio or video data is absent, then the corresponding probability is simply marginalized to a suitable value and the estimation of the posterior probabilities for the particles is done using only video or only audio data.

3. APPLICATIONS TO EVENT DETECTION

The described method of performing multi-modal sensor fusion can be used to track an object at it moves through space, and to observe and detect characteristic patterns in either video, audio or both modalities that correspond to events. This requires human interpretation of when an event occurs. We do this by building a statistical model of the object state and interstate transition (e.g., using HMMs) and using the measurements to find out the corresponding object state or sequence of the object states (e.g., using the Viterbi algorithm). We show two examples of the method applied to an audio-visual event detection and classification problem. In the first example, a free flying bat is capturing a prey in a flight room with sound absorbing walls, and its acoustical echo-locating emissions are captured along with
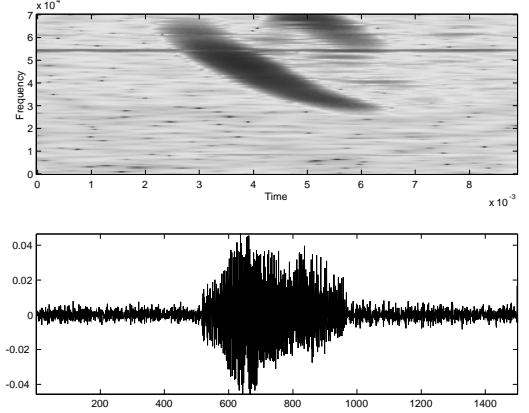

Figure 2: A spectrogram of one bat echolocation call.

video data. In the second example, a videoconference session or a meeting is taking place in the room, and changes of the active speaker are detected. The experimental setup and event detection model for these applications are described.

3.1. Flight room experimental setup

The flight room is setup primarily for biological behavioral study purposes. The room is relatively large (5m×5m×2.5m), and the room walls are covered by an acoustically absorbing material, thereby substantially mitigating acoustic reflection and multipath problems. The room is normally used to study bat behavior in the absence of visible light illumination, so that the bat is forced to use echolocation for navigation and prey hunting. Equipment in the room include two infrared cameras, several IR light sources, and a microphone array arranged on the floor along two corners of the room. The microphone array was not originally used for bat localization; only the energies of the signal in different spatially separated channels were used to study the direction and the width of the ultrasonic echolocation calls, and the bat trajectory was reconstructed from video data. We extended the system to perform multimodal bat tracking by incorporating the time-delay information from the microphone array data. The trajectory reconstruction is currently performed off-line using recorded audio and video data; work toward the on-line tracking is in progress.

The bat used in this study, the big brown bat (*Eptesicus fuscus*), emits ultrasonic chirps consisting of downward sweeping FM sounds. The plot and the spectrogram of an individual chirp are shown in Figure 2. The signal bandwidth extends from 50 kHz down to 20 kHz. The duration of the signals ranges from 2 to 20 ms. The bat was trained to fly in a flight room and capture a mealworm suspended from the ceiling by a microfilament. The bat's flight was recorded using two Kodak MotionCorder^{TM} digital cam-
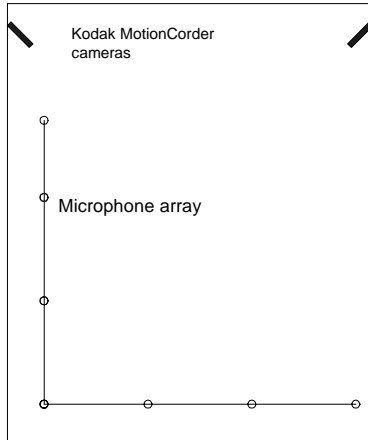
Figure 3: Schematic of flight room experimental set up.

Figure 4: A sequence of bat vocalizations during search, approach and capture.

eras running at 240 frames/sec. Vocalizations of the bat were recorded from six microphones (Knowles FG3329) arranged in an "L" shaped array. Sounds were digitized at 140 kHz/channel using an IoTech Wavebook^{TM}. The video and audio data were synchronized by running the acquisition off a common trigger. A schematic of the flight room is shown in Figure 3.

