MULTIMODAL TRACKING FOR SMART VIDEOCONFERENCING

Dmitry Zotkin, Ramani Duraiswami,Harsh Nanda, Larry S. Davis

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

ABSTRACT

Many interactive multimedia applications require the ability to track the 3-D motion of participants in a room. Particle filters are attractive for this since they do not require solution of the inverse problem of obtaining the state from measurements, and since the tracking can be easily extended to integrate multimodal measurements. We extend our previous work on smart videoconferencing to include a multimodal tracker of the session participants using multiple cameras and microphone arrays. We verify the correctness and robustness of the multimodal tracker using synthetic and real data. We also present practical details of how such a system can be implemented using off-the-shelf hardware and computers.

1. INTRODUCTION

Our perception of the environment is a strong function of our location relative to objects being perceived. It changes with our orientation, body posture, whether we are seated or standing, etc. Applications in virtual and augmented reality (VR/AR) that seek to create convincing experiences require an ability to track the motion of a person (especially the head, where most of the sensing modalities are) as he moves through a 3-D space. Applications that seek to provide telepresence (e.g., videoconferencing) need to determine the spatial distribution of people. This information must be obtained at a relatively high rate to minimize cognitive dissonance in VR applications and to quickly follow the motion of an active speaker in videoconferencing.

Multiple modalities can be used to acquire such positional information, including multi-perspective video based localization of inserted markers or of feature points obtained from image processing, audio localization using speech or transmitters placed on moving objects, magnetic tracking of installed tags, etc. Integrating information obtained from multiple sensors and across many time-steps can lead to improvements in both the accuracy and the sampling rate, and lead to a practical design for smart videoconferencing or VR/AR systems. The goal of the present paper is to achieve such integration for audio and video data via use of particle filters to perform 3-D tracking. The data are obtained using multiple cameras and microphone arrays. The algorithms are implemented in an enhanced version of a smart videoconferencing system presented earlier [1].

In §2 of the paper provides a brief introduction to the tracking algorithm; §3 introduces the video and audio tracking algorithms and their combination. In §4 we present some practical details of our system. In §5 we present results that verify the correctness and usefulness of the multimodal tracker using both synthetic and real data. We show results of several successful tracking experiments in different environments including results of a person moving with an ultrasonic sound source in a quiet room (with sound-absorbing walls), a flying echolocating bat in the same room, and a speaker in a typical office room. We also show that our algorithm is capable of successful handling of temporary absence of some measurements (the target being occluded from one or both cameras, or absence of audio data). §6 concludes the paper.

2. THE TRACKING ALGORITHM

The particle filtering tracker, known also as a CONDENSATION tracker, was first introduced for vision based tracking in [3]. The mathematical framework of the tracker provides a way of updating the variables in the state vector X_s of the tracked object (e.g., coordinates, velocities, Euler angles, color histogram etc.), on the basis of the measurement/observation vector X_m that consists of the values obtained from the sensors. The (unknown) true state vector for any given time corresponds to a point in a state space. Instead of working with actual values of the state vector, the method works with the probability distribution function (pdf) on the state space that represents the uncertainty in the knowledge of the state vector. The tracker maintains an explicit approximate pdf by computing its value at a set of randomly selected sample points (called *particles*) in the state space, and thus is able to work well when Kalman filtering fails due to the underlying pdf being non-Gaussian. A further advantage of the technique, which is likely to be more important in practice, and that is the focus of the present paper, is that since the technique does not require the construction of explicit inverse solutions, it allows one to mix modalities/measurements during the tracking relatively easily.

Algorithm overview: The particle set update algorithm used in this paper is very similar to the original algorithm, though it is pertinent to note there are several improvements of the original algorithm published. These include importance sampling [4], stratified sampling [5] and quasi-random sampling [6]. In the algorithm used in this paper, every particle in the set $\{x_i\}$, $i = 1...N$, in the state space X has a weight π_i associated with it. This set is called *properly weighted* if it approximates the true pdf $P(x)$, so that for every integrable function $H(x)$

$$E_{P(x)}(H(x)) = \lim_{N\to\infty} \frac{\sum_N H(x_i)\pi_i}{\sum_N \pi_i}. \qquad (1)$$

Given a properly weighted set of particles at time t with equal weights, it is possible to update it to reflect the new measurements obtained at time $t + \delta t$. The update algorithm is as follows:

1. Propagate each particle $\mathbf{x}_i$ in time using the object *motion model* to obtain an updated particle set $\{\mathbf{x}_i^*\}$.

