TRACKING OF MULTIPLE OBJECTS USING THE PHD FILTER

PHAM NAM TRUNG

(B.Sc., University of Natural Science, Ho Chi Minh City, Vietnam)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2007

Acknowledgements

I would like to express my deep and sincere gratitude to my supervisors, Professor Sim Heng Ong, Dr. Weimin Huang, Dr. Jian Kang Wu, and Dr. Tele Tan. Their wide knowledge has been of great value for me. Their understanding, encouraging and personal guidance have provided contributions for the present thesis.

I am deeply grateful to Professor Wing Kin Ma for his program codes on multiple-speaker tracking.

I warmly thank Mr. Ya Dong Wang for his valuable discussions during the time I spent at Institute for Infocomm Research.

I also wish to thank my friends at Singapore, my family for their sympathising and encouraging me to finish this work.

I owe my loving thanks to my girl friend Nguyen Thi Kim Tuyen for her encouraging, understanding, and loving support when I am studying abroad in Singapore.

I am grateful to both National University of Singapore and Institute for Infocomm Research for their generous financial assistance during my postgraduate study.

I would like to give my thanks for using the facility of Star Home [1] for data capturing of testing.

Last but not least, I gratefully acknowledge the support that was provided under EU project ASTRALS (FP6-IST-0028097).

Abstract

The random set approach opens a new direction for multiple-sensor multiple-object tracking. All aspects related to objects such as appearing, disappearing, moving, measurements, and clutter can be modeled by random finite sets. The probability hypothesis density (PHD) filter, proposed by Mahler, operates on a single-object state space and avoids the data association problem. Multiple-object tracking is thus made more practical but we need to formulate the problem under the random finite set framework to use the PHD filter in applications. These formulations are not straight-forward.

In this thesis, we investigated methods based on the PHD filter for multiple-object tracking. The contributions of this thesis include:

1) Proposing a method to maintain the track continuity in the PHD filter in Chapter 4. The method can be used to track multiple objects in applications with high density of clutter and varying number of objects that traditional methods such as JPDA or MHT find difficult to handle because of the computational complexity.

2) Giving an efficient method for multiple-speaker tracking using the PHD filter in Chapter 5. Our method is less computational and more reliable than some methods for multiple-speaker tracking. The proposed method is efficient for real time tracking of multiple speakers in a reverberant room.

3) Improving the performance of multiple-object tracking in video by using the PHD filter in Chapter 6. A PHD recursion for visual observations with color measurements is proposed. With this approach, the video tracking can work for varying number of objects in single-object state space. Moreover, we extend the method in multiple-camera multiple-object tracking with good performance in Chapter 7.

The experimental results in this thesis show that the PHD filter is a promising approach for multiple-object tracking applications.

List of Tables

List of Figures

Contents

Chapter 1

Introduction

1.1 Motivation

Object tracking is an important part of many applications, such as sports analysis [28], [72], surveillance [91], smart room [70], robot control [69], human computer interaction [12], and video conferencing [27]. It allows us to determine the states of objects and helps us in analyzing their behaviors. Because of the importance of object tracking, there are many researchers working in this area. Some of them have proposed approaches for tracking a single object [11], [83], [110]. However, in many applications, there are more than one object. There are many approaches for multiple-object tracking. Traditional approaches are based on data association between objects and measurements such as time delay of arrival, and range from sensors [6], [80]. If this data association is known in advance, the problem of

multiple-object tracking becomes one of tracking independent single objects. Otherwise, we need to consider the data association problem. This is because when the data association is not correct, the state estimates are not reliable. There are some approaches to data association problem such as multiple hypothesis tracking (MHT) [80] and joint probabilistic data association (JPDA) [6], [85]. However, the determination of association probabilities in these methods is an NP-hard problem [71].

There has been increasing research interest on using the random set theory for multiple-object tracking [37], [63]. In the random set approach, the states of objects, measurements, and clutter are modeled by random sets. Mahler [63] presented a probability hypothesis density (PHD) filter for multiple-object tracking by using the random set framework. This method operates on a single-object state space and avoids the combinatorial problem that arises from the data association between objects and measurements. Thus, the computation of the PHD filter is less than traditional methods such as MHT, and JPDA. The low cost of the computation in the PHD filter makes the random set approach more promising for multiple-object tracking applications.

In this thesis, we focus on multiple-object tracking by using the random set approach and concentrate on practical issues when using the PHD filter in applications. Through experimental results on applications, we also show that the PHD filter can handle non-trivial tasks in multiple-object tracking such as data

association, varying number of objects, multiple-sensor data fusion, and clutter handling.

1.2 Major contributions

The major contributions of this thesis are to develop methods using the PHD filter for multiple-object tracking in several directions:

- A reliable method to maintain the track continuity in the PHD filter.

  A reliable method to maintain the track continuity in the PHD filter is proposed. The track continuity is determined by the Hungarian algorithm. This is an exact method to determine the data association between tracks of the previous and the current time steps. The proposed method is more reliable than using heuristic methods to maintain track continuity. This method might be important for multiple-object tracking with high density of clutters and varying number of objects that traditional methods such as MHT and JPDA find difficult to handle. This is because the computational complexity of MHT and JPDA are known as NP-hard while the computational complexity of the PHD filter is $O(|Z| \cdot N)$, where $|Z|$ is the maximum number of measurements, and N is the maximum number of Gaussian components (for the Gaussian mixture PHD filter) or number of samples (for the particle PHD filter).

- An efficient method for multiple-speaker tracking

  An efficient technique for real-time tracking of multiple speakers in a reverberant room is proposed. To have an efficient method for multiple-speaker tracking, fusing measurements from microphone pairs with low cost of computation and high performance is a critical and challenging step. In this thesis, we fuse the time delay of arrival measurements in the Gaussian mixture probability hypothesis density filter. The method is more reliable and computationally tractable than some methods for multiple-speaker tracking. Moreover, our approach can be applied to other multiple-sensor multiple-object applications such as bearing and range tracking, and multiple-camera multiple-object tracking.

- A method using the PHD filter for color object tracking.

  When using the PHD filter, representing measurements as random sets is a mandatory step. Unfortunately, representing color measurements as random sets is a difficult task. We propose a method to obtain the color measurement random set and apply it in the PHD filter for color object tracking. It tracks multiple objects with video data in single-object state space. Moreover, it may be important in other applications, such as track-before-detect, where it is difficult to obtain a measurement random set.

- A method for multiple-camera multiple-object tracking using the PHD filter.

Tracking multiple objects in a multiple-camera environment is a challenging task. This is because of the data association of objects among cameras in the high dimensional state space. We propose a method for multiple-camera multiple-object tracking using the PHD filter. This method can track 3D object locations even when objects are undergoing complex interaction with multiple occlusions and merge-split in groups. Moreover, it avoids data association and tracks multiple objects in single-object state space. In the proposed method, both temporal and visual information are considered.

1.3 Organization of thesis

The organization of this thesis is as follows.

- Chapter 2: A literature review on object tracking by filtering approaches is given. This chapter discusses fundamentals of object tracking and data association for multiple-object tracking.

- Chapter 3: This chapter contains an introduction on the PHD filter. Random set formulations of multiple-object tracking are described. Implementations of the PHD filter such as the particle PHD filter, the Gaussian mixture PHD filter are also discussed.

- Chapter 4: The method for maintaining the track continuity in the PHD

filter is presented. Simulation resutls to show the efficiency of the method is detailed.

- Chapter 5: The method for multiple-speaker tracking and the implementation issues are described. Simulation results and comparisons between our method and others are shown to demonstrate the efficiency of the method.

- Chapter 6: A technique for tracking multiple objects by using the PHD filter and color measurements is proposed. Steps to obtain the color measurement random set and the implementation of the method are described.

- Chapter 7: A multiple-camera multiple-person tracking using the random set approach is presented. The method includes two stages of single-view tracking and multiple-camera fusion. These two stages are described and promising experimental results of the proposed method are shown.

- Chapter 8: This chapter summarizes contributions of our research and discusses future work.

Chapter 2

Review of object tracking

2.1 Introduction to object tracking

Tracking is the processing of measurements obtained from objects such as color [28], contour [11], time delay of arrival [112], bearing and range [81] to obtain the estimations of unknown object kinematics or states. The unknown object kinematics of interest are usually the position, velocity, and acceleration of the object in an appropriate coordinate system. Some examples of tracking include radar tracking of military vehicles [9], tracking of people for monitoring [41], surveillance systems [90]. Figure 2.1 shows the components of a typical tracking process.

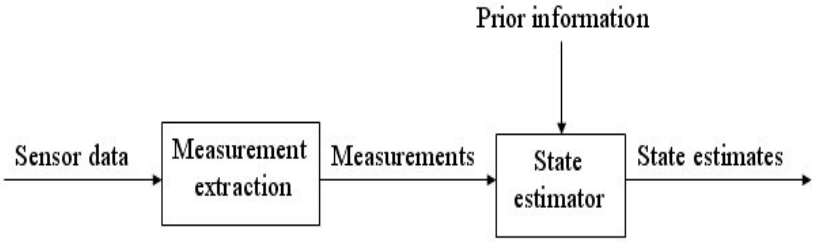

Figure 2.1: Typical components of an object tracking system

There are various techniques for object tracking. For example, some techniques transfer the tracking problem to a minimization problem by searching for the best match of the object with the previous estimation as an initialization [23], [34]. Some other object tracking techniques use neural networks [3], [59], fuzzy logic [94], and Bayes filter [47]. However, Bayesian theory remains the most widely accepted approach to object tracking.

2.2 Single-object tracking by the Bayes filter

We consider the scenario in which a single object is present. We assume that the state of object follows a Markov process on the state space $\mathcal{X} \subseteq \mathbb{R}^{n_x}$. Let x_k be the state of object at time k. The evolution of the state sequence of an object is given by

$$x_k = f_k(x_{k-1}, w_{k-1}) \tag{2.1}$$

where f_k is a transition function, w_{k-1} is the process noise. The probability density of a transition from x_{k-1} to x_k is $f_{k|k-1}(x_k|x_{k-1})$.

This Markov process is partially observed in the observation space $\mathcal{Z} \subseteq \mathbb{R}^{n_z}$, i.e., given a state x_k at time k, the observation of an object is

$$z_k = g_k(x_k, v_k) \tag{2.2}$$

where g_k is a measurement function, v_k is the measurement noise. The probability density of receiving the observation z_k given the state x_k is $g_k(z_k|x_k)$. It is also called the likelihood function.

The probability density of state x_k at time k given all observations $z_{1:k} = (z_1, \ldots, z_k)$ up to time k, denoted by

$$p_k(x_k|z_{1:k}), \tag{2.3}$$

is called the posterior density (or filtering density) at time k. This posterior density can be obtained by using the Bayes filter. This filter includes two steps. The first step is called the prediction step. From the posterior density at the previous time $k-1$, $p_{k-1}(x_{k-1}|z_{1:k-1})$, and the transition density $f_{k|k-1}(x_k|x_{k-1})$, we can obtain the predicted density $p_{k|k-1}(x_k|z_{1:k-1})$ via the prediction equation

$$p_{k|k-1}(x_k|z_{1:k-1}) = \int f_{k|k-1}(x_k|x_{k-1}) p_{k-1}(x_{k-1}|z_{1:k-1}) dx_{k-1} \tag{2.4}$$

The second part of the Bayes filter is the updating step. From the predicted density $p_{k|k-1}(x_k|z_{1:k-1})$ and the likelihood function $g_k(z_k|x_k)$, the posterior density can be

obtained by

$$p_k(x_k|z_{1:k}) = \frac{g_k(z_k|x_k)p_{k|k-1}(x_k|z_{1:k-1})}{\int g_k(z_k|x_k)p_{k|k-1}(x_k|z_{1:k-1})dx_k} \tag{2.5}$$

The state of object $\hat{x}_{k|k}$ can be estimated from the posterior density $p_k(x_k|z_{1:k})$ by taking either the maximum a posterior (MAP) estimation,

$$\hat{x}_{k|k}^{MAP} = \arg\max_{x_k} p_k(x_k|z_{1:k}) \tag{2.6}$$

or the expected a posteriori (EAP) estimation

$$\hat{x}_{k|k}^{EAP} = \int x p_k(x|z_{1:k})dx \tag{2.7}$$

The EAP estimation is also the minimum mean square error (MMSE) estimation of the state of object.

2.3 Kalman filter

In linear systems with Gaussian noises, the Bayes filter has a closed-form that is proposed by Kalman [4]. This closed-form is called the Kalman filter (KF). The Kalman filter assumes that the posterior density at every time step is a Gaussian and hence parameterized by a mean and a covariance matrix. In linear systems with Gaussian noises, if the posterior density at time $k-1$, $p_{k-1}(x_{k-1}|z_{1:k-1})$, is a Gaussian, then the posterior density at time k, $p_k(x_k|z_{1:k})$, is also a Gaussian. In

these cases, the state space model can be re-written as follows:

$$x_k = F_{k-1}x_{k-1} + w_{k-1} \tag{2.8}$$

$$z_k = H_k x_k + v_k \tag{2.9}$$

where F_{k-1} is a transition matrix, H_k is a measurement sensitive matrix, $w_{k-1} \sim \mathcal{N}(0, Q_{k-1})$ is the process noise and $v_k \sim \mathcal{N}(0, R_k)$ is the measurement noise. $\mathcal{N}(m, P)$ is a Gaussian density with mean m and covariance P.

The Kalman filter algorithm, derived by using (2.4) and (2.5), can be viewed as the following recursive relationship:

$$p_{k-1}(x_{k-1}|z_{1:k-1}) = \mathcal{N}\left(x_{k-1}; m_{k-1|k-1}, P_{k-1|k-1}\right) \tag{2.10}$$

$$p_{k|k-1}(x_k|z_{1:k-1}) = \mathcal{N}\left(x_k; m_{k|k-1}, P_{k|k-1}\right) \tag{2.11}$$

$$p_k(x_k|z_{1:k}) = \mathcal{N}\left(x_k; m_{k|k}, P_{k|k}\right) \tag{2.12}$$

where

$$m_{k|k-1} = F_{k-1}m_{k-1|k-1} \tag{2.13}$$

$$P_{k|k-1} = Q_{k-1} + F_{k-1}P_{k-1|k-1}F_{k-1}^T \tag{2.14}$$

$$m_{k|k} = m_{k|k-1} + K_k\left(z_k - H_k m_{k|k-1}\right) \tag{2.15}$$

$$P_{k|k} = P_{k|k-1} - K_k H_k P_{k|k-1} \tag{2.16}$$

and where $\mathcal{N}(x; m, P)$ is a Gaussian density with argument x, mean m and covariance P and

$$K_k = P_{k|k-1}H_k^T\left(H_k P_{k|k-1}H_k^T + R_k\right)^{-1} \tag{2.17}$$

In linear Gaussian systems, the Kalman filter is an optimal solution to the single-object tracking [4]. The implication is that no algorithm can do better than the Kalman filter in the linear Gaussian environment. If the system is not linear Gaussian, there are some extension from the Kalman filter such as the extended Kalman filter (EKF) [49] or the unscented Kalman filter (UKF) [109] that can be applied. The advantage of KF, EKF, UKF is the low computational burden. However, they are not appropriate when the posterior density is a binomal or multimodal probability density function.

2.4 Particle filter

The particle filter was first introduced by Gordon [38]. It is also known by different names, such as the condensation algorithm [46], the bootstrap filter [38], and the Monte Carlo filter [32]. The particle filter has been developed to solve non-Gaussian and non-linear problems. The key idea to solve non-Gaussian and non-linear problems is to represent the posterior density function by a set of random samples with associated weights and to compute estimates based on these samples and weights. When the number of samples goes to infinity, the particle filter will become the optimal Bayes filter. The proof of convergence is found in [25], [26]. The particle filter is briefly described as follows.

Assume at time $k-1$, a set of weighted particles $\left\{w_{k-1}^{(i)}, x_{k-1}^{(i)}\right\}_{i=1}^{N}$ representing

the posterior density $p_{k-1}(x_{k-1}|z_{1:k-1})$, i.e.,

$$p_{k-1}(x_{k-1}|z_{1:k-1}) \approx \sum_{i=1}^{N} w_{k-1}^{(i)} \delta_{x_{k-1}^{(i)}}(x_{k-1}) \tag{2.18}$$

The particle filter proceeds to approximate the posterior density $p_k(x_k|z_{1:k})$ by a new set of weighted particles $\left\{w_k^{(i)}, x_k^{(i)}\right\}_{i=1}^{N}$ as given in Figure 2.2.

At time $k \geq 1$,

Step 1: *Sampling Step*

For $i = 1, ..., N$

Sample $\tilde{x}_k^{(i)} \sim q_k\left(\cdot|x_{k-1}^{(i)}, z_k\right)$

Set $\tilde{w}_k^{(i)} = \dfrac{g_k\left(z_k|\tilde{x}_k^{(i)}\right) f_{k|k-1}\left(\tilde{x}_k^{(i)}|x_{k-1}^{(i)}\right)}{q_k\left(\tilde{x}_k^{(i)}|x_{k-1}^{(i)}, z_k\right)} w_{k-1}^{(i)}$

EndFor

Normalized weights: $\sum_{i=1}^{N} \tilde{w}_k^{(i)} = 1$.

Step2: *Resampling Step*

Resample $\left\{\tilde{w}_k^{(i)}, \tilde{x}_k^{(i)}\right\}_{i=1}^{N}$ to get $\left\{w_k^{(i)}, x_k^{(i)}\right\}_{i=1}^{N}$

Figure 2.2: Particle filter

The problem with the particle filter is that after a few time steps, one particle will dominate the particle cloud. To prevent this problem, we have to resample particles to guarantee that there is no sample having zero or very small probability. Another approach is to choose an importance sampling density $q_k\left(\cdot|x_{k-1}^{(i)}, z_k\right)$.

Having a good importance sampling density is critical for the performance of the particle filter. There are some related works, such as the unscented particle filter [67], the boosted particle filter [72], and the Markov chain Monte Carlo [36].

2.5 Multiple-object tracking

The formulations of single-object tracking in Section 2.2 can be extended to multiple-object tracking. Let $M(k)$ be the number of objects at time k, and $N(k)$ is the number of received measurements. The set of object states and measurements at time k can be denoted by

$$X_k = \left\{x_{k,1}, x_{k,2}, ..., x_{k,M(k)}\right\} \tag{2.19}$$

$$Z_k = \left\{z_{k,1}, z_{k,2}, ..., z_{k,N(k)}\right\} \tag{2.20}$$

We assume that each object moves according to the Markov dynamic model and generates observations according to Equations (2.1) and (2.2). There are two challenges in multiple-object tracking. The first challenge is the varying number of objects. In a tracking scenario, the number of objects can be time-varying, so the tracking algorithm has to detect the change of the number of objects, and automatically track new objects. The second challenge is the data association between measurements and objects. The data association problem can be defined as follows:

Let $\theta = \{\theta_{j,i},\ j = 1, ..., N(k), i = 1, ..., M(k)\}$ denote the association events between objects and measurements, where $\theta_{j,i}$ is the particular event which assigns measurement $z_{k,j}$ to the ith object. When the ith object moves, if we know in advance which measurement $z_{k,j}$ is originated from the ith object, then multiple-object tracking becomes independent tracking of single object. However, in applications, we do not know or are difficult to know this association.

Let Ω be the space of all possibilities for the data association θ. Each of $\theta \in \Omega$ is called a hypothesis association. The multiple-object tracking algorithm try to find the best hypothesis. The large number of possibilities for the data association affects the time for running tracking algorithms. Hence, data association is a challenge to multiple-object tracking.

Two famous approaches to multiple-object tracking are the multiple hypothesis tracking [80] and the joint probabilistic data association [6], [7]. Besides, there are approaches to multiple-object tracking, such as PMHT [92], [93], sequential Monte Carlo methods [44], and jump Markov model [33]. Random set approaches to multiple-object tracking have also attracted increasing attentions [19], [37], [61], [63], [102].

2.6 Multiple hypothesis tracking

The multiple hypothesis tracking [80] (MHT) tries to find the best hypothesis association between measurements and tracks. MHT does not need assumptions on the number of objects. Thus, it can track a varying number of objects at each time step. The idea of the method is based on enumerating all possible hypotheses over the number of most recent frames and choosing the most likely one.

Let $\Omega^k \triangleq \left\{\Omega_1^k, ..., \Omega_{I(k)}^k\right\}$ be the set of all association hypotheses at time k. For each Ω_m^{k-1}, a set of associations $\Psi^k \triangleq \left\{\Psi_1^k, ..., \Psi_{J(k)}^k\right\}$ between Z_k and X_k is defined. Hence, an association hypothesis Ω_i^k is the combination between an previous association hypothesis Ω_m^{k-1} and Ψ_n^k, i.e.,

$$\Omega_i^k = \left\{\Omega_m^{k-1}, \Psi_n^k\right\} \tag{2.21}$$

The probability of hypothesis Ω_i^k is $p\left(\Omega_i^k|Z_k\right)$ given by

$$\begin{aligned} p\left(\Omega_i^k|Z_k\right) &= p\left(\Omega_m^{k-1}, \Psi_n^k|Z_k\right) \\ &= \frac{1}{c} p\left(Z_k|\Omega_m^{k-1}, \Psi_n^k\right) p\left(\Psi_n^k|\Omega_m^{k-1}\right) p\left(\Omega_m^{k-1}\right) \end{aligned} \tag{2.22}$$

where c is a normalization constant. The first term $p\left(Z_k|\Omega_m^{k-1}, \Psi_n^k\right)$ can be derived by multiplications of likelihood function $g_k\left(z_k|x_k\right)$ and the clutter density. The second term is obtained from assumptions on the probability of detection, the clutter density, and the birth density. The last term is from previous step $k-1$, $p\left(\Omega_m^{k-1}|Z_{k-1}\right)$.

The disadvantage of MHT is computational expense due to the number of hypotheses growing exponentially over time. The computation of MHT is known to be NP-hard. Hence, there are some heuristic methods to reduce the number of hypotheses, such as gating [24], [80], N-scan pruning [10], PMHT [92], [93], clustering [108], and fast association technique [77]. However these heuristic methods are used at the expense of optimality and the algorithms can still suffer in a dense environment. Furthermore, the running time at each step of the algorithm cannot be bounded easily, making it difficult to deploy in a real-time system. Some examples of using MHT in applications are in [14], [40], [78].

2.7 Joint probabilistic data association

The joint probabilistic data association (JPDA) filter [6], [7] tries to calculate the state estimation based on the expectation of hypothesis associations beteen measurement and objects. JPDA assumes that there is a known number of objects. While MHT is concerned with the accumulated data set, JPDA aims to find an association between measurements and objects at the current time step based on enumerating all possible associations and computing the association probability.

The joint association event probability is

$$\begin{aligned} p\left(\theta|Z_{1:k}\right) &= p\left(\theta|Z_k, N(k), Z_{1:k-1}\right) \\ &= \frac{1}{c} p\left(Z_k|\theta, N(k), Z_{1:k-1}\right) p\left(\theta|Z_{1:k-1}, N(k)\right) \\ &= \frac{1}{c} p\left(Z_k|\theta, N(k), Z_{1:k-1}\right) p\left(\theta|N(k)\right) \end{aligned} \tag{2.23}$$

where c is a normalization constant. The first term $p\left(Z_k|\theta, N(k), Z_{1:k-1}\right)$ is the likelihood function of the measurements, given by

$$p\left(Z_k|\theta, N(k), Z_{1:k-1}\right) = p_{fa}^{\phi} \prod_{\theta_{ij}\in\theta} g_k\left(x_{k,i}|z_{k,j}\right) \tag{2.24}$$

where p_{fa} is the probability density of false-alarms, ϕ is the number of false-alarms. The second term $p\left(\theta|N(k)\right)$ is the prior probability of a joint association event, given by

$$p\left(\theta|N(k)\right) = p_D^{N(k)-\phi}\left(1-p_D\right)^{M(k)-(N(k)-\phi)} \mu_F\left(\phi\right) \frac{\phi!}{N(k)!} \tag{2.25}$$

where p_D is the probability of detection of an object, $\mu_F\left(\phi\right)$ is the probability of number of false alarms. Thus the probability of a joint association event is

$$p\left(\theta|Z_{1:k}\right) = \frac{1}{c}\frac{\phi!}{N(k)!} p_D^{N(k)-\phi}\left(1-p_D\right)^{M(k)-(N(k)-\phi)} \mu_F\left(\phi\right) p_{fa}^{\phi} \prod_{\theta_{ij}\in\theta} g_k\left(x_{k,i}|z_{k,j}\right) \tag{2.26}$$

The association probability for a particular association between measurement $z_{k,j}$ to the ith object is defined by

$$\begin{aligned} \beta_{j,i} &= p\left(\theta_{j,i}|Z_{1:k}\right) \\ &= \sum_{\theta:\theta_{j,i}\in\theta} p\left(\theta|Z_{1:k}\right) \end{aligned} \tag{2.27}$$

The state estimation of the ith object is

$$\begin{aligned} \hat{x}_{k,i} &= E\left(x_{k,i}|Z_{1:k}\right) \\ &= \sum_{j=1}^{N(k)} E\left(x_{k,i}|\theta_{j,i}, Z_{1:k}\right)\beta_{j,i} \\ &= \sum_{j=1}^{N(k)} \hat{x}_{k,i}^{j}\beta_{j,i} \end{aligned} \tag{2.28}$$

where $\hat{x}_{k,i}^{j}$ is the state estimation from the Kalman filter with the assumption on the associating between measurement $z_{k,j}$ to the ith object.

The limitation of JPDA is that JPDA cannot initiate or terminate tracks. There are restricted extensions to JPDA to allow the formation of a new track [86]. Moreover, the number of possible associations is $\sum_{i=1}^{\min(M,N)} C_M^i A_N^i$, so the computation of JPDA is expensive. It is an NP-hard problem. There are some methods to reduce the computation in JPDA, such as the Markov chain Monte Carlo data association [71], and near optimal JPDA [82]. Some examples of using JPDA in applications are in [35], [42], [48].

Discussion of Bayes rule in MHT and JPDA

In Section 2.5, Ω is the space of all hypothesis association between objects and measurements. Hence, Ω should depend on Z. If this is true, the application of the Bayes rule to estimate the probability of an hypothesis association in equations (2.22) and (2.23) might be not suitable. This is because the probability $p\left(\theta|Z\right)$ is

calculated by the following equation

$$p\left(\theta\left(Z\right)|Z\right) = \frac{p\left(Z|\theta\left(Z\right)\right)p\left(\theta\left(Z\right)\right)}{p\left(Z\right)} \tag{2.29}$$

However, the function $p\left(Z|\theta\left(Z\right)\right)$ is not a valid likelihood function. Hence, it is not clear when using the Bayes rule in MHT and JPDA [105].

2.8 Multiple-object tracking with visual data

Multiple-object tracking using visual data is an important task in applications including surveillance, vehicle tracking, augmented reality, human motion analysis, etc. Because visual data have many properties such as color, shape, and texture, there is a variety of approches for multiple-object tracking with visual data. First, methods based on filtering try to estimate the posterior multiple-object density with the assumption on the state space to model the dynamic of objects. Some methods can be listed as follows. Isard [47] proposed a Bayesian multiple blob tracker that used particle filter to infer the multiple-object state. Wu [113] introduced a particle filter incorporating with Bayesian network to solve the occlusion problem in visual tracking. Okuma [72] introduced a boosted particle filter and applied successfully in tracking hockey players. Some other approaches used MHT or JPDA in visual tracking [39], [87]. Besides, the random finite set approach is also applied in visual tracking [111]. These filtering methods used different observation models such as the color likelihood model [28], [91], the blob likelihood model

[47], and the edge likelihood model [16]. Second, some methods used non-filtering approach such as Bayes inference and MAP estimation. Zhao [117] proposed an Bayes inference to find the state of multiple objects. Yu [115] employed a variational distribution in multiple-object tracking. The above methods still have many open research problems because of the high computation cost and the difficulty in adapting to a changing of environment (e.g., lighting and scaling).

2.9 Multiple-speaker tracking

Speaker tracking is an important part of multimedia applications, e.g., video conferencing and robot control applications. The tracking is technically a very challenging task because of multiple paths, noise from difference sources, and simutaneously talking by speakers. The methods for multiple-speaker tracking are divided into two approaches: deterministic methods and stochastic methods. Deterministic methods try to find locations of speakers by minimizing the cost function such as beamforming [17]. Stochastic methods usually include two stages: extracting time delay of arrival measurements, and using filtering methods to track speakers. Some filter methods are applied in multiple-speaker tracking such as the Kalman filter [79], the particle filter [98]. These methods try to solve the data association between the time delay of arrival and states of speakers. Recently, there are some methods for multiple speaker tracking based on the random finite set [61], [104].

