STOCHASTIC METHODS FOR BAYESIAN FILTERING AND THEIR APPLICATIONS TO MULTICAMERA MULTITARGET TRACKING

WANG YADONG

(B.ENG., M.ENG., NORTHWESTERN POLYTECHNICAL UNIVERSITY, CHINA)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2007

To Zhang Tianxia

Acknowledgements

First of all, I would like to thank my supervisors, Dr. Wu Jiankang and Professor Ashraf A. Kassim, for invaluable guidance, inspiring discussions and support during my Ph.D. study.

I gratefully acknowledge the financial support for my Ph.D. study given by both National University of Singapore and Institute for Infocomm Research, Singapore.

I also thank Dr. Huang Weimin and Mr. Pham Nam Trung for their helps and discussions. I wish to thank Dr. N. de Freitas for providing the programs of particle filters on his webpage, Dr. Ronald P.S. Mahler for his literature and Dr. Ba-Ngu Vo for the programs of the probability hypothesis density filter.

Finally, I would like to thank my parents and brother for support during these years and my wife Tianxia for love and encouragement.

Wang Yadong

July 2007

Contents

Summary

Target tracking is an important key technology for many military and commercial applications. The tracking problems are usually formulated by using the *state space approach* for discrete-time dynamic systems. Under this framework, the tracking is to estimate the *state* x_t of target at time t, given the *measurement* sequence $y_{1:t}$ of sensor from time 1 to t, or equivalently to construct the conditional probability density function $p(x_t|y_{1:t})$. The theoretical optimal solution is provided by the *recursive Bayesian filter*. However, for multi-sensor multi-target tracking, there are many challenges to extend the single-sensor single-target Bayesian filter. In this thesis, the focus is on extending the Bayesian filter to multi-camera or multi-target visual tracking.

First, a *spatio-temporal recursive Bayesian filter* is formulated for tracking a target using multiple cameras. We propose an *adaptive mixed particle filter* for the implementation of the spatio-temporal recursive Bayesian filter for the dynamic system.

In particular, the *mixed importance sampling strategy* is used to fuse temporal information of dynamic systems and spatial information from multiple cameras. It is *adaptive* in sense that it automatically ranks data from multiple cameras and assigns weights according to data's quality in the fusion process. The results show that this method is able to recover a target's position even when it is completely occluded in a particular camera for some time.

Second, a multi-target Bayesian filter, the *probability hypothesis density* (PHD) filter, is designed to track unknown and variable number of targets in image sequences. Because the dimensions of state and observation are time-varying during the tracking process, the PHD filter employs the *random finite set* representation of multiple states and multiple measurements and the PHD is the 1st order moment of random finite set. The PHD filter is implemented using two methods: both *particle filter* and *Gaussian mixture*. For the particle PHD filter, two *importance functions* and correspondent weight functions are proposed for survival targets and new-birth targets, respectively. It is shown in the thesis that the importance function for survival targets theoretically extends the optimal importance function of the linear Gaussian model from single-measurement case to measurement-set (multi-measurement) case. Whereas the importance function for new-birth targets is a *data-driven* method which uses the current measurements in the sampling process of the particle PHD filter. For the Gaussian mixture PHD filter, a *scene-driven* method which incorporates the prior knowledge of scene into the PHD filter

is presented. The results show that these PHD filters are able to track a variable number of targets and derive their positions in image sequences.

This work suggests that stochastic methods for Bayesian filtering are powerful means for multi-sensor multi-target tracking.

List of Tables

List of Figures

Chapter 1

Introduction

Target tracking is a fundamental problem for many military and commercial applications such as battlefield monitoring, video surveillance, human motion analysis, and human-computer interface. Different applications have different scenarios and motivations. For example, in radar tracking for battlefield monitoring, the target (e.g., airplane, missile, or ship) usually appears as a spot on the radar screen with complex maneuvers such as acceleration, turns, or stops. Whereas in visual tracking for video surveillance, the target (e.g., person or vehicle) is usually captured in form of image sequences. Rich information such as intensity, color, or contour contained in target pictures can be used for distinguishing, tracking and other form of analysis.

The tracking problems are usually formulated by using the *state space approach* for

discrete-time dynamic systems. Under this framework, the tracking is to estimate the *state* of target x_t (e.g., position, velocity, and identification) at time t given the *measurement* sequence of sensor $y_{1:t}$ (e.g., image sequences captured by a camera) from time 1 to t, or equivalently to construct the conditional probability density function $p(x_t|y_{1:t})$. Successive estimates provide the track which describes the trajectory of a target.

A simple form of tracking is tracking a single target. There are two main groups of methods for tracking a single target: *filtering methods* and *likelihood functions*. Filtering methods are mostly used in radar tracking and generally used to capture the dynamics of targets. The commonly used methods include: i) *Kalman filter* for linear system and Gaussian noise [68] and its extensions such as the extended Kalman filter (EKF) [45, 5] and the unscented Kalman filter (UKF) [67]; ii) *interacting multiple models* (IMM) for multiple motion models [20]; and iii) *particle filters* for nonlinear and non-Gaussian problems [51, 39]. On another hand, likelihood functions are mostly used in visual tracking tasks and concentrate on how to differentiate the target from the background. The typical likelihood functions include intensity-based method [81], contour-based method [62], and color-based method [32].

As tracking a single target using one sensor has many limitations, there is a recent trends towards multi-sensor or multi-target tracking. There has been some research done on tracking using multiple cameras [21, 78, 90, 97] and on tracking multiple

targets [43, 107, 101, 36].

When tracking multiple targets, data association methods are generally used to associate observations of sensors with targets. For example, if there are two targets, a person and a car, and the camera detects three foreground blobs, data association must determine which blob belongs to the person, the car, or the clutter environment, i.e., there are multiple choices for association. The aim of data association is to find the best association scheme. There have been a few categories of data association methods: i) *joint probabilistic data association* (JPDA) [43] which uses the weighted average of functions of multiple observations to update the state of a target, ii) *multiple hypotheses tracking* (MHT) [107] which enumerates multiple possible association hypotheses during a period till one hypothesis can be verified, and iii) assignment algorithms [101, 36] which essentially perform constrained optimization problems to find an optimal association solution.

Another trend for tracking multiple targets is tracking a variable number of targets. When the target number is unknown and variable, data association must deal with the variable dimension of state or observation. Some methods have been proposed to overcome this difficulty: *jump-diffusion process* [89], *reversible jump Markov chain Monte Carlo method* (RJMCMC) [72], and *finite set statistics* (FISST) and *probability hypothesis density* (PHD) [49, 85].

1.1 Motivations

Two stochastic methods for Bayesian filtering are closely related to this thesis, *particle filter* and *probability hypothesis density*.

The particle filter, also called *sequential Monte Carlo method*, is a Monte Carlo simulation based method and can be applied to nonlinear or non-Gaussian problems. The particle filter consists of 2 basic parts: *importance sampling* and *resamping*. Gordon *et al.* proposed the first particle filter, the *bootstrap* algorithm [51]. Liu and Chen presented a general framework for applying Monte Carlo methods to dynamic systems [80]. Their framework includes importance sampling, resampling, rejection sampling, and Markov chain iterations. Doucet *et al.* provided a Bayesian filtering framework of sequential simulation based methods for nonlinear and non-Gaussian dynamic models [41]. Their other major contributions are summarizing the methods for selecting importance sampling functions.

Much work has been done on tracking a visual target using particle filters. Isard and Blake proposed the first particle filter based visual tracking algorithm, the *condensation* (CONditional DENSity propogATION) algorithm [62], and later combined it with the statistical technique of importance sampling [63]. They demonstrated their method using a hand tracker which combines color blob-tracking with a contour model.

There has some research on tracking multiple targets using particle filters. Isard

and MacCormick presented a Bayesian multiple-blob tracker, *BraMBle* [64], to track multiple persons using a particle filter. Vermaak *et al.* [121] introduced a mixed particle filter to model each component (mode or target) with an individual particle filter and form part of the mixture. Okuma *et al.* [98] combined Vermaak's method with the *Adaboost* algorithm [123] to track multiple hockey players.

While considerable work involving the particle filter has been done on tracking, there has not been much work on multicamera tracking using particle filters. Occlusion, especially long-time complete occlusion, is a serious problem for tracking using a single camera. Multiple cameras provide information of a moving target from multiple views. As such, occlusions do not occur in all cameras and fusion of data from multiple cameras enables tracking of a moving target with desirable performance. Both importance sampling and resampling strategies in particle filters provide a theoretical framework for information fusion of multiple cameras. Therefore, how to design adaptive particle filter to fuse information of multiple cameras remains a challenge.

Tracking becomes challenging when the number of targets is unknown and variable because the state and observation dimensions are time-varying under this situation. There has been some recent work that attempt to meet this challenge. Reid proposed *multiple hypothesis tracking* (MHT) algorithm which enumerates multiple track-to-measurement association hypotheses during a period till one hypothesis can be verified [107]. The problem of MHT is the potential combinatorial explosion

in the number of hypotheses. Miller *et al.* generated the conditional mean estimates of an unknown number of targets and target types via *jump-diffusion* process [89]. Musicki *et al.* proposed *integrated probabilistic data association* (IPDA) [95] as a recursive formula for both data association and probability of target existence. Vermaak et al. presented the *existence joint probabilistic data association filter* (E-JUDAH) to track a variable number of targets [122]. E-JUDAH associates with each target a binary existence variable that indicates whether the correspondent target is active or not and assumes that a large and fixed target number (including both active and inactive targets) is known in advance. Green proposed a *reversible jump Markov chain Monte Carlo* (RJMCMC) approach [52] to generate samples with different dimensions by "jump" operations in a Markov chain. Khan *et al.* used this method to track a variable number of interacting ants [71]. Smith *et al.* used RJMCMC to track varying numbers of interacting people [114]. To simplify the sampling procedure for "jump", [71] and [114] restrict proposals of RJMCMC to add or remove a single target. Mori and Chong gave a *point process* formalism for multitarget tracking problems [93].

