A BAYESIAN 3D PEOPLE TRACKER USING MULTIPLE CAMERAS AND A MICROPHONE ARRAY

Yeongseon Lee and Russell Mersereau

Georgia Institute of Technology
Atlanta, GA 30332-0250

ABSTRACT

In this paper, we consider the problem of tracking multiple people in a 3D world domain using a microphone array and multiple cameras. The data fusion is done using a particle filter. To support 3D tracking, we propose a new video data likelihood model using a camera calibration matrix that can be used for a moving camera without continuous camera calibration. Then we apply an independent partition particle filter for multiple objects in order to generate particles efficiently. To detect the current speaker, we use a simple cost function using the generated particles. Finally we implement this tracking algorithm as a real-time system.

Index Terms— Particle filter, Acoustic arrays, face detection, multiple target tracking, Monte Carlo methods

1. INTRODUCTION

Joint audio-visual tracking using a particle filter has been a popular approach for people tracking recently [1–3]. The use of multiple modalities can overcome the limitation of a single one, such as occlusion with visual tracking or reverberation with acoustic tracking. In addition, a particle filter can model hidden Markov processes and incorporate non-linear, non-Gaussian properties of the dynamically changing environment beyond what a Kalman filter can accomplish.

Most past research for audio-visual people tracking using particle filters has been done using 2D camera images [1,2,4], not 3D world coordinates. Indeed, 2D camera domain tracking is sufficient for many applications, but some applications, nonetheless, require a 3D world coordinate location of the object. As one example, we might track multiple people using static cameras, then let another camera follow the individual who is talking among the crowd. Zotkin [3] demonstrated joint audio-visual 3D speaker tracking in world coordinates but tracked only a single object.

In this paper, we expand Zotkin's 3D audio-visual tracker in several ways. First, we track multiple people. Second, we implement a video data likelihood model with a more flexible camera calibration matrix that can reflect camera movements such as panning and tilting without camera re-calibration. Next, we apply an independent partitioned particle filter [5] to generate particles efficiently as well as handle data association issues among multiple objects. In addition, we detect the current speaker using a simple cost function based on the least square error.

This paper is organized as follows. In Section 2, we describe our particle filter and data likelihood models. In Section 3, we describe in some detail our multiple object tracking algorithm and the method for detection of the current speaker. Finally we present our experimental setup and simulation results.

2. PARTICLE FILTER FRAMEWORK

2.1. State-Space Model

Consider each person in the scene (in our case, a conference room) as one 3D random process. The random process does not have a constant motion, but random movements within a certain range. Therefore, we formulate the state-space vector with only locations and the state space-transition model as a random walk process with Gaussian noise as follows,

$$\mathbf{X}_t = [\chi_t^1, \chi_t^2, ..., \chi_t^K]^T$$

$$p(\mathbf{X}_t|\mathbf{X}_{t-1}) \sim \mathcal{N}(\mathbf{X}_t|\mathbf{X}_{t-1}, \Sigma)$$

where $\chi_t^i = [x_t^i, y_t^i, z_t^i]$ is the position of the i^{th} individual at time t, and the K is the number of people in the room.

Then, the state-space vector $\mathbf{X}_t$ is concatenated with S_t indicating the current speaker among K people at time t. However, S_t is not tracked using a particle filter, but detected using a least square error cost function at each time t.

2.2. Video Data Likelihood

In people tracking using joint audio-visual information, the different observations should point to the same hidden object. Since acoustic data show the existence and the position of a sound source, we track the position of a face centered around lips.

Different image observations can be used to detect a face. Contours, face template matches, skin colors, and motion are

perhaps the most frequently used visual features. Skin color information is known to be one of the dominant features for localizing faces, so we used it here.

To detect the skin area in a scene, we use the Bhattacharyya similarity distance [2] in the UV color domain. While [2] used the Bhattacharyya similarity distance of the whole image as a video observation, we localize the face candidates from the Bhattacharyya similarity distance image in Fig. 1. This localization makes it easy to associate the same object from multiple cameras.