Support of NSF award 0086075 is gratefully acknowledged. We would also like to thank Kaushik Ghose and Cynthia Moss of the University of Maryland bat laboratory for providing us with the bat data.

2. Obtain a new measurement vector X_m and evaluate the *posterior probability density* π_i^* on $\{\mathbf{x}_i^*\}$, $\pi_i^* = p(x_i^*|X_m)$, which measures the probability of $\mathbf{x}_i^*$ given X_m. This can be written using Bayes' rule:

$$p(\mathbf{x}_i^*|X_m) = \frac{p(X_m|\mathbf{x}_i^*)p(\mathbf{x}_i^*)}{p(X_m)} \qquad (2)$$

   in which $p(X_m)$ is the prior probability of measurement, which is assumed to be a known constant, and $p(\mathbf{x}_i^*) = 1/N$. Thus, $p(\mathbf{x}_i^*|X_m) = Kp(X_m|\mathbf{x}_i^*)$ for some constant K, and $p(X_m|\mathbf{x}_i^*)$ can be computed without inversion of the measurement equations.

3. Resample from $\{\mathbf{x}_i^*\}$ with probabilities π_i^*, and generate a new *properly weighted* set $\{\mathbf{x}_i'\}$ with equal weights $1/N$ for each particle. Repeat steps 1-3 for subsequent times.

Instantiation of the particle filter In this section, we describe the state and measurement vectors, the motion model and the posterior estimation functions used. The state X_s consists of the coordinates and velocities of the tracked point: $X_s = [x\ y\ z\ \dot{x}\ \dot{y}\ \dot{z}]$. The motion model which is used to propagate it in time is

$$x(t+\delta t) = x(t) + \dot{x}(t)\delta t, \quad \dot{x}(t+\delta t) = \dot{x}(t) + F\delta t, \qquad (3)$$

and similar expressions for $y, z, \dot{y}, \dot{z}$. The F in the equation is a random acceleration applied to the particles and is dependent on the expected range of the object velocity. It provides an element of robustness to the tracker.

The measurement vector is made up of a video part consisting of the pairs $(\hat{u}_i, \hat{v}_i)$ of *image coordinates* of feature points on the tracked object for every camera in the system, and an audio part that consists of the values of *time differences of arrivals* (TDOA) $\hat{\tau}_{ij}$ of the acoustic source signal between different microphone pairs in the microphone array. Thus, N video cameras produce $2N$ components of the observation vector for each feature point, and for M microphones the number of audio observations is C_2^M per source. The transformation that converts the world coordinates into the image coordinates is pre-computed by a calibration procedure (described below), and is used to "project" the state vector to the image coordinate space and compute the posterior probability of the state given the observations. The corresponding transformation from the state space to the audio observation space is easy to compute and is also described below.

We define the video and audio error measures and the posterior probability as

$$\epsilon_v^2(X_s, X_m) = \frac{1}{N}\sum_{i=1}^{N}[(u_i - \hat{u}_i)^2 + (v_i - \hat{v}_i)^2], \qquad (4)$$

$$\epsilon_a^2(X_s, X_m) = \frac{1}{C_2^M}\sum_{\forall i,j \quad i<j}(\tau_{ij} - \hat{\tau}_{ij})^2, \qquad (5)$$

Here (u_i, v_i) are the image coordinates obtained by projecting the sample 3-D point locations $\mathbf{x}$ using the "projection function" Φ_v, while τ_{ij} is the TDOA induced between microphones i and j for the source at the sample location $\mathbf{x}$. Using these errors we can define the final component of the tracker, which is the likelihood function that determines how likely it is that a particular observation vector X_m at time t is caused by an object state X_s.

$$p(X_m|X_s) = \frac{\exp\left(-\frac{\epsilon_v^2}{2\sigma_v^2}\right)}{\sqrt{2\pi\sigma_v}}\frac{\exp\left(-\frac{\epsilon_a^2}{2\sigma_a^2}\right)}{\sqrt{2\pi\sigma_a}}. \qquad (6)$$

The parameters σ_v and σ_a are the audio/video standard deviations. Loosely speaking, they define how much trust is put on an individual measurement. If the measurements are known to be inaccurate, larger values of σ_a and σ_v should be used. However, if the value used is too large the filter is slow to learn the observed motion. Note that the $p(X_m|X_s)$ is a product of Gaussians formed from individual measurements. If at some time part of video or audio observation vector is unavailable, then that part of $p(X_m|X_s)$ is simply set to a constant, and the update is performed using the marginalized values.