The *FInite Set STatistics* (FISST) proposed by Mahler is the first systematic treatment of multisensor-multitarget tracking. FISST results in a systematic Bayesian unification of detection, classification, tracking, decision-making, sensor management, group-target processing, expert-systems theory and performance evaluation in multiplatform, multisource, multievidence, multitarget, multigroup problems

[49, 83]. The problem of FISST is its computational complexity when dealing with multiple sensors and multiple targets. To reduce the complexity, Mahler devised the *Probability Hypothesis Density* (PHD) filter as an approximation of multitarget filter [85]. There are two implementation methods for the PHD filter. One is particle filter implemented by Zajic [131], Sidenbladh [112] and Vo *et al.* [125]. Johansen *et al.* [66] and Clark and Bell [28] demonstrated the convergence property of the particle PHD filter respectively, which show that the empirical representation of the PHD converges to the true PHD. The other is Gaussian mixture proposed by Vo and Ma [124]. Clark and Vo [27] proved the convergence property of the Gaussian mixture PHD filter.

The particle PHD filter differs from the other particle filters. There has been much work on tracking multiple targets using particle filters. These works can mainly be divided into two categories: 1) one particle filter with the joint state space for multiple targets [60, 64, 72]; 2) one mixed particle filter, where each component (mode or cluster) is modelled with one individual particle filter that forms part of the mixture [121, 98]. The disadvantage of the 1st approach is that it is difficult to find an efficient importance sampling function when the target number is large and the dimension of the joint state space is high. The 2nd approach usually uses some heuristic methods to determine the target number firstly and then derives states of targets. For example, the boosted particle filter [98] adds, deletes, and merges targets according to the overlapping regions between the targets detected

by Adaboost algorithm and the existing targets (from the authors' programs [3]). The particle PHD filter is similar with the second approach but the particle PHD filter has an important property that the integral of the PHD over a region in a state space is the expected number of targets within this region. The PHD filter can automatically determine the target number by this property, which differs from the other multitarget particle filters.

There have been some applications of FISST and PHD. Sidenbladh tracked vehicles in terrain using the FISST particle filtering [113]. Tobias and Lanterman [118] applied the particle PHD filter for radar tracking problem. Clark and Bell [29] used the particle PHD filter in tracking in sonar images. Ikoma *et al.* filtered trajectories of feature points in images using the particle PHD filter [61]. Haworth *et al.* presented a system to detect and track metallic objects concealed on people in sequences of millimeter-wave images [55]. Clark *et al.* developed the Gaussian mixture PHD multitarget tracker [25] and demonstrated it on forward-looking sonar data [28]. While tracking people has wide applications and no work has been done on automatically tracking people or human groups using the PHD filter.

Some applications in business intelligence such as customer statistics only care about the number of people or groups near a store and do not need the identification information of them. The PHD filter is suitable for these scenarios. Under these cases, the current measurements for the PHD filter are not a single measurement but a random measurement set. Therefore, how to design importance function

of the particle PHD filter to incorporate the current measurement set remains a challenge.

1.2 Objective of this study

The goal of this thesis is to extend mathematical methods of stochastic processes, especially Bayesian filtering, to visual tracking problems. Two new developments of Bayesian filtering, the particle filter and the probability hypothesis density filter, are chosen them as our theoretical methods. The tracking scenarios are:

- The use of multiple cameras to track a target is investigated to deal with long-time full occlusion in a particular camera. The two cameras have a common overlapping field of view in the experiments. The target may be occluded by the environment such as tree or building in one camera while it can be seen by another camera. A spatio-temporal Bayesian filtering is designed to fuse the spatial information from both cameras and the temporal information of dynamic system. The spatio-temporal Bayesian filtering may be nonlinear and non-Gaussian, so it is implemented using an adaptive particle filter which can automatically rank data from two cameras and assigns weights according to the quality of data in the fusion process.

- When the number of targets are unknown and time-varying, the dimensions of

state and measurement of dynamic system are variable. Tracking pedestrians in a corridor of a shopping center is an example. To deal with this problem, tracking a variable number of people in image sequences using the probability hypothesis density filter is investigated. When people appear, merge, split, and disappear in the field of view of a camera, the aim is to track the time-varying number of targets and their position.

1.3 Contributions

The contributions of this thesis are summarized as below:

- A data fusion approach is proposed for visual tracking using multiple cameras with overlapping fields of view. A spatio-temporal recursive Bayesian filter is designed to fuse spatial information from multiple cameras and temporal information of dynamic systems. An adaptive mixed particle filter is formulated to realize the spatio-temporal recursive Bayesian filter. The mixed particle filter adapts to the dynamic change of data quality of two cameras. The algorithm can recover the target's position even under long-time complete occlusion in a camera.

- A multitarget recursive Bayesian filter, the Probability Hypothesis Density (PHD) filter, is applied to a visual tracking problem: tracking a variable number of people or human groups in image sequences. The PHD filter

is implemented using two methods: both sequential Monte Carlo method and Gaussian mixture. Two importance functions and weight functions of the particle PHD filter are developed. The importance function for survival targets theoretically extends the optimal importance function of the linear Gaussian model from single-measurement case to measurement-set (multi-measurement) case. Whereas the importance function for spontaneous birth targets is a data-driven method for spontaneous birth objects. A scene-driven method is also proposed to initialize the Gaussian mixture probability hypothesis density filter and model the birth of new objects. The results show when people or groups appear, merge, split, and disappear in the field of view, these PHD filters can track the variable number of objects and their positions.

1.4 Organization of the thesis

This thesis is organized as follows. Chapter 2 provides a literature review for target tracking. Chapter 3 presents an adaptive mixed particle filter for tracking and data fusion of multiple cameras. Tracking a variable number of pedestrians or human groups in image sequences using the probability hypothesis density filter is introduced in chapter 4. Chapter 5 concludes this thesis and provides the future work.

Chapter 2

Literature review

Tracking is a fundamental problem for many applications such as video surveillance [30, 106] and human motion analysis [23, 4, 44, 91, 126]. Radar tracking [8, 9, 10] and visual tracking [18] are two important research fields and have different scenarios and motivations. On one hand, the target (e.g., airplane, missile or ship) in radar tracking usually appears as a spot on the radar screen with complex maneuvers such as acceleration, turns, or stops. So research on radar tracking focuses on capturing dynamics of targets accurately. On the other hand, the target (e.g., person or vehicle) in visual tracking is usually captured in form of image sequences. Rich information such as intensity, color, or contour contained in target pictures can be used for distinguishing, tracking and other form of analysis. So research on visual tracking concentrates on building an likelihood function which can accurately differentiate the object from the background.

Fig. 2.1 gives an overview of target tracking methods reviewed in this chapter. Section 2.1 introduces the Bayesian filtering framework in target tracking. Section 2.2 presents the basic filtering technologies for modelling dynamics of targets. Section 2.3 describes some commonly used likelihood functions for visual tracking. Multicamera tracking methods are introduced in section 2.4. Multitarget tracking and tracking a variable number of targets is presented in section 2.5. A summary is provided in section 2.6.

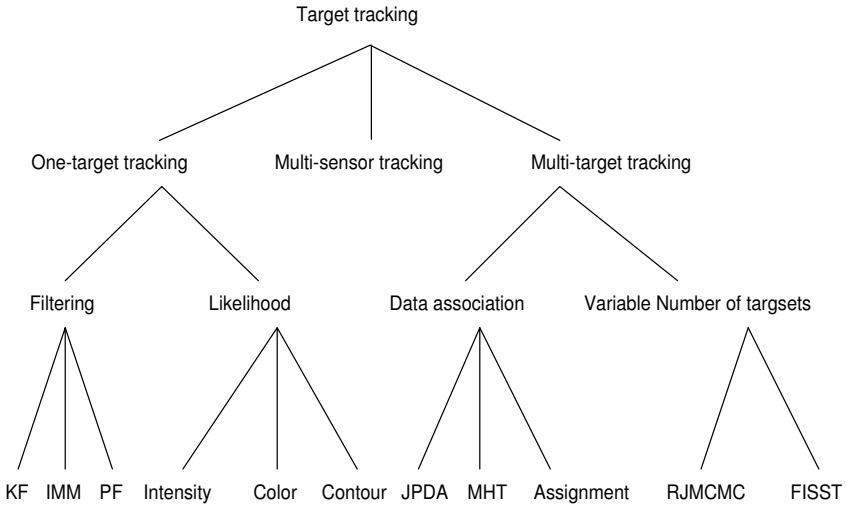


Figure 2.1: Overview of target tracking methods.