3.2. Capture event

Behavioral studies of bats show that a flying hunting bat uses ultrasonic echolocation pulses to navigate in the room, to localize the target and to distinguish between edible and inedible targets. The bat's hunting behavior is generally divided into three stages: search, approach and capture [4]. When the bat is in general flight and unaware of the presence of the target, its ultrasonic chirps are relatively rare and powerful (search mode). As the bat acquires the target it begins emitting more frequent vocalizations (approach mode). After the capture, indicated by a joining of the estimates of the bat track and the target track, the bat is silent for a while, and then begins to emit search mode clicks again. Figure 4 is a recording of a bat acoustic activity over the period of one second which includes all three hunting stages. The end of a frequent vocalization series in the plot signifies the capture event.

We use a simple sensor fusion method to detect the capture event. The bat trajectory must pass in the vicinity of the target position (obtained via video detection), and the bat acoustic activity must cease for a while (obtained via audio detection). If both these conditions are satisfied, we claim that the capture event took place.

3.3. Videoconferencing setup

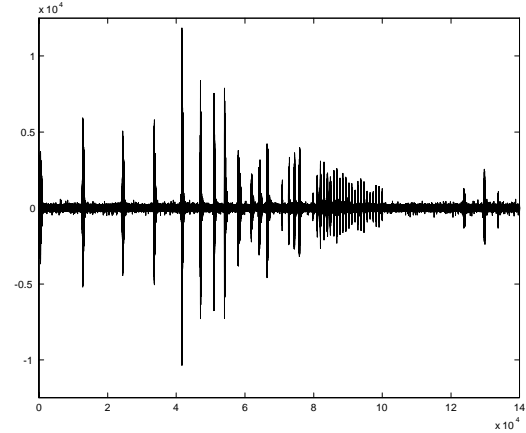To perform algorithm evaluation in noisy and reverberant environments, we collected and processed data in a regular office room. The setup is generally similar to the system described in [9] and consists of a tracking subsystem with two video cameras and two microphone arrays with seven microphones in each array, and a single video camera used for collecting videoconferencing data. The cameras are located at two corners along a wall of the room and are calibrated using the same calibration object as the cameras in the flight room. The microphone arrays are attached to the room wall between the cameras. One dual-processor Pentium III 933 MHz PC was used to acquire and process the audio and video data in real time. The capabilities of a single PC system bus throughput limited the operation rate of the tracker to approximately 8 updates per second.

We use both audio and video data to track the active source using the multimodal tracker described earlier, to detect an active speaker, detect the change in identity of the active speaker and to control a third camera (used to collect the videoconferencing stream) which follows the active speaker (if any) or zooms out when a period of silence is detected. Audio localization data is used to initialize the tracker. Subsequently, the speaker is followed by the active camera using multimodal tracking. The speakers are distinguished and identified by color histograms of their images. The details of audio-video processing and camera control algorithms are described in [9].

3.4. Speaker change event

When the audio localization data do not fall within a small tolerance region around the position of the current speaker, the system assumes that a speaker change event is taking place. This event is often referred as *turn-taking*. The record of speaker change events made over a relatively long time (minutes), combined with identities of individual speakers, form a turn-taking sequence of the meeting.
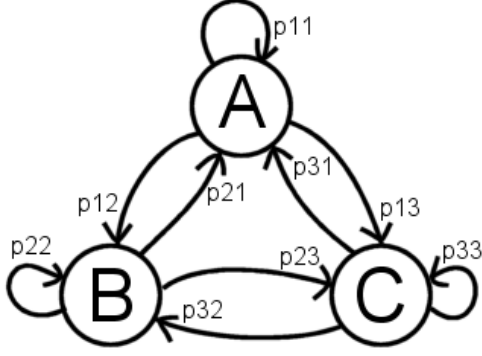
Figure 5: A simple Markov model for a turn-taking sequence.

The high-level structure of the scene can be recovered from this sequence by building a simple Markov model for the speaker change events. Figure 5 shows an example when three speakers are present.

The nine-element state transition probability matrix, P, fully describes the stochastic process corresponding to the model. Element p_{ij} of the matrix corresponds to the transition probability from state i to state j at the current time step. Only six model parameters are independent since the sum of the transition probabilities outgoing from one state must equal to one. The values of individual elements in the matrix bear relationship to the temporal structure of the corresponding stochastic process, e.g. to the average length of continuous speech by a single speaker and to the frequency of speaker change event.

