DATA ASSOCIATION FOR PEOPLE TRACKING USING MULTIPLE CAMERAS

Yeongseon Lee and Russell Mersereau

Georgia Institute of Technology
Atlanta, GA 30332-0250

ABSTRACT

In this paper, we present a data association algorithm for people tracking in a 3D world using multiple cameras. Our approach expands an independent partitioned particle filter with a data association vector. For the association parameter, we propose a proposal function using likelihood functions based on color and distance. This proposed algorithm solves the data association problem without dramatically increasing the computational complexity even in the case of trajectories that cross.

***Index Terms*—** Particle filter, data association, visual tracking, IPPF, proposal function

1. INTRODUCTION

A data association problem for unlabeled measurements always presents in single or multi-target tracking. In multi-target tracking, there is inevitable ambiguity even in the absence of clutter. Clutter complicates the problem considerably because we also have to deal with the problem of missing data and false alarms.

While some researches [1, 2] assume that measurements for each target are already known, others try to solve the data association problem with unlabeled measurements [3,4]. The most popular approaches are a multiple hypothesis filter (MHT) and a joint probability data association filter(JPDAF) [3]. The MHT approach attempts to advance all possible association hypotheses including new targets, through a measurement-to-target association, but this approach suffers from the complexity caused by the increasing number of possible association hypotheses with time. Since JPDAF, on the other hand, associates targets to measurements, it cannot deal with new targets, but it does generate a more feasible number of hypotheses. The final decision regarding which method is more applicable to a specific application depends mainly on the numbers of new targets, false alarms, etc.

Recently, a particle filter [1, 2] has been popular for target tracking because of its non-linear and non-Gaussian structure, but this particle filter also cannot inherently solve the data association problem. Moreover, in multiple target tracking, a standard particle filter also suffers because of the increased size of a target space of multiple targets. When this particle filter with a large state space has to perform data association, it requires so many particles to maintain the tracking that it cannot perform real-time tracking.

Our approach implements data association using particle filter, but to overcome the curse of dimensionality, we use independent partitioned particle filter proposed by Orton [5]. To perform the data association, we use target-to-measurement association similar JPDAF, but allow it to share one measurement among multiple targets based on the assumption that data association is independent among targets. In addition, we propose a proposal function for the data association parameter that uses a color-based data likelihood in addition to a distance-based data likelihood to prevent performance degradation when targets cross.

2. BASIC PARTICLE FILTER FRAMEWORK

A particle filter is a method to implement a sequential Bayesian estimation. The Bayesian estimation [2] is to do a statistical inference using probabilistic models. In this Bayesian inference, all variables are treated as random variables. It assumes that the true states are hidden. Then it estimates a posterior probability given measurements and prior information. The particle filter estimates the posterior probability using discrete samples called particles. This estimation occurs in time domain through target dynamics and a data likelihood.

2.1. Target dynamics and data likelihood

The target state of interest consists of 3D locations of people in a conference room. During the conference, people are assumed to sit on chairs without constant motion to a specific direction. Thus we define the target state and target dynamics as a random walk with Gaussian noise as follows:

$$\mathbf{X}_t = [\mathbf{x}_{1,t}, \ldots, \mathbf{x}_{k,t}, \ldots, \mathbf{x}_{K,t}] \tag{1}$$

$$p(\mathbf{X}_t|\mathbf{X}_{t-1}) \sim \mathcal{N}(\mathbf{X}_t|\mathbf{X}_{t-1}, \Sigma) \tag{2}$$

where $\mathbf{x}_{k,t} = [x_{k,t}, y_{k,t}, z_{k,t}]^T$ is the position of target k at time t, and K is the number of people in the room. Σ is chosen to cover possible movement of people.