2.1 Bayesian filtering framework

Most tracking problems are formulated using a dynamic system and a state space approach [8, 9, 10]. Under the formulation of a dynamic system, the *state* of a

target at time t is denoted as x_t, which may be its position, velocity, acceleration, width, height, etc. The *observation* or *measurement* of the sensor at time t is denoted as y_t, e.g., an image captured by a camera. The series of observations or measurements from time 1 to t are denoted as $y_{1:t}$. For simplicity, the dynamic system is usually modelled as a first-order Markov process, representing it as a dynamic equation:

$$x_t = f_t(x_{t-1}, u_t) \tag{2.1}$$

where $f_t : R^{n_x} \times R^{n_u} \rightarrow R^{n_x}$ is possibly a nonlinear function of the state, $\{u_t\}$ is an independent identical distribution (i.i.d) process noise sequence, and n_x, n_u are dimensions of the state and process noise vectors, respectively. The observation or measurement equation is:

$$y_t = h_t(x_t, v_t) \tag{2.2}$$

where $h_t : R^{n_x} \times R^{n_v} \rightarrow R^{n_y}$ is possibly a nonlinear function, $\{v_t\}$ is an i.i.d measurement noise sequence, and n_y, n_v are dimensions of the measurement and measurement noise vectors, respectively.

From a Bayesian perspective, the tracking problem is to recursively calculate some degree of belief in the state x_t at time t given the data $y_{1:t}$ up to time t, i.e., to construct the conditional probability density function (pdf):

$$p(x_t|y_{1:t}) \tag{2.3}$$

It is assumed that the initial pdf $p(x_0|y_0) \equiv p(x_0)$ is known as the prior. Then

the pdf $p(x_t|y_{1:t})$ can be recursively obtained in two stages of Bayesian filtering: *prediction* and *update*.

Suppose that the pdf at time $t-1$ is available. The prediction stage involves using the dynamic model (2.1) to obtain the prior probability density function of the state at time t via the Chapman-Kolmogorov equation [65]:

$$p(x_t|y_{1:t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1} \tag{2.4}$$

At time t, a measurement y_t becomes available and is used to update the prior pdf via the Bayes' rule:

$$p(x_t|y_{1:t}) = \frac{p(y_t|x_t)p(x_t|y_{1:t-1})}{p(y_t|y_{1:t-1})} \tag{2.5}$$

where the normalizing constant is:

$$p(y_t|y_{1:t-1}) = \int p(y_t|x_t)p(x_t|y_{1:t-1})dx_t \tag{2.6}$$

In this stage, the measurement y_t is used to modify the prior pdf to obtain the required posterior probability density function of the current state.

Equ. (2.4) and (2.5) comprise the *recursive Bayesian filtering*. The problem is that the above method is only a conceptual solution; since the integrals are not tractable in most cases.

2.2 Filtering methods

Targets in radar tracking are usually the maneuvering objects (e.g., airplane or missile) and have complicated dynamics. Much work (including linear and nonlinear filters) has been done to model dynamics of targets using the filtering technologies [16, 12]. The *Kalman filter* and the *interacting multiple model filter* are two examples of linear filters, whereas the *particle filter* is an example of nonlinear filters. Daum provided an review for nonlinear filters [35].

Kalman first described a recursive solution to the discrete-data linear filtering problem [68]. The Kalman filter is the standard algorithm for radar tracking scenarios. The Bayesian filtering (2.4) and (2.5) has a closed-form solution under these conditions: i) the dynamic function $f(\cdot)$ of the system in (2.1) is linear; ii) the measurement function $h(\cdot)$ of the system in (2.2) is linear; iii) the process noise u_t is Gaussian distribution; iv) the measurement noise v_t is Gaussian distribution; and v) the initial state error is Gaussian distribution. Under these conditions, The dynamic system (2.1) and (2.2) can be written as

$$x_t = F_t x_{t-1} + u_t \tag{2.7}$$

$$y_t = H_t x_t + v_t \tag{2.8}$$

where F_t and H_t are known matrices defining the linear functions. The covariance of u_t and v_t are Q_t and R_t respectively. The posterior density is Gaussian and can be parameterized by a mean and a covariance (only the first and second order

moments) [127]:

$$p(x_{t-1}|y_{1:t-1}) = N(x_{t-1}; m_{t-1|t-1}, P_{t-1|t-1}) \tag{2.9}$$

$$p(x_t|y_{1:t-1}) = N(x_t; m_{t|t-1}, P_{t|t-1}) \tag{2.10}$$

$$p(x_t|y_{1:t}) = N(x_t; m_{t|t}, P_{t|t}) \tag{2.11}$$

where

$$m_{t|t-1} = F_t m_{t-1|t-1} \tag{2.12}$$

$$P_{t|t-1} = Q_t + F_t P_{t-1|t-1} F_t^T \tag{2.13}$$

$$m_{t|t} = m_{t|t-1} + K_t(y_t - H_t m_{t|t-1}) \tag{2.14}$$

$$P_{t|t} = P_{t|t-1} - K_t H_t P_{t|t-1} \tag{2.15}$$

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

$$K_t = P_{t|t-1} H_t^T S_t^{-1} \tag{2.16}$$

$$S_t = H_t P_{t|t-1} H_t^T + R_t \tag{2.17}$$

K_t is the Kalman gain and S_t is the covariance of the innovation term $y_t - H_t m_{t|t-1}$. The transpose of a matrix F is denoted by F^T. The Kalman filter is an estimator with the minimum mean square error (MMSE) for linear systems with Gaussian noise. When the system functions $f(\cdot)$ and $h(\cdot)$ are non-linear, the *extended*

Kalman filter (EKF) uses their local linearization as an approximation of the optimal Bayesian filtering [45, 5]. The unscented transform has been used in a EKF framework and the resulted filter is called the *unscented Kalman filter* (UKF) [67]. The Kalman filter requires that the target has only one motion model. However, an actual maneuver target usually shows multiple motion behaviors. Blom and Bar-Shalom introduced an interacting multiple model (IMM) approach as a hybrid state estimation scheme to deal with this problem [20]. The main feature of IMM is its ability to estimate the state of a dynamic system with several behavior modes which can switch from one to another. IMM makes a good compromise between complexity and performance: its computational requirements are nearly linear in the size of the problem (number of models) while its performance is almost the same as that of an algorithm with quadratic complexity. Yeddanapudi *et al.* applied IMM estimator for tracking formation and maintenance in a multisensor air traffic surveillance scenario [129]. Kirubarajan *et al.* presented a variable structure interacting multiple model (VSIMM) estimator for tracking groups of ground targets on constrained paths [73]. Mazor *et al.* provided a survey on interacting multiple model methods for target tracking [88].

Particle filter, or called the sequential Monte Carlo method, [39, 37, 56], developed from the 1990s, is a Monte Carlo simulation based method and can be applied to solve nonlinear and non-Gaussian problems, which are usual for tracking under

complex environments. The basic idea of particle filter is that the posterior probability distribution can be approximated by a set of randomly chosen weighted samples (or particles). The first particle filter, *bootstrap*, was proposed by Gordon *et al.* [51]. Liu and Chen presented a general framework for applying Monte Carlo methods to dynamic systems [80]. Their framework includes importance sampling, resampling, rejection sampling, and Markov chain iterations. Doucet *et al.* provided a Bayesian filtering framework of sequential simulation based methods for nonlinear and non-Gaussian dynamic models [41]. Their other major contribution are summarizing the methods for selecting importance sampling functions.

The basic particle filter includes two components: *importance sampling* and *resampling*. Importance sampling introduces a new *importance function* (or *importance density*, *proposal density*) and draws samples from the importance function instead of the posterior distribution. The selection of the importance function is a key issue for the particle filter as it affects the sampling efficiency of the particle filter [41]. The bootstrap algorithm [51] uses the dynamic function (2.1) as the importance function. But this sampling method does not consider the information of the current measurement so that it may be inefficient. Many methods have been proposed to overcome this problem. For example, Doucet *et al.* presented a local linearization method for the importance function [41]. Thrun *et al.* proposed a hybrid importance function to improve the sampling efficiency [117]. van der Merwe *et al.* used the unscented Kalman filter to generate the importance function [120].

If only importance sampling is used, the particle filter has the *degeneracy* problem, i.e., after a few iterations, all but few particles will have negligible weights. Doucet proved that the variance of the weights increases over time [41]. Therefore, it is impossible to avoid the degeneracy problem. Resampling introduces a selection step to eliminate samples with low weights and multiply samples with high weights to reduce the variance of the weights. There are some resampling methods: *sampling importance resampling* (SIR) [51], *residual resampling* [80], and *systematic sampling* [74].

Resampling reduces the diversity of particles and this problem is known as *sample impoverishment*. To solve this problem, Gilks and Berzuini combined the *Markov chain Monte Carlo* (MCMC) method [47, 6] with the particle filter and proposed the *resample-move* algorithm [46].

There have been some new developments on particle filters. Pitt and Shephard proposed an *auxiliary particle filter* [104]. Kotecha and Djutic designed *Gaussian particle filter* [76] and *Gaussian sum particle filter* [77]. *Rao-Blackwellised particle filter* [80, 41] was used in dynamic Bayesian networks [40]. Particle filters have been widely used in radar tracking scenarios [50, 22, 54, 69, 57, 92].

Much work has been done on tracking a visual target using particle filters. Isard and Blake proposed the first particle filter based visual tracking algorithm, the *condensation* (CONditional DENSity propogATION) algorithm [62], and later combined

it with the statistical technique of importance sampling [63]. They demonstrated their method using a hand tracker which combines color blob-tracking with a contour model. Arnaud *et al.* [7] proposed a *conditional particle filter* for point tracking. Rui and Chen used the *unscented particle filter* [120] to obtain a better importance function [109]. Pérez *et al.* [103] introduced importance sampling for data fusion of multiple cues (colour and motion) and different sensors (camera and microphone).