Because the multiple-speaker tracking has very high clutter in measurements, it is still a difficult research problem.

2.10 Summary

In this chapter, we reviewed methods for single and multiple-object tracking. The Kalman filter and the particle filter were presented to solve the Bayes filtering for single-object tracking. In multiple-object tracking, the data association problem is most challenging. Two methods, MHT and JPDA, were introduced to solve the data association problem. Both the Kalman filter and the particle filter can be used to implement MHT and JPDA. However, MHT and JPDA are computationally expensive due to the possible data associations. In the next chapter, we will discuss random set approaches that can avoid the data association in multiple-object tracking.

Chapter 3

Probability hypothesis density filter

3.1 Introduction

MHT and JPDA are indirect estimation methods for multiple-object tracking because they concentrate on the computing of the probability of data associations before estimating the states of objects. In this chapter, we review multiple-object tracking methods based on the random set approach. These methods are direct estimation methods for multiple-object tracking.

Random sets are random elements whose values are sets. They are generalizations of the familiar concept of random variables (or random vectors) in probability theory. Recently, there has been increasing research interest in using random finite set theory to solve multiple-object tracking problem [37], [63], [66]. Here, the states of objects are represented as random sets. Using this model, the birth and death of

objects can be described in the tracking algorithm. Mahler [63] presented a probability hypothesis density (PHD) filter based on random finite set to approximate the first moment of the multiple-object posterior density. The PHD filter operates on a single-object state space and avoids the combinatorial problem that arises from data association.

3.2 Random finite set Bayesian filter for multiple-object tracking

The following RFS formulations for multiple-object tracking is described in [102]. From Section 2.5, we have multiple-object state X_k and measurements of multiple objects Z_k. Let the single-object state space be $\mathcal{X}$. The object states $x_{k,1}, \ldots, x_{k,M(k)} \in \mathcal{X}$ and multiple-object state $X_k \in \mathcal{F}(\mathcal{X})$, where $\mathcal{F}(\mathcal{X})$ denotes the collection of all finite subsets of space $\mathcal{X}$. Similarly, let the single-object measurement space be $\mathcal{Z}$. The measurements $z_{k,1}, \ldots, z_{k,N(k)} \in \mathcal{Z}$ and the multiple-object measurement $Z_k \in \mathcal{F}(\mathcal{Z})$.

Now, we describe an RFS model for the time evolution of the multiple-object state, which incorporates object motions, births and deaths of objects. For a given multiple-object state X_{k-1} at time $k-1$, each $x_{k-1} \in X_{k-1}$ either continues to exist at time k with probability $p_{S,k}(x_{k-1})$ or dies with probability $1 - p_{S,k}(x_{k-1})$. Conditional on the existence at time k, the probability density of a transition from

state x_{k-1} to x_k is $f_{k|k-1}(x_k|x_{k-1})$ (mentioned in Section 2.2). Consequently, the object survives or dies from a given state $x_{k-1} \in X_{k-1}$ can be described by an RFS $S_{k|k-1}(x_{k-1})$ that includes $\{x_k\}$ when the object survives, or $\emptyset$ when the object dies. Moreover, a new object at time k can arise either by spontaneous birth or by spawning from an object at time $k-1$. For a given X_{k-1} at time $k-1$, multiple-object state X_k at time k is given by the union of the surviving objects, the spawned objects and the spontaneous births:

$$X_k = \left[\bigcup_{x_{k-1} \in X_{k-1}} S_{k|k-1}(x_{k-1}) \right] \cup \left[\bigcup_{x_{k-1} \in X_{k-1}} B_{k|k-1}(x_{k-1}) \right] \cup \Gamma_k, \tag{3.1}$$

where

$$\begin{aligned} \Gamma_k &= \text{RFS of spontaneous births at time } k, \\ B_{k|k-1}(x_{k-1}) &= \text{RFS of objects spawned at time } k \text{ from an object with} \\ &\qquad \text{previous state } x_{k-1} \end{aligned}$$

The RFS measurement model, which includes detections and clutters, is described as follows. A given object $x_k \in X_k$ is either detected with probability $p_{D,k}(x_k)$ or missed with probability $1 - p_{D,k}(x_k)$. Conditional on detection, the probability density of obtaining an observation z_k from x_k is given by the likelihood function $g_k(z_k|x_k)$ (mentioned in Section 2.2). Consequently, at time k, the RFS of measurements $\Theta_k(x_k)$ that are generated from $x_k \in X_k$ include $\{z_k\}$ when the object is detected, or $\emptyset$ when the object is missed. Moreover, the sensor also receives a set K_k of false measurements, or clutters. Thus, given a multiple-object state X_k

at time k, the multiple-object measurement Z_k received at the sensor is formed by the union of object generated measurements and clutters

$$Z_k = K_k \cup \left[\bigcup_{x \in X_k} \Theta_k(x) \right] \tag{3.2}$$

With RFS formulations on multiple-object states and measurements, the Bayesian filter for multiple-object tracking is described as follows. Similar to single-object tracking in Section 2.2, the multiple-object transition is denoted by the function $f_{k|k-1}(X_k|X_{k-1})$ and the multiple-object likelihood is denoted by $g_k(Z_k|X_k)$, where $f_{k|k-1}(X_k|X_{k-1})$ and $g_k(Z_k|X_k)$ can be derived by using finite set statistics (FISST) [37]. Then, the RFS Bayes filter propagates the multiple-object posterior $p_k(X_k|Z_{1:k})$ in time via the recursion

$$p_{k|k-1}(X_k|Z_{1:k-1}) = \int f_{k|k-1}(X_k|X_{k-1}) p_{k-1}(X_{k-1}|Z_{1:k-1}) \mu_s(dX_{k-1}) \tag{3.3}$$

$$p_k(X_k|Z_{1:k}) = \frac{g_k(Z_k|X_k) p_{k|k-1}(X_k|Z_{1:k-1})}{\int g_k(Z_k|X_k) p_{k|k-1}(X_k|Z_{1:k-1}) \mu_s(dX_k)} \tag{3.4}$$

where μ_s is an appropriate reference measure on $\mathcal{F}(\mathcal{X})$ [103].

There is an implementation of the RFS Bayes filter by the sequential Monte Carlo method [61]. However, this method is computationally intensive due to the combinatorial nature of the densities, especially when the number of objects is large. Nonetheless, the RFS Bayes filter has been successfully applied to applications where the number of objects is small [61].

3.3 Probability hypothesis density (PHD) filter

The RFS Bayes filter in the previous section propagates the multiple-object posterior density recursively in time. However, the computational intractability is far more severe than the single-object case. A more tractable alternative to estimate the state of multiple objects is based on the probability hypothesis density filter [63]. The PHD is the first moment of the multiple-object posterior density. The PHD is defined as follows. For a RFS X on $\mathcal{X}$ with probability distribution P, the PHD is intensity $v(x)$ such that for each region $S \subseteq \mathcal{X}$, the integral of v over region S gives the expected number of elements of X that are in S:

$$\int \mid X \cap S \mid P(dX) = \int_S v(x)dx, \tag{3.5}$$

where X is a random set represented for a multiple-object state and x represents a state of a single object. Thus, we can estimate the state of objects by investigating high local maxima peaks of the PHD.

With the definition of PHD, the PHD filter is derived [63]. This is a recursive process from the PHD in the previous time under some assumptions. They are:

- Each object evolves and generates measurements independent of one another.
- The birth RFS and the surviving RFS are independent of each other.
- Clutter RFS is Poisson and independent of object-originated measurements.

These assumptions are commonly used in most tracking applications. Let

$\gamma_k(.)$ = intensity of the birth RFS Γ_k at time k

$\beta_{k|k-1}(.|\zeta)$ = the intensity of the RFS $B_{k|k-1}(\zeta)$ spawned at time k by an object with previous state ζ

$p_{S,k}(\zeta)$ = the probability that an object still exists at time k given that its previous state is ζ

$p_{D,k}(x)$ = the probability of detection given a state x at time k

$\kappa_k(.)$ = the intensity of the clutter RFS K_k at time k

Let v_k and $v_{k|k-1}$ denote the posterior and predicted intensities corresponding to the multiple-object posterior density $p_k(X_k|Z_{1:k})$ and the multiple-object predicted density $p_{k|k-1}(X_k|Z_{1:k-1})$ in the recursion. Under the above assumptions, the PHD filter propagates the posterior intensity in time via the PHD recursion:

$$\begin{aligned} v_{k|k-1}(x) &= \int p_{S,k}(\zeta) f_{k|k-1}(x|\zeta) v_{k-1}(\zeta) d\zeta \\ &\quad + \int \beta_{k|k-1}(x|\zeta) v_{k-1}(\zeta) d\zeta + \gamma_k(x) \end{aligned} \tag{3.6}$$

$$\begin{aligned} v_k(x) &= [1 - p_{D,k}(x)] v_{k|k-1}(x) \\ &\quad + \sum_{z \in Z_k} \frac{p_{D,k}(x) g_k(z|x) v_{k|k-1}(x)}{\kappa_k(z) + \int p_{D,k}(\xi) g_k(z|\xi) v_{k|k-1}(\xi)} \end{aligned} \tag{3.7}$$

Prediction Equation (3.6) includes three components. They can be intepreted as follows. The first component $\int p_{S,k}(\zeta) f_{k|k-1}(x|\zeta) v_{k-1}(\zeta) d\zeta$ is the predicted intensity of surviving objects from the previous step with a survival probability $p_{S,k}(\zeta)$. The second component $\int \beta_{k|k-1}(x|\zeta) v_{k-1}(\zeta) d\zeta$ is represented for the intensity of

spawning objects and the last component $\gamma_k(x)$ is the intensity of birth objects that is mentioned before. Update equation (3.7) has two components. The first component $[1-p_{D,k}(x)]v_{k|k-1}(x)$ is the intensity of objects with assuming that these objects are not detected. The second component $\sum_{z\in Z_k} \frac{p_{D,k}(x)g_k(z|x)v_{k|k-1}(x)}{\kappa_k(z)+\int p_{D,k}(\xi)g_k(z|\xi)v_{k|k-1}(\xi)}$ is represented for the intensity of objects that is caused by each measurement in the measurement random set.

Equations (3.6) and (3.7) are used in applications with one sensor. In the multiple-object tracking with multiple-sensor, the true PHD formula is difficult to obtain. Asynchronous sensor fusion in which the PHD is updated sequentially at each sensor has been proposed to deal with this case [104].

The PHD filter propagates the PHD in a single-object space over time steps, thus avoiding the high complexity computation from data association between measurements and objects. When using the intensity function to characterize the multiple-object posterior density, it is assumed that higher order moments are negligible. These assumptions are justifiable when measurement noise is small. Recently, to improve the performance of the PHD filter, Mahler [65] also presented a cardinalized probability hypothesis density (CPHD) filter that is a generalization of the PHD recursion. The CPHD filter jointly propagates the posterior intensity and the posterior cardinality distribution at time steps. Vo [106], [107] presented an implementation of the CPHD filter by using the Gaussian mixture.

3.4 Particle PHD filter

The earlier implementation of the PHD filter was based on sequential Monte Carlo methods [103]. This implementation is proved convergent and it is called the particle PHD filter. The particle PHD filter is summarized as follows.

Let a set of samples $\left\{w_k^{(i)}, x_k^{(i)}\right\}_{i=1}^{L_k}$ represent for the PHD $v_k(x)$, i.e.,

$$v_k(x) = \sum_{i=1}^{L_k} w_k^{(i)} \delta_{x_k^{(i)}}(x) \tag{3.8}$$

where L_k is the number of samples. From $\left\{w_{k-1}^{(i)}, x_{k-1}^{(i)}\right\}_{i=1}^{L_{k-1}}$ representing for $v_{k-1}(x)$, the prediction step is performed from Equation (3.6). Let J_k be the number of birth samples at time k, we have $\left\{\tilde{w}_{k|k-1}^{(i)}, \tilde{x}_k^{(i)}\right\}_{i=1}^{L_{k-1}+J_k}$ representing $v_{k|k-1}(x)$

$$v_{k|k-1}(x) = \sum_{i=1}^{L_{k-1}+J_k} \tilde{w}_{k|k-1}^{(i)} \delta_{\tilde{x}_k^{(i)}}(x) \tag{3.9}$$

Samples represented for predicted intensity $v_{k|k-1}(x)$ can be obtained from sampling functions $q\left(\cdot|x_{k-1}^{(i)}, Z_k\right)$ and $p_k(\cdot|Z_k)$. The weight for each birth sample is calculated from birth intensity $\gamma_k(x)$ and the weight for each predicted sample is from function $\phi_{k|k-1}(x,\zeta)$, where

$$\phi_{k|k-1}(x,\zeta) = p_{S,k}(\zeta) f_{k|k-1}(x|\zeta) + \beta_{k|k-1}(x|\zeta) \tag{3.10}$$

Then the update step is implemented from Equation (3.7) to obtain the $\left\{\tilde{w}_k^{(i)}, \tilde{x}_k^{(i)}\right\}_{i=1}^{L_{k-1}+J_k}$ representing $v_k(x)$. We eliminate particles with low weights and multiply particles with high weights to focus on the important zone of the space by the resampling

step. Figure 3.1 shows the details of the particle PHD filter. From the set of particles $\left\{w_k^{(i)}, x_k^{(i)}\right\}_{i=1}^{L_k}$, clustering techniques are performed to obtain the state estimates of objects. There are some works on obtaining the estimations from the set of particles such as approximation Gaussian mixtures [97] and K-means [103].

3.5 Gaussian mixture probability hypothesis density (GMPHD) filter

The limitations of the particle PHD filter are the large number of particles and the unreliability of clustering techniques for extracting state estimates. Hence, Vo [102] proposed an analytic solution to the PHD filter for linear Gaussian systems. It is called the Gaussian mixture probability hypothesis density (GMPHD) filter. The advantages of the GMPHD filter are the great reliability in extracting state estimates and lower cost of computation than the particle PHD filter. Some extensions of the GMPHD filter for non-linear Gaussian systems are also proposed. The GMPHD filter is summarized as follows.

First, we consider some assumptions. The transition function of each object follows a linear Gaussian model, i.e.,

$$f_{k|k-1}(x|\zeta) = \mathcal{N}(x; F_{k-1}\zeta, Q_{k-1}) \tag{3.11}$$

where $\mathcal{N}(.; m, P)$ denotes a Gaussian density with mean m and covariance P, F_{k-1}

At time $k \geq 1$,

Step 1: *Prediction Step*

For $i = 1, ..., L_{k-1}$

Sample $\tilde{x}_k^{(i)} \sim q\left(\cdot|x_{k-1}^{(i)}, Z_k\right)$ and compute the predicted weights

$$\tilde{w}_{k|k-1}^{(i)} = \frac{\phi_{k|k-1}\left(\tilde{x}_k^{(i)}, x_{k-1}^{(i)}\right)}{q_k\left(\tilde{x}_k^{(i)}|x_{k-1}^{(i)}, Z_k\right)} w_{k-1}^{(i)}$$

For $i = L_{k-1} + 1, ..., L_{k-1} + J_k$

Sample $\tilde{x}_k^{(i)} \sim p_k\left(\cdot|Z_k\right)$ and compute the weights of new born particles

$$\tilde{w}_{k|k-1}^{(i)} = \frac{1}{J_k} \frac{\gamma_k\left(\tilde{x}_k^{(i)}\right)}{p_k\left(\tilde{x}_k^{(i)}|Z_k\right)}$$

Step 2: *Update Step*

For each $z \in Z_k$, compute

$$C_k\left(z\right) = \sum_{j=1}^{L_{k-1}+J_k} p_{D,k}(\tilde{x}_k^{(j)}) g_k(z|\tilde{x}_k^{(j)}) \tilde{w}_{k|k-1}^{(j)}$$

For $i = 1, ..., L_{k-1} + J_k$

$$\tilde{w}_k^{(i)} = \left[\left(1 - p_{D,k}(\tilde{x}_k^{(i)})\right) + \sum_{z \in Z_k} \frac{p_{D,k}(\tilde{x}_k^{(i)}) g_k(z|\tilde{x}_k^{(i)})}{\kappa_k\left(z\right) + C_k\left(z\right)}\right] \tilde{w}_{k|k-1}^{(i)}$$

Step 3: Resampling *Step*

Compute the total mass $\hat{N}_{k|k} = \sum_{j=1}^{L_{k-1}+J_k} \tilde{w}_k^{(j)}$

Resample $\left\{\frac{\tilde{w}_k^{(i)}}{\hat{N}_{k|k}}, \tilde{x}_k^{(i)}\right\}_{i=1}^{L_{k-1}+J_k}$ to get $\left\{\frac{w_k^{(i)}}{\hat{N}_{k|k}}, x_k^{(i)}\right\}_{i=1}^{L_k}$

Rescale the weights by $\hat{N}_{k|k}$ to get $\left\{w_k^{(i)}, x_k^{(i)}\right\}_{i=1}^{L_k}$

Figure 3.1: Particle PHD filter

is a state transition matrix, and Q_{k-1} is a process noise covariance. The likelihood function is also a linear Gaussian model, i.e.,

$$g_k(z|x) = \mathcal{N}(z; H_k x, R_k) \tag{3.12}$$

where H_k is an observation matrix, and R_k is an observation noise covariance. The survival and detection probabilities are

$$p_{S,k}(x) = p_{S,k} \tag{3.13}$$

$$p_{D,k}(x) = p_{D,k} \tag{3.14}$$

The intensity of the spontaneous birth RFS is

$$\gamma_k(x) = \sum_{i=1}^{J_{\gamma,k}} w_{\gamma,k}^{(i)} \mathcal{N}(x; m_{\gamma,k}^{(i)}, P_{\gamma,k}^{(i)}) \tag{3.15}$$

The posterior intensity at time $k-1$ is a Gaussian mixture of the form

$$v_{k-1}(x) = \sum_{i=1}^{J_{k-1}} w_{k-1}^{(i)} \mathcal{N}(x; m_{k-1}^{(i)}, P_{k-1}^{(i)}) \tag{3.16}$$

With these assumptions, it may be proven that if the initial prior intensity is a Gaussian mixture, then the posterior intensity at any subsequent time step is also a Gaussian mixture. The details of the GMPHD filter are now described.

The predicted intensity to time k is a Gaussian mixture, and is given by

$$v_{k|k-1}(x) = v_{S,k|k-1}(x) + \gamma_k(x) \tag{3.17}$$

where

$$\begin{aligned} v_{S,k|k-1}(x) &= p_{S,k} \sum_{j=1}^{J_{k-1}} w_{k-1}^{(j)} \mathcal{N}(x; m_{S,k|k-1}^{(j)}, P_{S,k|k-1}^{(j)}), \\ m_{S,k|k-1}^{(j)} &= F_{k-1} m_{k-1}^{(j)}, \\ P_{S,k|k-1}^{(j)} &= Q_{k-1} + F_{k-1} P_{k-1}^{(j)} F_{k-1}^T. \end{aligned}$$

Because $v_{S,k|k-1}(x)$ and $\gamma_k(x)$ are Gaussian mixtures, $v_{k|k-1}(x)$ can be expressed as a Gaussian mixture of the form

$$v_{k|k-1}(x) = \sum_{i=1}^{J_{k|k-1}} w_{k|k-1}^{(i)} \mathcal{N}(x; m_{k|k-1}^{(i)}, P_{k|k-1}^{(i)}) \tag{3.18}$$

Then, the posterior intensity at time k is also a Gaussian mixture, and is given by

$$v_k(x) = (1 - p_{D,k}) v_{k|k-1}(x) + \sum_{z \in Z_k} v_{D,k}(x; z) \tag{3.19}$$

where

$$\begin{aligned} v_{D,k}(x; z) &= \sum_{j=1}^{J_{k|k-1}} w_k^{(j)}(z) \mathcal{N}(x; m_{k|k}^{(j)}, P_{k|k}^{(j)}), \\ w_k^{(j)}(z) &= \frac{p_{D,k} w_{k|k-1}^{(j)} q_k^{(j)}(z)}{\kappa_k(z) + p_{D,k} \sum_{l=1}^{J_{k|k-1}} w_{k|k-1}^{(l)} q_k^{(l)}(z)}, \\ q_k^{(j)}(z) &= \mathcal{N}(z; H_k m_{k|k-1}^{(j)}, R_k + H_k P_{k|k-1}^{(j)} H_k^T), \\ m_{k|k}^{(j)} &= m_{k|k-1}^{(j)} + K_k^{(j)}(z - H_k m_{k|k-1}^{(j)}), \\ P_{k|k}^{(j)} &= [I - K_k^{(j)} H_k] P_{k|k-1}^{(j)}, \\ K_k^{(j)} &= P_{k|k-1}^{(j)} H_k^T (H_k P_{k|k-1}^{(j)} H_k^T + R_k)^{-1}. \end{aligned}$$

3.6 Summary

In this chapter, we reviewed random set approaches for multiple-object tracking. The RFS Bayes filter can be used when the number of objects is small. However, when there are a large number of objects, the computation of the RFS Bayes filter is intractable. This is because the number of computations of likelihood function in the RFS Bayes filter using the finite set statistics is about $\sum_{i=1}^{\min(M,N)} C_M^i A_N^i N$. Hence, the PHD filter, the first moment of the multiple-object posterior density, was proposed. Two implementations of the PHD filter, the particle PHD filter and the GMPHD filter, were summarized. The PHD filter operates on a single-object state space and avoids the data association problem. This can help to reduce the computation when tracking multiple objects. In subsequent chapters, we will propose our methods and applications for multiple-object tracking that employ the PHD filter approach.

Chapter 4

Maintaining track continuity in the GMPHD filter

4.1 Introduction

PHD implementations such as the particle PHD filter and the GMPHD filter do not include object identities. In many cases, we need to know the track continuity of objects in order for post processing such as behavior of objects and activity recognition. There are methods to obtain the object identities for the PHD filter. Firstly, some methods use the particle PHD filter for pre-filtering the data input to other methods, such as the multiple hypothesis tracker [75] and assignment algorithms [60]. There are also methods that analyze the propagation of particles to maintain track continuity [20], [74]. Because of the unreliability of clustering methods in the

particle PHD filter, the performances of these approaches are affected.

Recently, Clark [21] introduced a technique to identify the state estimates of objects in the GMPHD filter. In this method, each Gaussian component is identified by a label. After the updating step in the GMPHD filter, if two or more components have the same label, then this label is given to the one with the largest weight and new labels are assigned to the other components. This method was successfully applied to sonar image tracking [22]. However, a limitation is that it does not include temporal information, which adversely affects the performance. For example in Figure 4.1, at time $k-1$, the first object (square) and the second object (circle) are at positions A and C, respectively. At time k, the first object moves to B and the second object moves to D. If we do not consider temporal information, the weight of the Gaussian component with label 'circle' at position B may be higher than the weight of the Gaussian component with the same label at position D. Hence, the state of the second object is estimated at B and a new label is assigned for the Gaussian component at D. These estimates are not the desired estimates.

In this chapter, we propose a method for maintaining the continuity of state estimates of objects in the GMPHD filter. To identify the states of objects, the set of labels from Gaussian components is used to create hypotheses for the label association process. This method reduces a large number of label association hypotheses compared with methods in [60], [75]. Moreover, we employ the Hungarian

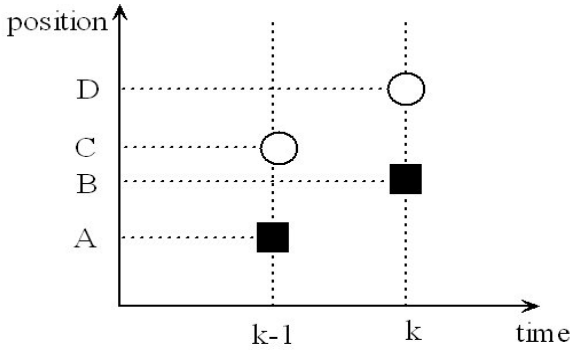

Figure 4.1: An example when two objects are close

algorithm [56] for optimizing the search for the best hypothesis association. The method can be applied in real-time tracking applications that the MHT or JPDA method performs with difficulty because of computational expense. Since the proposed method considers the temporal information from the previous step to the current step, it could achieve better performance than the identifications by the method in [21].

4.2 GMPHD filter with label association

To maintain the temporal continuity of state estimates of objects in the GMPHD filter, we propose a method with two stages. The first stage is to build hypothesis labels for Gaussian components. The set of labels for Gaussian components of the posterior intensity at time $k-1$ is $L_{k-1} = \left(l_{1,k-1}, ..., l_{J_{k-1},k-1}\right)$, where $l_{i,k-1}$ is the label of the ith Gaussian component. The labels of state estimates are

$L^E_{k-1} = \left(l^E_{1,k-1}, ..., l^E_{N_{k-1},k-1}\right)$, where N_{k-1} is the number of objects at time $k-1$. At time k, in the prediction step of the GMPHD filter (Equation (3.17)), the set of predicted labels is

$$L_{k|k-1} = L_{k-1} \cup L_\Gamma \tag{4.1}$$

where L_Γ is the set of birth labels for birth Gaussian components. Then, in the updating step of the GMPHD filter (Equation (3.19)), Gaussian components are updated with the measurement set to obtain $v_k(x)$. These Gaussian components are attached with labels from $L_{k|k-1}$. These labels are called hypothesis labels $\tilde{L}_k = \left(\tilde{l}_{1,k}, ..., \tilde{l}_{J_k,k}\right)$. We notice that because some Gaussian components are merged in the updating step, $\tilde{l}_{i,k}$ can have more than one label. After that, state estimates are extracted from intensity $v_k(x)$. From $\tilde{L}_k = \left(\tilde{l}_{1,k}, ..., \tilde{l}_{J_k,k}\right)$, the set of hypothesis labels for state estimates $\tilde{L}^E_k = \left(\tilde{l}^E_{1,k}, ..., \tilde{l}^E_{N_k,k}\right)$ is obtained.

In the second stage, each Gaussian component will be assigned a label. From $\tilde{L}^E_k$ and L^E_{k-1}, we construct a bipartite graph $G = (V, E)$ where $V = V_L \cup V_R$, vertices V_L are state predictions $\hat{x}_{k|k-1}$ from state estimates $\hat{x}_{k-1}$at time $k-1$, vertices V_R are state estimates $\hat{x}_k$ at time k, and E is the set of weight edges e_{ij} such that

$$e_{ij} = \begin{cases} \left\|\hat{x}_{i,k} - \hat{x}_{j,k|k-1}\right\| & \text{, if } l^E_{j,k-1} \in \tilde{l}^E_{i,k} \\ & \text{and } i \in V_R, j \in V_L \\ \infty & \text{otherwise} \end{cases} \tag{4.2}$$

A matching in a bipartite graph is assigning vertices in V_L to vertices in V_R. We

have to find the optimal match with minimum edges' weights. Then, if vertex $i \in V_R$ that is not matched with any vertex in V_L, we will assign vertex i with the label that has the largest weight in $\tilde{l}^E_{i,k}$. Thus, the label association for the GMPHD filter can be described in Figure 4.2.

4.3 Matching with minimum total distance for label association

In the context of label association for the GMPHD filter, we have to find a matching with minimum edge weights in a bipartite graph. If we choose a match by selecting edges that have minimum weights, there is no guarantee that the number of matched vertices is maximum. Therefore, some labels of state estimates at time k may not be the same as labels of state estimates at time $k-1$. In other words, selecting edges that have minimum weight favors good local matches. In the global view, this method is not an optimal method. For example, in Figure 4.3, if we choose {(b,d),(c,e)} (edges that have minimum weights), the vertices a and f are not matched. In this case, we have to choose {(a,d),(b,e),(c,f)}.