For the data likelihood model, our measurements are defined as the locations of the centers of faces. We denote a measurement vector as $\mathbf{Y}_t$, consisting of all face candidates from M_o cameras at time t. Then $\mathbf{Y}_t$ can be written as a matrix as

$$\mathbf{Y}_t = [\mathbf{y}_t^1, \ldots, \mathbf{y}_t^i, \ldots, \mathbf{y}_t^{M_o}]. \quad (3)$$

Here, the measurements from camera i are $\mathbf{y}_t^i = [y_t^1, \ldots, y_t^{M^i}]^T = [(u_t^1, v_t^1), \ldots, (u_t^{M^i}, v_t^{M^i})]^T$, and M^i is the number of measurements from camera i.

Then the data likelihood function, $p(\mathbf{Y}_t|\mathbf{X}_t)$, is formed from a multivariate Gaussian density as

$$p(\mathbf{Y}_t|\mathbf{X}_t) \propto \exp^{\left(-\frac{1}{2}\left[\mathbf{Y}_t - \hat{\mathbf{Y}}_t(\mathbf{X}_t,\mathbf{P})\right]^T \Sigma^{-1}\left[\mathbf{Y}_t - \hat{\mathbf{Y}}_t(\mathbf{X}_t,\mathbf{P})\right]\right)}$$

where $\hat{Y}_t(\mathbf{X}_t, \mathbf{P})$ is a matrix of hypothesized measurements calculated with particles and camera calibration matrices, $\mathbf{P}$. The relationship between a point in 3D world coordinates, $\mathbf{x} = (x, y, z)^T$, and its corresponding 2D image coordinates, $y^i = (u^i, v^i)^T$, is described using the camera calibration matrix of the camera i, $\mathbf{p}^i$, as $\mathbf{y}^i = k\mathbf{p}^i\mathbf{x}$.

3. PARTICLE FILTER WITH DATA ASSOCIATION

The curse of dimensionality using a standard particle filter for multiple target tracking is well explained by Orton [5]. One solution is to use partitioned sampling [6]. Partitioned sampling is a generic term for a strategy that divides the state space into two or more partitions and sequentially generates particles for each partition followed by an appropriate weighted re-sampling operation. Orton partitioned each target of multiple targets as an individual partition and implemented a weighted re-sampling function, which he called IPPF. Our approach extends Orton's IPPF algorithm with a data association parameter in order to implement a posterior probability. In this section, we introduce an association parameter and then propose proposal functions for the association parameter and the target state to determine how to distribute new particles, which is the most important issue in particle filter.

3.1. Data association

To deal with the data association problem, we introduce a set of data association parameters as done in [3, 4]. As the association parameter we use a target to a measurement, based on the assumption that the number of targets has not changed, or that a new target is detected outside of the particle filter.

We denote the target-to-measurement hypotheses for measurements from M^o cameras at time t as $\tilde{\Lambda} = (\tilde{\lambda}^1, \ldots, \tilde{\lambda}^{M^o})$, where $\tilde{\lambda}_t^i = (\tilde{\mathbf{r}}^i, M_C^i, M_T^i)$ is the target-to-measurement hypotheses for the measurements from camera i, $\tilde{\mathbf{r}}^i$ is an association vector, and M_C^i and M_T^i are the number of clutter measurements and target measurements. The values of M_T^i and M_C^i can be extracted from $\tilde{\mathbf{r}}^i$. The association vector $\tilde{\mathbf{r}}^i = (\tilde{r}_1^i, \ldots, \tilde{r}_K^i)$ is given by

$$\tilde{r}_k^i = \begin{cases} 0 & \text{if target k is not detected} \\ j \in \{1, \ldots, M^i\} & \text{if target } k \text{ generates measurement j.} \end{cases}$$