There has been much work on tracking multiple visual targets using particle filters. These works can mainly be divided into two categories: i) one particle filter with the joint state space for multiple targets [64, 72]; ii) one mixed particle filter, where each component (mode or cluster) is modelled with one individual particle filter that forms part of the mixture [121, 98]. Isard and MacCormick presented a Bayesian multiple-blob tracker, *BraMBle* [64], to track multiple persons using a particle filter. Khan *et al.* used the *trans-dimensional Markov chain Monte Carlo* method to track a variable number of ants [72]. Vermaak *et al.* [121] introduced a mixed particle filter to model each component (mode or target) with an individual particle filter and form part of the mixture. Okuma *et al.* [98] combined Vermaak's method with the *Adaboost* algorithm [123] to track multiple hockey players.

2.3 Likelihood functions for visual tracking

Visual tracking focuses on the likelihood functions which represent objects in images. Blake [18] and Yilmaz *et al.* [130] provided the surveys for object tracking methods respectively. The typical likelihood functions include intensity based methods, contour based methods, color based methods, motion feature based methods, spatio-temporal consistency based methods, and object priors based methods.

Template matching is an intensity-based method and to match a template on an image to minimize the misregistration error [81, 119, 111]. Lucas and Kanade used the spatial intensity gradient of images as feature to find a matching by the Newton-Raphson iteration [81]. Tomasi and Kanade designed a method to determine the feature windows that are best suitable for tracking [119]. Shi and Tomasi proposed an optimal feature selection criterion and a feature monitoring method that can detect occlusions [111].

Edge, contour and shape are important image features and can be used in visual tracking. Isard and Blake parameterized the contour using spline functions [19] and used contour as feature for tracking [62]. Paragios and Deriche applied geodesic active contours and *level sets* method to detect and track moving objects [99]. Mansouri used the level sets approach to region tracking [87]. Zhou *et al.* presented an information framework for robust shape tracking [133].

Color is usually selected as feature for tracking because it is rotation and scale

invariant to a certain extent. Comaniciu *et al.* combined the *mean shift* algorithm with the color histogram for visual tracking [32]. Pérez *et al.* [102] combined the particle filter with the color histogram and proposed the color-based probabilistic tracking. Nummiaro *et al.* [96] presented an adaptive color-based particle filter.

Motion features such as optical flow [59] are widely used in object tracking. Barron *et al.* [14] evaluated the performances of different optical flow techniques which include differential, matching, energy-based, and phase-based methods. Their experiments showed that the first-order, local differential method of Lucas and Kanade [81] and the local phase-based method of Fleet and Jepson [42] were the most reliable optical flow methods.

The spatio-temporal consistency is also used for moving object segmentation and tracking. Zhong and Chan [132] combined edge and color information to improve the object motion estimation result. Then they used the long-term spatio-temporal constraints to track objects over long sequences.

The prior knowledge of objects has been used for constraining the object segmentation/tracking process. For example, Rosenhaln *et al.* [108] integrated 3D shape knowledge into a variational model for level set based image segmentation and contour based 3D pose tracking.

2.4 Multicamera tracking methods

Tracking using multiple cameras has been done in much work in recent years. Multicamera tracking can be categorized into 2 classes: overlapping field with view and non-overlapping field with view. Kettnaker and Zabih [70] and Pasula *et al.* [100] introduced 2 multicamera tracking methods with non-overlapping field of view respectively. As for multicamera tracking with overlapping field of view, the commonly used methods include camera switching, geometry constraint and appearance matching.

Nummiaro *et al.* [97] presented a color-based multiview tracking method. The camera with the highest similarity for face's color histogram is selected and switched to carry on the tracking task. Cai and Aggarwal [21] presented a framework for tracking coarse human models from sequences of synchronized monocular grayscale images in multiple camera coordinates. When the system predicted that the active camera would no longer have a good view of the subject of interest, tracking would be switched to another camera which provides a better view and requires the least switching to continue tracking.

Homography is an important geometry constraint for points in a plane and can be used for multicamera tracking. Black and Ellis [15] presented a method for multicamera image tracking in the context of image surveillance. Viewpoint correspondence between the detected objects was established by using the ground

plane homography constraint. M2Tracker developed by Mittal and Davis [90] was a multiview approach to segmenting and tracking people in a cluttered scene using a region-based stereo algorithm. The DARPA VSAM project [31] at CMU used site model, camera calibration and model-based geolocation for video surveillance. Chang and Gong [24] presented a multicamera system based on Bayesian modality fusion to track multiple people in an indoor environment. Bayesian networks were used to combine geometry-based modalities with recognition-based modalities for matching subjects between consecutive image frames and between multiple camera views. Krumm *et al.* [78] created a practical person-tracking system using 2 sets of color stereo cameras. The stereo images were used to locate people, whereas the color images are used to maintain the identities of people.

2.5 Multitarget tracking methods

When tracking multiple targets, one needs to use *data association* method to associate observations of sensors with targets. For example, if there are two targets, a person and a car, and the camera detects three foreground blobs, data association must determine which blob belongs to the person, the car, or the clutter environment. As a result, there are multiple choices for association. The aim of data association is to find the best association scheme. Bar-Shalom and Li introduced multitarget multisensor tracking methods [11].

Bar-Shalom and Tse presented a *probabilistic data association* (PDA) scheme to calculate the association probability for each observation at the current time to the target of interest [13]. PDA assumes that: 1) there is only one target of interest; 2) at most one of observation can be target-originated; 3) the other observations are due to false alarm or clutter. On the basis of PDA, Fortmann and Bar-Shalom proposed a *joint probabilistic data association* (JPDA) approach [43]. JPDA can track multiple targets and assumes that: 1) the number of targets is known; 2) each target has been initialized; 3)a target can generate at most one measurement; and 4) a measurement could be originated from at most one target. JPDA allows a target's state to be updated by a weighted sum of all observations in its gate scope. Therefore, JPDA is a spatial information fusion method.

Reid proposed a *multiple hypothesis tracking* (MHT) approach for data association [107]. MHT is a deferred decision which forms multiple data association hypotheses when observation-to-target are uncertain. Rather than selecting the best hypothesis or combining multiple hypotheses as JPDA, the hypotheses are propagated into the future until the subsequent data can resolve the uncertainty. Therefore, MHT is a temporal information fusion method. MHT enumerates the exhausted hypotheses and the computational complexity increases exponentially with time. Cox and Hingorani [33] described a method to find m-best hypotheses using Murty's algorithm [94]. Blackman gave a summary of MHT for multiple target tracking [17].

The assignment algorithm [101, 36] essentially perform constrained optimization problems to find an optimal data association solution. Pattipati *et al.* [101] developed a Lagrangian relaxation technique to solve the 2-D assignment problem. Deb *et al.* [36] presented a generalized S-D assignment algorithm.

When the number of targets varies, the dimensions of the state and observation vectors are time-varying. Many approaches have been proposed to solve this problem. Miller *et al.* generated the conditional mean estimates of an unknown number of targets and target types via *jump-diffusion process* [89]. Mori and Chong gave a *point process* [34, 116] formalism for multitarget tracking problems [93]. Green proposed a *reversible jump Markov chain Monte Carlo* (RJMCMC) approach to deal with the problems where the dynamic variable of the simulation does not have fixed dimension [52, 53]. Godsill and Vermaak applied RJMCMC for tracking tasks where that state process arrives at unknown times that differ from the observation arrival times [48]. Khan *et al.* used RJMCMC to track a variable number of interacting ants [72].

The *finite set statistics* (FISST) proposed by Mahler is the first systematic treatment of multisensor multitarget tracking. It contributes to a unified framework of data fusion [49, 83, 84, 86]. Under this theory, the state of a target (e.g., position and velocity) and the measurement of a sensor are represented by state and measurement vector respectively; the state set of multiple targets and the measurement set of multiple sensors are represented as *Random Finite Sets* (RFS).

The problem of FISST is its computational complexity when dealing with multiple sensors and multiple targets. To reduce the complexity, Mahler devised the *Probability Hypothesis Density* (PHD) filter as an approximation of multitarget filter [85]. The PHD filter can jointly estimate the time-varying number of objects and their states from a sequence of measurement sets. The PHD filter was implemented using particle filters (Zajic [131], Sidenbladh [112], and Vo *et al.* [125]) and Gaussian mixture (Vo and Ma [124]). Johansen *et al.* [66] and Clark and Bell [26] demonstrated the convergence property of the particle PHD filter respectively, which show that the empirical representation of the PHD converges to the true PHD. Clark and Vo [27] proved the convergence property of the Gaussian mixture PHD filter.

There have been some applications of FISST and PHD. Sidenbladh tracked vehicles in terrain using the FISST particle filtering [113]. Tobias and Lanterman [118] applied the particle PHD filter for the radar tracking problem. Clark and Bell [29] used the particle PHD filter in tracking in sonar images. Ikoma *et al.* filtered trajectories of feature points in images using the particle PHD filter [61]. Haworth *et al.* presented a system to detect and track metallic objects concealed on people in sequences of millimeter-wave images [55]. Clark *et al.* developed the Gaussian mixture PHD multitarget tracker [25] and demonstrated it on forward-looking sonar data [28].

2.6 Summary

From the review of this chapter, target tracking has developed from single-sensor single-target tracking to multisensor multitarget tracking. For multisensor multitarget tracking, stochastic processes and Bayesian filtering provide powerful tools. However, there are many unsolved problems, e.g., the information fusion of multiple cameras and tracking unknown and time-varying number of targets. In the following two chapters, I present our contributions to solve these challenges.

Chapter 3

Adaptive particle filter for tracking

3.1 Introduction

Occlusion, especially complete occlusion, is a difficult problem for visual tracking using a single camera. Multiple cameras provide information of a moving target from multiple views. As such, occlusions do not occur in all cameras and fusion of data from multiple cameras enables tracking of a moving target with desirable performance. The objective of this chapter is developing a method for combining information from multiple cameras with emphasis on dealing with the problem of occlusion.