From the observed turn-taking sequence we can obtain the matrix P, and use it to classify meetings. If N_{ij} is the number of times when the arc from state i to the state j was visited (i.e. when person j became the active speaker after person i was speaking), then we define

$$p_{ij} = (N_{ij} + 1)/\sum_i (N_{ij} + 1). \qquad (13)$$

An extra visit is added to every arc to ensure that no probability is zero. This is desirable if one wants to create an artificial turn-taking sequence using P.

Having computed the elements of P, the values and the ratios of the elements on its main diagonal can be used to classify the turn-taking sequence into three general groups – lecture, discussion and conversation. In a sense, the value of p_{ii} is related to the average speaking time of ith speaker. When one p_{ii} it is close to 1.00, the conversation is likely to be *a lecture* with occasional interruptions by listeners; when three values of p_{ii} are similar, the roles of speakers are likely to be equal. Further examination of probability distribution over a row can distinguish between a *discussion-type*

meeting and *informal conversation* which correspond to the relatively small and relatively large non-diagonal elements of P, respectively.

4. RESULTS

We implemented the algorithm to perform multi-modal tracking and event detection in two different setups. For the bat in free flight, we first test the algorithm performance by using synthesized audio/video traces using the same room and sensor geometry and parameters as for real data. We then apply the algorithm to real bat flight trials. We also implemented the algorithm for audio-visual tracking of video-conferencing participants. Preliminary results on automatic meeting classification are presented. These results may be used for content classification and retrieval purposes.

4.1. Synthetic Data

To numerically evaluate the performance of the algorithm in the case when ground truth is available, we tested it with synthetic data. A synthetic dataset was generated with parameters corresponding to the conditions in the real data acquired from the flying bat experiments. An object is assumed to move in a spiral motion for one second along the trajectory $x = \sin(2\pi t)$, $y = 2.0 - t$, $z = \cos(2\pi t)$, $t \in [0, 1]$. The frame rate was set to be 240 fps, the discretization rate to 140 kHz, and all the geometric parameters of the system were kept the same as in the real data obtained in the quiet room. In every frame, the object position in the two camera images, (u_1, v_1) and (u_2, v_2), are obtained. In addition, the values of time delays τ_{ij} between all pairs of microphones are also computed. Independent Gaussian noise is added to the image coordinates and to the time delays with zero mean and variances of 3 pixels and 10 samples, respectively. These 25 values constitute the observation vector for a given time instant and are fed to the condensation tracker initialized with the correct initial state vector $[0.0\ 2.0\ 1.0\ 0.0\ 0.0\ 0.0]$. For every frame, the Euclidean distance between the object coordinates obtained from the tracker and the true object coordinates is computed and the average distance over all frames is taken to be the error measure. This is repeated several times with different random number generator seeds, and the average result is presented. In addition, the tracker performance is also measured in absence of one of the two modalities (i.e., for audio data only and for video data only), and the performance is plotted versus the number of particles in the particle filter (Figure 6).

The results show that the performance improves when the number of particles is increased, as might be expected. The audio data alone gives substantially higher residual error, which drops significantly as the number of particles is increased. That can be attributed to the high dimensionality of the observation space and to the fact that all the microphones lie in one plane, which diminishes the accuracy of 3-D coordinate determination. The video data alone show less
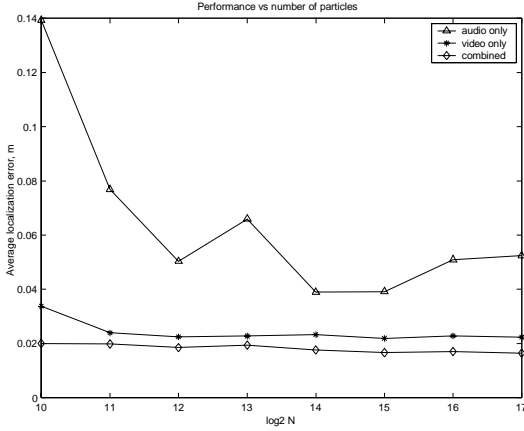
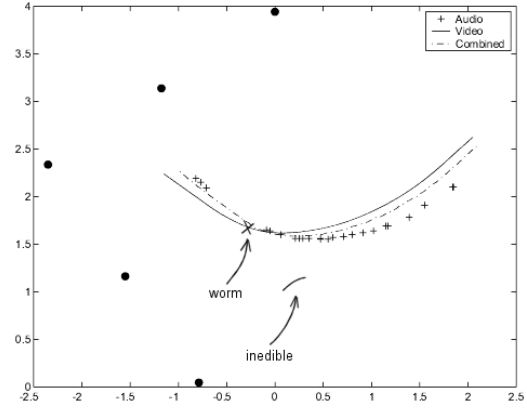Figure 6: Effect of the variable number of particles.