Video Tracking Our video setup consist of two video cameras in two corners of the room. We approximate the camera projection equations using the *Direct Linear Transformation* (DLT). The DLT uses a 3x4 matrix P to take the 3-D coordinates $\mathbf{x}$ into image coordinates (u_i, v_i) in camera i as

$$\left\{\begin{array}{c} u_i \\ v_i \end{array}\right\} = \left\{\begin{array}{c} \frac{p_{11}x+p_{12}y+p_{13}z+p_{14}}{p_{31}x+p_{32}y+p_{33}z+1} \\ \frac{p_{21}x+p_{22}y+p_{23}z+p_{24}}{p_{31}x+p_{32}y+p_{33}z+1} \end{array}\right\}. \qquad (7)$$

Obtaining the coefficients p_{ij} is the calibration step for the video measurements. To do this, we used a Peak calibration frame with 25 white balls on a black frame that is approximately [2m]^3. The 3-D coordinates of all balls, (x_j, y_j, z_j), are known to an accuracy of ~5 mm. The frame is placed in the region visible from both cameras, and the corresponding image coordinates of balls (u_i, v_i) obtained. The 25 point correspondences are plugged into the DLT equations to obtain an overdetermined system of 50 equations in 11 unknowns, which is solved using least-squares. Knowing the coefficients Equation (7) can be used to project the sample state locations to the measurement space and determine the error (4) and probability (6).

Audio tracking The audio algorithms for estimating the TDOAs (the audio measurements) have been described in our earlier work [1, 2]. Given the location of the sample points in the state space we can project them into the measurement space by explicitly computing the TDOAs using

$$\tau_{ij} = \frac{||\mathbf{x} - \mathbf{m}_j|| - ||\mathbf{x} - \mathbf{m}_i||}{c}, \qquad (8)$$

where $\mathbf{m}_i$ and $\mathbf{m}_j$ are the coordinates of i^{th} and j^{th} microphones, and c the sound speed. Obtaining these quantities is the audio calibration problem. Algorithms to obtain the source location from a set of TDOAs have been described in [1] and [8]. They were used to verify the tracker performance.

3. EXPERIMENTAL SETUPS

Videoconferencing setup For the videoconferencing trials, two color Sony EVI-D30 cameras were used at a resolution of 320×240 with Matrox Meteor II frame grabbers. The videoconferencing room audio setup is the same as in [1]. Two arrays of seven Panasonic button microphones each are used. Each array has one microphone at the center and six along the circumference of a circle of diameter 12". The microphones are connected to a custom-made low-noise low-distortion preamplifier based on a AD797 chip, and the signal is digitized at 22.05 kHz per channel using a PowerDAQ board. To ensure a good match between the audio and video coordinate systems the calibration frame is set up with its X and Z axis parallel to the room walls. The distance from the origin of the audio coordinate system to the central ball of the frame is obtained by direct measurement.

Implementation of the videoconferencing system The tracking system for the videoconferencing is implemented on a dual PIII-933MHz PC running Windows NT 4.0. Multithreaded programming allows utilization of both processors. The controlling software runs in a loop, performing audio capture and analysis, video capture and analysis, and state update via the particle filter. Algorithms described in [1] are used to control the third camera which is used to transmit images to the remote site at the full frame rate (30 fps). The processing power and the bus throughput of the single computer limits the tracking rate. Our ongoing research is exploring the use of a cluster of computers networked via gigabit ethernet to speed up the process [9]. This should significantly improve the fidelity of the tracker and make it suitable for use in virtual reality applications to track human location and pose [7].

Quiet room setup To assist biologists studying bat behavior [8], while at the same time collecting interesting data for verifying the joint-tracking algorithm performance, we acquired multi-channel audio and video recordings of a flying echolocating bat hunting a tethered mealworm prey, using a quiet room. In this room the video is recorded using two Kodak MotionCorder digital infrared cameras at a resolution of 640×480 and 240 fps. Near-infrared illumination was used during the recording, as the biologists wished to ensure that the bat flew using echolocation alone. The video stream was recorded at the digital video recorder with embedded timestamps. The bat and the ultrasonic sound source used in the quiet room produce ultrasonic chirps of 20 kHz - 50 kHz. To capture these signals seven Knowles FG3329 microphones were arranged on a horizontal plane in an L-shaped frame. The signal was preamplified using a home-made circuit and digitized at 140 kHz per channel using a IoTech Wavebook board. The positions of the individual microphones for audio processing were obtained from the video images (they are visible in the image as small dots), which introduced some calibration error in the audio results in this setup.

4. RESULTS

We performed the evaluation of the developed multimodal tracker on several sets of synthetic and real data obtained using the two setups. The synthetic data were used to verify the algorithm performance when ground truth is available. We are able to show that the performance of a multi-modal tracker is better that the performance of both the audio and the video trackers taken separately.