The best known polynomial time-bound algorithm for weighted bipartite matching is the classical Hungarian algorithm due to Kuhn [56], which runs in time $O(|V|(|E| + |V|log|V|)$. Weighted bipartite matching algorithms can be implemented efficiently, and can be applied to graphs of reasonably large size (about

Step 1: *Prediction Step*

$i = 0$

For $j = 1, ..., J_{\gamma,k}$ (birth Gaussians)

$i = i + 1$

Obtain weight, mean, covariance for the ith birth Gaussian

$l_{k|k-1}^{(i)} =$ birth label

For $j = 1, ..., J_{k-1}$ (existing Gaussians)

$i = i + 1$

Obtain weight, mean, covariance for the ith predicted Gaussian

$l_{k|k-1}^{(i)} = l_{k-1}^{(j)}$

Step 2: *Updating Step*

For $j = 1, ..., J_{k|k-1}$

$\tilde{l}_{j,k} = l_{k|k-1}^{(j)}$

$n = 0$

For each $z \in Z_k$

$n = n + 1$

For $j = 1, ..., J_{k|k-1}$

Obtain weight, mean, covariance for the $(nJ_{k|k-1} + j)$-th update Gaussian

$\tilde{l}_{nJ_{k|k-1}+j,k} = \tilde{l}_{j,k}$

Pruning and merging Gaussian components

Construct the label association graph $G = (V_L \cup V_R, E)$

Find a matching with minimum total distance in the graph

Assign labels for matching and non-matching Gaussian components

Figure 4.2: Label association for the GMPHD filter

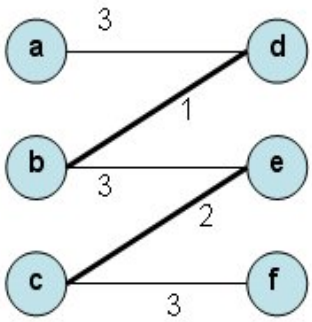

Figure 4.3: An example for wrong matching

100,000 vertices) [50]. Thus, we applied the Hungarian algorithm to find the minimum edge weight matching in the bipartite graph that is mentioned in the previous section. The Hungarian algorithm for label association can be summarized in Figure 4.4. After using the Hungarian algorithm to find the matching with minimum edge weights, we have label associations of state estimates of the previous and current time steps. If a state estimate at the current step is matched, we assign the corresponding Gaussian component for the label of the matched state estimate at the previous step. Otherwise, this Gaussian component is assigned for the label of the largest weight Gaussian contributing to this component. For example, in Figure 4.3, after using the Hungarian algorithm, the labels of Gaussian components represented for vertices d, e, f are the labels of Gaussian components represented for vertices a, b, c, respectively.

Input: $\mathbf{G} = (V_L \cup V_R, E)$

Step 1: Initialize $M \longleftarrow 0$

Step 2:

For each unmatched $x^* \in V$

Find path D from x^* in which matched vertex and unmatched vertex are alternated (see [84] for more details)

$M = (M \setminus (M \cap D)) \cap (D \setminus (M \cap D))$

EndFor

Output: M

Figure 4.4: Hungarian algorithm for label association

4.4 Simulation experiments

Results of simulations are presented to demonstrate the effectiveness of our approach. There are two examples. In Example 1, a maximum of two objects appear and disappear at different times. Each object has a survival probability $p_{S,k} = 0.99$ and follows a nonlinear nearly constant turn model in which the object state takes the form $x_k = \left[y_k^T, \omega_k\right]^T$, where $y_k = [p_{x,k}, p_{y,k}, \dot{p}_{x,k}, \dot{p}_{y,k}]^T$ is the coordinate (x, y)

and velocity in each dimension of object, and ω_k is the turn rate. The state dynamic equations are given by

$$\begin{aligned} y_k &= F(\omega_{k-1})y_{k-1} + G\omega_{k-1}, \\ \omega_k &= \omega_{k-1} + \Delta u_{k-1}, \end{aligned}$$

where

$$F(\omega) = \begin{bmatrix} 1 & 0 & \frac{\sin\omega\Delta}{\omega} & -\frac{1-\cos\omega\Delta}{\omega} \\ 0 & 1 & \frac{1-\cos\omega\Delta}{\omega} & \frac{\sin\omega\Delta}{\omega} \\ 0 & 0 & \cos\omega\Delta & -\sin\omega\Delta \\ 0 & 0 & \sin\omega\Delta & \cos\omega\Delta \end{bmatrix}, G = \begin{bmatrix} \frac{\Delta^2}{2} & 0 \\ 0 & \frac{\Delta^2}{2} \\ \Delta & 0 \\ 0 & \Delta \end{bmatrix}$$

$\Delta = 1s$, $\omega_k \sim \mathcal{N}(.; 0, \sigma_w^2 I_2)$, $\sigma_w = 10^2$, $u_k \sim \mathcal{N}(.; 0, \sigma_u^2)$, and $\sigma_u = 2\pi/180$. We assumed no spawning and that the spontaneous birth RFS is Poisson with intensity

$$\gamma_k(x) = 0.1\mathcal{N}(x; m_\gamma^{(1)}, P_\gamma)$$

where

$$\begin{aligned} m_\gamma^{(1)} &= [500, 500, 0, 0, 0]^T \\ P_\gamma &= \text{diag}([4000, 4000, 4000, 4000, (6\pi/180)^2]^T). \end{aligned}$$

Each object has a probability of detection $p_{D,k} = 0.98$. An observation consists of bearing and range measurements. The observation model is given by

$$z_k = \begin{bmatrix} \arctan(p_{x,k}/p_{y,k}) \\ \sqrt{(p_{x,k}^2 + p_{y,k}^2)} \end{bmatrix} + \epsilon_k,$$

where $\epsilon_k \sim N(.; 0, R_k)$ with $R_k = ([\sigma_\theta^2, \sigma_r^2]^T)$, $\sigma_\theta = (\pi/300)$ and $\sigma_r = 10$. The clutter RFS follows the uniform Poisson model over the surveillance region $[-\pi/2, \pi/2] \times [0, 2000]$, with $\lambda_c = 1.6 \times 10^{-3}$ (i.e., an average of 10 clutter returns on the surveillance region). The pruning parameters for the GMPHD filters are $T = 5 \times 10^{-3}$, merging threshold $U = 10$, and maximum number of Gaussian components $J_{max} = 100$.

Figures 4.5 and 4.6 show the track continuity from the method in [21] and our method, respectively. In these figures, the ground-truth is represented by lines and state estimates are represented by shapes. Two estimations having the same shape are from the same object. In Figure 4.5, the results indicate that the identities of the objects change at time steps 6 and 65. This is because the track continuity method in [21] chooses the labels of Gaussian components that have the largest weights. This method does not consider the temporal information, and it is based on the heuristic method. Hence, the labels of state estimates are not correct at time steps 6 and 65. In Figure 4.6, our method considers the minimum total distance between the prediction of previous state estimates and the current state estimates to assign labels. Thus, its performance is better in this example.

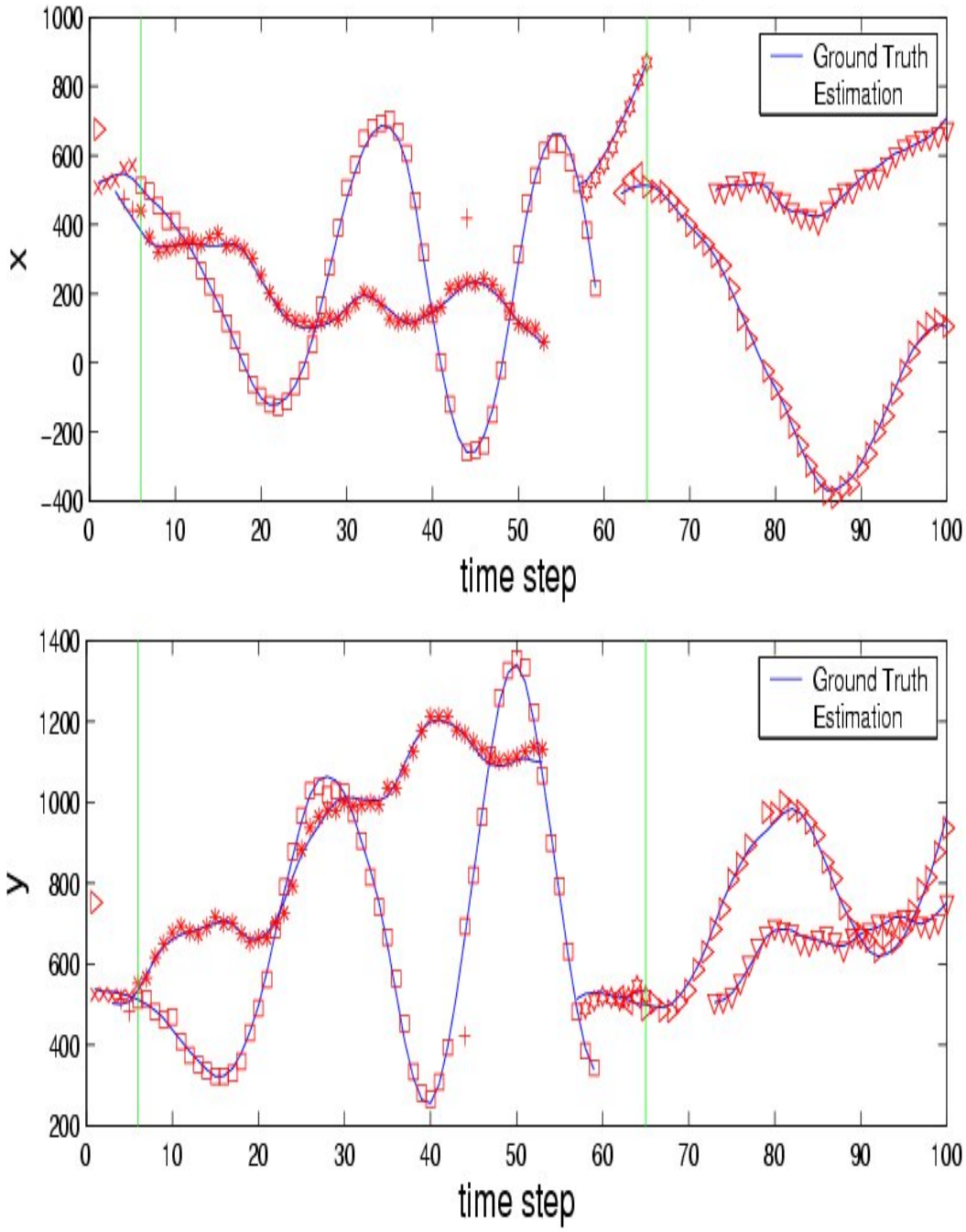

Figure 4.5: Track continuity with the method in [21]

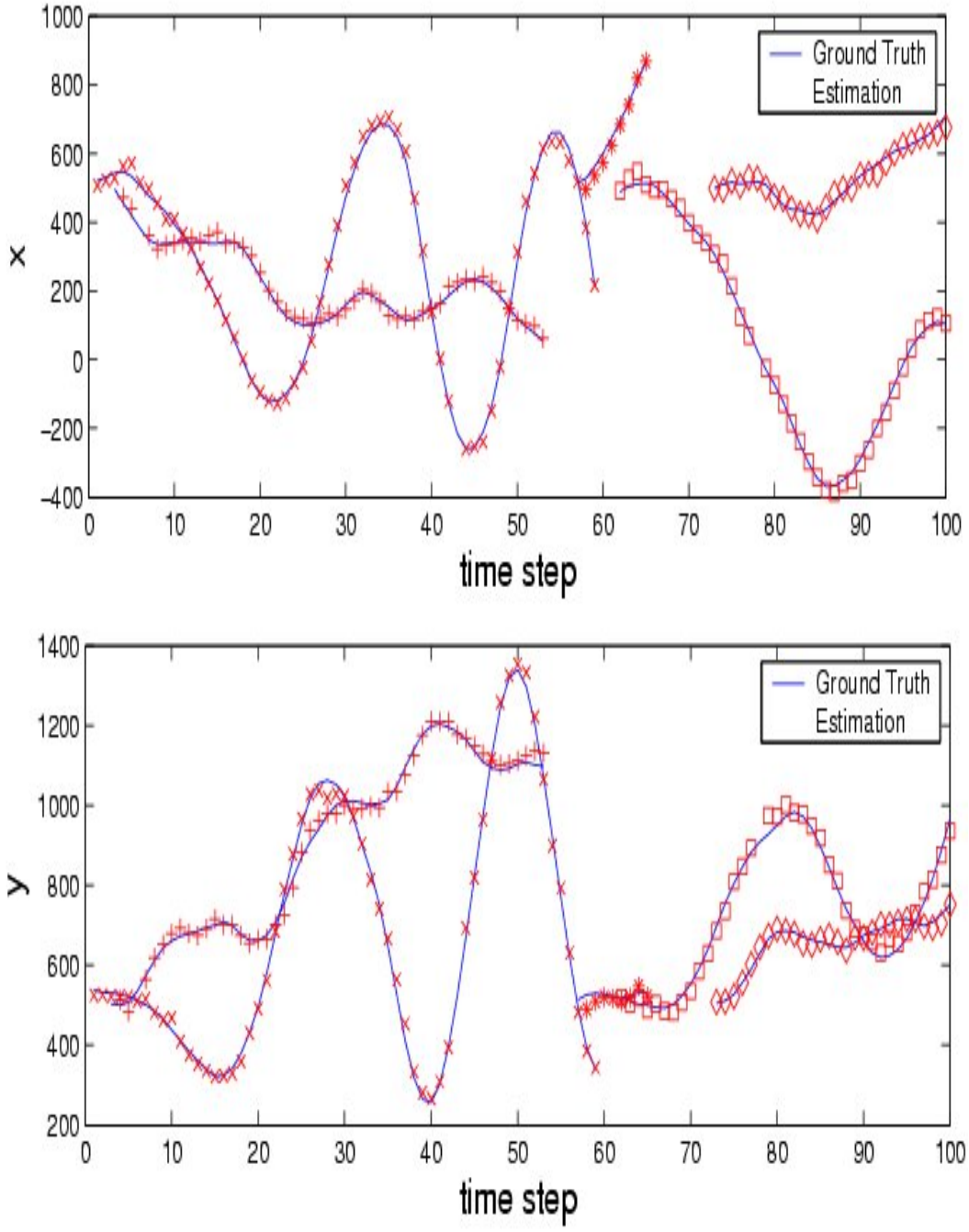

Figure 4.6: Track continuity with our method

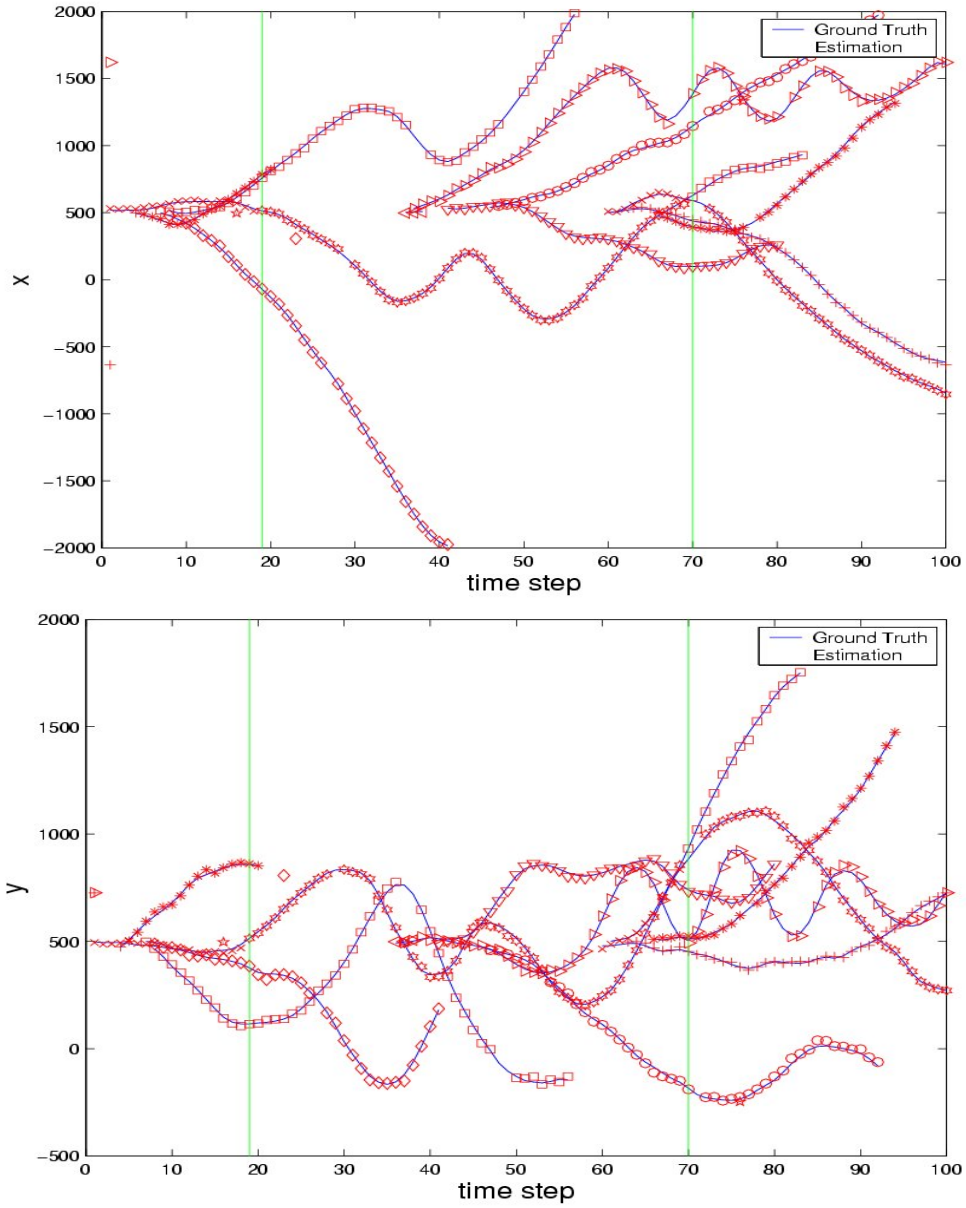

Figure 4.7: Track continuity with the method in [21]

In Example 2, there are a maximum seven objects that appear and disappear at different times. Figures 4.7 and 4.8 show the track continuity from the method in [21] and our method, respectively. The method in [21] changes the identities of

the objects at time steps 19 and 70 in Figure 4.7. Especially, at time step 70, this method gives wrong identifications of two tracks. In Figure 4.8, our method has a good performance in all time steps.

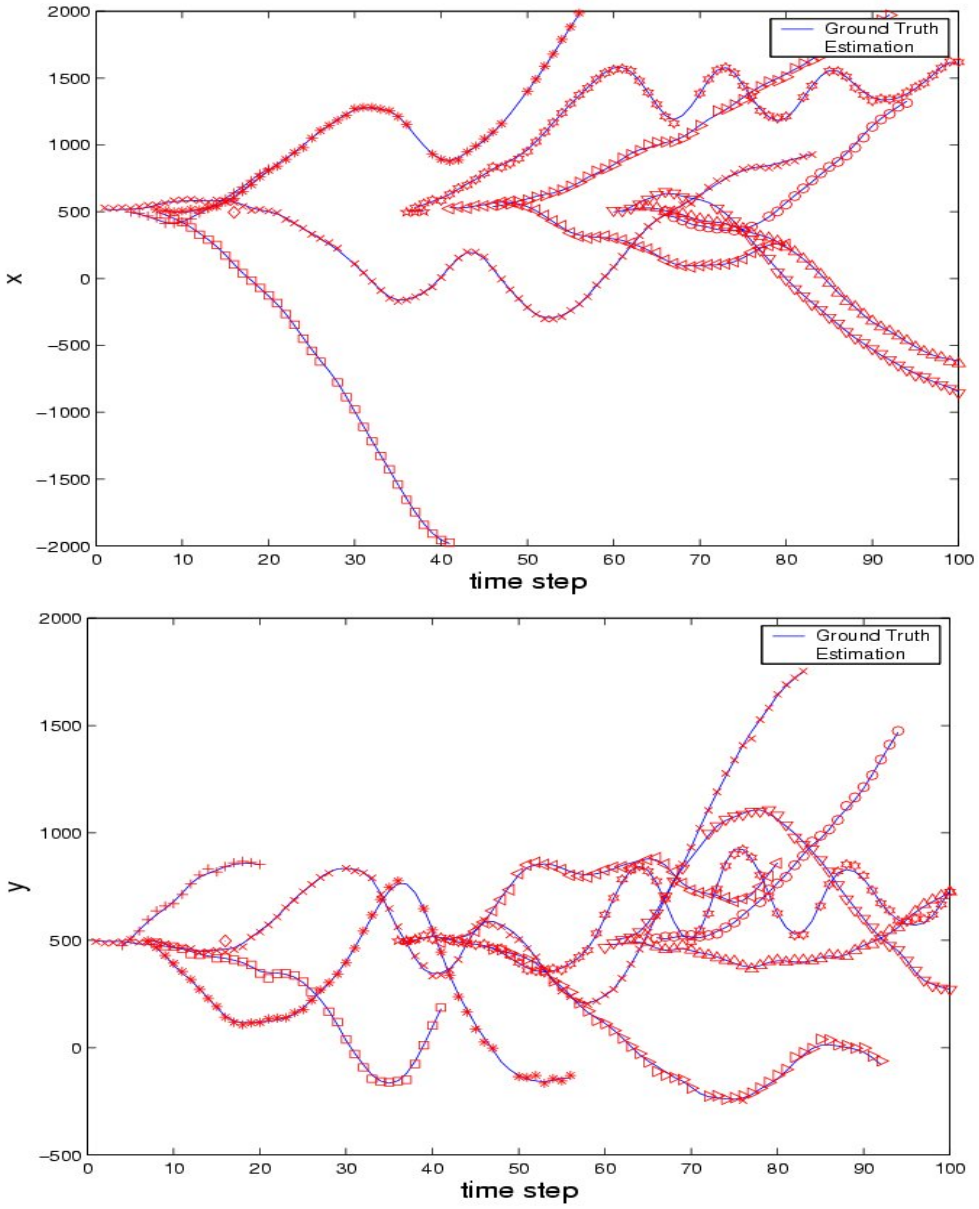

Figure 4.8: Track continuity with our method

In this example, in some time steps, there are 7 objects and 10 measurements. If we use methods such as MHT [80] or JPDA [6], the number of hypotheses is about $\sum_{i=1}^{7} C_7^i A_{10}^i$. The computation by JPDA or MHT in this example is very complicated. However, the run-time of our method for 100 time steps is 3.7s on Matlab 6.0, with a Pentium IV 2.6 GHz PC.

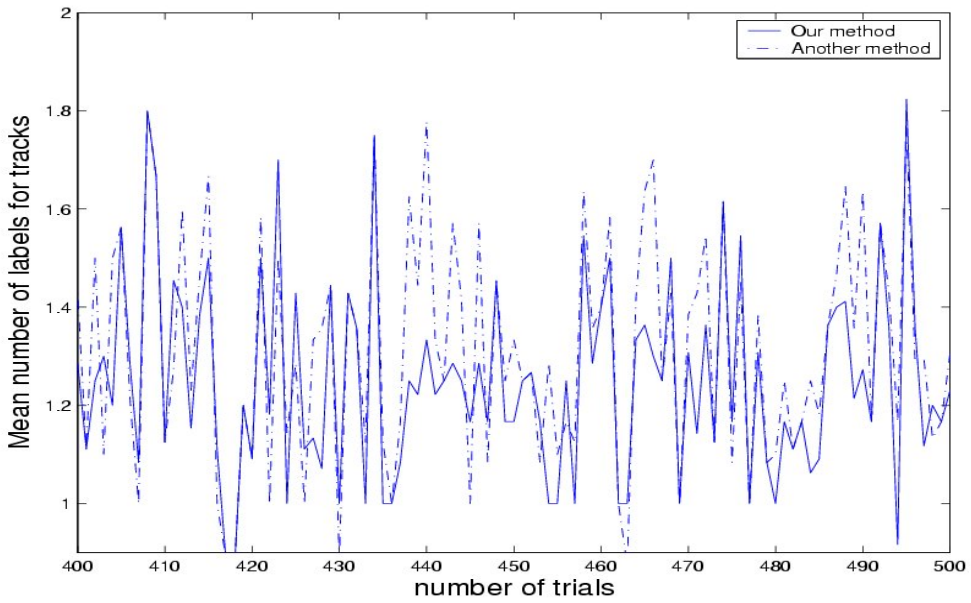

Figure 4.9: Mean number of labels for tracks of our method and the method in [21]

The above results are the performances for two trials. For performance generalization, we test the algorithm for 1000 trials. For ease of visualization, Figure 4.9 shows the results of the mean number of labels for tracks between our method and the method in [21] from the 400th trial to the 500th trial. The mean number of labels for tracks is the mean number of labels used to identify each track. The

results indicate that in most of the trials, the mean number of labels for tracks with our method is nearer to the true value (1) than the method in [21]. Moreover, the overall mean number of labels for tracks in 1000 trials with our method is 1.26 compared with 1.33 of the method in [21]. Hence, the proposed method for continuity tracking is more efficient.

4.5 Summary

An algorithm for maintaining the continuity of state estimates of objects in the GMPHD filter has been presented. It can be used to track multiple objects in applications with high density of clutters and varying number of objects that traditional methods such as JPDA or MHT are difficult to handle because of the computational complexity. Our method reduces the number of hypotheses remarkably based on the property of Gaussian components in the GMPHD filter. Furthermore, our method considers temporal information and it performs better than the method without using temporal feature [21]. It has been shown that our method is efficient for multiple-object tracking.

Chapter 5

Multiple-speaker tracking using the PHD filter

5.1 Introduction

Speaker tracking is an important part of multimedia applications. For example, in video conferencing [18], [27], and robot control applications [69], it can be used to determine the spatial location of a speaker so that the camera is steered toward that speaker. The essential requirement for a speaker tracking system is to estimate the state of speaker within the acoustic environment based on measurements that are collected from several microphones. Speaker tracking is challenging because of the effects of reverberations, and noise from different sources.

There are many approaches to speaker tracking. Traditional approaches are

classified as one-stage or two-stage methods. One-stage methods are direct methods such as steered beam-forming [17], [30]. They scan the search space by an acoustic beam to find the position with the highest beam output energy. These methods suffer from the poor resolution and require a search over a highly nonlinear surface. Moreover, they are computationally intensive, which may be impractical for real-time applications.

Two-stage methods include the time-estimation stage and the localization stage. Firstly, the time delays of arrival are extracted from data frames. A well-known method for time delay estimation is based on the generalized cross correlation function [54]. The localization stage can be done by a least squares or Gauss-Newton iteration method [34]. In general, these methods transform the received data frames into a localization function that exhibits a peak in the location due to the speaker. However, reverberations cause spurious peaks in the localization function that may have greater magnitudes than the peak associated with the speaker. Thus, these traditional methods may not be efficient in a reverberant acoustic environment.

Recently, some approaches for speaker tracking problem using the particle filter has been proposed to cope with the effects of reverberations [99], [112]. In these approaches, the speaker tracking problem is formulated within a state-space estimation framework. The key idea of these methods is that the state of a speaker

follows a dynamic model from frame to frame. The performances of these approaches outperform traditional methods. They assume that only one speaker is active at a time, while in many applications, there are many people speaking simultaneously at a time. Thus, it is necessary to have methods to track multiple speakers. Unfortunately, tracking multiple speakers is a challenging problem because it is difficult to obtain the measurements for multiple speakers and the number of speakers varies in the tracking periods. Multiple-sensor data fusion, high clutter, and data association also need to be considered.

With the development of multiple-object tracking methods, recently, there are some approaches for multiple-speaker tracking. In [79], the authors assume that a single array is unable to track two simultaneously active speakers, whereas the complementary provision of data in the multiple-array framework makes multiple-speaker tracking possible. This method assumes a fixed number of speakers. In [98], the authors propose a data association method for multiple-speaker tracking by using the particle filter . However, this method has a limitation when there is clutter. There are two methods for multiple-speaker tracking based on the random finite set [61], [104]. These methods have good performance when tracking multiple speakers.

The objective of this chapter is to develop an efficient technique for real-time tracking of multiple speakers in a reverberant room. We use the idea of approximating multiple-sensor PHD update in [104] for the GMPHD filter. The method is

performed in a simulation environment with 4 microphone pairs to do the multiple-speaker tracking. Because the GMPHD filter is a closed-form of the PHD filter, it avoids the need for data association of time delay of arrival measurements and speakers. Moreover, the state estimates of speakers are obtained from the means of Gaussian components. Thus, we do not need to use clustering techniques to extract state estimates. The advantages of our method are lower computational expense and higher reliability than other methods in [61], [98], [104] for multiple-speaker tracking.