(a) (b)

Fig. 1. (a) The original image marked with face localization with yellow stars. (b) The Bhattacharyya similarity distance image of the left image.

We denote the video feature observation $\mathbf{Y}_v$ in contrast with audio observation $\mathbf{Y}_a$. Then $\mathbf{Y}_v$ is defined as a vector of all face candidates at time t as

$$\mathbf{Y}_v = \{(u_{ij}, v_{ij})^T : i = 1, \ldots, K, j = 1, \ldots, M\}. \quad (1)$$

Here (u_{ij}, v_{ij}) are image coordinates of the i^{th} face candidate in the camera j, K is the number of face candidates, and M is the number of cameras. We assume that all cameras have the same number of face candidates at time t.

Next, we derive the relationship between a 3D world coordinate $\mathbf{x} = (x_i, y_i, z_i)^T$ and its corresponding 2D image coordinate $\mathbf{u} = (u_{ij}, v_{ij})^T$ at camera j. $\mathbf{u}$ is derived as a function in terms of $\mathbf{x}$ and the camera calibration matrix $\mathbf{C}$.

$$\mathbf{u} = f(\mathbf{C}, \mathbf{x}) = const \cdot \mathbf{C} \cdot \mathbf{x}. \quad (2)$$

The camera calibration matrix $\mathbf{C}$ describes the relationship between the 3D world coordinates and the 2D image coordinates of the camera and can be decomposed into 4 components as

$$\mathbf{C} = \mathbf{C_{Intrinsic}} \cdot \mathbf{C_{Tilt}} \cdot \mathbf{C_{Pan}} \cdot \mathbf{C_{Loc}}.$$

$\mathbf{C_{Intrinsic}}$ is an intrinsic camera calibration matrix, $\mathbf{C_{Tilt}}$ is the camera tilt matrix, $\mathbf{C_{Pan}}$ is the camera pan matrix, and $\mathbf{C_{Loc}}$ is the camera location matrix, which is fixed. This configuration allows us to move the camera with pans and tilts without re-calibration.

$\mathbf{C_{Intrinsic}}$ can be determined from various camera calibration tools [6] and does not change unless the focal length changes. On the other hand, $\mathbf{C_{Tilt}}$ and $\mathbf{C_{Pan}}$ are always updated whenever a pan or a tilt occurs. If θ is the panning angle, ϕ is the tilting angle, and the location of a camera is $[x_p, y_p, z_p]$, then $\mathbf{C_{Tilt}}$, $\mathbf{C_{Pan}}$, and $\mathbf{C_{Loc}}$ are as follows:

$$\mathbf{C_{Tilt}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(\phi) & -sin(\phi) & 0 \\ 0 & sin(\phi) & cos(\phi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{C_{Pan}} = \begin{bmatrix} cos(\theta) & 0 & -sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ sin(\theta) & 0 & cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{C_{Loc}} = \begin{bmatrix} 1 & 0 & 0 & -x_p \\ 0 & 1 & 0 & -y_p \\ 0 & 0 & 1 & -z_p \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Finally we formulate the video data likelihood function $p(\mathbf{Y}_v|\mathbf{X}_t)$ as

$$p(\mathbf{Y}_v|\mathbf{X}_t) \propto exp(-[\mathbf{Y}_v - \hat{\mathbf{U}}(\mathbf{X}_t)]^T \Sigma_v^{-1} [\mathbf{Y}_v - \hat{\mathbf{U}}(\mathbf{X}_t)]/2).$$

Here $\hat{\mathbf{U}}(\mathbf{X}_t)$ is calculated by (2) using the corresponding camera calibration matrix.

2.3. Audio Data Likelihood

Since a speech signal is wide-band and non-stationary, narrow band beamformers using the multiple signal classification (MUSIC) algorithm [7] do not work well. Even wide-band MUSIC can not handle room reverberation. For wide-band signals in reverberant environments, a time delay of arrival (TDOA) is more stable and is used here as audio measurements. TDOA is computed using the phase transform (PHAT) which shows better performance than the general cross-correlation method. Although an estimated TDOA has some error, it is still effective in the particle filter framework because the pooling of the measurements mitigates the effect of occasional errors.