There have been some research efforts on tracking using multiple cameras. In most of these works, information from different cameras are analyzed and the one with the best view is selected to overcome the problem of occlusion. Cai and Aggarwal [21], for example, selected a camera using three criteria: i) ability to track the object in the future; ii) robust spatial matching between cameras; and iii) ability to maintain objects over the most number of frames. Nummiaro *et al.* [97] selected the camera with the highest similarity for face's colour histogram. These switching criteria are often heuristic and have no theoretical basis. Only one camera's information is used although if multiple cameras can observe the target. The key challenge is to design a data fusion method to fuse information from multiple cameras.

An adaptive importance sampling strategy for the particle filter is proposed here, which can automatically rank data from multiple cameras and assign weights according to the quality of data in the fusion process.

This chapter is organized as follows. Section 3.2 provides a basic introduction to the spatio-temporal recursive Bayesian filter. Section 3.3 introduces the particle filter. Section 3.4 presents an adaptive particle filter which can combine information from multiple cameras. The experimental results, presented in section 3.5, show that the adaptive particle filter is able to recover the location of the occluded target. The details are discussed in section 3.6 and the summary is provided in section 3.7.

3.2 Spatio-temporal recursive Bayesian filter

Tracking a moving target using multiple cameras is represented as a dynamic system. The state of a moving target at time t, which is its position and size, is denoted as x_t. The target observed in an image captured by a camera at time t is denoted as y_t and the series of observations (or measurements) from time 1 to t are denoted as $Y_{1:t} = \{y_j : j = 1, \ldots, t\}$

For simplicity, the dynamical system is modelled as a first-order Markov process, representing it as a dynamic function (or a state transition function):

$$x_t = f_t(x_{t-1}, u_t) \tag{3.1}$$

where f_t is possibly a nonlinear function of the state x_{t-1}, $\{u_t\}$ is an independent identical distribution (i.i.d) noise sequence. The observation or measurement function is:

$$y_t = h_t(x_t, v_t) \tag{3.2}$$

where h_t is possibly a nonlinear function and $\{v_t\}$ is an i.i.d noise sequence.

In this visual tracking task, the state x_t is defined as:

$$x_t = (location_t, size_t) \tag{3.3}$$

where $location_t$ is the image coordinate at time t of the top left corner of the bounding box of the moving object, and $size_t$ is the width & height at time t of the bounding box of the moving object. The dynamic function is assumed to follow

a constant position model:

$$x_t = x_{t-1} + u_t \tag{3.4}$$

where u_t is a zero-mean Gaussian white noise vector with variance Σ_u.

As defined, the measurement y_t of a target from a camera is the target's image in the video frame at time t. Object recognition techniques are used to locate targets from video frames. The *mean shift* algorithm [32] is used to locate the target's image in each camera. The measurement y_t is a bounding box indicating the target's candidate location:

$$y_t = (location_t, size_t) \tag{3.5}$$

The resulting measurement y_t is:

$$y_t = x_t + v_t \tag{3.6}$$

v_t is a zero mean Gaussian white noise vector with variance Σ_v.

The number of cameras is denoted as C and $y_{t,c}$ is the measurement from the c^{th} camera at time t. $Y_{t,1:C}$ are measurements of all cameras at time t and $Y_{1:t,1:C}$ are measurements of all cameras from time 1 to t, i.e.,

$$Y_{1:t,1:C} = \{Y_{1,1:C}, Y_{2,1:C}, \ldots, Y_{t,1:C}\}$$

Our objective is to estimate the target's state x_t given all measurements from multiple cameras $Y_{1:t,1:C}$, i.e., to construct the conditional probability:

$$p(x_t|Y_{1:t,1:C}) \tag{3.7}$$

Suppose that the probability density function (pdf) $p(x_{t-1}|Y_{1:t-1,1:C})$ at time $t-1$ is available. The recursive Bayesian filter consists of two stages: *prediction* and *update*. The prediction stage involves obtaining the prior pdf of the state at time t via the state transition function. The resulting prediction equation is:

$$p(x_t|Y_{1:t-1,1:C}) = \int p(x_t|x_{t-1})p(x_{t-1}|Y_{1:t-1,1:C})dx_{t-1} \tag{3.8}$$

At time t, measurements $Y_{t,1:C}$ become available and are used to update the prior pdf via the Bayes' rule as follows:

$$p(x_t|Y_{1:t,1:C}) = \frac{p(Y_{t,1:C}|x_t)p(x_t|Y_{1:t-1,1:C})}{p(Y_{t,1:C}|Y_{1:t-1,1:C})} \tag{3.9}$$

The denominator $p(Y_{t,1:C}|Y_{1:t-1,1:C})$ is called the *evidence* and it is determined as follows:

$$p(Y_{t,1:C}|Y_{1:t-1,1:C}) = \int p(Y_{t,1:C}|x_t)p(x_t|Y_{1:t-1,1:C})dx_t \tag{3.10}$$

Assume that all measurements are conditionally independent given the state because different measurements come from different cameras.

$$p(Y_{t,1:C}|x_t) = \prod_{c=1}^{C} p(y_{t,c}|x_t) \tag{3.11}$$

Using (3.11) in (3.9), the update stage becomes

$$p(x_t|Y_{1:t,1:C}) = \frac{\prod_{c=1}^{C} p(y_{t,c}|x_t)p(x_t|Y_{1:t-1,1:C})}{p(Y_{t,1:C}|Y_{1:t-1,1:C})} \tag{3.12}$$

Equations (3.8) and (3.12) comprise the spatio-temporal recursive Bayesian filter.

3.3 Particle filter

Particle filter (the *sequential Monte Carlo method*) [39, 37, 56], developed from the 1990s, is a Monte Carlo simulation based method and can be applied to solve nonlinear and non-Gaussian problems, which are usual for tracking under complex environments. The first particle filter, *bootstrap*, was proposed by Gordon *et al.* [51]. The basic idea of particle filter is that the posterior probability distribution can be approximated by a set of N randomly chosen weighted samples or particles $\{x_{0:t}^{(i)}, w_t^{(i)}\}_{i=1}^N$ as follows:

$$\widehat{p(x_{0:t}|y_{1:t})} \approx \sum_{i=1}^{N} w_t^{(i)} \delta(x_{0:t} - x_{0:t}^{(i)}) \tag{3.13}$$

where δ is Dirac delta function. Random samples $\{x_{0:t}^{(i)}\}_{i=1}^N$ are drawn from the posterior distribution. Given the samples, the expectation of the function of the state can be approximated as follows:

$$\widehat{E(f(x_{0:t}))} = \sum_{i=1}^{N} w_t^{(i)} f(x_{0:t}^{(i)}) \tag{3.14}$$

Unfortunately, sampling directly from the posterior distribution is often impossible. To overcome this difficulty, the basic particle filter uses two sampling methods: *importance sampling* and *resampling*. Liu and Chen presented a general framework for applying Monte Carlo methods to dynamic systems [80]. Their framework includes importance sampling, resampling, rejection sampling, and Markov chain iterations. Doucet *et al.* provided a Bayesian filtering framework of sequential

simulation based methods for nonlinear and non-Gaussian dynamic models [41]. Their other major contribution are summarizing the methods for selecting the importance sampling function.

Section 3.3.1 introduces the importance sampling technologies. Section 3.3.2 presents the resampling methods. The generic particle filter is summarized in section 3.3.3.

3.3.1 Importance sampling

Importance sampling introduces a new *importance function* (or *importance density*, *proposal density*) $q(x_{0:t}|y_{1:t})$ and draws samples from the importance function instead of the posterior distribution. Then the weights in (3.13) are defined as

$$w_t^{(i)} \propto \frac{p(x_{0:t}^{(i)}|y_{1:t})}{q(x_{0:t}^{(i)}|y_{1:t})} \tag{3.15}$$

As for the sequential case, at each iteration, one could have samples constituting an approximation to $p(x_{0:t-1}|y_{1:t-1})$ and want to generate a new set of samples to approximate $p(x_{0:t}|y_{1:t})$. If the importance function can be factorized:

$$q(x_{0:t}|y_{1:t}) = q(x_0)\prod_{k=1}^{t} q(x_k|x_{0:k-1}, y_{1:k}) \tag{3.16}$$

then one can obtain new samples as shown in Fig. 3.1:

The selection of the importance function is a key issue for the particle filter as it affects the sampling efficiency of the particle filter [41]. The bootstrap algorithm [51] uses the dynamic function (3.1) as the importance function. But this sampling

For times $t = 1, 2, \cdots$

- For $i = 1, \cdots, N$, sample $x_t^{(i)}$ from $q(x_t|x_{0:t-1}, y_{1:t})$ and set $x_{0:t}^{(i)} = (x_{0:t-1}^{(i)}, x_t^{(i)})$.
- For $i = 1, \cdots, N$, evaluate the importance weights up to a normalizing constant:

$$w_t^{(i)} = w_{t-1}^{(i)} \frac{p(y_t|x_t^{(i)}) p(x_t^{(i)}|x_{0:t-1}^{(i)})}{q(x_t|x_{0:t-1}, y_{1:t})} \tag{3.17}$$

- For $i = 1, \cdots, N$, normalize the importance weights:

$$\tilde{w}_t^{(i)} = \frac{w_t^{(i)}}{\sum_{j=1}^{N} w_t^{(j)}} \tag{3.18}$$

Figure 3.1: Sequential importance sampling algorithm

method does not consider the information of the current measurement y_t so that it may be inefficient. Many methods have been proposed to overcome this problem. For example, Doucet *et al.* presented a local linearization method for the importance density [41]. Thrun *et al.* proposed a hybrid importance density to improve the sampling efficiency [117]. van der Merwe *et al.* used the unscented Kalman filter to generate the importance density [120].