Table 1. Standard particle filter with data association

For n= $1, \ldots, N$, Generate particles for $(\tilde{\Lambda}_t^{(n)}, \mathbf{x}_t^{(n)})$ $\propto q(\tilde{\Lambda}_t, \mathbf{X}_t|\mathbf{X}_{t-1}^{(n)}, \mathbf{Y}_t) = q(\tilde{\Lambda}_t|\mathbf{X}_{k,t}, y_t)q(\mathbf{X}_t|\mathbf{X}_{t-1}^{(n)}, \mathbf{Y}_t)$ For n= $1, \ldots, N$, $\omega_t^{(n)} \approx \omega_{t-1}^{(n)} \frac{p(\tilde{\Lambda}_t^{(n)})p(\mathbf{Y}_t|\mathbf{X}_t^{(n)}, \tilde{\Lambda}_t^{(n)})p(\mathbf{X}_t^{(n)}|\mathbf{X}_{t-1}^{(n)})}{q(\tilde{\Lambda}_t^{(n)}|\mathbf{X}_t^{(n)}, \mathbf{Y}_t)q(\mathbf{X}_t^{(n)}||\mathbf{X}_{t-1}^{(n)}, \mathbf{Y}_t)}$ $\sum_{n=1}^N \omega_t^{(n)} = 1$ If re-sampling is required, for $n = 1 \ldots N$, Sample an index $m(n) \approx \{\omega_t^{(l)}\}_{l=1}^N$ and replace $\{\omega_t^{(n)}, \mathbf{X}_t^{(n)}, \tilde{\Lambda}_t^{(n)}\} = \{N^{-1}, x_t^{(m(n))}, \tilde{\Lambda}_t^{m(n)}\}$

Then, $\tilde{\Lambda}_t$ is added into the target space of a standard particle filter. The final posterior probability, $p(\mathbf{X}_t, \tilde{\Lambda}_t|\mathbf{Y}_{1:t})$ is

$$p(\mathbf{X}_t, \tilde{\Lambda}_t|\mathbf{Y}_{1:t}) \propto p(\tilde{\Lambda}_t)p(\mathbf{Y}_t|\mathbf{X}_t, \tilde{\Lambda}_t) \int p(\mathbf{X}_t|\mathbf{X}_{t-1})p(\mathbf{X}_{t-1}|\mathbf{Y}_{1:t-1})d\mathbf{X}_{t-1}. \quad (4)$$

The association prior, $p(\tilde{\Lambda}_t)$, and the conditional data likelihood, $p(\mathbf{Y}_t|\mathbf{X}_t, \tilde{\Lambda}_t)$, are defined in [3] as follows:

$$p(\tilde{\Lambda}_t) = \prod_{i=1}^{M^o} \left(p(M_c^i) \prod_{k=1}^{K} p(\tilde{r}_k^i|\tilde{r}_{1:k-1}^i) \right) \quad (5)$$

with

$$p(\tilde{r}_k^i|\tilde{r}_{1:k-1}^i) \propto \begin{cases} 1 - P_D & \text{if j = 0} \\ \frac{P_D}{M_k^i} & \text{otherwise} \end{cases}$$

and

$$p(M_c^i) = (\lambda_c^i)^{M_c} \exp(-\lambda_c^i)/M_c^i!,$$
$$M_i^i = M^i - |l : \tilde{r_l}^i = 1, \ldots, k-1|,$$

$$p(\mathbf{Y}_t|\mathbf{X}_t, \tilde{\Lambda}_t) = \prod_{i=1}^{M^o} \left((V^i)^{-M_c^i} \prod_{k=1}^{K} p^i(y_{\tilde{r}_k}^i|\mathbf{x}_k) \right) \quad (6)$$

with

$$p^i(y_{\tilde{r}_k}^i|\mathbf{x}_k) \propto \begin{cases} 1 & \text{if } \tilde{r}_k^i = 0 \\ p_T^i(y_j^i|\mathbf{x}_k) & \text{if } \tilde{r}_k^i = j \in \{1, \ldots, M^i\} \end{cases} \quad (7)$$

where $p_T^i(y_{j,t}^i|\mathbf{x}_t)$ is defined as $N(y_j^i|\hat{y}_k^i(\mathbf{x}_{k,t}), \Sigma_y^i)$.

The straight implementation of this particle filter in Table 1 suffers from the curse of dimensionality and needs a large number of particles to maintain acceptable performance. For this reason, we, instead, use IPPF and extend it with data association.

3.2. IPPF with data association

IPPF assumes that each target is independent. We, in addition, assume that the data association is also independent among the targets. In this case, one measurement can be assigned into two targets with a probability. Since this can cause an incorrect association for very closely located targets, we expand the proposal function for the association vector with a color similarity-based data likelihood.