Synthetic Data We created a synthetic dataset by simulating an object moving in a spiral motion over the trajectory given by

$$x = \sin(2\pi t),\ y = 2.0 - t,\ z = \cos(2\pi t) \tag{9}$$

where $t \in [0, 1]$. The frame rate was set to 240 fps, the audio sampling frequency to 140 kHz, and all geometric parameters of the system were taken to be those of the setup in the quiet room. In every frame, we obtained the feature-point coordinates in image frames and the values of TDOAs. Then zero-mean Gaussian noise with variances of 3 pixels and 10 samples was added respectively to the image coordinates and TDOA values were. The tracker was initialized with the correct source position at $t = 0$ and zero velocity. The σ_v and σ_a for the tracker were set to 3 pixels and 10 samples, corresponding to the true measurement noise. (Any change in these values resulted in an increase in the estimation error). We performed several runs of the tracker with different number of particles. Average distances between the estimated and true object positions are shown in Figure 1.

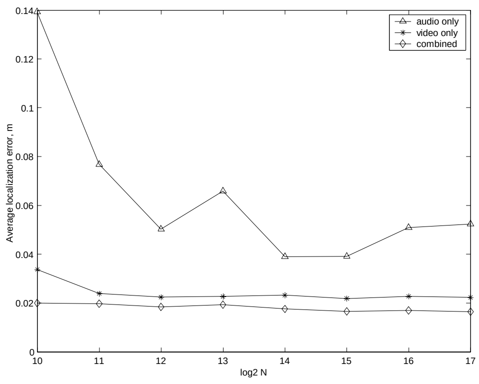

Figure 1: Performance vs log number of particles

It can be seen that the performance improves with the number of particles. The audio tracker performance alone is not very good. This can be attributed to the fact that all the microphones lie in the same horizontal plane which decreases the accuracy of object height determination. The performance of video tracker alone is better, and the performance for the combined tracker is improved even more. The smallest tracking error obtained in experiments is approximately 1.64cm, which is 2.5 times less than the error obtained for single-frame video object detection (about 3.83cm), which shows the effect of learning the object motion parameters.

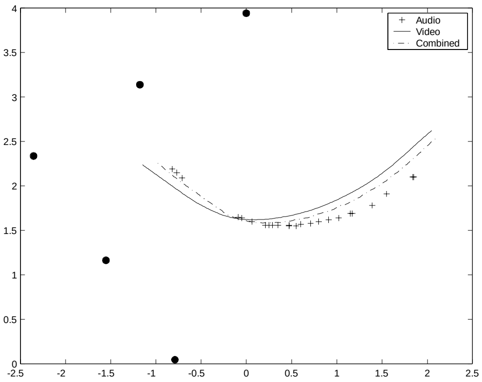

Figure 2: Bat flight path and microphone locations (quiet room).

Tracking ultrasonic sources in the quiet room Obtaining correspondences from images in the quiet room was relatively easy due to the experimental set-up. The images are taken with near-infrared light since the biologists who use this room want the bat being imaged to only use echolocation to detect its prey. In addition the walls are covered with black audio-absorbing material. Thus, only a few bright spots are seen in the infrared image, and simple background subtraction suffices to determine corresponding points. In the case of trials with a flying bat, the head of the bat is visible and was hand selected in each image frame. We also acquired data of a person carrying an ultrasonic sound source in the room. Here some reflecting tape was placed on the source, and its image could be easily located using background subtraction.

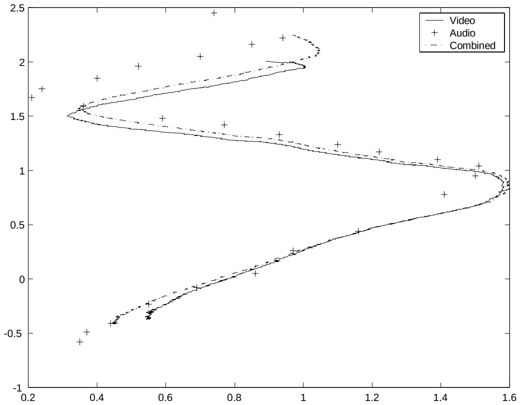

Figure 3: The trajectory of the ultrasonic sound source (quiet room)

The object position in the frame for both cameras and TDOA values were used as input to our tracker. The quantity F in Eq. (3) was set to $\mathcal{N}(0, 10)$. In addition, the video trajectory and audio data points was determined from video and audio data independently. In Figures 2 and 3, the video trajectory, audio data points corresponding to the individual echolocating bat calls and the output of the tracker are shown for two trials. The results show that the independently obtained video and audio trajectories are in good agreement. The misalignment between them is likely due to the fact that the microphone coordinates used in the audio algorithms were obtained from the video images. These can be inaccurate because the microphones lie far from the area where the calibration frame was placed. The output of the multi-modal tracker integrates the audio and video information and lies between the tracks, as expected. No ground truth data is available for these runs. We plan to verify the microphone coordinates by other means or find other reasons for this bias.