3.3.2 Resampling

If only importance sampling is used, the particle filter has the *degeneracy* problem, i.e., after a few iterations, all but few particles will have negligible weights. Doucet showed that the variance of the weights increases over time [41]. Therefore, it is impossible to avoid the degeneracy problem. Resampling introduces a selection step to eliminate samples with low weights and multiply samples with high weights to reduce the variance of the weights. There are some resampling methods: *sampling importance resampling* [51], *residual resampling* [80], and *systematic sampling* [74]. The basic resampling algorithm is described in Fig. 3.2.

Resampling reduces the diversity of particles and this problem is known as *sample impoverishment*. To solve this problem, Gilks and Berzuini combined the Markov chain Monte Carlo (MCMC) method [47, 6] with the particle filter and proposed the *resample-move* algorithm [46].

$$[\{x_t^{(j*)}, w_t^{(j)}, i^{(j)}\}_{j=1}^{N}] = RESAMPLE[\{x_t^{(i)}, w_t^{(i)}\}_{i=1}^{N}]$$

- Initialize the cumulative density function (CDF): $c_1 = 0$,
- FOR $i = 2 : N$,

  Construct the CDF: $c_i = c_{i-1} + w_t^{(i)}$
- END FOR
- Start at the bottom of the CDF: i=1
- Draw a starting point $u_1 \sim Uniform[0, N^{-1}]$.
- FOR $j = 1 : N$,

  Moving along the CDF: $u_j = u_1 + N^{-1}(j-1)$

  WHILE $u_j > c_i$

  $i = i + 1$

  END WHILE

  Assign sample: $x_t^{(j*)} = x_t^{(i)}$

  Assign weight: $w_t^{(j)} = N^{-1}$

  Assign parent: $i^{(j)} = i$
- END FOR

Figure 3.2: Resampling algorithm

3.3.3 Generic particle filter

A suitable measurement of the degeneracy problem of particle filter is the *effective sample size* N_{eff} introduced in [75] and defined as

$$N_{eff} = \frac{N}{1 + \mathrm{var}(w_t^{*(i)})} \tag{3.19}$$

where $w_t^{*(i)} = p(x_t^{(i)}|y_{1:t})/q(x_t^{(i)}|x_{t-1}^{(i)}, y_t)$ is referred as the "true weight". This can not be evaluated exactly, but an estimate $\widehat{N_{eff}}$ of N_{eff} can be obtained in [38] as follows:

$$\widehat{N_{eff}} = \frac{1}{\sum_{i=1}^{N}(w_t^{(i)})^2} \tag{3.20}$$

where $w_t^{(i)}$ is the normalized weight obtained using (3.17). Notice that $N_{eff} \leq N$, and the greater the effective sample size $\widehat{N_{eff}}$, the better the sampling efficiency of the algorithm. The generic particle filter uses the effective sampling size as the condition of resampling to implement the adaptive resampling. If the effective sampling size is under a threshold (e.g. half of the sample number, $N/2$), the particle filter does the resampling. Else the particle filter skips resampling procedure and iterates to the next time instant.

The generic particle filter is summarized in Fig. 3.3:

$[\{x_t^{(i)}, w_t^{(i)}\}_{i=1}^N] = ParticleFilter[\{x_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{j=1}^N, y_t]$

- FOR $i = 1 : N$,

  Draw $x_t^{(i)} \sim q(x_t | x_{t-1}^{(i)}, y_t)$

  Assign the particle a weight, $w_t^{(i)}$, according to (3.17)

- END FOR

- Calculate total weight: W=SUM$[\{w_t^{(i)}\}_{i=1}^N]$

- FOR $i = 1 : N$,

  Normalize: $w_t^{(i)} = W^{-1} w_t^{(i)}$,

- END FOR

- Calculate $\widehat{N_{eff}}$ using (3.20)

- IF $\widehat{N_{eff}} < N_T$ (a threshold of sampling efficiency)

  Resample as the resampling algorithm (Fig. 3.2)

- END IF

Figure 3.3: Generic particle filter

3.4 Adaptive mixed particle filter for multicamera tracking

In this section our adaptive mixed particle filter is introduced. Section 3.4.1 provides an overview of the adaptive mixed particle filter. Section 3.4.2 introduces the object segmentation and detection methods. Section 3.4.3 presents the likelihood function for evaluating particles. Section 3.4.4 proposes the mixed importance sampling strategy of particle filter. The weight function of particle filter is introduced in section 3.4.5. An adaptive importance sampling method is presented in section 3.4.6. Section 3.4.7 summaries the algorithm.

3.4.1 Algorithm overview

Our algorithm takes as input, images from two wide baseline fixed cameras that have an overlapping field of view. In addition to the two images, I_1 from the 1^{st} camera and I_2 from the 2^{nd} camera (Fig. 3.5), the following are also input in our algorithm:

- the target's appearance model: $16 \times 16 \times 16$ bins (RGB) colour histogram $\{hist(target)\}_{u=1}^{4096}$, where u is the bin index;
- the calibrated coordinate transform $f : x_2 \rightarrow x_1$ and $f^{-1} : x_1 \rightarrow x_2$, where x_1 is the target's location in the 1^{st} camera and x_2 is

the target's location in the 2^{nd} camera.

We make some assumptions for our algorithm: i) the target is in the same ground plane; and ii) the target's colour distribution does not change during tracking. The homography transformation [15] is used to implement the coordinate transform between cameras:

$$\tilde{x}_1 = H\tilde{x}_2 \tag{3.21}$$

where H is a 3×3 homogeneous matrix and $\tilde{x}_1$ and $\tilde{x}_2$ are the homogeneous coordinates in two cameras.

The output of our algorithm is the target's location x_1 in the 1^{st} view or x_2 in the 2^{nd} view.

3.4.2 Object segmentation

We use the *background subtraction* algorithm [31] to obtain the foreground object. We are able to do this effectively because the camera is fixed and the background image is therefore easily obtained. Let $P(x, y)$ and $B(x, y)$ represent a pixel intensity value and the background intensity value at position (x, y). Then pixel (x, y) belongs to the foreground region if:

$$|P(x, y) - B(x, y)| > Th \tag{3.22}$$

where the threshold Th is set by the experiments. The foreground image obtained using background subtraction and thresholding is usually noisy and morphological

operations are performed to remove noise. Dilation and erosion are applied to the binary foreground images. In order to eliminate "noise objects" that are not eliminated by morphological operations in the foreground image, small objects of an area smaller than a threshold are eliminated from the foreground. The resulting foreground object is our detected object. The object is occluded in a camera if the object is in the overlapping fields of view of two cameras and only one camera detects the object.

We use the *mean shift* algorithm [32] to obtain the current measurement. The approximation of the estimated position $\hat{x}$ of a target is obtained iteratively as follows:

$$\hat{x}_t = \frac{1}{C} \sum_x x w(x) g(||x - \hat{x}_{t-1}||^2) \tag{3.23}$$

where $C = \sum_x w(x) g(||x - \hat{x}_{t-1}||^2)$, g is the derivative of a particular kernel function used to build the spatial density function, and $w(x)$ is a weight that measures the degree of prevalence of the colour of pixel x in the target template relative to its prevalence in the test target.

3.4.3 Likelihood function

To evaluate how likely an image at the candidate location represents the real target, we define the likelihood of the particle by using the color likelihood in [102]. Colour measure is selected because it is rotation and scale invariant to a certain extent.

The colour histogram of the candidate image $x_t^{(i)}$, determined by our algorithm, is $\{hist(x_t^{(i)})\}_{u=1}^{4096}$. The Bhattacharyya coefficient ρ represents the similarity between the candidate image and the target image, defined as follows:

$$\rho[hist(x_t^{(i)}), hist(target)] = \sum_{u=1}^{4096} \sqrt{hist(x_t^{(i)})_u hist(target)_u} \tag{3.24}$$

The distance measure between two colour histograms is

$$Distance[hist(x_t^{(i)}), hist(target)] = \sqrt{1 - \rho[hist(x_t^{(i)}), hist(target)]} \tag{3.25}$$

This distance has several desirable properties: i) it is nearly optimal; ii) it imposes a metric structure; and iii) it is invariable to the scale of the target etc. The likelihood, which represents the similarity between the target's template and the particle's region, is defined as follows:

$$p(y_t|x_t^{(i)}) \propto -\lambda Distance^2[hist(x_t^{(i)}), hist(target)] \tag{3.26}$$

The greater a particle's likelihood, the more likely the candidate image is the real target. Pérez *et al.* [102] set $\lambda = 20$ empirically. In our experiments, our algorithm works for values of λ from 20 to 100.

3.4.4 Mixed importance sampling

Our objective is the posterior distribution:

$$p(x_t|x_{0:t-1}, y_{t,1}, y_{t,2}) \propto p(y_{t,1}|x_t)p(y_{t,2}|x_t)p(x_t|x_{t-1}) \tag{3.27}$$

In general this posterior distribution is complicated and has no closed-form solution; so we cannot directly sample from it. To overcome this difficulty, we use the importance sampling strategy of the particle filter. Traditional particle filters such as [51, 62] only use the dynamic function (3.1) as the importance function. But this function does not consider the current measurement y_t and its sampling efficiency may be low [41]. We propose our mixed importance sampling method which generates the particles from both the dynamic function and the current measurement as shown in Fig. 3.4.