The audio data likelihood $p(\mathbf{Y_a}|\mathbf{X}_t)$ is then formulated as

$$p(\mathbf{Y}_a|\mathbf{X}_t) \propto exp\left(-[\mathbf{Y}_a - \hat{\tau}(\mathbf{X}_t)]^T \Sigma_a^{-1} [\mathbf{Y}_a - \hat{\tau}(\mathbf{X}_t)]/2\right) \quad (3)$$

where $\mathbf{Y}_a$ is the vector of the TDOAs estimated from the audio data, and $\hat{\tau}(\mathbf{X}_t)$ is the vector of the TDOAs estimated from $\mathbf{X}_t$ and the known microphone array geometry.

In contrast to video observations, audio observations at time t correspond to one person. Hence, $p(\mathbf{Y}_a|\chi_k)$ for a non-talking individual k is quite small, so that the total data likelihood $p(\mathbf{Y}_t|\chi_k)$ approaches zero. Therefore we need a correction term as below,

$$p(\mathbf{Y}_a|\mathbf{X}_t) \propto p_c + (1-p_c) exp\left(-[\mathbf{Y}_a - \hat{\tau}(\mathbf{X_t})]^T \Sigma_a^{-1} [\mathbf{Y}_a - \hat{\tau}(\mathbf{X}_t)]/2\right) \quad (4)$$

where p_c is a small constant that makes the data likelihood a constant when the exponential term is almost zero. Then, the audio data likelihood of (4) does not affect the non-talking individuals.

Under the assumption that the audio and video observations are conditionally independent given $\mathbf{X_t}$, the joint data likelihood is finally described as below,

$$p(\mathbf{Y}_t|\mathbf{X}_t) \propto p(\mathbf{Y}_v|\mathbf{X}_t)p(\mathbf{Y}_a|\mathbf{X}_t)$$

3. JOINT MULTIPLE TARGET TRACKING

3.1. Importance Sampling for Multiple Objects

One of the most important issues in particle filter implementations is determining how to distribute particles to support real target states. Generally, the optimum selection is known as a posterior, so we use it here. Since we defined the state transition model and video and audio data likelihood models as Gaussian functions, we can describe the posteriori proposal function π_t analytically using a gradient and a Hessian at its mode [5, 8]. First, we define our proposal function as

$$\pi_t \propto p(\mathbf{X}_t|\mathbf{Y}_t, \mathbf{X}_{t-1}) \propto p(\mathbf{Y}_t|\mathbf{X}_t)p(\mathbf{X}_t|\mathbf{X}_{t-1}).$$

Let $L_y^a(\mathbf{X}_t)$,$L_y^v(\mathbf{X}_t)$, and $L_x(\mathbf{X}_t)$ denote the logarithms of $p(\mathbf{Y}_a|\mathbf{X}_t)$, $p(\mathbf{Y}_v|\mathbf{X}_t)$, and $p(\mathbf{X}_t|\mathbf{X}_{t-1})$, then

$$log(\pi) \propto L_y^a(\mathbf{X}_t) + L_y^v(\mathbf{X}_t) + L_x(\mathbf{X}_t).$$

Using a Taylor Series Expansion around its mode $\mathbf{X}_t$, the distribution of the proposal function is also a Gaussian distribution, which has its mean and variance as

$$m(\mathbf{X}_t) = \mathbf{X}_t + \Sigma(\mathbf{X}_t)\left(\bigtriangledown L_y^a(\mathbf{X}_t) + \bigtriangledown L_y^v(\mathbf{X}_t) + \bigtriangledown L_x(\mathbf{X}_t)\right)$$

$$\Sigma(\mathbf{X}_t) = -\left(\bigtriangledown^2 L_y^a(\mathbf{X}_t) + \bigtriangledown^2 L_y^v(\mathbf{X}_t) + \bigtriangledown^2 L_x(\mathbf{X}_t)\right)^{-1}.$$

3.2. Independent Partition Sampled Particle Filter

In spite of the well-defined proposal function, the higher dimension of the state vector required for multiple objects can decrease system performance. To overcome this problem, we used a independent partition particle filter (IPPF) [5, 8]. Table1 summarizes the particle filter tracking algorithm.