Videoconferencing with occlusions In the videoconferencing room, we made audio and video recording of a single speaker moving in several patterns in the field of view of a tracking system. Acquiring corresponding feature points between the images is more complicated here. The image of the head for a person 3.5 meters from the camera is at most 15×15 pixels, so the level of detail is insufficient to use a face template. We used a simple background substraction to roughly locate the head of the person. That approach provides sufficient accuracy for videoconferencing purposes since we only require approximate centering of the frame around the source. However, for the trials presented here we perform more accurate position estimation by placing a colored marker on the face of the speaker.

Agreement of the video and audio trajectory to 50 mm is achieved. A low discretization frequency, relatively small intermicrophone distance within the audio arrays and large distance from the microphone arrays to the speaker contributed to relatively low accuracy of audio data, so the video data is primary source of information for the run. The audio error is however less than 10 cm, which is sufficient to correctly locate the speaker and put her into a frame for transmission.

We also tested the algorithm robustness to occlusions. Normally, the speaker is visible to both the tracking cameras. When only one camera can see the speaker the video information alone can not be used to recover the speaker coordinates. The multimodal tracker however continues to track the speaker correctly using audio data and marginalizing the video information. In Figure 4 we show the tracking results with a simulated occlusion. For the part of the track within the area marked “Occlusion”, the video data from one of the cameras were treated as missing. The track stays near the video trajectory, although it is influenced more by audio data. When the video data becomes available again, the tracking error decreases.

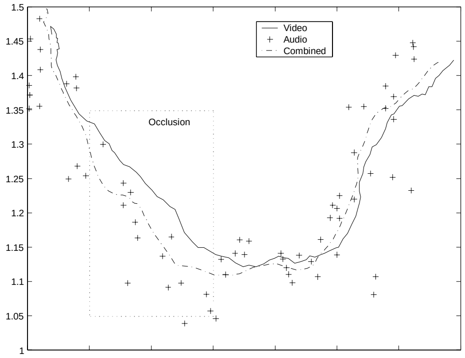

Figure 4: Multimodal speaker tracking with occlusions

5. CONCLUSIONS

The described tracker allows a natural framework to integrate multimodal information for tracking and performs robustly even when information from one of the channels is missing (video occlusions or audio noise). We anticipate continued development of the system and achievement of a high frame-rate, high-fidelity, real-time tracker for videoconferencing and virtual reality applications.

6. REFERENCES

[1] D. Zotkin, R. Duraiswami, V. Philomin, L. Davis.“Smart Videoconferencing”. Proc. ICME, New York, Aug. 2000.

[2] D. Zotkin, R. Duraiswami, L. Davis, I.Haritaoglu. “An audio-video front end for multimedia applications”, Proc. SMC2000, Nashville, TN, Oct. 2000.

[3] M. Isard, A. Blake. “CONDENSATION conditional density propagation for visual tracking”. Intl. J. of Comput. Vision, 28, 1996.

[4] M. Isard, A. Blake. “ICONDENSATION: Unifying low-level and high-level tracking in a stochastic framework”. Proc. ECCV, Freiburg, Germany, 1998.

[5] J. Carpenter, P. Clifford, P. Fearnhead. “An improved particle filter for non-linear problems”. IEEE Proc. Radar, Sonar and Navigation, vol. 146, 1999.

[6] V. Philomin, R. Duraiswami, L.Davis. “Quasi-random sampling for CONDENSATION”. Proc. ECCV, Dublin, 2000.

[7] R. Duraiswami et al. “Individualized HRTFs using computer vision and computational acoustics”.J. Acoust. Soc. Am., vol. 108, p. 2597, 2000.

[8] K. Ghose, D. Zotkin, R. Duraiswami, C. Moss. “Multimodal localization of a flying bat”. accepted ICASSP, Salt Lake City, May 2001.

[9] R. Duraiswami, D. Zotkin, L.S. Davis. “Active speech source localization by a dual coarse-to-fine search”. accepted ICASSP, Salt Lake City, May 2001.