Fig. 1. Black and white boxes show approximate windows.

3.2.1. *Proposal function of data association*

Here we define a proposal function for the data association vector for a target k, $\tilde{r}_{k,t}$, given $\mathbf{x}_{k,t}$ and y_t^i. Basically, the proposal function of the data association vector depends upon the distance between a hypothesized measurement and a real measurement in (8) because $p_T^i(y_{j,t}^i|\mathbf{x}_t)$ is $N(y_j^i|\hat{y}_k^i(\mathbf{x}_{k,t}), \Sigma_y^i)$. This distance-based proposal function generally works well, but accuracy is greatly degraded when the measurements from multiple targets are close or overlapping. To deal with this issue, we add color similarity to a proposal function for the association vector.

$$q(\tilde{r}_{k,t}^i = j|\mathbf{x}_{k,t}, y_t^i) = q(y_t^i|\tilde{r}_k^i = j, \mathbf{x}_{k,t})q(\tilde{r}_{k,t}^i = j) \quad (8)$$

with

$$q(y_t^i|\tilde{r}_k^i = j, \mathbf{x}_k) = \begin{cases} V^{-M^i}, & \text{if } j = 0 \\ V^{-(M^i-1)} P_T(y_{j,t}^i|\mathbf{x}_{k,t}) & \text{if } j \in \{1 \dots M^i\} \end{cases}$$

$$q(\tilde{r}_{k,t}^i = j) = \begin{cases} 1 - P_D, & \text{if } j = 0 \\ \frac{P_D}{K} & \text{if } j \in \{1 \dots M^i\}. \end{cases}$$

This color similarity is calculated using a Bhattacharyya coefficient [7] from the approximate upper-body areas from each measurement y_t^i at camera i and the current target state, $\mathbf{x}_{k,t}^{(n)}$. However, since calculating histograms for all particles is relatively expensive, we calculate a histogram from $\mathbf{x}_{k,t-1}$ instead of $\mathbf{x}_{k,t}^{(n)}$.

$$d(h_1(y_t^i), h_2(\hat{y}(\mathbf{x}_{k,t-1}, \mathbf{p}^i))) = \sqrt{1 - \Sigma_u \Sigma_v (h_1(u,v) h_2(u,v))} \quad (9)$$

where $h_1(y_k^j)$ and $h_2(\hat{y})$ are normalized color histograms of two windows chosen to cover the upper bodies from y_t^i and $\hat{y}(\mathbf{x}_{k,t-1})$. In Figure 1, black boxes are from $\hat{y}(\mathbf{x}_{k,t-1})$ of target k, and white boxes are from measurements. The color similarity is calculated between a black box and a white box using (9). The proposal function based on the color similarity is given by

$$q(y_t^i|\tilde{r}_k^i = j, \mathbf{x}_k) \propto exp(-\frac{d(h_1(y_t^i = j), h_2(\hat{y}(\mathbf{x}_{k,t-1}, \mathbf{p}^i)))}{2\sigma^2}). \quad (10)$$

where σ is empirically determined. Therefore, the final proposal function to generate data association particles consists of both (8) and (10).