3.3. Current Speaker Detection

Before detecting the current speaker among tracked multiple people, we must determine whether or not the fragmented audio frame is silent. Then, the state S_t to describe who is speaking is determined from the least square error below,

$$S_t = \arg\min_k \left\{ c \sum_{n=1}^{P} \sum_{\substack{1<i<j \\ i<j<M}} (\tau_{ij} - \hat{\tau}_{ij}(\chi_k^{(n)}))^2 \right\}, k = 1, ..., K.$$

Here c is a normalization constant, and P is the number of particles for each object at time t.

Table 1. Algorithm for the independent partition particle filter

Define $\mathbf{X_t^i} = [\chi_1^i, \ldots, \chi_K^i]^T$
At time t=0 :
  Initialize all Particles with Initial values.
For time t > 0 :
  For k=1,..., K, χ_k^i
    Sample $\chi_k^i \sim \pi_k(\chi_k|\chi_k^i, Y_k)$
    Calculate the partition weight function $q_k^i(\chi_k^i)$,
    $q_k^i(\chi_k^i) = \pi_k(\chi_k|\chi_k^i, Y_k)$
    Normalize $q_k^{(i)}$ for each partition
    Re-sample χ_k^i with $q_k^{(i)}$ and re-index $\chi_{t-1,k}$
  For $X_t^{(i)}$:
    Calculate the importance weights using
    $w_t^{(i)} = w_{t-1}^{(i)} \frac{p(Y_t|X_t^{(i)})p(X_t^{(i)}|X_{t-1}^{(i)})}{\pi(X_t|X_{t-1}^{(i)}, Y_k^t)}$
    Normalize the importance weights $w_t^{(i)}$
    Re-sample the Particle $X_t^{(i)}$

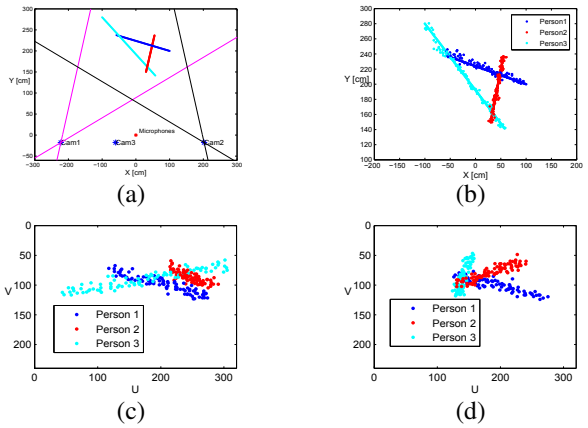
4. SIMULATIONS

To test the proposed algorithms, we equipped a conference room with three EVI-D30 cameras and six omni-microphones. Two static cameras, Cam1 and Cam2, were used for the tracking, but Cam3 was steered to track the final speaker. Fig. 2(a) shows the room configuration in the $x-y$ domain. The room size is 590 cm x 360 cm x 240 cm. Cam1 and Cam2 are aimed $-40°$ and $40°$ clockwise to cover the room as possible, but they have a physical limitation due to their limited field of view. The horizontal and vertical field of views of these cameras are $48.8°$ and $29°$, respectively. The center of the microphone array is the reference point $(0, 0, 0)$ of the whole 3D room coordinate system.

We simulated the proposed algorithms in two ways. The first used synthetic data generated using the actual camera calibration matrices and microphone geometry. In this test, we assumed there were three people moving inside of the tracking range in the 3D domain as in Fig. 2 (a). The positions of the people moving were converted into the 2D image coordinates, and then degraded with a $\sigma^2 = 20$ Gaussian noise to represent face detection inaccuracy in Fig. 2(c) and (d). Likewise, the positions of the people were also converted into TDOAs and degraded with a $\sigma^2 = 5$ uniform noise.

The final tracking result is shown in Fig. 2(b). The estimates follow the ground truth trajectories quite well, and the average difference between the tracked positions and the ground truth for all three people is less than $4\ cm$.