Figure 7: Trial 1: Audio, video and combined tracks of the bat flight.

Figure 8: Trial 2: Audio and video tracks for the second trial with the moving target.

error and shows some improvement as the number of particles increases. The combined data show even better performance which is improved by a small factor (about 15%) as the number of particles grows. The smallest tracking error for combined audio and video tracking (about 1.64 cm) is approximately 2.5 times less than the error obtained by the pure object detection in every frame without any tracking involved (about 3.83 cm). This shows the effect that learning the object motion model has on the localization error.

To summarize, the combined audio and video tracking provides significantly better error rate than either of these two modalities separately, and is achieved with relatively small number of particles. The algorithm is therefore well-suited for real-time implementation.

4.2. Real Data

We present results from two trials of a bat moving towards a tethered mealworm prey. There is also an inedible distracting target located in proximity to the edible target. The bat flies in from the right towards the target located at the left in these figures (which show a plan view of the room). Figure 7 provides both audio and video tracks for the first trial as well as output from the multimodal tracker. The microphone positions are also shown as large black dots. The video coordinates are available in every frame, while the audio track is available when the bat emits vocalizations, and consists of a scattered set of points. The audio localization is done using the CL1 algorithm described in [9]. As can be seen, the audio solution tracks the video data quite well. The output of the multi-modal tracker lies between the audio and video tracks, as also can be expected. The slight constant disagreement between audio and video is likely due to inaccuracies in determination of microphone coordinates. The coordinates are determined from the locations of the microphone receivers in stereo images using the
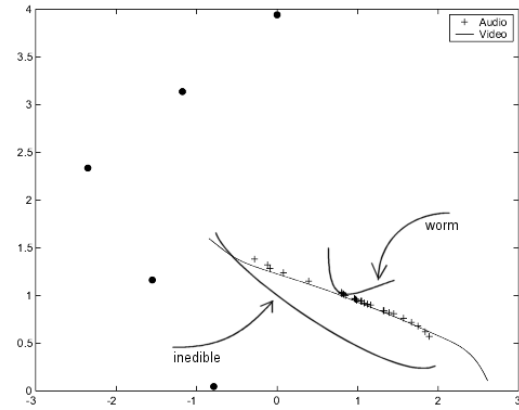
DLT transformation described earlier, and the calibration for DLT is accurate only in a limited space region occupied by the calibration object. The microphones lie far from this well-calibrated region.

A second trial is shown in Figure 8. In this trial both the prey and the distracting target are moving, also from right to left. The bat is able to come near the correct target, but misses it. Again, the audio and video tracks are in a very good agreement, and the combined track differs very little from the video trajectory and therefore is not plotted.

In both plots the temporal density of the echolocation calls changes during the flight sequence. As described before, the end of the frequent vocalization pulse series with relatively low signal power signifies the prey capture event. It can be seen that this corresponds to the bat trajectory passing in a small neighborhood of a prey location. These two cues combined are used to identify the moment of capture.
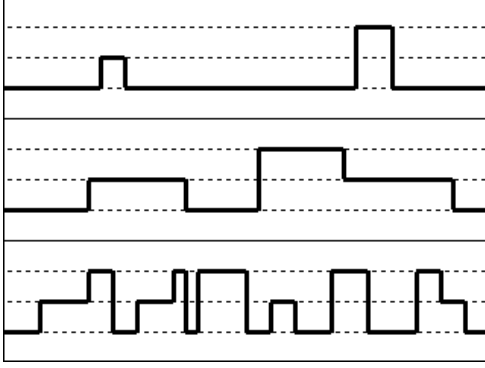
Figure 9: Three sample turn-taking sequences.