5.2 Random finite set for multiple-sensor multiple-object tracking

Multiple-speaker tracking is a particular problem in multiple-sensor multiple-object tracking. The multiple-sensor multiple-object tracking problem can be modeled by a random finite set (RFS) framework. Similar to Section 3.2, given a multiple-object state X_{k-1} at time $k-1$, the multiple-object state X_k at time k is given by the union of surviving objects and new objects, which is

$$X_k = \left[\bigcup_{x_{k-1} \in X_{k-1}} S_k(x_{k-1}) \right] \cup \left[\bigcup_{x_{k-1} \in X_{k-1}} B_{k|k-1}(x_{k-1}) \right] \cup [\Gamma_k] \tag{5.1}$$

Let $\mathcal{Z}^i$ be the measurement space of a single object at the ith sensor, then measurements collected from the ith sensor at time k is $Z_k^i \in \mathcal{F}(\mathcal{Z}^i)$. A given

object state $x_k \in X_k$ is either detected with a probability p_D or missed with a probability $(1 - p_D)$. Conditional on detection, the measurement from x_k at the ith sensor is defined by the RFS $\Theta_k^i(x_k)$. The ith sensor can also receive a set of clutters C_k^i. So, given a multiple-object state X_k at time k, the measurement set from the ith sensor at time k is formed by the union of object generated measurements and clutters,

$$Z_k^i = \left[\bigcup_{x_k \in X_k} \Theta_k^i(x_k) \right] \cup C_k^i \tag{5.2}$$

Assuming that we have Q sensors, the RFS of measurements at time k is modelled by

$$Z_k = \left[Z_k^1; Z_k^2; \ldots; Z_k^Q \right] \tag{5.3}$$

The multiple-sensor multiple-object tracking problem can be posed as follows: given a set of measurements $Z_{1:k}$ collected from sensors up to time k, the problem is to find $\hat{X}_k$ that is the expectation or maximization of the posterior density function $p(X_k|Z_{1:k})$. In the next section, we will present a method to obtain the PHD of the posterior density function $p(X_k|Z_{1:k})$ in the multiple-sensor multiple-object tracking environment by the GMPHD filter.

5.3 Gaussian mixture probability hypothesis density filter with multiple sensors

5.3.1 Assumptions

First, we consider some assumptions. The transition function of each object follows a linear Gaussian model, i.e.,

$$f_{k|k-1}(x|\zeta) = \mathcal{N}(x; F_{k-1}\zeta, Q_{k-1}) \tag{5.4}$$

where $\mathcal{N}(.; m, P)$ is a Gaussian density with mean m and covariance P, F_{k-1} is a state transition matrix, and Q_{k-1} is the process noise covariance. The likelihood function at each of Q sensors is a linear Gaussian model

$$g_k^i(z|x) = \mathcal{N}(z; H_k^i x, R_k^i) \tag{5.5}$$

where H_k^i is an observation matrix of the ith sensor, and R_k^i is an observation noise covariance of the ith sensor. The survival and detection probabilities are, respectively,

$$p_{S,k}(x) = p_{S,k} \tag{5.6}$$

$$p_{D,k}(x) = p_{D,k} \tag{5.7}$$

The intensity of the spontaneous birth RFS is

$$\gamma_k(x) = \sum_{i=1}^{J_{\gamma,k}} w_{\gamma,k}^{(i)} \mathcal{N}(x; m_{\gamma,k}^{(i)}, P_{\gamma,k}^{(i)}) \tag{5.8}$$

where $J_{\gamma,k}$ is the number of birth Gaussian components at time k. The posterior intensity at time $k-1$ is a Gaussian mixture of the form

$$v_{k-1}(x) = \sum_{i=1}^{J_{k-1}} w_{k-1}^{(i)} \mathcal{N}(x; m_{k-1}^{(i)}, P_{k-1}^{(i)}) \tag{5.9}$$

where J_{k-1} is the number of Gaussians of posterior intensity $v_{k-1}(x)$

5.3.2 GMPHD filter with multiple sensors

When there are many sensors, Vo [104] gave an idea of approximating multiple-sensor PHD update. Now, we implement this idea to fuse data from multiple sensors in the GMPHD filter. The algorithm is described below.

With the assumptions in 5.3.1, at time $k-1$ we have

$$v_{k-1}(x) = \sum_{i=1}^{J_{k-1}} m_{k-1}^{(i)} \mathcal{N}(x; w_{k-1}^{(i)}, P_{k-1}^{(i)}) \tag{5.10}$$

First, we use assumptions on state equation (5.4), measurement equation (5.5) and $v_{k-1}(x)$ to predict intensity $v_{k|k-1}^1(x)$ at sensor 1 by using Equation (3.17). Then, predicted PHD $v_{k|k-1}^1(x)$ is updated with measurement set Z_k^1 by Equation (3.19) to obtain the PHD at time k on sensor 1, $v_k^1(x)$. Since $v_{k-1}(x)$ is a Gaussian mixture, $v_k^1(x)$ is also a Gaussian mixture and has the form

$$v_k^1(x) = \sum_{i=1}^{J_k^1} w_{1,k}^{(i)} \mathcal{N}(x; m_{1,k}^{(i)}, P_{1,k}^{(i)}) \tag{5.11}$$

Now, at sensor 2, $v_k^1(x)$ is considered as the predicted PHD for sensor 2. Similar

to Equation (3.19), we have

$$v_k^2(x) = (1 - p_{D,k})v_k^1(x) + \sum_{z \in Z_k^2} v_{D,k}(x; z) \tag{5.12}$$

Hence, $v_k^2(x)$ also has the Gaussian mixture form

$$v_k^2(x) = \sum_{i=1}^{J_k^2} w_{2,k}^{(i)} \mathcal{N}(x; m_{2,k}^{(i)}, P_{2,k}^{(i)}) \tag{5.13}$$

We repeat this process with Q sensors. At the Qth sensor, we obtain $v_k^Q(x)$, and it has the form

$$v_k^Q(x) = \sum_{i=1}^{J_k^Q} w_{Q,k}^{(i)} \mathcal{N}(x; m_{Q,k}^{(i)}, P_{Q,k}^{(i)}) \tag{5.14}$$

The PHD for the multiple-sensor multiple-object posterior density will be

$$v_k(x) = v_k^Q(x) \tag{5.15}$$

The number of objects is estimated by

$$\begin{aligned} \hat{N}_{k|k} &= \int v_k(x)dx \\ &= \int \sum_{i=1}^{J_k^Q} w_{Q,k}^{(i)} \mathcal{N}(x; m_{Q,k}^{(i)}, P_{Q,k}^{(i)})dx \\ &= \sum_{i=1}^{J_k^Q} w_{Q,k}^{(i)} \end{aligned} \tag{5.16}$$

Thus, the properties of the GMPHD filter in the case of multiple sensors are similar to the single-sensor case. This means that in the multiple-sensor multiple-object tracking problem, under the assumptions in 5.3.1, if the initial prior intensity of multiple-sensor multiple-object tracking is a Gaussian mixture, the posterior

intensity for asynchronous sensor fusion method at any subsequent time step will be a Gaussian mixture.

5.3.3 Implementation issues

The state estimates of objects are the means of Gaussian components that have high weights (above 0.5) in $v_k(x)$. This estimation method is more efficient than the particle PHD filter. This is because in the particle PHD filter, we obtain the number of objects $\hat{N}_{k|k}$ then partition particles into $\hat{N}_{k|k}$ clusters. If $\hat{N}_{k|k}$ is not correct, the tracking performance will be affected.

Now, we investigate the number of Gaussian components in $v_k(x)$. At the first sensor, the number of Gaussian components is

$$J_k^1 = (J_{k-1} + J_{\gamma,k})(1 + |Z_k^1|) \tag{5.17}$$

At the second sensor, the number of Gaussian components is

$$J_k^2 = J_k^1(1 + |Z_k^2|) = (J_{k-1} + J_{\gamma,k})(1 + |Z_k^1|)(1 + |Z_k^2|) \tag{5.18}$$

Hence, the number of of Gaussian components in $v_k(x)$ is

$$J_k = J_k^Q = (J_{k-1} + J_{\gamma,k})(1 + |Z_k^1|) \cdots (1 + |Z_k^Q|) \tag{5.19}$$

The number of Gaussian components J_k in the GMPHD filter with multiple sensors increases with the time. This causes high computations. So, at each time, methods to reduce the number of Gaussian components are required. There are some rules

to reduce the number of Gaussians, such as those that have small weights will be discarded, those that are close together will be merged into one, and if the number of Gaussian components is over a threshold L, the first L Gaussian components with high weights will be chosen for propagating in the next iteration (see [102] for more details of these rules).

5.4 Time delay of arrival measurement for multiple-speaker tracking

There are many methods to estimate the time delay of arrival (TDOA) measurement for each pair of microphones, such as the adaptive eigenvalue decomposition algorithm [8], and the well-known generalized cross correlation function (GCC) [54]. However, these techniques are applied for estimating the TDOA for one speaker. In [61], the authors extended the GCC method to collect measurements for multiple-speaker tracking. The technique is described as follows.

Let $s_n(t)$ be the signal due to speaker n, and $y_1(t)$, $y_2(t)$ are the signals received at the first and second microphones of a microphone pair. Assuming there are N speakers, the received signals $y_1(t)$ and $y_2(t)$ can be modeled as

$$y_1(t) = \sum_{n=1}^{N} s_n(t - \Delta_{1,n}) + v_1(t) \tag{5.20}$$

$$y_2(t) = \sum_{n=1}^{N} s_n(t - \Delta_{2,n}) + v_2(t) \tag{5.21}$$

where $v_i(t)$ is a noise signal present at microphone i, and $\Delta_{i,n}$ is the time it takes the sound to propagate from speaker n to microphone i. The time delay of arrival (TDOA) due to speaker n is defined for a given microphone pair as:

$$\tau_n = \Delta_{2,n} - \Delta_{1,n} \tag{5.22}$$

The goal of the collecting measurement step is to find these TDOAs due to multiple speakers.

The GCC method is applied to find the TDOA of multiple speakers. The GCC function is obtained as:

$$\hat{R}(\tau) = \int_{-\infty}^{+\infty} \psi_{12}(w) Y_1(w) Y_2^*(w) e^{jw\tau} dw \tag{5.23}$$

where $Y_1(w), Y_2(w)$ are the Fourier transforms of $y_1(t), y_2(t)$ respectively, τ is the time delay, and $\psi_{12}(w) = \dfrac{1}{|Y_1(w)Y_2^*(w)|}$. In the presence of multiple speakers, there are multi-path signal propagations and the GCC function in Equation (5.23) is composed of cross correlations of the various paths. Hence, some of the peaks of the GCC function are expected to be contributed by the direct path components of speaker sources. By collecting some local maximum peaks in the GCC function, we have a set of measurements for multiple-speaker tracking. Figure 5.1 shows an example to collect TDOA measurements at a microphone pair (for example microphone pair 2).

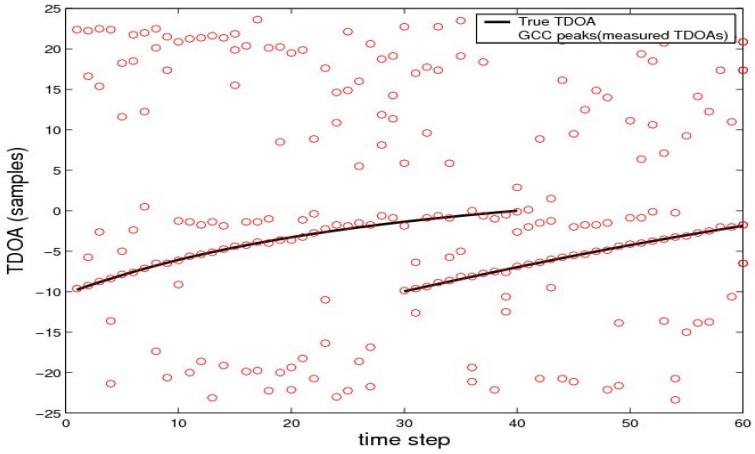

Figure 5.1: TDOA measurements for multiple speaker tracking

5.5 GMPHD filter for multiple-speaker tracking

Firstly, we define the state space model for multiple-speaker tracking. Each speaker follows a dynamic model equation

$$x_k = Ax_{k-1} + w_{k-1} \tag{5.24}$$

where A is a pre-specified matrix, and w_{k-1} is an uncorrelated noise. We assume $A = [I]$ and $w_{k-1} \sim \mathcal{N}([0;0],\text{diag}([0.01;0.01]))$. This means the average distance from the previous time $k-1$ to k of a speaker is about 10 cm. Given a speaker state x_k, TDOA measurement z_k^q is measured from the qth microphone pair at time k. The measurement equation is

$$z_k^q = \frac{\|x_k - p_{2,q}\| - \|x_k - p_{1,q}\|}{c} + v_k^q, \qquad q = 1, ..., Q \tag{5.25}$$

where $p_{i,q}$ is the position of microphone i of pair q, c is the speed of sound, and v_k^q is an uncorrelated noise and follows a Gaussian distribution with zero mean and variance σ_v^2. In this context, we assume $v_k^q \sim \mathcal{N}(0; 4 \times 10^{-9})$. The sampling frequency is 16000 Hz. This means the average time delay noise is the time for delay of 1 sample.

Secondly, at each time k, RFS measurement Z_k is obtained by collecting TDOA measurements at microphone pairs. The method to collect TDOA measurements is described in Section 5.4. From the PHD at time $k-1$, $v_{k-1}(x)$ and RFS measurement Z_k, we apply the GMPHD filter for multiple sensors that was proposed in Section 5.3 to obtain the state estimates of speakers. Because measurement equation for speaker tracking (5.25) is not linear, we have to apply an unscented transform to approximate a linear system [102].

5.6 Experimental results

5.6.1 GMPHD filter with multiple sensors for bearing and range tracking

First, we consider a bearing and range tracking application to demonstrate the effectiveness of the GMPHD filter with multiple sensors. There are objects that appear and disappear at different times. Each object has the survival probability

$p_{S,k} = 0.99$ and follows a nonlinear nearly constant turn model [102] in which the object state takes the form $x_k = \left[y_k^T, \omega_k\right]^T$, where $y_k = [p_{x,k}, p_{y,k}, \dot{p}_{x,k}, \dot{p}_{y,k}]^T$ is the coordinate (x, y) and velocity in each dimension of object, and ω_k is the turn rate. The state dynamic equations are given by

$$\begin{aligned} y_k &= F(\omega_{k-1})y_{k-1} + G\omega_{k-1}, \\ \omega_k &= \omega_{k-1} + \Delta u_{k-1}, \end{aligned} \tag{5.26}$$

where $\Delta = 1s$, $\omega_k \sim \mathcal{N}(\cdot; 0, \sigma_w^2 I_2)$, $\sigma_w = 15$ m/s^2, $u_k \sim \mathcal{N}(\cdot; 0, \sigma_u^2)$, $\sigma_u = \pi/180$ rad/s,

$$F(\omega) = \begin{bmatrix} 1 & 0 & \frac{\sin\omega\Delta}{\omega} & -\frac{1-\cos\omega\Delta}{\omega} \\ 0 & 1 & \frac{1-\cos\omega\Delta}{\omega} & \frac{\sin\omega\Delta}{\omega} \\ 0 & 0 & \cos\omega\Delta & -\sin\omega\Delta \\ 0 & 0 & \sin\omega\Delta & \cos\omega\Delta \end{bmatrix}, G = \begin{bmatrix} \frac{\Delta^2}{2} & 0 \\ 0 & \frac{\Delta^2}{2} \\ \Delta & 0 \\ 0 & \Delta \end{bmatrix}$$

We assume no spawning and that the spontaneous birth RFS is Poisson with intensity

$$\gamma_k(x) = 0.1\mathcal{N}(x; m_\gamma, P_\gamma)$$

where

$$\begin{aligned} m_\gamma &= [0; 0; 2000; 0; 0]^T, \\ P_\gamma &= \text{diag}([2500, 2500, 2500, 2500, (6\pi/180)^2]^T). \end{aligned}$$

Each object has a probability of detection $p_{D,k} = 0.98$. The observations consist of bearing and range measurements from two sensors. The positions of the sensors

are

$$p_s^1 = [0; 0] \tag{5.27}$$

$$p_s^2 = [1000; 1000] \tag{5.28}$$

The observation model at sensor i is given by

$$z_k^i = \begin{bmatrix} \arctan\left(\frac{p_{x,k}-p_{s,x}^i}{p_{y,k}-p_{s,y}^i}\right) \\ \sqrt{(p_{x,k}-p_{s,x}^i)^2 + (p_{y,k}-p_{s,y}^i)^2} \end{bmatrix} + \epsilon_k, \tag{5.29}$$

where $\epsilon_k \sim N(.; 0, R_k)$ with R_k =diag($[\sigma_\theta^2, \sigma_r^2]^T$), $\sigma_\theta = \pi/30$ rad/s and $\sigma_r = 10$ m. The clutter RFS follows the uniform Poisson model over the surveillance region $[-\pi/2, \pi/2]$ rad $\times[0, 3000]$ m, with $\lambda_c = 1.1 \times 10^{-3}\text{radm}^{-1}$ (i.e., an average of 10 clutter returns on the surveillance region). The pruning parameters for the GMPHD filters are $T = 10^{-5}$, merging threshold $U = 4$, and maximum number of Gaussian components $J_{max} = 100$.

Figure 5.2 and 5.3 show the position estimations with measurements from sensor 1 and 2, respectively. Because of the high clutter and high noise, there are some errors in the filter outputs. Figure 5.4 shows the position estimations of the GMPHD filter with multiple sensors. The performance of the GMPHD filter with fusion of multiple sensors outperform with the GMPHD filter with single sensor.

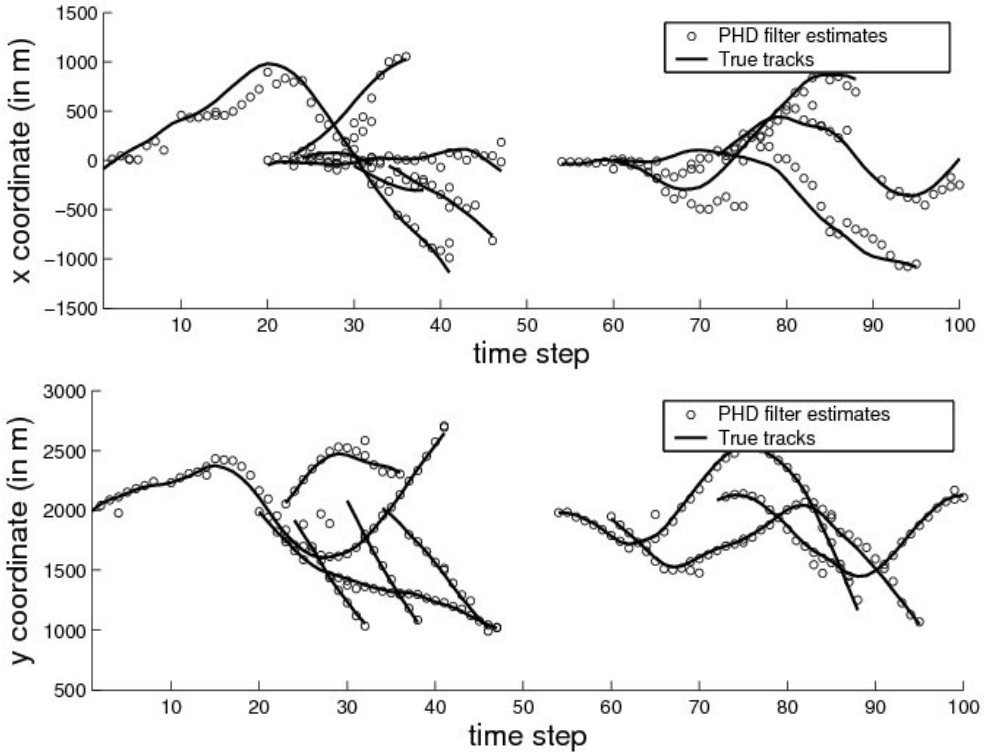

Figure 5.2: Position (x, y) of objects with measurements from sensor 1

5.6.2 GMPHD filter for multiple-speaker tracking

We simulate an acoustic room to test the performance of the GMPHD filter in tracking multiple speakers. The dimensions of the room are 3m × 3m × 2.5m. There are four microphone pairs, each of them has an inter-sensor spacing of 0.5m. The speaker sources are all female. The acoustic image method [2] is used to simulate the room impulse responses. The reverberation time of the room impulse responses is about $T_{60} = 0.15$s. The speech signal to noise ratio is about 20dB. There are 60 frames. The time frame length for measuring TDOA is 256ms, and

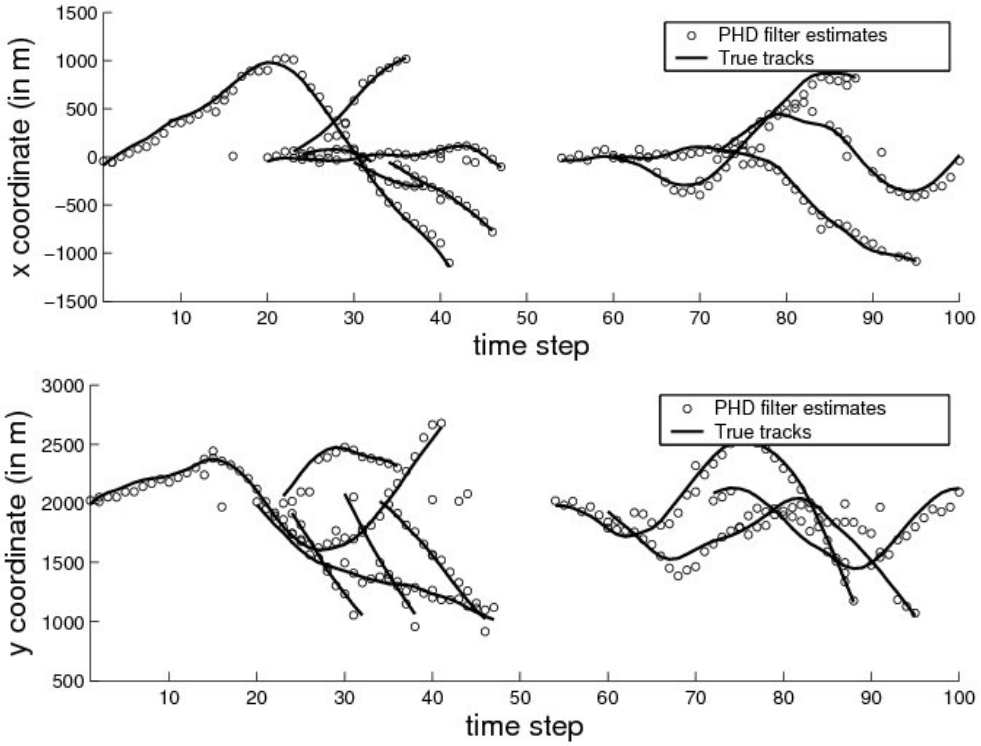

Figure 5.3: Position (x, y) of objects with measurements from sensor 2

they are non-overlapping. There are two speakers. They appear and disappear at different times.

The parameter settings for the GMPHD filter are as follows. The probability of survival $p_S = 0.95$. The probability of detection $p_D = 0.7$ that are set by experiments. The pruning parameters for the GMPHD filter is $T = 10^{-5}$, the merging threshold $U = 10$, and the maximum number of Gaussian components $J_{max} = 30$. These parameters are set for reducing the number of Gaussian components that helps to improve the speed of the algorithm. The clutter density is the uniform distribution on the range of TDOA of microphone pairs.

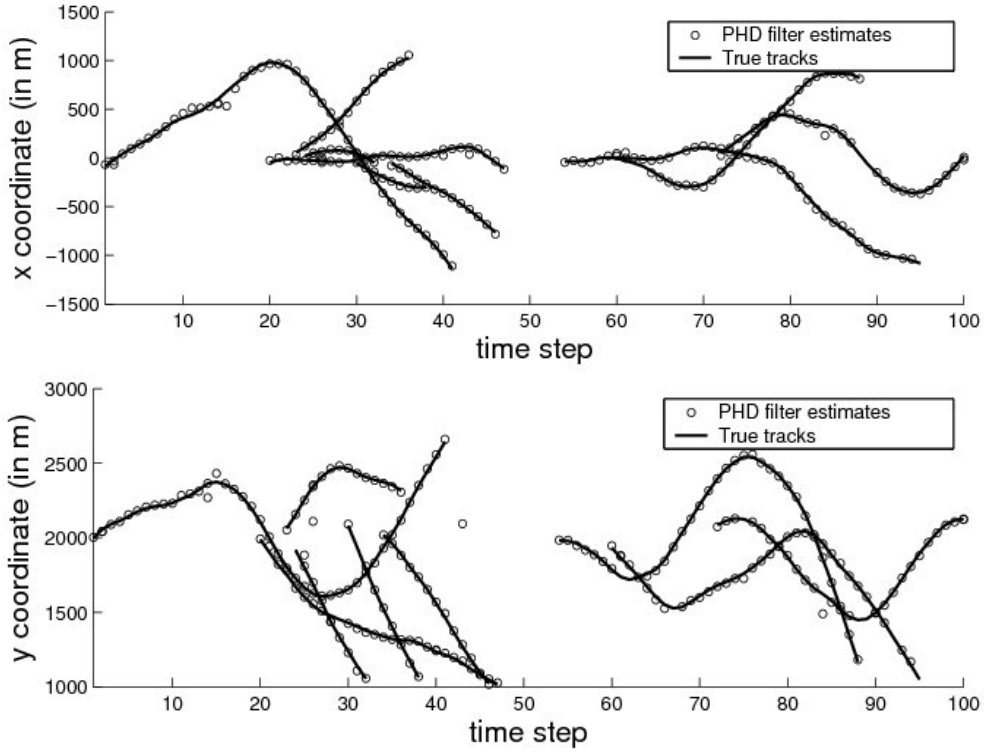

Figure 5.4: Position (x, y) of objects with the fusion method

Figures 5.5 shows the multiple-speaker tracking performance of the particle filter in [98]. Because the simulated acoustic room is reverberant, the steer-beamforming method to detect measurements in [98] is not efficient due to multi-path. In particular, when two people speaks simultaneously in this data set, the measurements from speakers are not correct. Thus, this tracking method does not perform well.

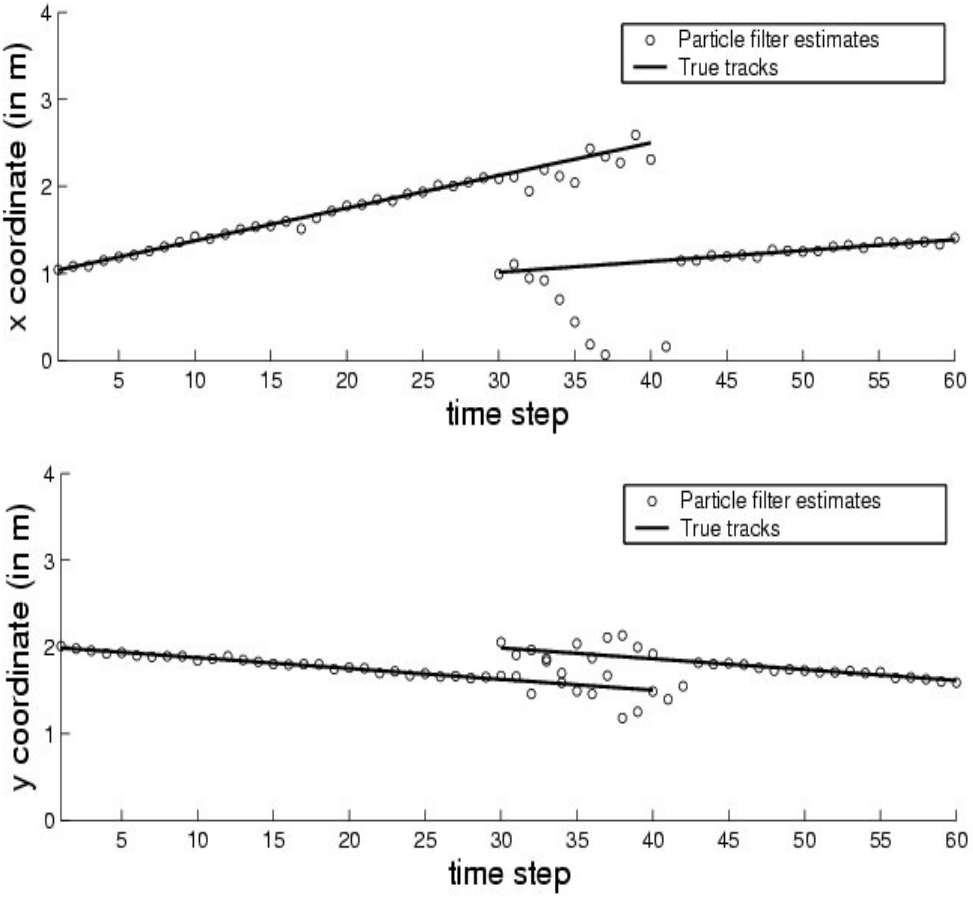

Figure 5.5: Position (x, y) of speakers with the particle filter in [98]

Figures 5.6 and 5.7 show the multiple-speaker tracking performance of the particle PHD filter [104]. Because of the unreliability in the clustering, the state estimates are affected.