The next simulation used real data captured from the conference room. Two people were located at $(-59, 238, 51)$ and $(55, 238, 59)$. The individuals were not allowed to walk but could move their bodies. The face localization results for 13 sec are clustered around the ground truth in Fig. 4(a) and (b). The average difference between the tracked positions and



(a) (b) (c) (d)

Fig. 2. (a) Conference room configuration in the $x - y$ domain. (b) The final tracking results using the particle filter around the ground truth. The dots show the tracking results and solid lines shows the ground truth. (c) Video observations at Cam1. (d) Video observations at Cam2.

the ground truth for the two people is less than 7 cm. However, in these images it was difficult to localize face candidates because of the the small size of the faces, other skin area around the arms, and the background color similar to the skin color.

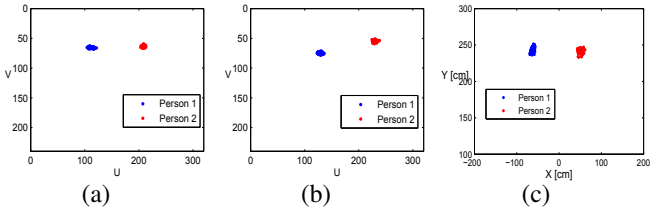
Finally, we checked the speaker detection results using the synthetic data and real data. In both synthetic and real data, the people took turns speaking. The result of speaker detection worked well except that their locations in the synthetic data were slightly biased in the direction of the other speaker.

(a) (b)

Fig. 3. (a) Camera1 image. (b) Camera2 image.

5. CONCLUSIONS

In this paper, we introduced multiple people tracking in the 3D world domain using a microphone array and multiple cameras, incorporating a particle filter. We proposed a 3D video data likelihood with a changeable camera calibration matrix and applied an independent partition particle filter. We also evaluated our proposed algorithm using synthetic and real data. However, our system is still preliminary. It has certain limitations and needs further improvements. First of all, we want to



(a) (b) (c)

Fig. 4. (a) Face localization result at Cam1. (b) Face localization result at Cam2. (c) The final tracking results using the particle filter, which are clustered around the ground truth.

change our video data measurement to use the whole image instead of a localized face candidates in order to remove errors from the face localization. Another requirement is to use a different feature for the person. Skin color has the strong restriction that people have to face the camera, which may not always be possible. The contour of the head and shoulders might be a more robust feature for multiple cameras.

6. REFERENCES

[1] P. Perez, J. Vermaak, and A. Blake, "Data fusion for visual tracking with particles," *Proceedings of the IEEE*, vol. 92, pp. 495–513, 2004.

[2] J. Vermaak, M. Gangnet, A. Blake, and P. Perez, "Sequential monte carlo fusion of sound and vision for speaker tracking," in *Proc. IEEE Intl. Conf. on Computer Vision*, 2001.

[3] D. N. Zotkin, R. Duraiswami, and L. S. Davis, "Joint audio-visual tracking using particle filters," *EURASIP Journal on Applied Signal Processing*, vol. 11, pp. 1154–1164, 2002.

[4] N. Checka, K. Wilson, V. Rangarajan, and T. Darrell, "A probabilistic framework for multi-modal multiperson tracking," in *Proc. of CVPR Workshop on Multi-Object Tracking*, 2003.

[5] M. Orton and W. Fitzgerald, "A bayesian approach to tracking multiple targets using sensor arrays and particle filters," *IEEE Trans. Signal Processing*, vol. 50, pp. 216–223, 2002.

[6] D. Stoyanov, "Camera calibration tool box and browsing," `http://www.vision.caltech.edu/bouguetj/calib_doc/`.

[7] Y. Huang and M. Barkat, "Near-field multiple source localization by passive sensor array," *IEEE Trans. on Antennas and Propagation*, vol. 39, pp. 968–975, 1991.

[8] V. Cevher, *A Bayesian Framework for Target Tracking using Acoustic and Image Measurements*, Ph.D. thesis, Georgia Institute of Technology, Atlanta, GA, 2005.