For $i = 1, \ldots, N$, generating a uniformly distributed random number $r \in [0, 1)$:

- if $0 \leq r < \alpha_1$, generate a process noise $u_t^{(i)}$ and a sample $x_t^{(i)} = x_{t-1} + u_t^{(i)}$ according to the dynamic function (3.4);
- if $\alpha_1 \leq r < \alpha_1 + \alpha_2$, generate a measurement noise $v_{t,1}^{(i)}$ and a sample $x_t^{(i)} = y_{t,1} - v_{t,1}^{(i)}$ according to the current measurement of the 1^{st} camera(3.6);
- if $\alpha_1 + \alpha_2 \leq r < 1$, generate a measurement noise $v_{t,2}^{(i)}$ and a sample $x_t^{(i)} = y_{t,2} - v_{t,2}^{(i)}$ according to the current measurement of the 2^{nd} camera (3.6);

Figure 3.4: Adaptive mixed importance sampling.

Let N be the total number of samples, and the coefficients α_1, α_2, and α_3 (where

$\alpha_1 + \alpha_2 + \alpha_3 = 1$) determine the respective contributions of the dynamic function, the measurement of the 1^{st} camera, and the measurement of the 2^{nd} camera.

3.4.5 Weight function of particle filter

The weights of particles are updated in the update stage of the Bayesian filter (3.12). From (3.12), the weight of a particle should be proportional to the product of two likelihoods if the target is visible in the two cameras:

$$w_t^{(i)} \propto p(y_{t,1}|x_t^{(i)})p(y_{t,2}|x_t^{(i)}) \tag{3.28}$$

where $p(y_t|x_t^{(i)})$ is defined in our likelihood model (3.26). But if a target becomes occluded in a camera, a particle may have a high likelihood for the visible camera and a low likelihood for the occluded camera. If the weight function is the product of two likelihoods, the weight of a particle is mainly affected by the low likelihood of the occluded camera, which is not desirable. Our fusion method chooses the high likelihood to update the weight and discard the low likelihood to reduce the influence of occlusion. In summary, if the target is visible in the two cameras, we use the product of two likelihoods to update the weight of a particle as follows:

$$w_t^{(i)} = p(y_{t,1}|x_t^{(i)})p(y_{t,2}|x_t^{(i)}) \tag{3.29}$$

while if the target is occluded in a camera, we use the greater likelihood to update the weight of a particle as follows:

$$w_t^{(i)} = \max_c p(y_{t,c}|x_t^{(i)}) \tag{3.30}$$

The object segmentation stage (section 3.2) determines whether the object is visible in the two cameras or is occluded in a camera.

3.4.6 Adaptive importance sampling

Suppose we have no prior information on the mixed weights α_1, α_2, and α_3 in the importance sampling (Fig. 3.4), they may be uniform distributed, i.e.,

$$\alpha_1 = \alpha_2 = \alpha_3 = \frac{1}{3} \tag{3.31}$$

Because the "quality of data" changes over time, the importance sampling method should adapt to this change. For example, when occlusions occur in the 1^{st} camera, samples from the measurement of the 1^{st} camera should be reduced. The goal of this section is to propose an adaptive importance sampling method to track this change.

The variance of measurement noise of sensor reflects the signal-to-noise ratio (SNR) of measurement of sensor. The smaller the variance, the higher the SNR of measurement. Therefore, the variance of measurement noise is helpful to adapt the importance sampling.

Let $\Sigma_{v,1}$ and $\Sigma_{v,2}$ denote the variances of measurement noise of the 1^{st} and 2^{nd} cameras respectively. From the measurement model (3.6), the linear estimator of the position of target from measurements of the two cameras is

$$x_t = (1-\beta)(y_{t,1} - v_{t,1}) + \beta(y_{t,2} - v_{t,2}) \tag{3.32}$$

where β is the parameter and can be varying during the tracking process. Thus,

$$E(x_t) = (1-\beta)y_{t,1} + \beta y_{t,2} \tag{3.33}$$

$$\text{var}(x_t) = E(x_t^2) - E^2(x_t) = (1-\beta)^2\Sigma_{v,1} + \beta^2\Sigma_{v,2} \tag{3.34}$$

$$\frac{\partial \text{var}(x_t)}{\partial \beta} = 2(\beta-1)\Sigma_{v,1} + 2\beta\Sigma_{v,2} \tag{3.35}$$

then we get

$$\beta = \frac{\Sigma_{v,1}}{\Sigma_{v,1} + \Sigma_{v,2}} \tag{3.36}$$

Therefore, the optimal linear estimator of x_t from the measurements of two cameras is

$$E(x_t) = \frac{\Sigma_{v,2}}{(\Sigma_{v,1} + \Sigma_{v,2})}y_{t,1} + \frac{\Sigma_{v,1}}{(\Sigma_{v,1} + \Sigma_{v,2})}y_{t,2} \tag{3.37}$$

Now the problem is to determine the variance of measurements noise of cameras. In particle filter, the weights of samples reflects the qualities of samples. For example, if a measurement has high signal-to-noise ratio, the samples from that measurement should have high weights and small weight variance. Therefore, the variance of weights of samples is a suitable measure of the variance of measurement noise of sensors.

The mean of weights of samples drawn from the measurement of the 1^{st} camera is:

$$\bar{w}_1 = \frac{1}{|I_1|}\sum_{i \in I_1} w_t^{(i)} \tag{3.38}$$

where I_1 is the index set of samples drawn from the measurement of the 1^{st} camera. The variance of measurement noise of the 1^{st} camera is estimated by the variance

of weights of samples drawn from the measurement of the 1^{st} camera:

$$\Sigma_{v,1} = \frac{1}{|I_1|}\sum_{i\in I_1}(w_t^{(i)} - \bar{w}_1)^2 \tag{3.39}$$

The mean of weights of samples from the measurement of the 2^{nd} camera is:

$$\bar{w}_2 = \frac{1}{|I_2|}\sum_{i\in I_2} w_t^{(i)} \tag{3.40}$$

where I_2 is the index set of samples drawn from the measurement of 2^{nd} camera. The variance of measurement noise of the 2^{nd} camera is estimated by the variance of weights of samples drawn from the measurement of the 2^{nd} camera:

$$\Sigma_{v,2} = \frac{1}{|I_2|}\sum_{i\in I_2}(w_t^{(i)} - \bar{w}_2)^2 \tag{3.41}$$

Thus, the adaptive importance sampling is proposed as follows:

$$\alpha_1 = \frac{1}{3}, \quad \alpha_2 = \frac{2\Sigma_{v,2}}{3(\Sigma_{v,1}+\Sigma_{v,2})}, \quad \alpha_3 = \frac{2\Sigma_{v,1}}{3(\Sigma_{v,1}+\Sigma_{v,2})} \tag{3.42}$$

Our importance sampling uses both the state at the previous time and the current measurements of two cameras to generate samples of the target's state (Fig. 3.5). This method may be generalized to information fusion of $C(> 2)$ cameras. The mean of weights of samples drawn from the measurement of the c^{th} camera is:

$$\bar{w}_c = \frac{1}{|I_c|}\sum_{i\in I_c} w_t^{(i)} \tag{3.43}$$

where I_c is the index set of samples drawn from the measurement of the c^{th} camera. The variance of measurement noise of the c^{th} camera is estimated by the variance

(a) Image I_1 from the 1^{st} camera (b) Image I_2 from the 2^{nd} camera

Figure 3.5: The samples (the target's position and size) are generated by the adaptive importance sampling method. The red boxes are samples generated from the dynamic function, the green boxes are samples generated from the measurement of the 1^{st} camera, and the blue boxes are samples generated from the measurement of the 2^{nd} camera.

of weights of samples drawn from the measurement of the c^{th} camera:

$$\Sigma_{v,c} = \frac{1}{|I_c|} \sum_{i \in I_c} (w_t^{(i)} - \bar{w}_c)^2 \tag{3.44}$$

3.4.7 Algorithm summary

Our algorithm for data fusion of two cameras is summarized below.

1. Initialize the target models for two views when the target enters the field of view. Set $\alpha_1 = \alpha_2 = \alpha_3 = \frac{1}{3}$.