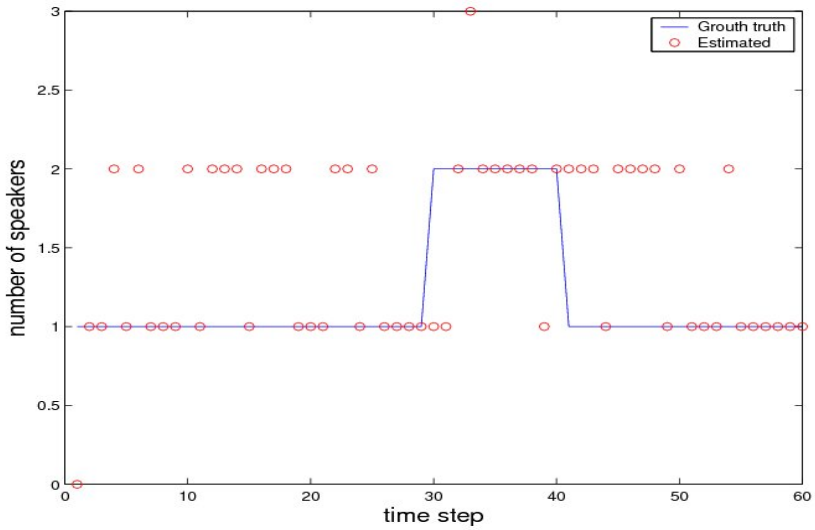

Figure 5.6: Number of speakers by the particle PHD filter

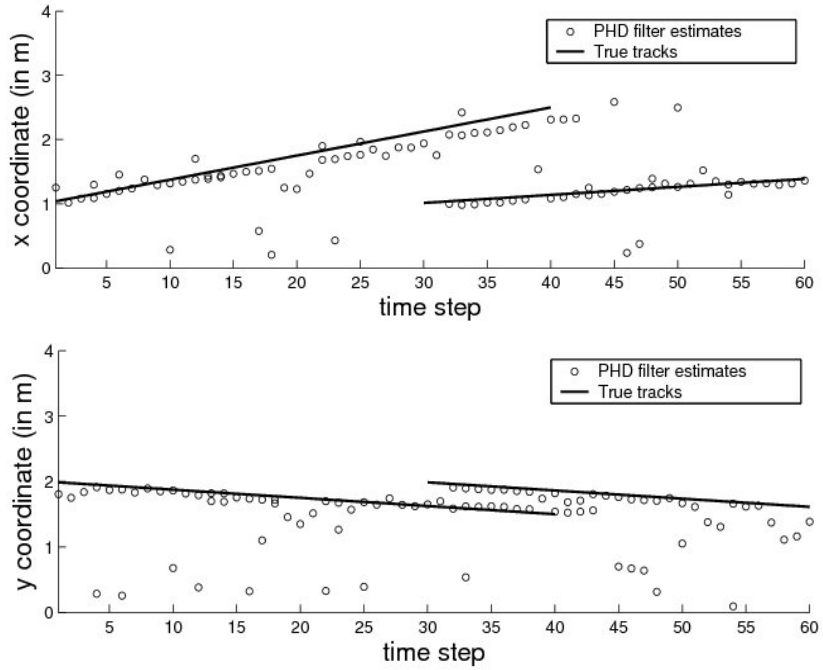

Figure 5.7: Position (x, y) of speakers with the particle PHD filter

Figures 5.8 and 5.9 show the multiple-speaker tracking performance of the RFS sequential Monte Carlo (RFS-SMC) Bayes filter [61]. The results show that this method is better than the particle PHD filter. However, the RFS-SMC Bayes filter is computationally expensive due to the large number of samples and the calculation of the RFS likelihood function by using a finite set statistic. The computation of the RFS-SMC Bayes is exponentially growing with the number of speakers or measurements.

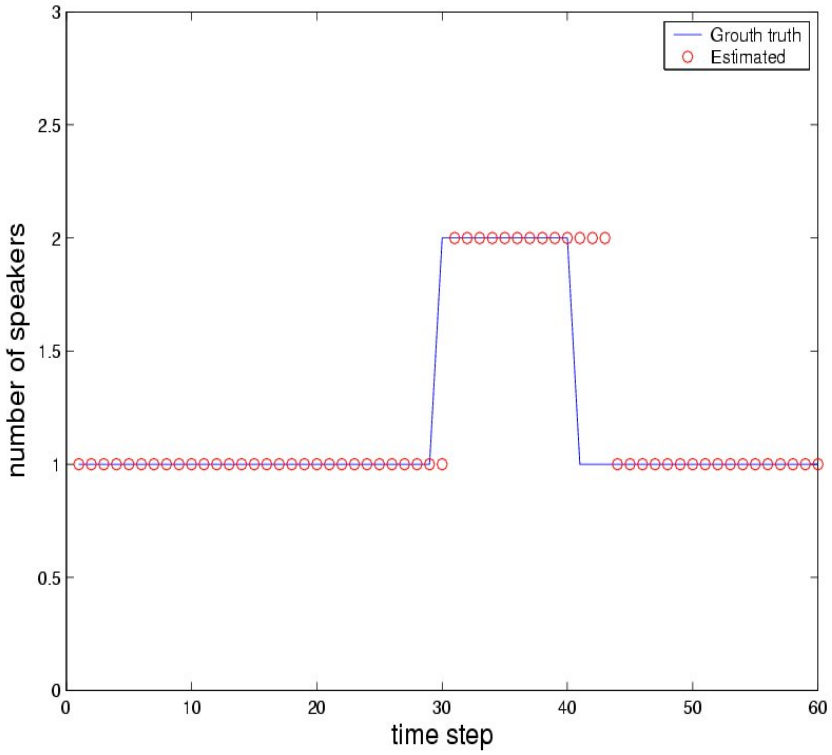

Figure 5.8: Number of speakers by the RFS-SMC Bayes filter

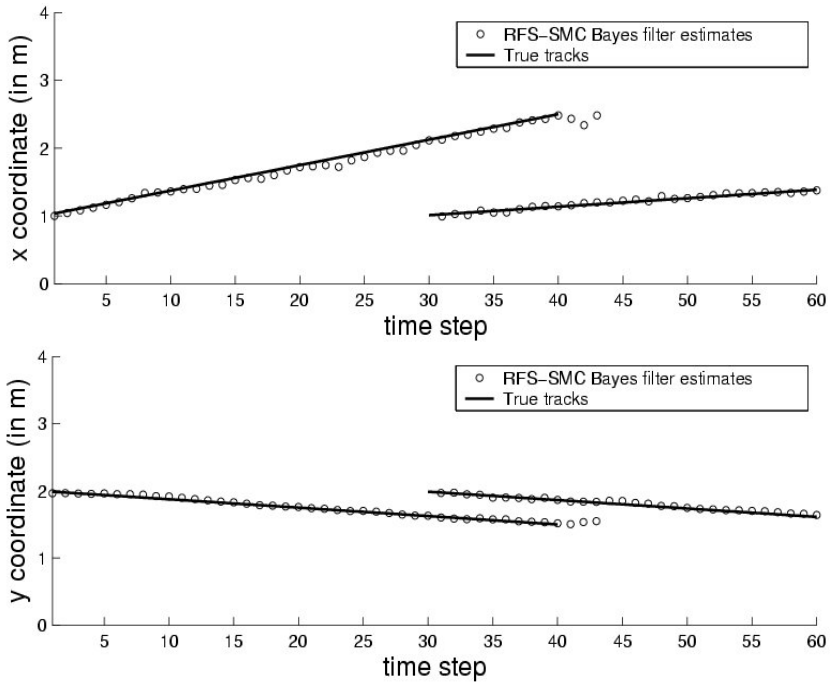

Figure 5.9: Position (x, y) of speakers with the RFS-SMC Bayes filter

Figures 5.10 and 5.11 show the multiple-speaker tracking performance of our method. This performance is better than the method in [98], the particle PHD filter and is similar to the RFS-SMC Bayes filter under the same parameters. In most of the time when two persons speak simultaneously, our method can give reliable estimations. This is because the state estimates in the GMPHD filter are the means of Gaussian components that have high weights. Hence, this method is not affected by errors from clustering techniques. Moreover, the complexity of the GMPHD filter in the multiple-sensor environment is $O\left(Q.\left|Z\right|.J_{\max}\right)$, where Q is the number of sensors, $|Z|$ is the maximum number of measurements, and

$J_{\max}$ is the maximum number of Gaussian components. It is less computationally expensive than the RFS-SMC Bayes filter $QN_t \sum_{i=1}^{\min(M,|Z|)} C_M^i A_{|Z|}^i |Z|$, where N_t is the number of samples, and the particle PHD filter $O(Q \cdot N_t \cdot |Z|)$.

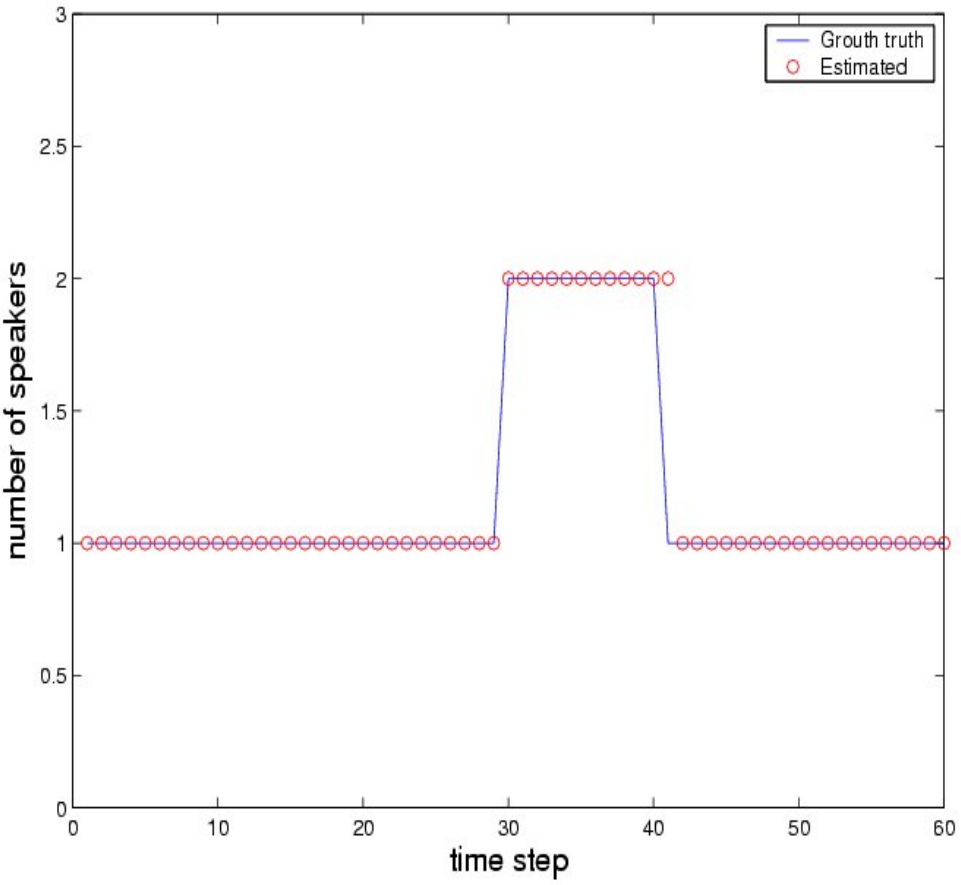

Figure 5.10: Number of speakers by the GMPHD filter

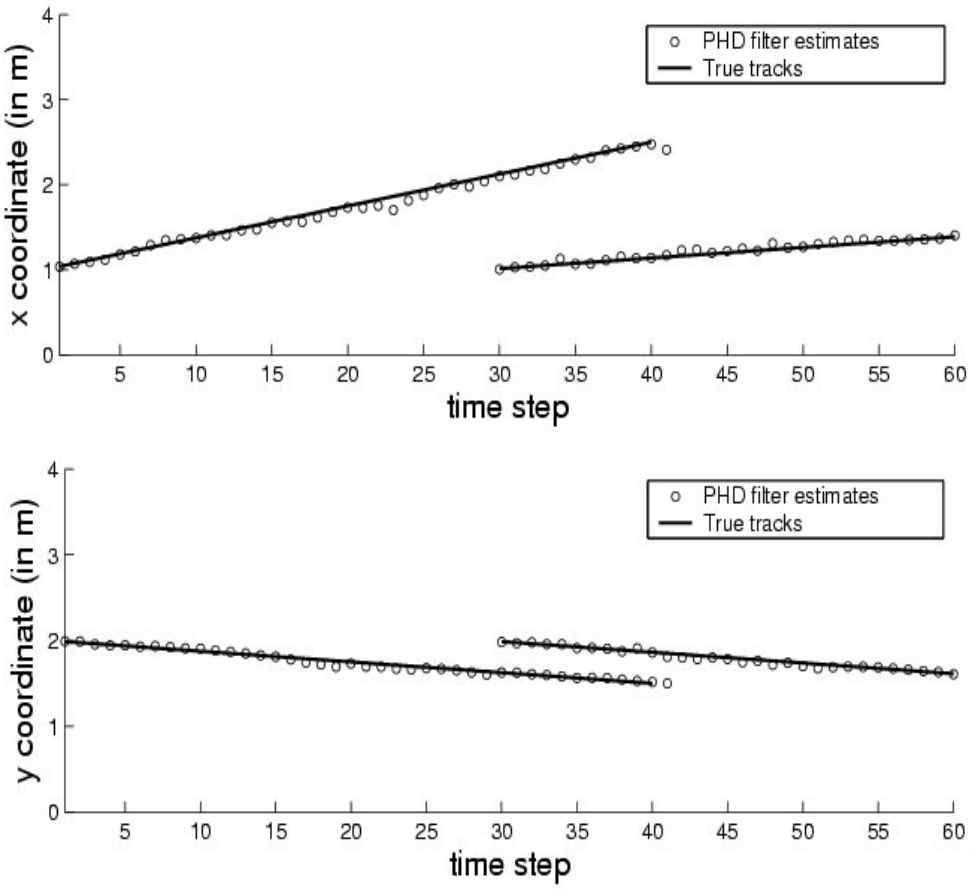

Figure 5.11: Position (x, y) of speakers with the GMPHD filter

The above result is the performance for one trial. To measure the average performance, we use the performance measurement from [61]. It includes the probability of correct speaker number, expected absolute error on the number of speaker and conditional mean distance error by Wasserstein distance. The probability of correct speaker number is defined by

$$P(|\hat{X}_k| = |X_k|) = \frac{\text{Number of } |\hat{X}_k| = |X_k|}{\text{Number of trials}} \tag{5.30}$$

where $\hat{X}_k$ is the estimation of multi-speaker state and X_k is ground-truth. The expected absolute error on the number of speaker is

$$E(|\hat{X}_k| - |X_k|) \tag{5.31}$$

When $|\hat{X}_k| = |X_k|$, the Wasserstein distance between $\hat{X}_k$ and X_k is defined as follows

$$d(X_k, \hat{X}_k) = \inf_{C} \left(\sum_{x_i \in X_k} \sum_{\hat{x}_j \in \hat{X}_k} d(x_i, \hat{x}_j)^P \right)^{1/P} \tag{5.32}$$

where C represents an $|\hat{X}_k| \times |X_k|$. The conditional mean distance error is defined

$$E\{d(X_k, \hat{X}_k) | \text{correct speaker number estimate}\} \tag{5.33}$$

We test the performance with 500 trials. Each trial is a new signal and a new TDOA measurement set. Figures 5.12 and 5.13 show the probability of correct speaker number and expected absolute error in estimation of number of speaker compared between our method and the particle PHD filter, the RFS-SMC Bayes filter. Our method is more stable than others. The main error in our method occurs due to TDOA measurements are not reliable in some time steps for example when two people speak simultaneously. Figures 5.14 shows the conditional mean distance error of speaker tracking. Our method is also more accurate than the others.

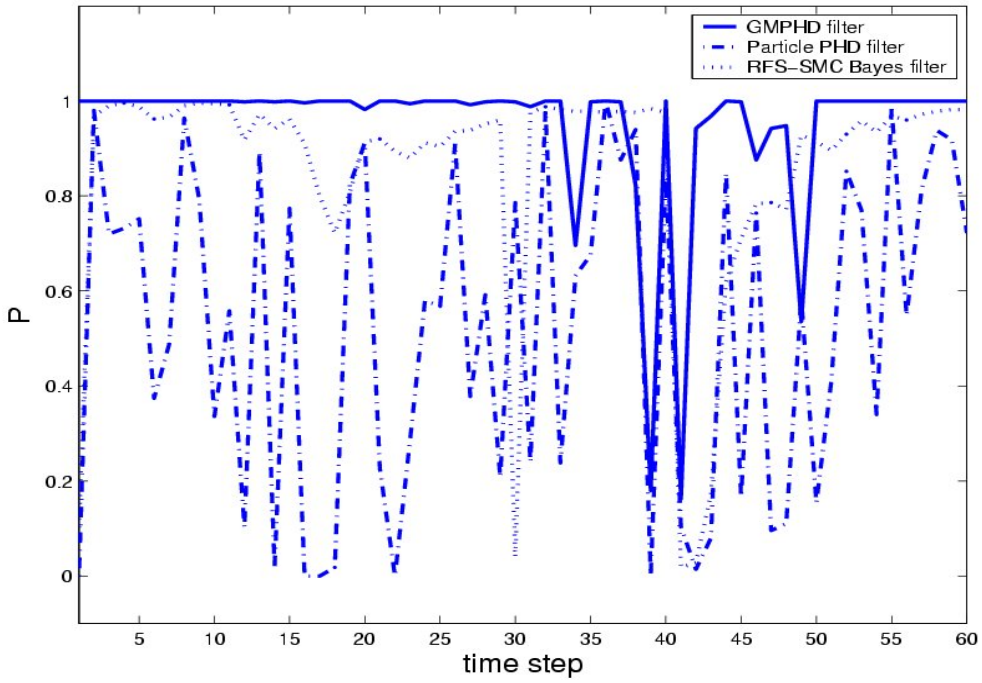

Figure 5.12: Probability of correct speaker number

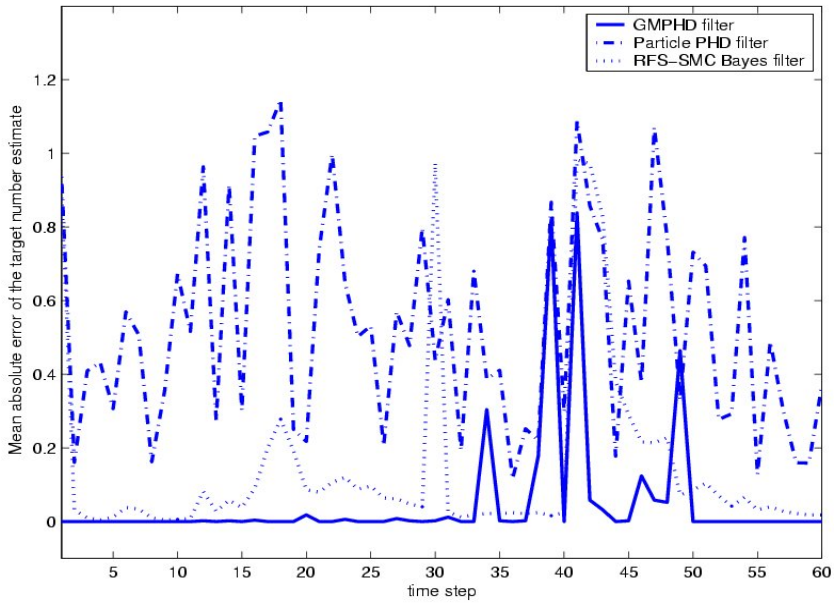

Figure 5.13: Absolute error on the number of speaker

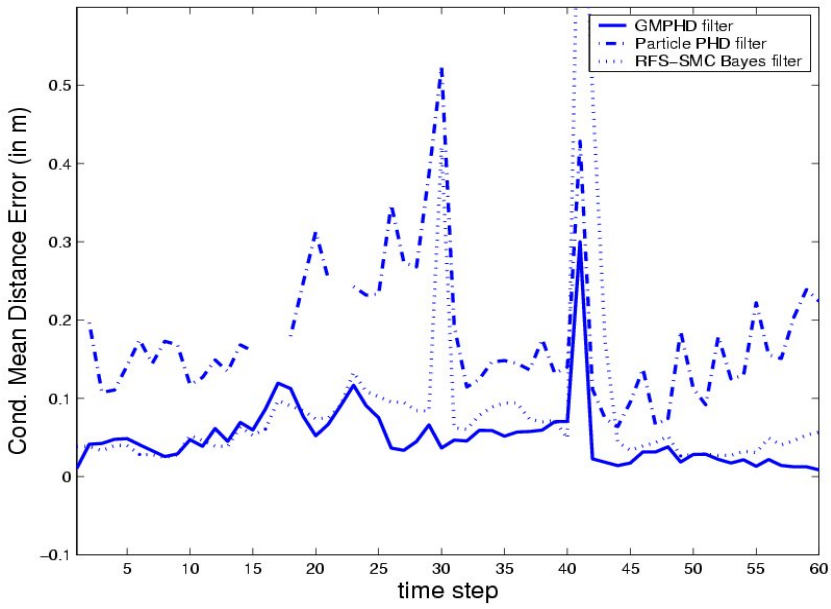

Figure 5.14: Conditional mean distance error of multiple-speaker tracking

Our implementation runs under Matlab 6.0 on a Pentium IV 2.6 GHz, 512M RAM computer. The run-time for 60 time steps is 11.2 s. This means 0.19 s for one frame (0.256 s). The run-time of the RFS-SMC Bayes filter and the particle PHD filter are 14.2 s and 31.5 s, respectively. Hence, this method is fast and it can be used in real-time applications.

5.7 Summary

Tracking multiple speakers is a challenging problem. In this chapter, we developed a reliable and computationally tractable approach to multiple-speaker tracking. The GMPHD filter was applied in multiple-speaker tracking. Using simulation

data, we demonstrated that the GMPHD filter was more efficient than some other methods in a reverberant acoustic environment. To improve the performance of the GMPHD filter in multiple-speaker tracking, the investigation on accurate and robust acoustic measurements is needed.

Chapter 6

Multiple-object tracking using the PHD filter and color measurements

6.1 Introduction

Tracking moving objects in video sequences is important in many applications, e.g., tracking players in sport sequences [72], [100], surveillance [91], and many more. Video tracking is challenging due to many factors, including measurement noise, inaccurate modelling, and clutter problem. They cause the uncertainty in the estimations of object states. To adequately capture the uncertainty due to these factors, a probabilistic framework can be used.

One particularly popular approach is the Bayes filter. In this filter, the state

estimates are obtained from the posterior density function. The Bayes filter propagates the posterior density function with the time. The Markov dynamic model is assumed to describe how the object state evolves, and a model to evaluate the likelihood of a hypothesised state given the observed data is assumed. However, the Bayes filter in multiple-object tracking is computationally expensive and sometimes cannot be represented analytically. Hence, approximation methods are often used.

Among approximation methods for the Bayes filter, methods using the sequential Monte Carlo implementation attract substantial interest. In this approach, the posterior density function is represented by a set of particles. These particles are weighted by observation models that can be obtained from background model [47], [91], or color model [28]. Some of these methods operate on a single-object state space [72], [100]. In these methods, the mixture filtering distribution is defined from the filtering distribution of each object and coefficient weights. The mixture filtering distribution is approximated to maintain multi-modality by the particle filter. However, a common limitation of these methods is that if objects are close to each other and particles from one specified object have very high weight, the particles representing the remaining objects are often suppressed. In addition, there are methods using a joint state space for tracking [28], [47]. The number of dimensions of a multiple-object state space is the multiplication between the number of objects and the number of dimensions of single-object state space. For example,

if the number of objects is 9 and the number of dimensions of single-object state space is 4, the number of dimensions of multiple-object state space is 36. Hence, the state space of multiple-object tracking by using the joint state space is very large. Sampling particles from the joint state space can become inefficient as the number of dimension of the space increases. Although there are some attempts to reduce the number of particles such as [91], it is still computationally demanding.

In this chapter, we propose a method for tracking multiple objects from video data using the probability hypothesis density filter on color measurements. A method to obtain the PHD with color histogram measurements is presented. This method is based on a hypothesis intensity function that is used to obtain the color measurement set. We assume that we have color histogram models of objects under tracking which can be obtained from the training stage or initialization stage. Then, the proposed tracking can be efficiently applied for tracking varying number of objects. The advantages of the method are that it operates on single-object state space and can be employed in applications that methods based on background subtraction fails due to a lot of clutters. The proposed method can be used for the analysis of different type of video, such as sports video, home video and surveillance video.

6.2 Color likelihood

The state of single object is described by $x = \{x_c, y_c, H_x, H_y\}$. This is a rectangle with the center and size defined by $\{x_c, y_c\}$ and $\{H_x, H_y\}$, respectively. Let the color histogram of object be denoted as $p(u)$, the color histogram of template as $q(u)$. The similarity function between an object and a template is measured by the Bhattacharyya distance [23],

$$D = \sqrt{1 - \int \sqrt{p(u)q(u)}du} \tag{6.1}$$

In multiple-object tracking, we can have many color models of templates, and let these models be as $\{q_1(u), q_2(u), ..., q_n(u)\}$. The similarity function between an object and the templates is modified by

$$D = \min_i \left(\sqrt{1 - \int \sqrt{p(u)q_i(u)}du} \right) \tag{6.2}$$

We use the RGB color system, so the distance D is

$$D = \min_i \left(D_R^i + D_G^i + D_B^i \right) \tag{6.3}$$

where D_R^i, D_G^i, D_B^i are the Bhattacharyya distances between the object model and templates on the R, G, B color channels, respectively.

The color likelihood function is defined by

$$p(z|x) = l_z(x) = \mathcal{N}(D; 0, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} exp\left\{ -\frac{D^2}{2\sigma^2} \right\} \tag{6.4}$$

where z is the current image and σ^2 is a variance of noise.

6.3 Random finite set formulation for color object tracking

By assuming that the state of object does not change a lot between frames, each object in multiple-object tracking is evolved from a dynamic moving equation as follows:

$$x_k = x_{k-1} + w_k \tag{6.5}$$

where w_k is the process noise. Let the multiple-object state be $X_k = \{x_{1,k}, x_{2,k}..., x_{N_k,k}\} \in \mathcal{F}(\mathcal{X})$, where $\mathcal{F}(\mathcal{X})$ denotes the collection of all finite subsets of the single-object state space $\mathcal{X}$. Let Z_k be the image frame at time k. Color tracking is to track objects described by specified color representations q^*, e.g., histograms. More specific, the multiple-object tracking problem is to find the multiple-object state estimate $\hat{X}_k$ from the posterior density function $p(X_k|Z_{1:k})$,where the objects have the color histograms similar to q^*. However, it is not easy to obtain the posterior density function $p(X_k|Z_{1:k})$ when the state space is too large. Fortunately, it can be approximately recovered from the first moment of this distribution, the PHD. To obtain the PHD, we need to represent measurements as a random finite set. It is difficult to represent color histograms as a RFS directly. The following sections in this chapter will present a method to obtain the PHD with color measurements in video tracking.

6.4 Hypothesis intensity function for color tracking

In this section, we propose a hypothesis intensity function that is used for the multiple-object tracking algorithm with video data. The generating probability functional (gpf) of the posterior density is defined in [29] as

$$G_k[h] = E\left[\prod_{x_k \in X_k} h(x_k)\right] \tag{6.6}$$

We let $\prod_X[h] = \prod_{x \in X} h(x)$, the gpf $G_k[h]$ becomes

$$\begin{aligned} G_k[h] &= E\left[\prod_{X_k}[h]\right] \\ &= \int \prod_{X_k}[h]\, p(X_k|Z_{1:k})\, \mu(X_k) \end{aligned} \tag{6.7}$$

where μ is a dominating measure [103]. From the Bayes rule, we have

$$p(X_k|Z_{1:k}) = \frac{g(Z_k|X_k)\, p(X_k|Z_{1:k-1})}{\int g(Z_k|X)\, p(X|Z_{1:k-1})\, \mu(X)} \tag{6.8}$$

Hence,

$$\begin{aligned} G_k[h] &= \int \prod_{X_k}[h]\, p(X_k|Z_{1:k})\, \mu(X_k) \\ &= \frac{1}{\int g(Z_k|X)\, p(X|Z_{1:k-1})\, \mu(X)} \int \prod_{X_k}[h]\, g(Z_k|X_k)\, p(X_k|Z_{1:k-1})\, \mu(X_k) \end{aligned} \tag{6.9}$$

We let $K = \int g(Z_k|X)\, p(X|Z_{1:k-1})\, \mu(X)$,

$$G_k[h] = \frac{1}{K} \int \prod_{X_k}[h]\, g(Z_k|X_k)\, p(X_k|Z_{1:k-1})\, \mu(X_k) \tag{6.10}$$

From the assumption in [28], the multiple-object color likelihood is as follows

$$\begin{aligned} g\left(Z_k|X_k\right) &\propto \prod_{x_k \in X_k} g(Z_k|x_k) \\ &\propto \prod_{x_k \in X_k} l_{Z_k}(x_k) \end{aligned} \tag{6.11}$$

This means given an image Z_k, the function $l_{Z_k}(x_k)$ described in Section 6.2 represents the dependence of a state x_k on image Z_k. The multiple-object color likelihood in Equation (6.11) is the multiplication of all $l_{Z_k}(x_k)$ where $x_k \in X_k$.