Table 2. IPPF with data association

For $k = 1 \dots K, n = 1 \dots N$,
    Generate particles for target states
    $\mathbf{x}_{k,t}^{(n)} \propto q(\mathbf{x}_{k,t}^{(n)}|\mathbf{x}_{k,t-1}^{(n)}, \mathbf{Y}_t)$
For $k = 1 \dots K, i = 1 \dots N_o, n = 1 \dots N$,
    Generate particles for target-to-measurement association
    $\tilde{r}_{k,t}^{i(n)} \propto q(\tilde{r}_{k,t}^{i(n)}|, \mathbf{x}_{k,t}^{(n)}, \mathbf{y}_t^i)$
For $k = 1 \dots K, n = 1 \dots N$,
    Compute and normalize individual target weights
    $\alpha_{k,t}^{(n)} \propto \frac{p_k(x_{k,t}^{(n)}|\mathbf{x}_{k,t-1}^{(n)})}{q_k(\mathbf{x}_{k,t}^{(n)}|\mathbf{x}_{k,t-1}^{(n)}, \mathbf{Y}_t)} \prod_{i=1}^{N_o} \frac{p^i(\mathbf{y}^i_{\tilde{r}_{k,t}^{i(n)}}|\mathbf{x}_{k,t}^{(n)}) p(\tilde{r}_{k,t}^{i(n)})}{q(\tilde{r}_{k,t}^{i(n)}|\mathbf{x}_{k,t}^{(n)}, \mathbf{y}_t^i)}$
    $\alpha_{k,t}^{(n)} = \sum_{n=1}^{N} \alpha_{k,t}^{(n)} = 1$
    Sample an index $m_k(n) \propto \{\alpha_{k,t}^{(l)}\}_{l=1}^{N}$ and replace
$\{x_{k,t}^{(n)}, \{\tilde{r}_{k,t}^{i(n)}\}_{i=1}^{N_o}\} \leftarrow \{x_{k,t}^{m_k(n)}, \{\tilde{r}_{k,t}^{im_k(n)}\}\}$
For $n = 1 \dots N$, Compute and normalize particle weights
    $\omega_t^{(n)} \approx \omega_{t-1}^{(n)} \frac{p(\tilde{\Lambda}_t^{(n)}) p(\mathbf{Y}_t|\mathbf{X}_t^{(n)}, \tilde{\Lambda}_t^{(n)}) p(\mathbf{X}_t^{(n)}|\mathbf{X}_{t-1}^{(n)})}{q(\tilde{\Lambda}_t^{(n)}|\mathbf{X}_t^{(n)}, \mathbf{Y}_t) q(\mathbf{X}_t^{(n)}|\mathbf{X}_{t-1}^{(n)}, \mathbf{Y}_t)}$
If re-sampling is required, sample an index $m(n) \propto \{\omega_t^{(l)}\}_{l=1}^{N}$
and replace $\{\omega_t^{(n)}, x_t^{(n)}\} \leftarrow \{N^{-1}, x_t^{m(n)}\}$

3.2.2. *Proposal function of a target state*

The optimal proposal function for a target state is its posterior probability [2]. However, since the relationship between $\mathbf{x}_{k,t}$ and $\mathbf{Y}_t$ is not known, we cannot derive the proposal function of a target state without an association vector. However, if we define it as a mixture of posteriors of all measurements and a target, then we remove the association vector for it. After extended with a state transition to cover the case of missing data, the final proposal function is as follows:

$$q_k \propto \alpha_1 p(\mathbf{x}_{k,t}|\mathbf{x}_{k,t-1}) + (1 - \alpha_1) \sum_{i=1}^{M^0} \sum_{j=1}^{M^i} p_T(y_{j,t}^i|\mathbf{x}_{k,t}) p(\mathbf{x}_{k,t}|\mathbf{x}_{k,t-1}).$$

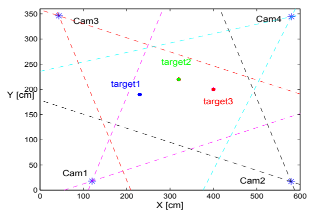
Here, the mixture coefficient, α_1, is empirically determined.

This mixture of posteriors of all measurements sometimes requires to increase the number of particles when they have too many measurements. Then, we use a gating [3] to delete distributions that have measurements far from the hypothesized measurement.

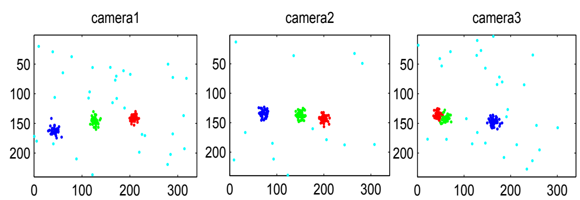
4. SIMULATIONS

For our simulation, we equipped a conference room with four EVI-D30 cameras. The size of the room is 5.9 m x 3.6 m x 2.4 m. All cameras are steered to cover the center of the room as much as possible shown as Figure 2(a), but the tracking area is still limited due to their field of view. All the cameras were calibrated off-line. The size of the images captured from the cameras is 320x240. One set of sample images captured from three cameras is shown in Figure 1.