The capture event is identified correctly in both cases. The event detection in this experiment is thus rather simple.

4.3. Videoconferencing setup

The accuracy of the audio localization in the videoconferencing setup is sufficiently lower than in the quiet room due to the smaller microphone array baseline, lower sampling frequency, noise and reverberations. Thus, the width of the Gaussian kernel for the audio distribution in the multi-modal tracker is increased to avoid jerky tracker outputs, and audio information serves as the supporting modality for the more accurate video data. The details of the data processing are given in [9].

To test the applicability of the multi-modal tracker for the event detection application, we obtained data from several types of simulated meetings. We selected three data sets from the recordings. The visual depiction of turn-taking sequences in these three cases is presented in Figure 9. The figure graphically shows the time segments of activity of different participants. From these plots, it can be inferred that the first sequence is a lecture-type session since one person is a primary speaker. The second and the third patterns can be classified as a discussion meeting (regular speaker changes) and a less formal (or more active) conversation (frequent speaker change and interruption). The matrices P_1, P_2, P_3 for these three turn-taking sequences computed according to the rule described earlier are shown below:

$$P_1 = \begin{bmatrix} 0.90 & 0.05 & 0.05 \\ 0.40 & 0.40 & 0.20 \\ 0.33 & 0.17 & 0.50 \end{bmatrix} P_2 = \begin{bmatrix} 0.78 & 0.11 & 0.11 \\ 0.15 & 0.80 & 0.05 \\ 0.10 & 0.20 & 0.70 \end{bmatrix}$$

$$P_3 = \begin{bmatrix} 0.58 & 0.21 & 0.21 \\ 0.21 & 0.58 & 0.21 \\ 0.34 & 0.13 & 0.53 \end{bmatrix}$$

It can be seen that the elements of the state transition probability matrix can be used to describe the nature of the

meeting and that the classification according to the rules described in section 3.4 agrees with the correct classification. Again, this is in no way an extensive study but a simple application showing that the developed multi-modal tracker can be successfully used for the event detection and classification problems.

5. CONCLUSIONS

We have developed a multi-modal sensor fusion tracking algorithm based on particle filtering which is able to integrate multiple modalities and cope with temporary absence of some measurements. The tracking is essential to the event detection problem since it allows the system to selectively look and listen at potentially interesting locations. We applied the tracking method for two specific simple event detection problems and found that the results are promising.

6. REFERENCES

[1] M. Isard, A. Blake. "CONDENSATION conditional density propagation for visual tracking". International J. Computer Vision, 28(1), 1998.

[2] M. Isard, A. Blake. "ICONDENSATION: Unifying low-level and high-level tracking in a stochastic framework". Proc. ECCV1998, Freiburg, Germany.

[3] D. Forsyth, J. Ponce. "Computer Vision: A modern approach". (pre-publication copy available at http://www.cs.berkeley.edu/~daf/book.html).

[4] C. Moss, H. Schnitzler, "Behavioral studies of auditory information processing" in A. Popper and R. Fay (Ed) "Hearing by bats", Springer Handbook of Auditory Research, Vol. 5, pp. 87-145, 1995.

[5] J. MacCormick, A. Blake. "Probabilistic exclusion and partitioned sampling for multiple object tracking". International J. Comp. Vision, 39(1), 2000.

[6] C. Slama, C. Theurer, S. Henriksen, editors. "Manual of Photogrammetry. Fourth edition." American Society for Photogrammetry and Remote Sensing, Falls Church, Virginia 1980.

[7] D. Feitelson, A. Weil. "A robust method for speech signal time-delay estimation in reverberant rooms". Proc. ICASSP96, Atlanta, GA.

[8] R. Hartley, A. Zisserman. "Multiple View Geometry in Computer Vision". Cambridge Press.

[9] D. Zotkin, R. Duraiswami, L.S. Davis & I. Haritaoglu. "An audio-video front-end for multimedia applications," Proc. IEEE SMC 2000, Nashville, pp. 786-791, 2000.

[10] V. Philomin, R. Duraiswami, L. Davis. "Quasi-random sampling for Condensation". Proc. ECCV2000, Dublin, Ireland.

[11] B.Li, R.Chellappa. "Simultaneous Tracking and Verification via Sequential Posterior Estimation". Proc. CVPR2000, Hilton Head, SC.