The gpf $G_k\left[h\right]$ can be re-written by

$$\begin{aligned} G_k\left[h\right] &\propto \frac{1}{K}\int \prod_{X_k}\left[h\right]\prod_{X_k}\left[l_{Z_k}\right] p\left(X_k|Z_{1:k-1}\right)\mu\left(X_k\right) \\ &\propto \frac{1}{K}\int \prod_{X_k}\left[hl_{Z_k}\right] p\left(X_k|Z_{1:k-1}\right)\mu\left(X_k\right) \\ &\propto \frac{1}{K} G_{k|k-1}\left[hl_{Z_k}\right] \end{aligned} \tag{6.12}$$

We assume that $p\left(X_k|Z_{1:k-1}\right)$ is Poisson [63], the generating function $G_{k|k-1}\left[h\right]$ has the form

$$G_{k|k-1}\left[h\right] = e^{v_{k|k-1}\cdot(h-1)} \tag{6.13}$$

where $v_{k|k-1}$ is the predicted intensity function and $v_{k|k-1}\cdot h = \int v_{k|k-1}\left(dx\right)h\left(x\right)$. Hence,

$$G_k\left[h\right] \propto \frac{1}{K} e^{v_{k|k-1}\cdot\left(hl_{Z_k}-1\right)}$$

Let $\Phi\left[h\right] = v_{k|k-1}\cdot\left(hl_{Z_k}-1\right)$, the derivative functional of $\Phi\left[h\right]$ is

$$\left(d\Phi\right)_h\left[\varsigma\right] = v_{k|k-1}\cdot\left(\varsigma l_{Z_k}\right) \tag{6.14}$$

Thus, the derivative functional of $G_k[h]$ is

$$(dG_k)_h[\varsigma] \propto \frac{1}{K} v_{k|k-1} \cdot (\varsigma l_{Z_k}) \, e^{v_{k|k-1} \cdot \left(h l_{Z_k} - 1\right)} \tag{6.15}$$

We know from [29] that

$$v_k(x) = (dG_k)_1[\delta_x] \tag{6.16}$$

From Equation (6.15), we have

$$\begin{aligned} (dG_k)_1[\delta_x] &\propto \frac{1}{K} v_{k|k-1}(x) l_{Z_k}(x) \, e^{\int v_{k|k-1}(dx)\left(h(x) l_{Z_k}(x) - 1\right)} \\ &\propto v_{k|k-1}(x) l_{Z_k}(x) \end{aligned} \tag{6.17}$$

Let $\tilde{v}_k(x) = v_{k|k-1}(x) l_{Z_k}(x)$, from Equations (6.17), and (6.16), we can conclude

$$v_k(x) \propto \tilde{v}_k(x) = v_{k|k-1}(x) l_{Z_k}(x) \tag{6.18}$$

From the joint state space, we have found a function $\tilde{v}_k(x)$ in the single-object state space that is a proportion to PHD $v_k(x)$. $\tilde{v}_k(x)$ is called hypothesis intensity function for color tracking.

6.5 GMPHD filter for color multiple-object tracking

We cannot apply directly the GMPHD filter with color measurements because obtaining the measurement random set from video is not straight-forward. Here, we

propose a PHD recursion for color measurements. This PHD recursion is described in Figure 6.1.

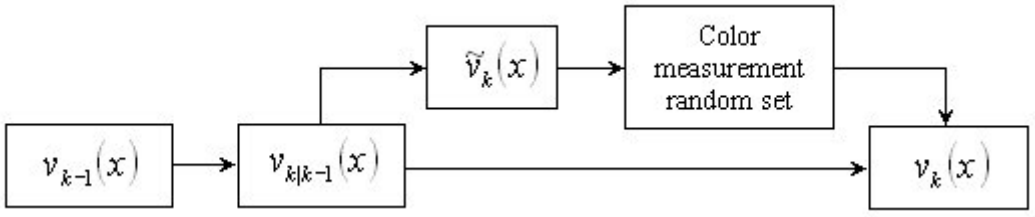

Figure 6.1: PHD recursion for color multiple-object tracking

Firstly, color models of objects are obtained from template images. The color models are the color histograms of objects. These models are used when we evaluate the color likelihood that is described in Section 6.2. From previous posterior density $v_{k-1}(x)$, prediction equation (3.17) is performed to obtain the predicted intensity at time k, $v_{k|k-1}(x)$. We propose a method to obtain $\tilde{v}_k(x)$ and color measurement random set by the Monte Carlo technique. Predicted intensity $v_{k|k-1}(x)$ can be expressed in the form

$$v_{k|k-1}(x) = \sum_{i=1}^{N} \psi^i \delta_{x^i}(x) \tag{6.19}$$

where sample x^i is drawn from $v_{k|k-1}(x)$ and has weight ψ^i, N is the number of

samples. From Equation 6.18, we have

$$\begin{aligned} \tilde{v}_k(x) &= v_{k|k-1}(x) l_{Z_k}(x) \\ &= l_{Z_k}(x) \sum_{i=1}^{N} \psi^i \delta_{x^i}(x) \\ &= \sum_{i=1}^{N} \psi^i l_{Z_k}(x) \delta_{x^i}(x) \end{aligned} \tag{6.20}$$

Hence, $\tilde{v}_k$ can be represented as

$$\tilde{v}_k(x) = \sum_{i=1}^{N} \xi^i \delta_{x^i}(x) \tag{6.21}$$

where

$$\xi^i = l_{Z_k}(x^i)\psi^i \tag{6.22}$$

Next, resample $\{x^i, \xi^i\}_{i=1}^{N}$, and group these samples into clusters. A simple clustering technique is performed. Samples that are close to each other are grouped to form m clusters. Then, the K-means algorithm is applied to adjust centers of these clusters. After that, these centers of clusters are used to create the color measurement random set

$$Z_k = \{z_1, ..., z_m\} \tag{6.23}$$

where z_i is the center of the ith cluster.

Secondly, from predicted intensity $v_{k|k-1}(x)$ and color measurement random set Z_k, we apply the updating step in the GMPHD filter to obtain posterior intensity $v_k(x)$. Then we find Gaussian components whose weights are larger than a

threshold (0.5) in posterior intensity $v_k(x)$. The set of means of these Gaussian components are state estimations.

With assumptions in Section 3.5, the tracking method is detailed as follows:

- Step 1. Prediction

The predicted intensity to time k is given by

$$v_{k|k-1}(x) = v_{S,k|k-1}(x) + \gamma_k(x) \tag{6.24}$$

where

$$\begin{aligned} v_{S,k|k-1}(x) &= p_{S,k} \sum_{j=1}^{J_{k-1}} w_{k-1}^{(j)} \mathcal{N}(x; m_{S,k|k-1}^{(j)}, P_{S,k|k-1}^{(j)}), \\ m_{S,k|k-1}^{(j)} &= F_{k-1} m_{k-1}^{(j)}, \\ P_{S,k|k-1}^{(j)} &= Q_{k-1} + F_{k-1} P_{k-1}^{(j)} F_{k-1}^T. \end{aligned}$$

$v_{S,k|k-1}(x)$ and $\gamma_k(x)$ are Gaussian mixtures, so $v_{k|k-1}(x)$ can be expressed as a Gaussian mixture of the form

$$v_{k|k-1}(x) = \sum_{i=1}^{J_{k|k-1}} w_{k|k-1}^{(i)} \mathcal{N}(x; m_{k|k-1}^{(i)}, P_{k|k-1}^{(i)}) \tag{6.25}$$

From $v_{k|k-1}$, we obtain $\tilde{v}_k$ by the Monte Carlo technique. $v_{k|k-1}$ are represented by N samples $\{x^i, \psi^i\}_{i=1}^N$. From Equation (6.21), $\tilde{v}_k$ will be represented by $\{x^i, \xi^i\}$. Then, resamples $\{x^i, \xi^i\}$, and groups these samples into clusters. The centers of clusters $Z_k = \{z_1, ..., z_m\}$ will be measurements for the next updating step.

- Step 2. Update

The posterior intensity at time k is a Gaussian mixture, and is given by

$$v_k(x) = (1 - p_{D,k})v_{k|k-1}(x) + \sum_{z \in Z_k} v_{D,k}(x; z) \tag{6.26}$$

where

$$\begin{aligned}
v_{D,k}(x; z) &= \sum_{j=1}^{J_{k|k-1}} w_k^{(j)}(z)\mathcal{N}(x; m_{k|k}^{(j)}, P_{k|k}^{(j)}), \\
w_k^{(j)}(z) &= \frac{p_{D,k} w_{k|k-1}^{(j)} q_k^{(j)}(z)}{\kappa_k(z) + p_{D,k} \sum_{l=1}^{J_{k|k-1}} w_{k|k-1}^{(l)} q_k^{(l)}(z)}, \\
q_k^{(j)}(z) &= \mathcal{N}\left(z; H_k m_{k|k-1}^{(j)}, R_k + H_k P_{k|k-1}^{(j)} H_k^T\right), \\
m_{k|k}^{(j)} &= m_{k|k-1}^{(j)} + K_k^{(j)}(z - H_k m_{k|k-1}^{(j)}), \\
P_{k|k}^{(j)} &= [I - K_k^{(j)} H_k] P_{k|k-1}^{(j)}, \\
K_k^{(j)} &= P_{k|k-1}^{(j)} H_k^T (H_k P_{k|k-1}^{(j)} H_k^T + R_k)^{-1}.
\end{aligned}$$

Note that we have associated each Gaussian component with each label by using the method in [21]. These labels are also the object identifications. Gaussian components that are near each other and have the same label are merged after the updating step. At the end of each iteration, among Gaussian components having the same label, we keep the Gaussian component that has largest weight. Another notice is that if a new object appears, this object will create a peak in hypothesis intensity function $\tilde{v}_k(x)$ because of the high value in the likelihood function with assuming that we knew the color model of this object before. Hence, the color measurement random set includes the mean of the cluster that contains this peak. Then, after the updating step, the state estimate of this new object is obtained.

In a similar manner, when an object disappears, no peak in $\tilde{v}_k(x)$ caused by this object is obtained. This leads to the small weight of the Gaussian component represented for this object and this Gaussian component is removed. Hence, the label of this object is removed. Moreover, when objects are occluded together, peaks can also be detected and used to update Gaussian components. Thus, this method can estimate states of objects when occlusions occur.

6.6 Experimental results

We test the performance of the proposed method in sequences from [28], [58], and [91]. There is a total about 9500 frames. The Wasserstein distance in [43], [61] is used to measure the performance. The errors of estimations are shown in Table 6.1

We use 400 samples to represent for the hypothesis intensity function. The maximum of Gaussian mixture components are 30. For the football sequence, we divide the tracking area into 15 parts (grid 3×5). The birth intensity is the mixture of Gaussian components whose means are centers of these parts. For other sequences, we assume that persons enter the tracking areas from entrances. Hence, the birth intensity is the mixture of Gaussian components whose means are the locations at these entrances. The clutter density is an uniform distribution of the size of image and the range of radius H_x and H_y. The probability of survival is

Sequences	Mean error (pixel)
seq24-2p-0111-cam1	7.2
seq24-2p-0111-cam2	4.8
seq35-2p-1111-cam1	4.8
seq35-2p-1111-cam2	3.9
seq44-3p-1111-cam1	8.2
seq44-3p-1111-cam2	6.1
football	7.2
seq16	9

Table 6.1: Error of estimation

$p_S = 0.99$ and the probability of detection is $p_D = 0.98$. These parameters are set by experiments. In these testing sequences, the detected number of objects are correct except that there are some delays at frames when objects begin to enter the tracking area.

Figure 6.2 shows the comparison between our method and the boosted particle filter [72] that we implemented. For the boosted particle filter, we assume that we have very good detections (from groudtruth) and the proposal coefficient $\alpha = 0.8$ (80% particles from detection distribution which means majority of particles are around the real state). However, because the likelihoods of the particles near the black person are too high, the boosted particle filter is ambiguous between

two persons and one track is lost. The results show that our method is better to maintain the tracking through the occlusion. This is because we can detect the peak caused by the white person although this peak is smaller than the peak caused by the black person. If detected peaks are not caused by persons, these peaks will be false alarms. The weights of Gaussian components caused by these peaks are small and they are removed. Otherwise, if peaks are caused by persons, state estimates of these persons will be obtained. Hence, in this case, our method is better than the boosted particle filter.

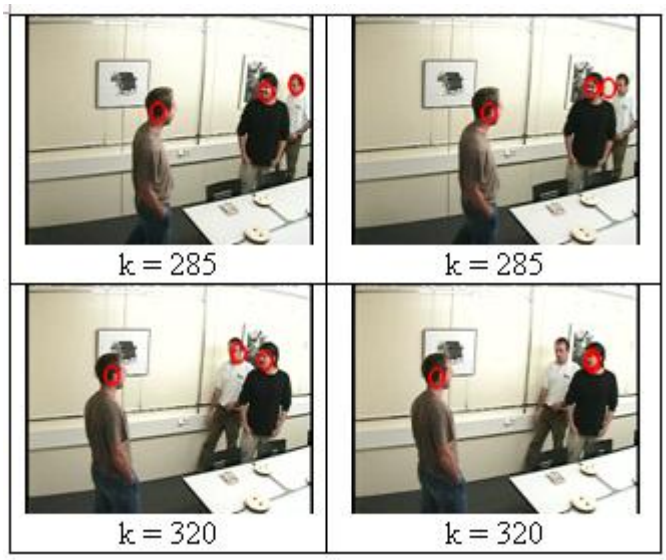

Figure 6.2: Comparison between our method (left) and the boosted particle filter (right)

Figure 6.3 shows the results of tracking white football players. In this sequence, the number of white players changes during the tracking period. When a white

player appears in the camera view, the likelihood function is high at the position of this player and we can collect the color measurement by using the hypothesis intensity function. This measurement increases the weight of a birth Gaussian and the state estimate of this player is obtained. In this sequence, the camera is moving when capturing. Hence, segmentation methods are difficult to apply. In [28], Czyz used 5000 joint state space samples for this sequence. However, we use 400 particles to obtain the color measurement random set.

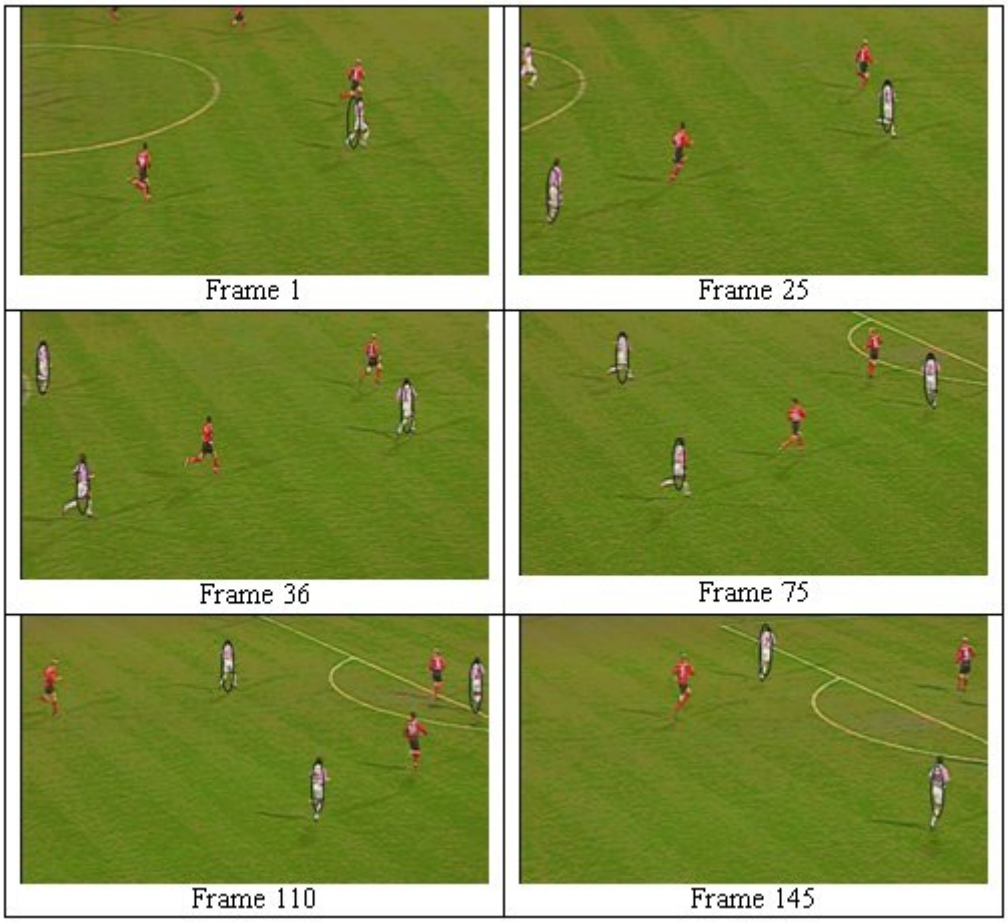

Figure 6.3: Tracking multiple players in the football sequence

Figure 6.4 shows some results of the tracking in the seq16 from [91]. In this sequence, at the beginning, there is no one in the scene. At frame 34, 78, 135 and 141, the first, second, third and fourth person enter the tracking area, respectively. They walk in two opposite directions and occlusions may occur. The results show that our method can track varying number of people in this sequence.

6.7 Summary

The chapter described a method using the GMPHD filter to track multiple objects by incorporating the color representation. It is proved that the PHD is proportional to our hypothesis intensity function for color tracking, which helps to define the color measurement random set. A PHD recursion for visual observations with color measurements is proposed. With this approach, the experiments show that the video tracking works for varying number of objects in a single-object state space, which is efficient and promising for real-time applications.

Figure 6.4: Tracking multiple persons in seq16

Chapter 7

Multiple-camera multiple-object tracking using the PHD filter

7.1 Introduction

Tracking moving objects is an important part of many applications. Some people proposed methods to track objects by using one camera [23], [28], [83]. However, when persons might be occluded by other persons in the scene, using one camera to track these persons is difficult. This is because information of these persons from one camera is not enough to solve the occlusion problem. An idea to solve this problem is to use multiple cameras to recover information that might be missing from a particular camera. Furthermore, multiple cameras can be used to recover the 3D information of objects.

There are some approaches for tracking using multiple cameras. Most of them have two stages. They are single-view stage and multiple-view data fusion stage. In the single-view stage, observations and estimations are extracted by the Kalman filter [31], the particle filter [70], segmentation methods [68], or detection methods [13]. Then in the second stage, these data are fused to obtain the final results. Some methods propose to track one object using multiple cameras [13], [70]. These methods track an object and switch to another camera when the system predicts that the current camera no longer has a good view of the object. However, these methods need to consider data association when extending from tracking one object to multiple objects. Some other methods can track multiple objects [15], [31], [53], [68]. Among them, some methods match objects between different camera views [15], [68] or incorporate classification methods [53] to do the data association between observations and objects in multiple views. These methods can collaborate multiple cameras for multiple-object tracking. However, when the appearances of objects are similar or occlusions occur, these methods might not be suitable. This is because some wrong matches may occur. An example is shown in Figure 7.1. In this figure, the color of the brown person's face in camera 1 is similar to the color of the white person's face in camera 2. Hence, the wrong match has occurred. The other idea is to find 3D observations that correspond with observations from different views [31]. However, the association of observations from different views can increase computational cost in 3D observation searching.

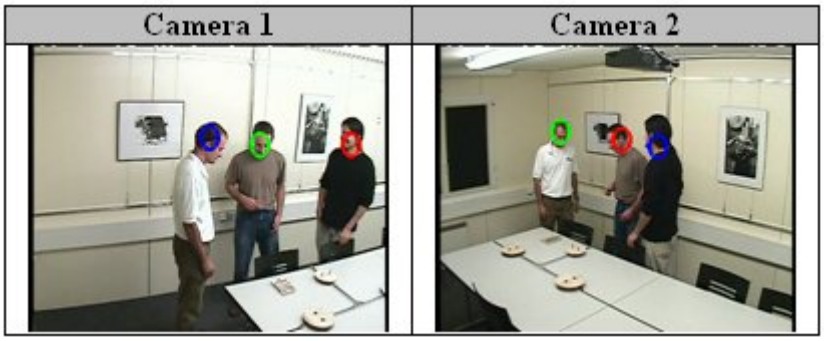

Figure 7.1: An example for wrong matching based on the apperance

Data association between observations and objects in multiple views is a challenging problem in multiple-camera multiple-object tracking. If data association is based on the appearance of objects, the changing of appearance between cameras will affect the performance. To avoid the data association problem, the PHD filter can be used. There are some approaches that use the PHD filter for video tracking [62], [111].

In this chapter, we employ the GMPHD filter with multiple sensors (more details are in Section 5.3) to track several people using multiple cameras in a room. The method includes two stages: single-view tracking and multiple-camera fusion. These two stages are based on the GMPHD filter. It is assumed that we have projection matrices from 3D space to the cameras. Our method can recover the 3D object locations and handle the occlusion at each camera. We assume that color models are available. Then, the proposed tracking method can be efficiently applied to track a varying number of objects. Our method fails when an object

is occluded in all cameras views, but it can be tracked once the occlusion is over. Further, because the fusion stage of multiple cameras to obtain 3D object locations is based on the GMPHD filter, it reduces the amount of computation compared with other methods such as search based methods or the particle filter.

7.2 System overview

We propose a method to track 3D locations of heads of people using multiple cameras with assumptions that the cameras are calibrated and the fields of views of the cameras overlap. The proposed method, as shown in Figure 7.2, consists of two major components: single view tracking and multiple-camera fusion. In the first component, at each camera at time k, we find color observations and then use the GMPHD filter to estimate the 2D locations of objects. Let $Y_k^i = \{y_{1,k}^i, ..., y_{m,k}^i\}$ be the set of 2D estimations of objects at time k, view i. We have n single views, so the set of 2D estimations of objects at time k can be defined by

$$Y_k = \left[Y_k^1; Y_k^2; ...; Y_k^n\right] \tag{7.1}$$

More details on the first step will be shown in Section 7.3.

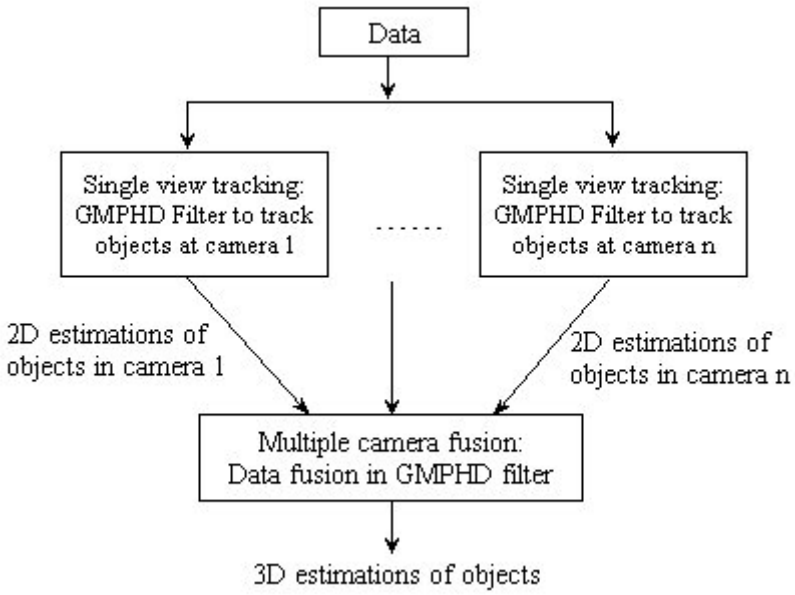

Figure 7.2: The sketch of our system for multiple object tracking using multiple cameras

In the second component, we consider the set of 2D estimations of objects Y_k as observations for a data fusion step to estimate the 3D information of objects by the GMPHD filter. This method can avoid the data association between observations and state of objects. More details of the second step will be shown in Section 7.4.

7.3 Single-view tracking

At each single view, we assume that the object state does not change much between frames and each object in multiple-object tracking is evolved from a dynamic

moving equation

$$x_k = x_{k-1} + w_{k-1} \tag{7.2}$$

where the state of an object in a single view $x_k = \{x_c, y_c, H_x, H_y\}$ is a rectangle with center $\{x_c, y_c\}$ and size $\{H_x, H_y\}$, w_k is the process noise.

Single-view tracking consists of two parts: obtaining the color measurement random set and using these color measurements to obtain the PHD. First, we train color models of the heads from template images. The color model is the color histogram of the head. These models are used when we evaluate the color likelihood. Now, we consider the ith camera. Let $v_k^i(x)$ be the PHD of the ith camera at time k and $v_{k|k-1}^i(x)$ be the predicted PHD of the ith camera at time k. From $v_{k-1}^i(x)$, we can predict the PHD $v_{k|k-1}(x)$ by assumptions on the state dynamic equation and the prediction step in the GMPHD filter. We prove that

$$v_k^i(x) \propto \tilde{v}_k^i(x) = l_z(x) v_{k|k-1}^i(x) \tag{7.3}$$

where $l_z(x)$ is the color likelihood that is defined in Section 6.2. $\tilde{v}_k^i(x)$ is the hypothesis intensity function. After that, we use Monte Carlo samples to find peaks in $\tilde{v}_k^i(x)$. These peaks are also peaks in $v_k^i(x)$. The set of these peaks is considered to be the color measurement random set.

Secondly, we use the color measurement random set to update the PHD by the updating step in the GMPHD filter. After updating the predicted PHD $v_{k|k-1}^i(x)$ with the color measurement random set, we obtain the PHD $v_k^i(x)$. From PHD

$v_k^i(x)$, we find Gaussian components whose weights are larger than a threshold (0.5). The set of means of these Gaussian components are 2D estimations of objects at the ith camera. They are denoted as $Y_k^i = \{y_{1,k}^i, ..., y_{m,k}^i\}$. (See Chapter 6 for more details of single-view tracking.)

7.4 Multiple-camera fusion

We assume that the dynamic moving equation for 3D tracking is

$$x_k = x_{k-1} + w_{k-1} \tag{7.4}$$

where the state of an object $x_k = \{x_{1,k}, x_{2,k}, x_{3,k}\}$ is a 3D coordinate, w_{k-1} is the process noise.

The observations are 2D estimations from multiple cameras. So, the measurement equation at the ith camera is described by

$$\begin{aligned}
\begin{pmatrix} l_{1,k} \\ l_{2,k} \\ l_{3,k} \end{pmatrix} &= \begin{pmatrix} a_{11}^i & a_{12}^i & a_{13}^i & a_{14}^i \\ a_{21}^i & a_{22}^i & a_{23}^i & a_{24}^i \\ a_{31}^i & a_{32}^i & a_{33}^i & a_{34}^i \end{pmatrix} \begin{pmatrix} x_{1,k} \\ x_{2,k} \\ x_{3,k} \\ 1 \end{pmatrix} \\
\begin{pmatrix} y_{1,k}^i \\ y_{2,k}^i \end{pmatrix} &= \begin{pmatrix} l_{1,k}/l_{3,k} \\ l_{2,k}/l_{3,k} \end{pmatrix} + u_k
\end{aligned} \tag{7.5}$$

where u_k is the measurement noise, and a_{mn}^i are projection parameters from 3D

coordinate to the ith camera plane. Assuming that cameras are calibrated, we have projection parameters a^i_{mn}.