Because of the difficulty of showing results using real data, we show results using synthetic data generated using real camera calibration matrices and real image size. The parameters used here are $P_d = 0.98$ for the detection rate, $\lambda_c = 0.1$ for the Poisson clutter process in (5), $\sigma^2 = 20$ pixel for measurement noise. All simulations are evaluated with a mean absolute difference (MAD) between the true target states and estimated target states with 30 repetitions of Monte Carlo simulation. The number of particles is 100 for each target. All target states in the simulation has $3D$ values, but here, we depict only the $x - y$ coordinates.



(a)

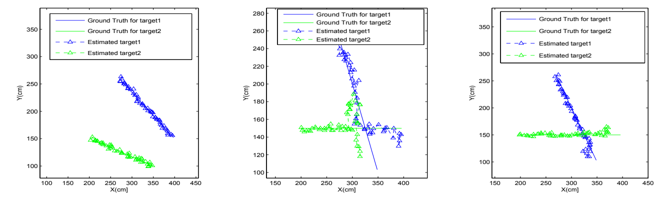


(b)

Fig. 2. Conference room in the $x - y$ domain with (a) three people, (b) measurements from three cameras. Each blue, green, and magenta, cyan dots correspond to target 1, target 2, target 3, and false data for 50 images.

First, we applied our proposed algorithm to a scenario of static people as shown in Figure 2(a). In this scenario, three people sit in chairs and talk. Surprisingly, while the MAD using two cameras is 2.7 cm, the MAD using three cameras is 3.6 cm. This contradicts the general assumption that tracking performance is better when we have more sensors. However, The reason becomes clear when we see the measurements from each camera in Figure 2(b). In camera 3, the measurements of target 1 and 2 are very close. When the measurements for two targets are too close or overlapping, many of the association vectors of target particles for target 1 associate measurements from target 2 and vice versa. This causes the final tracking location to go other target location. This becomes much clearer when we run the simulation with non-crossing moving targets and crossing moving targets.

Figure 3 shows the results of two scenarios of non-crossing targets and crossing targets. In Figure 3(a) for non-crossing targets, the MAD using two cameras is 6.6 cm, and the MAD using three cameras is 3.1 cm. For the target crossing scenario, however, the proposal function based on a distance cannot associate the true data relations, so the tracking after crossing is incorrect as in Figure 3(b). However, when we add color-similarity to the proposal function as (10), this is prevented as shown in Figure 3(c). The Bhattacharyya coefficient, d, is 0.5 for the true association and 0.8 for the incorrect association for the simulation of Figure 3(c).



(a) (b) (c)

Fig. 3. A simulation result. (a) not-crossing targets scenario using (8), (b) two crossing target scenario using (8), (c) two crossing target scenario using (8) and (10).

5. CONCLUSIONS

In this paper, we proposed data association algorithm for a particle filter using multiple cameras. We expanded Orton's IPPF with a data association vector and proposed a proposal function for the data association parameter with color-based data likelihood. This proposed algorithm can solve the data association problem from unlabeled measurement for multiple target tracking, without dramatically increasing computational complexity.

6. REFERENCES

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

[2] A. Doucet, N. Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*, Springer-Verlag, 2001.

[3] J. Vermaak, S. J. Godsill, and P. Perez, "Monte carlo filtering for multi-target tracking and data association," in *Aerospace and Electronic Systems, IEEE Transactions on*, Jan. 2005, vol. 41, pp. 309 – 332.

[4] Y. Bar-Shalom and T. E. Fortman, *Tracking and Data Association*, Mathematics in Science and Engineering, 1988.

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

[6] J. MacCormick and Michael Isard, "Partitioned sampling, articulated objects, and interface-quality hand tracking," in *ECCV*, 2000, pp. 3–19.

[7] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification, second edition*, A Wiley-interscience Publication, 2001.