The idea of fusing data from multiple cameras is to use the approximation of multiple-sensor PHD update in GMPHD filter. The idea of approximation of multiple-sensor PHD update is described in [104]. Let $V_k(x)$ be the PHD for multiple-camera multiple-object tracking at time step k. The overview of the idea is shown in Figure 7.3. Now, we describe the details of the algorithm

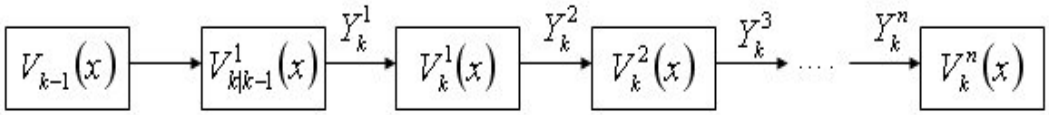

Figure 7.3: Sequential updating for PHD at cameras

- Step 1: Assuming that we have the PHDs of previous time step $k-1$ of multiple-camera fusion stage $V_{k-1}(x)$ and single-view tracking stage $v^1_{k-1}(x)$ at camera 1, we employ the method described in Section 7.3 to obtain the set of 2D estimations of objects, Y^1_k, and PHD $v^1_k(x)$. Then, from $V_{k-1}(x)$, we use dynamic state equation (7.4) and measurement equation (7.5) to predict $V^1_{k|k-1}(x)$ at camera 1 by Equation (3.17). Because measurement equation (7.5) is not linear, we have to use the unscented transform in the prediction step (more details are in [102]). Then, the set of 2D estimations of objects at camera 1, Y^1_k, is used to update $V_{k|k-1}(x)$ to $V^1_k(x)$ by the updating step in the GMPHD filter (Equation (3.19)). From assumptions on the GMPHD

filter, $V_{k-1}(x)$ is a Gaussian mixture, so $V_k^1(x)$ is also a Gaussian mixture.

- Step 2: Set $i = 2$

- Step 3: At camera i, set $V_{k|k-1}^i(x) = V_k^{i-1}(x)$. Assuming that we have the PHD of previous time step $k-1$ of single-view tracking stage at camera i, $v_{k-1}^i(x)$, the method described in Section 7.3 is performed to obtain the set of 2D estimations of objects at camera i, Y_k^i, and PHD $v_k^i(x)$. Because $V_{k|k-1}^i(x)$ is a Gaussian mixture, we can use the updating step of the GMPHD filter to update $V_{k|k-1}^i(x)$ with observations in Y_k^i. This means

$$V_k^i(x) = (1 - p_{D,k})V_k^{i-1}(x) + \sum_{y \in Y_k^i} V_{D,k}(x; y) \tag{7.6}$$

  Then, we can obtain the $V_k^i(x)$.

- Step 4: Set $i = i + 1$. If $i \leq n$ then we repeat step 3. Otherwise, we have $V_k^n(x)$. The PHD of the system is $V_k(x) = V_k^n(x)$. For estimating the 3D object locations, we investigate the PHD of the system $V_k(x)$ and choose Gaussian components whose weights are larger than a threshold (0.5) to obtain the 3D estimations of objects. (See Section 5.3 for more details of the asynchronous sensor updating in the GMPHD filter.)

We note that the GMPHD filter in [102] did not include the track labels of objects. For label tracking, our method is described as follows. Each Gaussian component is associated with a label. For birth Gaussian components, we assign

them a special label (for example -1). After the updating step in the first camera, Gaussian components with labels become the predicted Gaussian components for the second camera and then they are used to update the PHD in the second camera. At the last camera, for each label, we choose the Gaussian component that has the largest weight. The estimations of object locations are from the means of these largest Gaussian components. If a Gaussian component has a special label and its weight is large enough, we assign it a new label. This means a new person occurs. Hence, the identifications of people are defined in the tracking. This label tracking method is extended from the work in [22] from single sensor to multiple sensors and then applied in multiple-camera multiple-object tracking.

7.5 Experimental results

First, we test the performance of our method with data from the first and second cameras in scenarios seq24-2p-0111, seq35-2p-1111, and seq44-3p-1111 in the test database [58]. There are about 4500 time steps for each camera (9000 image frames for two cameras). We use the Wasserstein distance in [43], [61] to measure performance. The errors of 3D estimations are listed in Table 7.1.

Scenarios	Mean error (m)
seq24-2p-0111	0.06
seq35-2p-1111	0.05
seq44-3p-1111	0.07

Table 7.1: Error of 3D estimation

For visualization, we show the results from test case 'seq44-3p-1111' in Figures 7.4 and 7.5. In this scenario, there are three persons. They appear and disappear at different times. This scenario is challenging because occlusions occur between persons when they cross together. Moreover, in this scenario, the lighting of the room changes through the tracking, so it is difficult to apply segmentation methods. In addition, because the color models of heads are different between views, it is sometimes difficult to apply methods such as stereo matching to find the correspondences. Hence, the 3D reconstructions from correspondences are not reliable in this data. However, our method successfully tracks 3D object locations in this scenario.

At each camera, we use 400 samples to represent for the hypothesis intensity function at single-view tracking stage. The maximum of Gaussian mixture components are 30. We assume that persons enter the tracking areas from two entrances. Hence, the birth intensity is the mixture of Gaussian components whose means are the locations at these entrances. The clutter density in the multiple-view camera

fusion is an uniform distribution on the tracking area 3m $\times$ 2m $\times$ 2m which is the visible space in the 3D tracking and the clutter density in the single-view tracking stage is an uniform distribution of the size of image (it is also the projection from tracking area to cameras) and the range of radius H_x and H_y ([5,15]). The probability of survival is $p_S = 0.99$ and the probability of detection is $p_D = 0.98$. These parameters are set by experiments.

Figure 7.4 shows the performance of 3D people tracking. The dots are the ground-truth and the lines are the estimates from our method. The results indicate that tracks of people are maintained. The x and y components are reliable while the z component has some errors, for example at steps 600 to 700. This is because at steps 600 to 700, the color of the background near the person's location at the camera 2 is similar to the color of the templates. However, these errors are quite small. In this sequence, when a person moves out of the view and then moves back, we will assign it a new label, which is treated as correct detection.

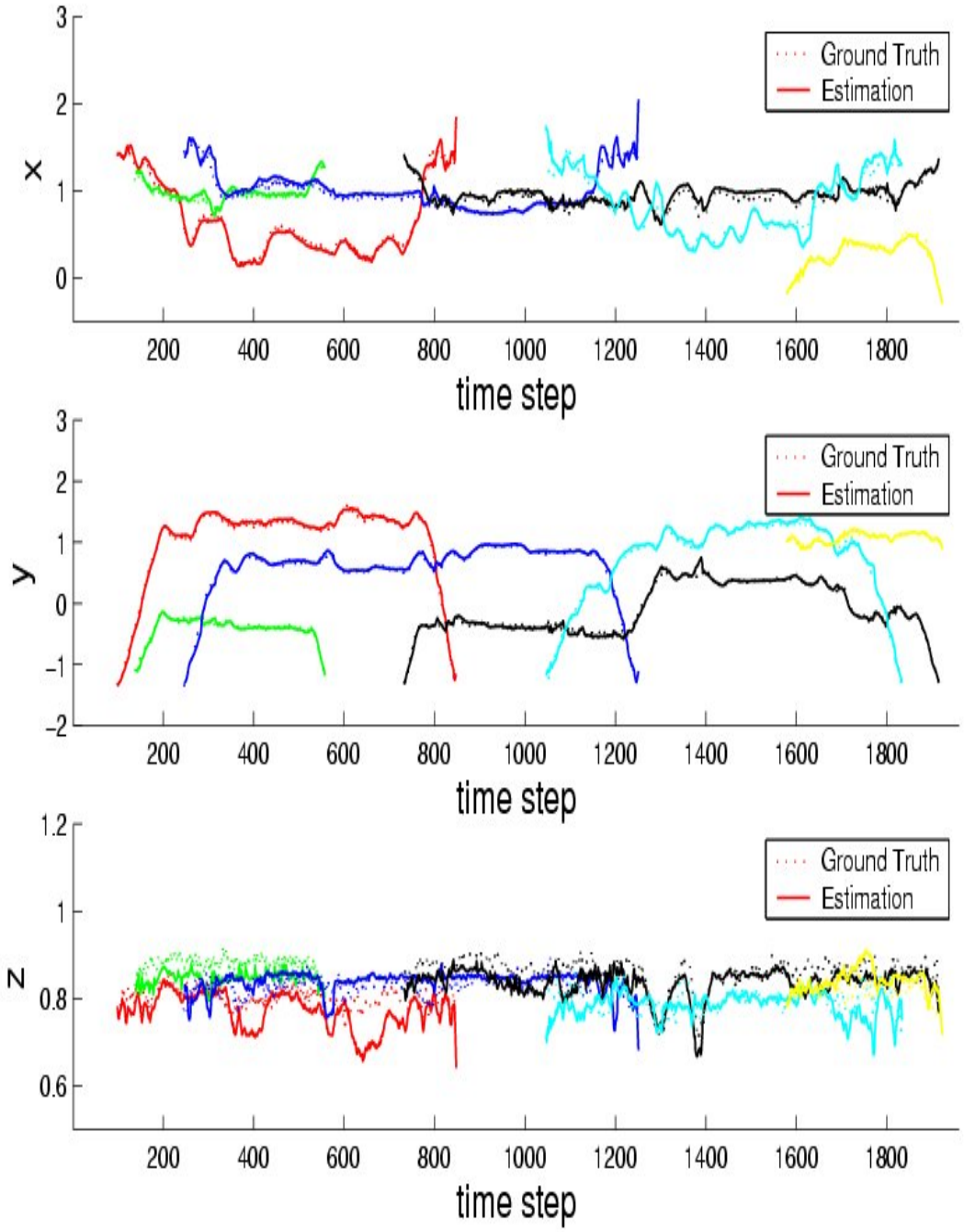

Figure 7.4: 3D results of tracking multiple people using the PHD filter

Figure 7.5 shows the results when we project 3D locations to the camera plane.

Each cell in the figure has two images. The left image is from camera 1 and the right image is from camera 2. In this figure, we can see that at time $k = 99, 144, 247$, the first, second, and third persons appear in the overlapped region sequentially. They are detected and tracked automatically. At time $k = 264, 295$, the occlusion between the second and third person occurs in camera 1 and 2. However, the tracks are maintained after the occlusion. At time $k = 809$, the occlusion between the first and third person occurs at camera 1 and the occlusion between the first and second person occurs at camera 2. We can see in the figure that our method can handle these cases. This is because the PHD from camera 1 is a good prediction for the PHD at the camera 2. Information from two cameras is fused to obtain the reliable 3D estimations without using data association methods.

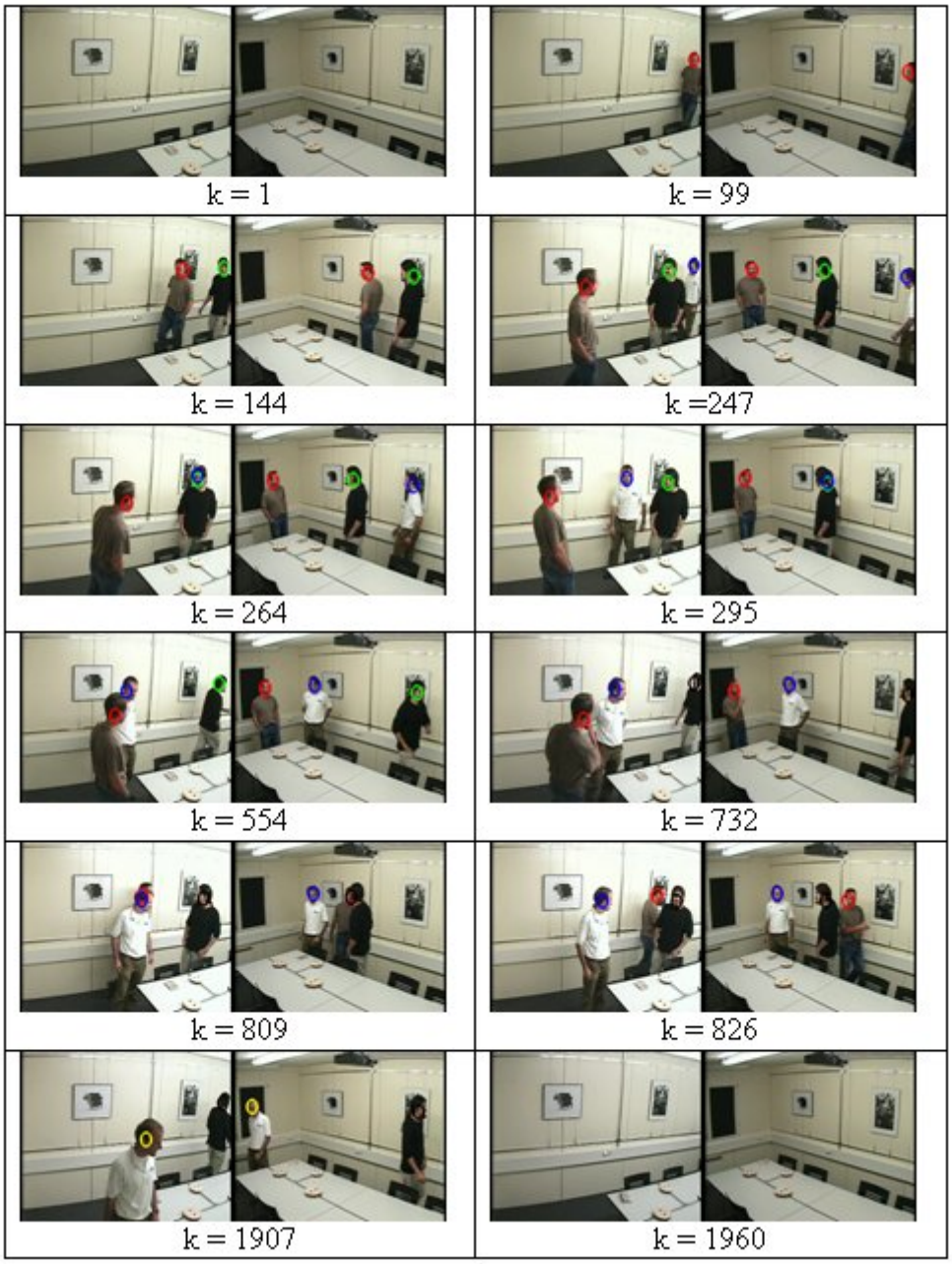

Figure 7.5: Projection 3D estimations to two camera planes

We also compare our method with the stereo matching method that is based on epipolar constraints [13], [68]. Figure 7.6 shows the performance of 3D people

tracking using the stereo matching. The results indicate that the performance of our method in Figure 7.4 is better than the stereo matching method in this data.

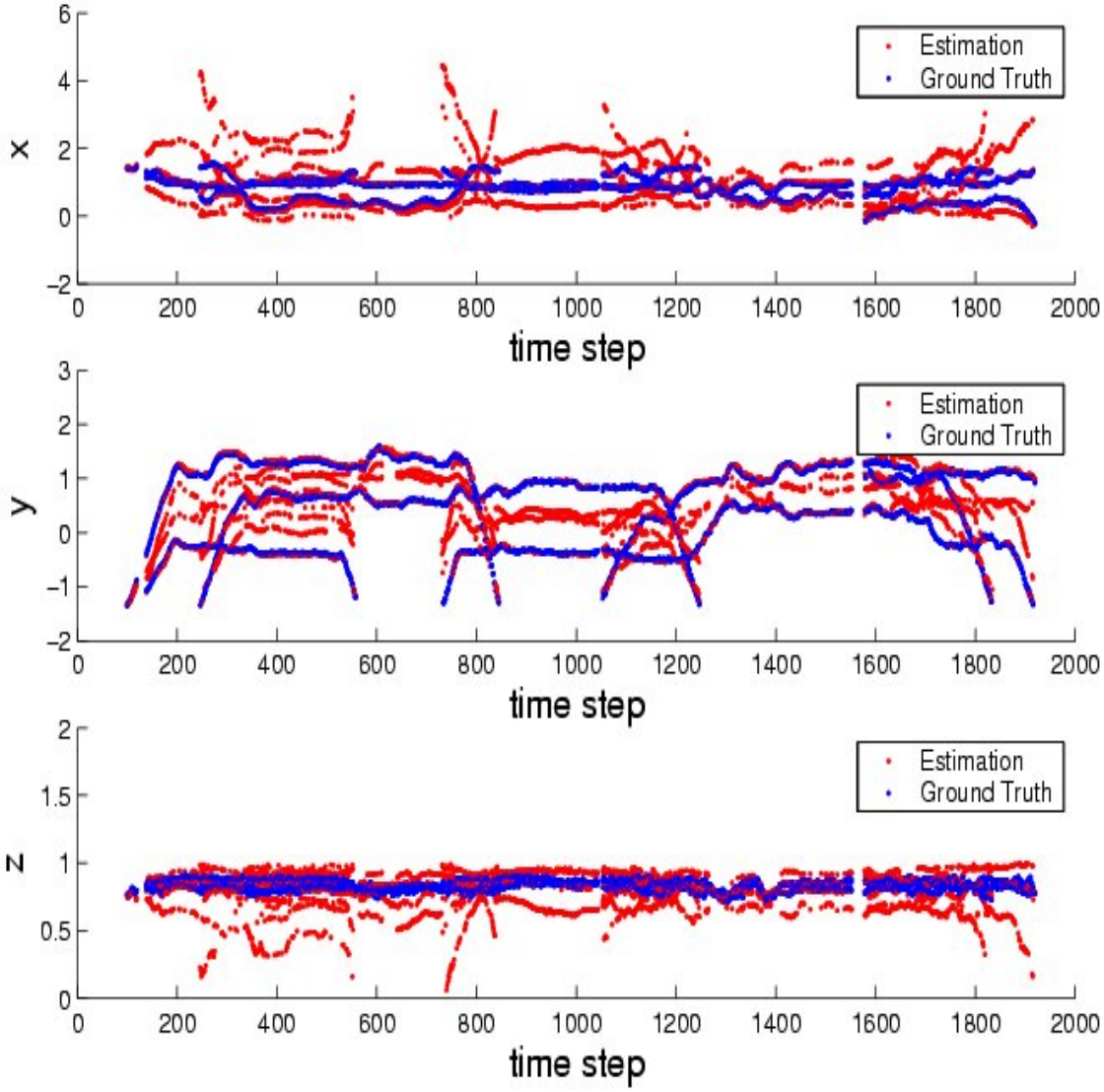

Figure 7.6: 3D results of tracking multiple people using Stereo Matching

Figure 7.7 and 7.8 show some comparison frame examples from stereo matching method and our method. In Figure 7.7, the stereo matching to reconstruct 3D

coordinates was not successful because the color models of the same object in two cameras are different from each other. Thus, matching cannot give the desired correspondences. The results in Figure 7.8 show that our method is successful in the 3D reconstruction. This is because we considered the temporal information of each object in two cameras. In our method, the current state estimates are predicted and updated from the previous state estimates and observations. This avoids sudden changes due to the errors of appearance matching.

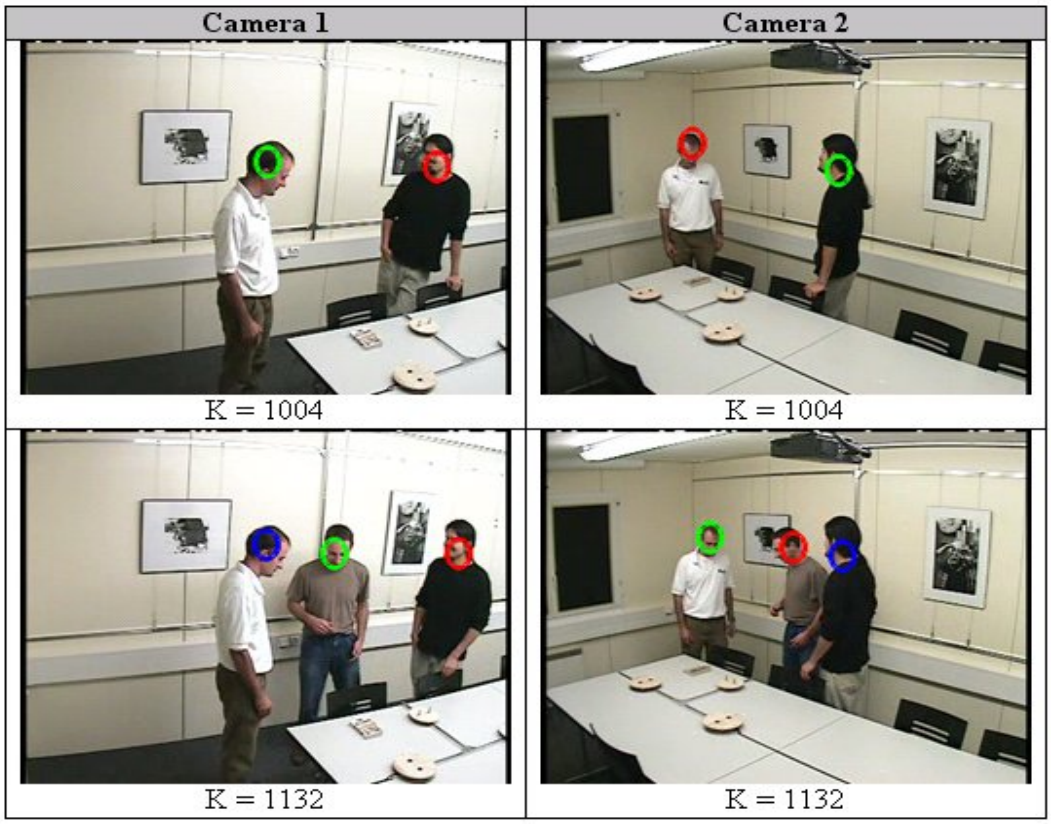

Figure 7.7: Some frame results from the Stereo Matching method

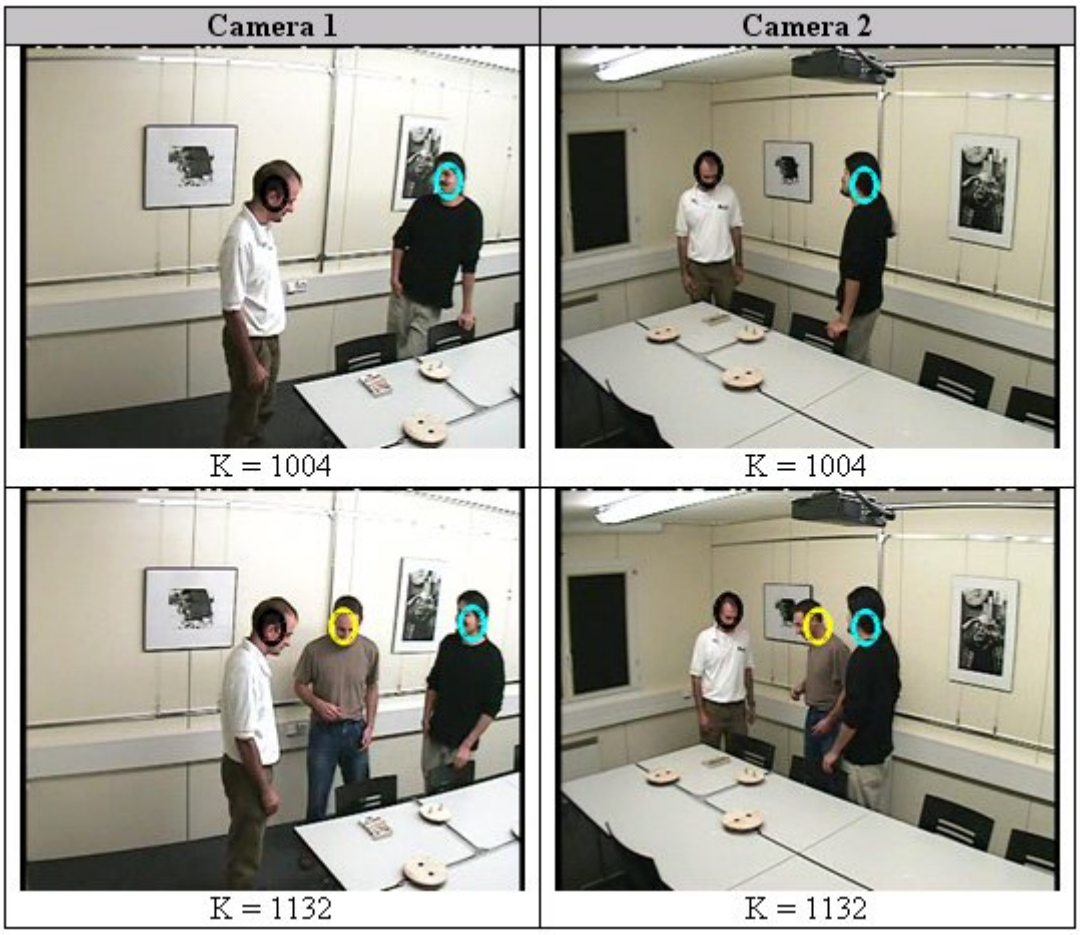

Figure 7.8: Some frame results from our method

To further test the performance, we set up a multiple-camera system in our lab 'StarHome' [1]. We collect some video sequences with 5000 image frames (2500 time steps). In this data, when persons move to the television, the tracking is lost because the color models of the screen and head are similar. This error is common for color object tracking. Hence, we use a simple background subtraction method to remove local background areas that have the similar color to the head. The

background is obtained before any object moves in the camera view. Of course, if the object is the same as the local background, it will not be able to be tracked correctly. In this case, we can use auxiliary information to maintain the tracking, or the system will recover the track when the object moves out of the region. Figure 7.9 shows results in the first sequence.

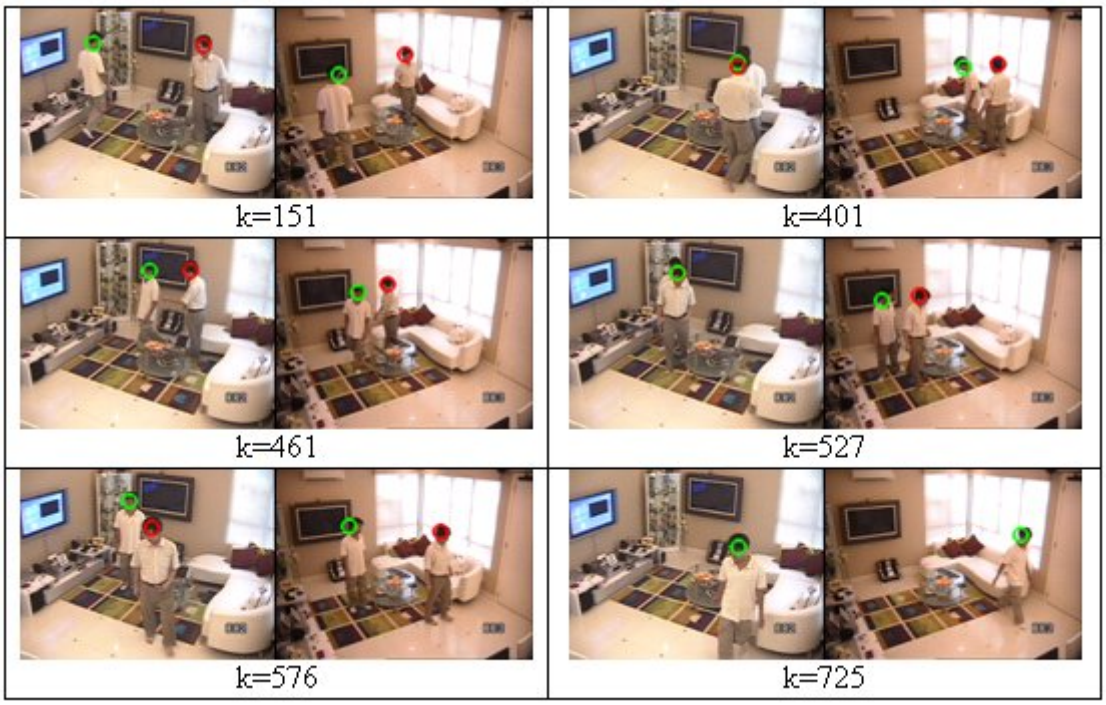

Figure 7.9: 3D results of tracking multiple people in sequence 1

The occlusions between two persons occur at time steps $k = 401, 527$ in camera 1. However, in camera 2, two persons are not occluded in these time steps. Thus, the method can track these persons correctly. In the second sequence, there are three persons. This sequence is challenging because three persons are occluded at

some time steps, for example time step $k = 625$. Figure 7.10 shows results in the second sequence.

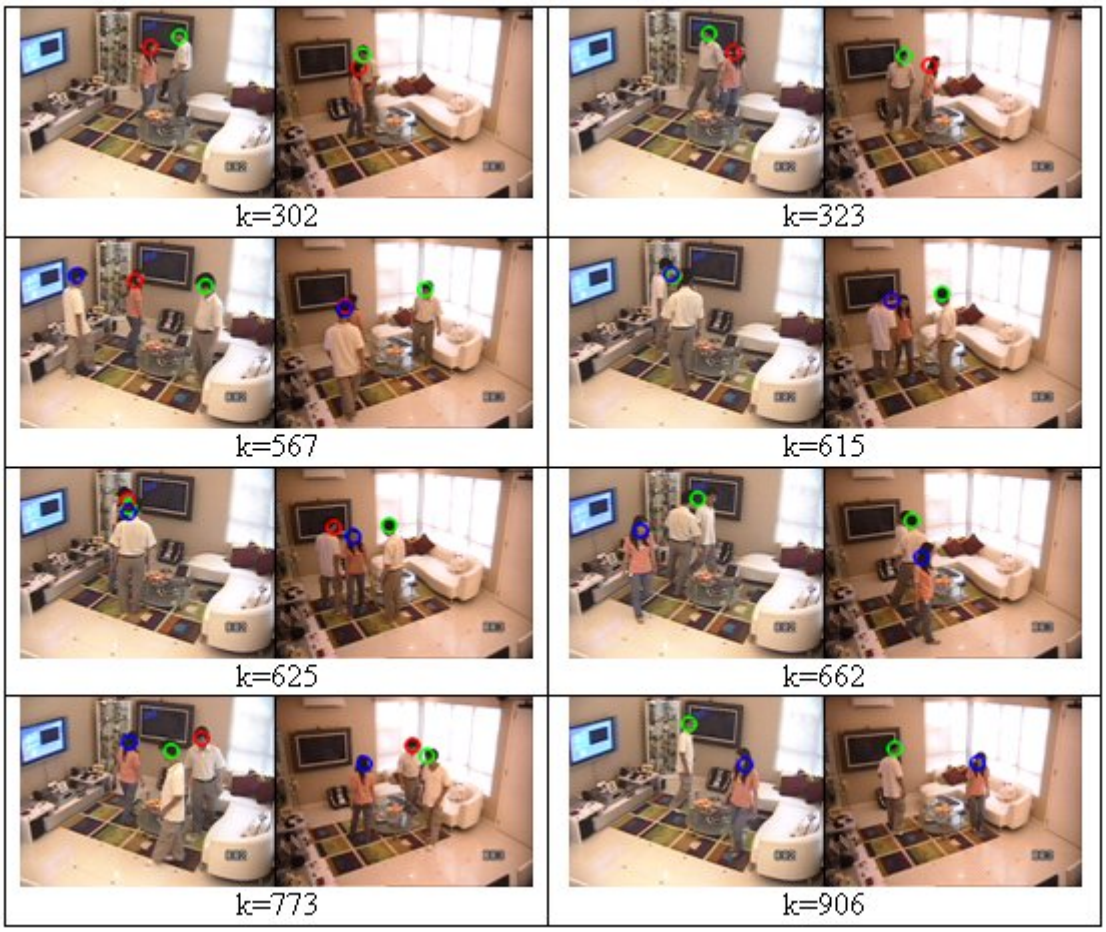

Figure 7.10: 3D results of tracking multiple people in sequence 2

Our method has good state estimates in most of the time in this sequence. However, at time step $k = 615$, because persons are near each other, the state estimates of the white person and orange person are the same at both cameras. This causes errors of identifications of persons. These errors can be recovered by

using auxiliary information such as cloth color [114].

7.6 Summary

The chapter described a method of using the GMPHD filter to track 3D locations of objects. The method can track a varying number of objects. Moreover, it can solve some occlusion problems for which single camera system has difficulty. The fusion stage using the GMPHD filter reduced a lot of computations compared with the methods that search whole state space or the particle filter method with multiple objects. Experimental results have shown that the proposed approach is promising.

Chapter 8

Conclusion and future work

In this study, we applied successfully the PHD filter in visual tracking and speaker tracking. The GMPHD filter is proved efficiently in multiple-sensor scenarios such as microphone pairs or multiple cameras. Moreover, a method for maintaining track continuity in methods using the PHD filter for multiple-object tracking was also proposed. These contributions and discussions are detailed as follows.

Firstly, an efficient method for maintaining the track continuity in the GM-PHD filter is proposed. Our experimental results show that the GMPHD filter could identify the trajectory of each object in multiple-object tracking with a high accuracy. In the results, even when the measurement detection is not very reliable, the labels of objects are kept correctly. The results also show that the performance of our method is better than existing methods. This is because our method propagates the identifications of objects with Gaussian components and uses them to

create a label association search tree. A lot of branches in this search tree are reduced. Then, an exact search method is employed rather than approximation methods. Therefore, our algorithm has a high accuracy and can deal with a large number of objects in multiple-object tracking. The results of this study demonstrate that our method can be used in applications requiring real time processing with high density clutter and variable number of objects that traditional methods such as JPDA or MHT find difficult to handle due to the computational complexity. One limitation is when two objects are close or occluded for a long time, the track label could be wrong. Therefore, it is necessary to develop a method to deal with these cases. A suggestion is that we can detect when the occlusions occur and then we can solve the occlusion problem based on data from previous times.

Secondly, an efficient method for multiple-speaker tracking using the random finite set frame work is also addressed. We use the GMPHD filter for multiple-speaker tracking in a reverberant environment. The results show that the positions of speakers have a small error during the tracking period. Moreover, the method successfully handles the varying of number of speakers and false alarms in a reverberant environment. The results also demonstrate that the GMPHD filter is much more efficient than some methods for multiple-speaker tracking in a reverberant acoustic environment. Furthermore, it is shown that our method works well in other applications, such as tracking with bearing and range measurements. By applying the approximation of multiple-sensor PHD update in the GMPHD filter,

which proved to be a closed-form of the PHD filter, the efficiency is maintained. Because the state estimates of objects are peaks in Gaussian components, they are more exact than those by using clustering techniques to extract state estimates from particles. However, in the speaker tracking application that we implemented in this thesis, the extracting measurement method is based on the GCC TDOA method. It is easy to have false detections when there are a large number of speakers. To deal with this problem, we need to investigate some other methods to obtain better measurements. Another issue also can be considered. It is to integrate the acoustic and visual data in multiple-speaker tracking. With the help from visual processing, the accuracy of estimations of state of speakers will be improved.

Thirdly, in this study, a method is developed for tracking multiple objects using the PHD filter and color measurements. The method operates in single-object state space. It requires less computation than methods using multiple-object state space and it provides an alternative way to obtain the visual measurement random set which is not straight-forward sometimes. For example, when the camera is moving, methods to define the visual measurement random set by using detection methods such as background subtraction are difficult to be performed. By proposing a hypothesis intensity from color likelihood, a color measurement random set is obtained. From this color measurement random set, we can formulate the video tracking in the random finite set framework. With our method, the experimental results demonstrate that state estimates of objects are more reliable during

the tracking period. Moreover, our method can detect the appearance and disappearance of objects without using other additional methods. In the experimental results, the objects interact each other many times, but they are tracked with high accuracy by using our method. This is because the random finite set approach can overcome the varying number of objects automatically. The method can be a general framework for other applications such as radar tracking where we do not have a method to define measurement random set with raw data. However, two issues need to be considered. The first issue is that the method requires an amount of computation when using particles to obtain the measurement set. Therefore, it is worth developing a closed-form to obtain the measurement set to improve the speed of the method. The second issue is that in the proof to obtain the hypothesis intensity function, we assume that the multiple-object likelihood is the products of each single-object likelihood. This assumption is from [28] and can be used in the video tracking. It is not always correct in other applications. Hence, proving the formula of the hypothesis intensity function without using this assumption need to be considered.

Lastly, a two-stage method for multiple-camera multiple-object tracking is proposed. The method divides into two stages. These two stages operate in a single-object state space by using the GMPHD filter. The results show that our method can track 3D locations of objects even when the occlusions occur in some camera views and can track varying number of objects. Moreover, as a comparison,

the results indicate that the performance of our method is better than the stereo matching method because of using temporal information to avoid sudden changes. One limitation is that when two or more objects are occluded at all camera views, the identifications of these objects may be not correct. This is because the information of these objects at all camera views are the same and the proposed method cannot differentiate them. A suggestion is that we can use auxiliary information such as cloth color to recover the identifications of these objects. However, if two objects have similar appearances, the differentiation between them is a challenging task. Another limitation is the changing of the object colors. This is because limitations of of using color in single object tracking of the first stage. Therefore, it is worth developing methods to adapt to color changes in objects.

Appendix: Publications

[1] N. T. Pham, W. H. Huang, and S. H. Ong. Tracking multiple objects using probability hypothesis density filter and color measurements. In *International Conference on Multimedia and Expo*, Beijing, China, 2007.

[2] N. T. Pham, W. H. Huang, and S. H. Ong. Multiple sensor multiple object tracking with GMPHD filter. In *International Conference on Information Fusion*, Quebec, Canada, 2007.

[3] N. T. Pham, W. H. Huang, and S. H. Ong. Probability hypothesis density approach for multi-camera multi-object tracking. In *Asian Conference on Computer Vision*, Japan, 2007.

[4] N. T. Pham, W. H. Huang, and S. H. Ong. Tracking multiple speakers using CPHD filter. In *ACM Multimedia*, Germany, 2007.

[5] N. T. Pham, W. H. Huang, and S. H. Ong. Maintaining track continuity in GMPHD filter. In *International Conference on Information, Communications and Signal Processing*, Singapore, 2007.

[6] N. T. Pham, J. K. Wu, and S. H. Ong. Fusing color and contour in visual tracking. In *IAPR Conference on Machine Vision Application*, Japan, 2005.

Bibliography

[1] http://starhome.i2r.a-star.edu.sg.

[2] J. Allen and D. Berkley. Image method for efficiently simulating small room acoustic. *Journal of the Acoustical Society of America*, 65:943–950, 1979.

[3] F. Amoozegar and M. Sudareshan. Target tracking by neural network maneuver modeling. In *Proceedings of IEEE World Congress of Computational Intelligence*, pages 3932–3937, 1994.

[4] B. Anderson and J. Moore. *Optimal Filtering.* Prentice Hall, New Jersey, 1979.

[5] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle

filters for on-line non-linear/non-Gaussian Bayesian tracking. *IEEE Transaction on Signal Processing, Special Issue on Monte Carlo Methods*, pages 174–188, 2002.

[6] Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association*. Academic Press, San Diego, 1988.

[7] Y. Bar-Shalom and X. Li. *Multitarget-Multisensor Tracking: Principles and Techniques*. YBS Publishing, 1995.

[8] J. Benesty. Adaptive eigenvalue decomposition algorithm for passive acoustic source localization. *Journal of the Acoustical Society of America*, 107(1):384–391, 2000.

[9] S. Blackman. *Multiple-Target Tracking with Radar Application*. Artech House, 1986.

[10] S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking System*. Artech House, 1999.

[11] A. Blake and M. Isard. *Active Contours*. Springer-Verlag, 1998.

[12] L. Bretzner and T. Lindeberg. Use your hand as a 3-D mouse or relative orientation from extended sequences of sparse point and line correspondances using the affine trifocal tensor. In *Proceedings of European Conference on Computer Vision*, pages 141–157, 1998.

[13] Q. Cai and J. K. Aggarwal. Automatic tracking of human motion in indoor scenes across multiple synchronized video streams. In *International Conference on Computer Vision*, pages 356 – 362, 1998.

[14] T. J. Cham and J. M. Rehg. A multiple hypothesis approach to figure tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 239–244, 1998.

[15] T. Chang and S. Gong. Tracking multiple people with a multi-camera system. In *IEEE Workshop on Multi-Object Tracking*, pages 19–26, 2001.

[16] Y. Changjiang, R. Duraiswami, and L. Davis. Fast multiple object tracking via a hierarchical particle filter. In *IEEE Conference on Computer Vision*, 2005.

[17] J. C. Chen, K. Yao, and R. E. Hudson. Source localization and beamforming. *IEEE Signal Processing Magazine*, 19:30–39, 2002.

[18] Y. Chen and Y. Rui. Real-time speaker tracking using particle filter sensor fusion. *Proceedings of the IEEE*, 92(3):485–494, 2004.

[19] D. E. Clark. *Multiple Target Tracking with The Probability Hypothesis Density Filter*. PhD thesis, Department of Electrical Engineering, Heriot-Watt University, 2006.

[20] D. E. Clark and J. Bell. Data association for the PHD filter. In *Second International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, Melbourne, December 2005.

[21] D. E. Clark, K. Panta, and B. N. Vo. The GM-PHD filter multiple target tracker. In *International Conference on Information Fusion*, pages 1–8, 2006.

[22] D. E. Clark, B. N. Vo, and J. Bell. The GM-PHD filter multitarget tracking in sonar images. In *Proceedings of SPIE Signal Processing, Sensor Fusion and Target Recognition XV*, pages 62350R.1–62350R.8, 2006.

[23] D. Comaniciu and P. Meer. Mean shift analysis and applications. In *International Conference on Computer Vision*, pages 1197–1203, 1999.

[24] I. Cox and S. Hingorani. An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 18(2):138–150, 1996.

[25] D. Crisan and A. Doucet. Convergence of sequential Monte Carlo methods. In *CUED/F-INFENG/TR381*, 2000.

[26] D. Crisan and A. Doucet. A survey of convergence results on particle filtering methods for practitioners. *IEEE Transaction on Signal Processing*, 50(3):736–746, 2002.

[27] R. Cutler, Y. Rui, A. Gupta, J. J. Cadiz, I. Tashev, L.-W. He, A. Colburn, Z. Zhang, Z. Liu, and S. Silverberg. Distributed meetings: a meeting capture and broadcasting system. In *Proceedings of ACM Multimedia*, pages 503 – 512, 2002.

[28] J. Czyz, B. Ristic, and B. Macq. A color-based particle filter for joint detection and tracking of multiple objects. In *International Conference on Acoustic, Speech, and Signal Processing*, pages 217–220, 2005.

[29] D. J. Daley and D. Vere-Jones. *An Introduction to the Theory of Point Processes*. Springer, 1988.

[30] J. H. Dibiase, H. F. Silverman, and M. S. Brandstein. Robust localization in reverberant rooms. In M. S. Brandstein and D. B. Ward, editors, *Microphone Arrays: Signal Processing Techniques and Applications*, chapter 8, pages 157–180. Springer-Verlag, Berlin, Germany, 2001.

[31] S. Dockstader and A. M. Tekalp. Multiple camera tracking of interacting and occluded human motion. *Proceedings of the IEEE*, 89(10), 2001.

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

[33] A. Doucet, B. N. Vo, C. Andrieu, and M. Davy. Particle filtering for multi-target tracking and sensor management. In *International Conference on Information Fusion*, pages 474–481, 2002.

[34] T. Dvorkind. Speaker localization in a reverberant and noisy environment. Master's thesis, Israel Institue of Technology, Israel, 2003.

[35] T. Gehrig, U. Klee, J. McDonough, S. Ikbal, M. Wolfel, and C. Fugen. Tracking and beamforming for multiple simultaneous speakers with probabilistic data association filters. In *Proceedings of Interspeech*, 2006.

[36] W. R. Gilks, S. Richardson, and D. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman and Hall CRC, 1995.

[37] I. R. Goodman, R. Mahler, and H. T. Nguyen. *Mathematics of Data Fusion*. Kluwer Academic Publishers, 1997.

[38] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/nonGaussian Bayesian state estimation. In *IEE Procedings F*, pages 107–113, 1993.

[39] M. Han, W. Xu, H. Tao, and Y.H. Gong. An algorithm for multiple object trajectory tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[40] B. Hansson, C. Forster, and E. Torebjork. Matched filtering and multiple hypothesis tracking applied to C-fiber action potentials recorded in human nerves. In *Proceedings of Signal and Data Processing of Small Targets*, pages 582–593, 1998.

[41] I. Haritaoglu, D. Harwood, and L. Davis. Who, when, where, what: a real time system for detecting and tracking people. In *Proceedings of the Third Face and Gesture Recognition Conference*, pages 222–227, 1998.

[42] S. Herman and P. Moulin. A particle filtering approach to FM-band passive radar tracking and automatic target recognition. In *Proceedings of the IEEE Aerospace Conference*, 2002.

[43] J. R. Hoffman and R. Mahler. Multitarget miss distance via optimal assignment. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 34(3):327–336, 2004.

[44] C. Hue, J. Cadre, and P. Perez. Sequential Monte Carlo methods for multiple target tracking and data fusion. *IEEE Transactions on Signal Processing*, 50(2):309–325, 2002.

[45] N. Ikoma, T. Uchino, and H. Maeda. Tracking of feature points in image sequence by SMC implementation of PHD filter. In *SICE Annual Conference in Sapporo*, pages 1696– 1701, 2004.

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

[47] M. Isard and J. MacCormick. BraMBLe: a Bayesian multiple-blob tracker. In *International Conference on Computer Vision*, pages 34–41, 2001.

[48] M. Jaward, L. Mihaylova, N. Canagarajah, and D. Bull. Multiple objects tracking using particle filters in video sequences. In *Proceedings of the IEEE Aerospace Conference*, 2005.

[49] A. Jazwinski. *Stochastic Processes and Filtering Theory.* Academic Press, 1970.

[50] D. S. Johnson and C. McGeoch. Network flows and matching: First DIMACS implementation challenge. *American Mathematical Society*, 1993.

[51] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transaction of the ASME-Journal of Basic Engineering*, 82D(1):34–45, 1960.

[52] S. M. Khan and M. Shah. A multiview approach to tracking people in crowded scenes using a planar homography constraint. In *European Conference on Computer Vision*, pages 133–146, 2006.

[53] K. Kim and L. S. Davis. Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filtering. In *European Conference on Computer Vision*, pages 98–109, 2006.

[54] C. Knapp and G. Carter. The generalized correlation method of estimation of time delay. *IEEE Transaction on Acoustic, Speech, Signal Processing*, ASSP-24(4):320–327, 1976.

[55] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer. Multi-camera multi-person tracking for easyliving. In *IEEE International Workshop on Visual Surveillance*, pages 3–10, 2000.

[56] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, pages 83–97, 1955.

[57] G. Lathoud and M. Magimai. A sector based frequency domain approach to detection and localization of multiple speakers. In *International Conference on Acoustic, Speech, Signal Processing*, pages 265–268, 2005.

[58] G. Lathoud, J. Odobez, and D. Perez. AV16.3: an audio-visual corpus for speaker localization and tracking. In *Proceedings of the MLMI Workshop*, pages 182–195, 2004.

[59] H. Leung and M. Blanchette. Data association for multiple target tracking using Hopfield neural network. In *Proceedings of International Symposium on Speech, Image Processing and Neural Networks*, pages 280–283, 1994.

[60] L. Lin, Y. Bar-Shalom, and T. Kirubarajan. Data association comnined with the probability hypothesis density filter for multitarget tracking. In *Proceedings of SPIE*, pages 464–475, 2004.

[61] W. K. Ma, B. N. Vo, S. Singh, and A. Baddeley. Tracking an unknown time-varying number of speakers using TDOA measurements: a random finite set approach. *IEEE Transaction on Signal Processing*, 54(9), 2006.

[62] E. Maggio, E. Piccardo, C. Regazzoni, and A. Cavallaro. Particle PHD filtering for multi-target visual tracking. In *International Conference on Acoustic, Speech, Signal Processing*, pages 1101–1104, 2007.

[63] R. Mahler. Multi-target Bayes filtering via first-order multi-target moments. *IEEE Transaction on Aerospace and Electronic Systems*, 39(4):1152–1178, 2003.

[64] R. Mahler. Statistic 101 for multitarget, multisensor data fusion. *IEEE Magazine on Aerospace and Electronic Systems*, 19(1):53–64, 2004.

[65] R. Mahler. Theory of PHD filters for higher order in target number. In *Proceedings of SPIE Signal Processing, Sensor Fusion and Target Recognition XV*, 2006.

[66] R. Mahler. *Statistical Multisource-Multitarget Information Fusion*. Artech House Publishers, 2007.

[67] R. V. D. Merwe, N. Freitas, A. Doucet, and E. Wan. The unscented particle filter. In *Advances in Neural Information Processing Systems 13*, 2001.

[68] A. Mittal and L. S. Davis. M2Tracker: a multi-view approach to segmenting and tracking people in a cluttered scene. *International Journal of Computer Vision*, 51(3):189–203, 2003.

[69] K. Nakadai, K. Hidai, H. Mizoguchi, H. G. Okuno, and H. Kitano. Real-time auditory and visual multiple-object tracking for robots. In *International Joint Conference on Artificial Intelligence*, pages 1425–1436, 2001.

[70] K. Nummiaro, E. Koller-Meier, T. Svoboda, D. Roth, and L. V. Gool. Color-based object tracking in multi-camera environments. In *Symposium for Pattern Recognition of the DAGM*, pages 591–599, 2003.

[71] S. Oh and S. Sastry. A polynomial-time approximation algorithm for joint probabilistic data association. In *Proceedings of the American Control Conference*, pages 1283–1288, 2005.

[72] K. Okuma, A. Taleghani, N. Freitas, J. J. Little, and D. G. Lowe. A boosted particle filter: multitarget detection and tracking. In *European Conference on Computer Vision*, pages 28–39, 2004.

[73] K. Panta and B. N. Vo. An efficient track management scheme for the gaussian mixture probability hypothesis density tracker. In *International Conference on Intelligent Sensing and Information Processing*, 2006.

[74] K. Panta, B. N. Vo, and S. Singh. Improved probability hypothesis density filter (PHD) for multitarget tracking. In *Proceedings of Intelligent Sensing and Information Processing*, pages 213–218, 2005.

[75] K. Panta, B. N. Vo, S. Singh, and A. Doucet. Probability hypothesis density filter versus multiple hypothesis tracking. In *Proceedings of SPIE*, pages 284–295, 2004.

[76] N. T. Pham, J. K. Wu, and S. H. Ong. Fusing color and contour in visual tracking. In *IAPR Conference on Machine Vision Applications*, 2005.

[77] A. B. Poore, A. J. Robertson, and P. J. Shea. A new class of Lagrangian relaxation based algorithms for fast data association in multiple hypothesis tracking applications. In *Proceedings of SPIE*, pages 184–194, 1995.

[78] R. Popoli, S. Blackman, and M. Busch. Application of multiple hypothesis tracking to agile beam radar tracking. In *Proceedings of SPIE*, pages 418–428, 1996.

[79] I. Potamitis, H. Chen, and G. Tremoulis. Tracking of multiple moving speakers with multiple microphone arrays. *IEEE Transaction on Speech Audio Processing*, 12(5):309–325, 2002.

[80] D. Reid. An algorithm for tracking multiple targets. *IEEE Transaction on Automatic Control*, 24(6):84–90, 1979.

[81] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2004.

[82] J. A. Roecker. Approximate joint probabilistic data association algorithms. In *Proceedings of SPIE*, pages 331–340, 1993.

[83] Y. Rui and Y. Chen. Better proposal distributions: object tracking using unscented particle filter. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 86–793, 2001.

[84] K. Ruohonen. Graph theory. www.math.tut.fi/ruohonen/GT_English.pdf.

[85] D. Schulz, W. Burgard, D. Fox, and A.Cremers. Tracking multiple moving targets with a mobile robot using particle filters and statistical data association. In *International Conference on Robotics, Automation*, pages 1665–1670, 2001.

[86] D. Schulz, W. Burgard, D. Fox, and A.B. Cremers. People tracking with a mobile robot using sample-based joint probabilistic data association filters. *International Journal of Robotics Research*, 22(2):99–116, 2003.

[87] I. O. Sebe, S. You, and U. Neumann. Globally optimum multiple object tracking. In *Proceedings of SPIE*, 2005.

[88] H. Sidenbladh. Multi-target particle filtering for the probability hypothesis density. In *International Conference on Information Fusion*, pages 800–806, 2003.

[89] H. Sidenbladh and S. Wirkander. Tracking random sets of vehicles in terrain. In *Proceedings of IEEE Workshop on Multi-Object Tracking*, 2003.

[90] N. T. Siebel and S. J. Maybank. The ADVISOR visual surveillance system. In *Proceedings of the ECCV Workshop on Applications of Computer Vision*, 2004.

[91] K. Smith, D. Perez, and J. M. Odobez. Using particles to track varying numbers of interacting people. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 962 – 969, 2005.

[92] R. L. Streit and T. E. Luginbuhl. A probabilistic multi-hypothesis tracking algorithm without enumeration and pruning. In *Proceedings of the Sixth Joint Service Data Fusion Symposium Laurel*, 1993.

[93] R. L. Streit and T. E. Luginbuhl. Maximum likelihood method for probabilistic multi-hypothesis tracking. In *Proceedings of SPIE International Symposium, Signal and Data Processing of Small Targets*, pages 394–405, 1994.

[94] C. Tao, W. Thompson, and J. Taur. A fuzzy logic approach to multidimensional target tracking. In *Proceedings of IEEE International Conference on Fuzzy Systems*, pages 1350–1355, 1993.

[95] M. Tobias. *Probability Hypothesis Densities for Multitarget, Multisensor Tracking with Application to Passive Radar*. PhD thesis, School of Electrical and Computer Engineering, Georgia Institue of Technology, 2006.

[96] M. Tobias and A. D. Lanterman. Multitarget tracking using multiple bistatic range measurements with probability hypothesis densities. In *Proceedings of SPIE Signal Processing, Sensor Fusion and Target Recognition XIII*, pages 296–305, 2004.

[97] M. Tobias and A. D. Lanterman. Probability hypothesis density based multitarget tracking with bistatic range and Doppler observations. In *IEE Proceedings on Radar, Sonar and Navigation*, pages 195–205, 2005.

[98] J. M. Valin, F. Michaud, and J. Rouat. Robust 3D localization and tracking of sound sources using beamforming and particle filtering. In *International*

Conference on Acoustic, Speech, Signal Processing, pages 841–844, 2006.

[99] J. Vermaak and A. Blake. Nonlinear filtering for speaker tracking in noisy and reverberant environments. In *International Conference on Acoustic, Speech, Signal Processing*, pages 3021–3024, 2001.

[100] J. Vermaak, A. Doucet, and P. Perez. Maintaining multimodality through mixture tracking. In *International Conference on Computer Vision*, pages 1110–1116, 2003.

[101] M. Vihola. Random set particle filter for bearings-only multitarget tracking. In *Proceeding SPIE Signal Processing, Sensor Fusion and Target Recognition XIV*, pages 301–312, 2005.

[102] B. N. Vo and W. K. Ma. The Gaussian mixture probability hypothesis density filter. *IEEE Transaction on Signal Processing*, 54(11):4091–4104, 2006.

[103] B. N. Vo, S. Singh, and A. Doucet. Sequential Monte Carlo methods for Bayesian multi-target filtering with random finite sets. *IEEE Transaction on Aerospace and Electronic Systems*, 41(4):1224–1245, 2005.

[104] B. N. Vo, S. Singh, and W. K. Ma. Tracking multiple speakers using random sets. In *International Conference on Acoustic, Speech, Signal Processing*, pages 357–360, 2004.

[105] B. T. Vo. *Random finite sets in multi-object filtering.* PhD thesis, School of Electrical, Electronic and Computer Engineering, University of Western Australia, 2008.

[106] B. T. Vo, B. N. Vo, and A. Cantoni. The cardinalized probability hypothesis density filter for linear Gaussian multi-target models. In *Proceedings of 40th Conference on Information Sciences and Systems*, pages 681–686, 2006.

[107] B. T. Vo, B. N. Vo, and A. Cantoni. Analytic implementations of the cardinalized probability hypothesis density filter. *IEEE Transaction on Signal Processing*, 55(7):3553–3567, 2007.

[108] H. W. Waard. An improved clustering concept for MHT applications. In *IEE Target Tracking: Algorithms and Applications*, pages 9/1–9/8, 2002.

[109] E. Wan and R. Merwe. The unscented Kalman filter for nonlinear estimation. In *Proceedings of Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pages 153–158, 2000.

[110] G. Wang, R. Rabensenstein, N. Strobel, and S. Spors. Object localization by joint audio video signal processing. In *Vision Modelling and Visualization*, pages 97–104, 2000.

[111] Y. D. Wang, J. K. Wu, and W. M. Huang. Tracking a variable number of human groups in video using probability hypothesis density. In *International Conference on Pattern Recognition*, pages 1127 – 1130, 2006.

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

[113] Y. Wu, T. Yu, and G. Hua. Tracking appearances with occlusions. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[114] M. Yang, Y. Wu, and S. Lao. Intelligent collaborative tracking by mining auxiliary objects. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 697–704, 2006.

[115] T. Yu and Y. Wu. Collaborative tracking of multiple targets. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[116] T. Zajic and R. Mahler. A particle system implement of the PHD multi-target tracking filter. In *Signal Processing, Sensor Fusion and Target Recognition XII, SPIE Proceedings*, pages 291–299, 2003.

[117] T. Zhao and R. Nevatia. Tracking multiple humans in crowded environment. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2004.