2. For $c = 1, 2$, obtain the measurements $y_{t,c}$ of the c^{th} camera.

3. For $i = 1, \ldots, N$, generating a uniformly distributed random number $r \in [0, 1)$.

   - if $0 \leq r < \alpha_1$, generate a process noise $u_t^{(i)}$ and a sample $\tilde{x}_t^{(i)} = x_{t-1}^{(i)} + u_t^{(i)}$ according to (3.4);
   - if $\alpha_1 \leq r < \alpha_1 + \alpha_2$, generate a measurement noise $v_{t,1}^{(i)}$ of the 1^{st} camera and a sample $\tilde{x}_t^{(i)} = y_{t,1} - v_{t,1}^{(i)}$ according to (3.6);
   - if $\alpha_1 + \alpha_2 \leq r < 1$, generate a measurement noise $v_{t,2}^{(i)}$ of the 2^{nd} camera and a sample $\tilde{x}_t^{(i)} = y_{t,2} - v_{t,2}^{(i)}$ according to (3.6);

4. For $i = 1, \ldots, N$, evaluate the importance weights as (3.29) or (3.30).

5. Normalize the importance weights:

$$w_t^{(i)} = \frac{\tilde{w}_t^{(i)}}{\sum_{j=1}^{N} \tilde{w}_t^{(j)}} \tag{3.45}$$

6. The current location is

$$\widehat{E(x_t)} = \sum_{i=1}^{N} w_t^{(i)} \tilde{x}_t^{(i)} \tag{3.46}$$

7. Compute the means and variances of weights of samples as (3.38)-(3.41). Set α_1, α_2 and α_3 as (3.42).

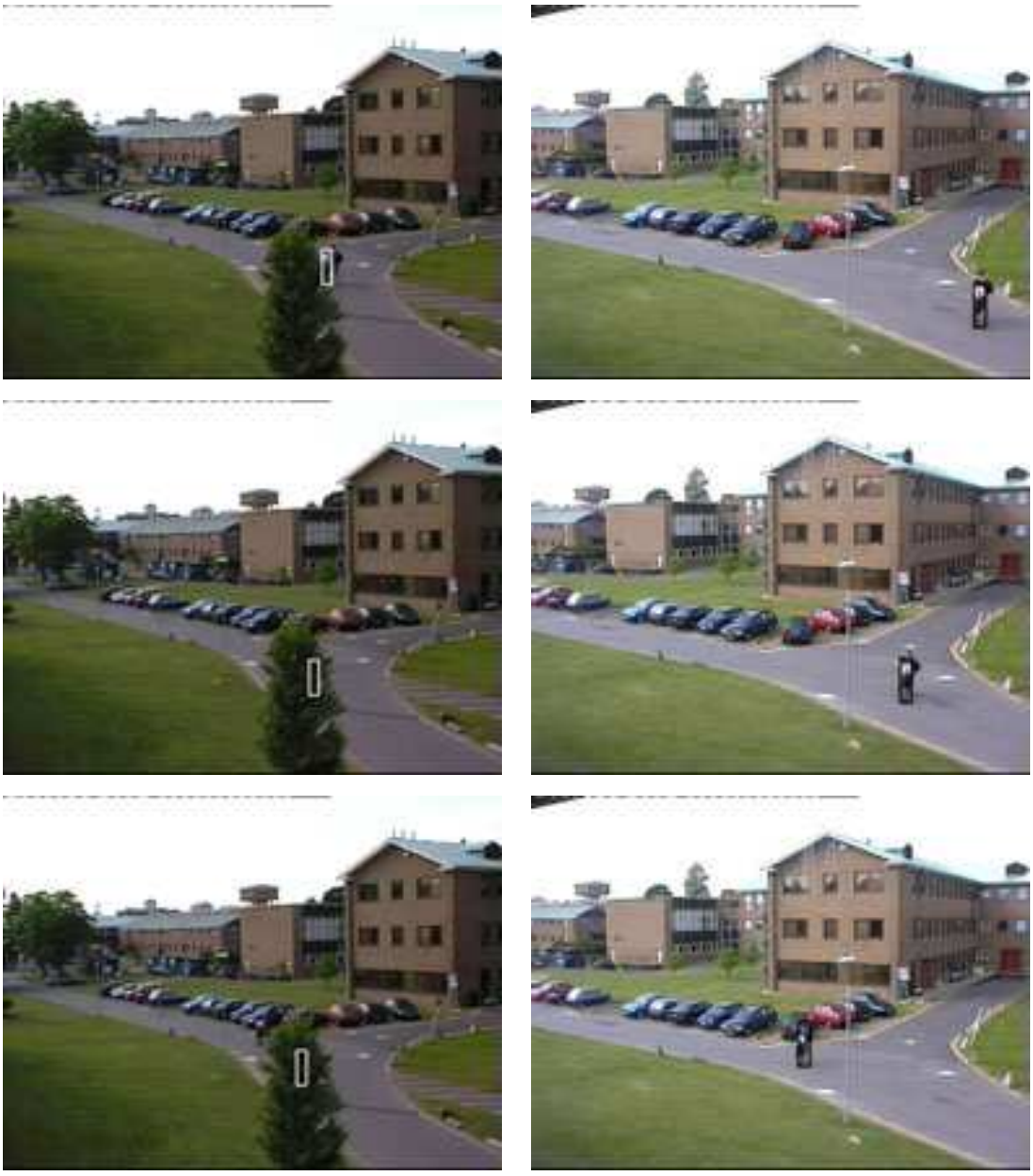
8. Resample $\tilde{x}_t^{(i)}$ to get $x_t^{(i)}$ and assign $w_t^{(i)} = 1/N$.

9. $t \leftarrow t + 1$. Go to step 2 till the last frame of the image sequence.

3.5 Experimental results

We test our adaptive particle filter using the PETS2001 dataset [1] and compare our algorithm with the mean shift algorithm [32] and the condensation algorithm [62]. The PETS2001 dataset has image sequences of two cameras from different views. For the mean shift algorithm and the condensation algorithms, tracking is carried separately for separate cameras (e.g., only using the image sequence of the 1^{st} camera for the 1^{st} camera's tracking). In contrast, our algorithm tracks a target using image sequences obtained from both cameras. The mean shift algorithm generates an estimated position at each time while both our algorithm and the condensation algorithm generate 50 candidate positions (i.e., 50 samples) at each time. These three tracking algorithms are implemented in Matlab.

The mean shift algorithm fails to track the person for frames 352 and 465 of the 1^{st} camera (Fig. 3.6(a)) but there is no problem in tracking the person for frames 352 and 465 of the 2^{nd} camera (Fig. 3.6(b)). For the 1^{st} camera, the person is completely occluded by the tree in frame 352 and the target is lost in frame 352 because no information about the target is available. When the person reappears in frame 465, the mean shift algorithm is not able to track him. As shown in Fig. 3.6(b), the mean shift algorithm is a good approach for tracking using a single camera when there is no occlusion, but it has difficulties in tracking a completely occluded target.

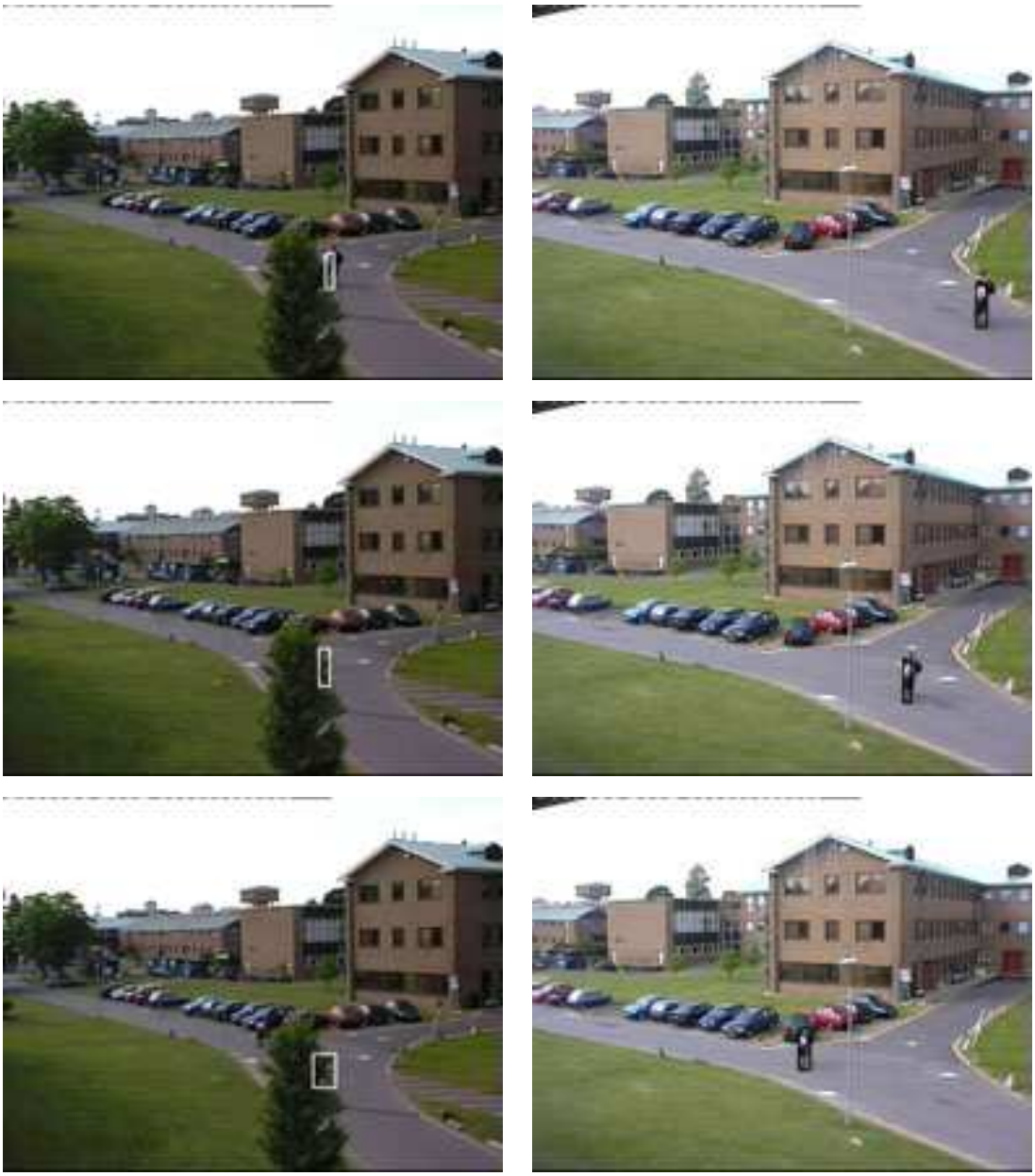


(a) Camera 1 (b) Camera 2

Figure 3.6: Tracking results using the mean shift algorithm for frames 293, 352 and 465 superimposed on images obtained from two cameras.

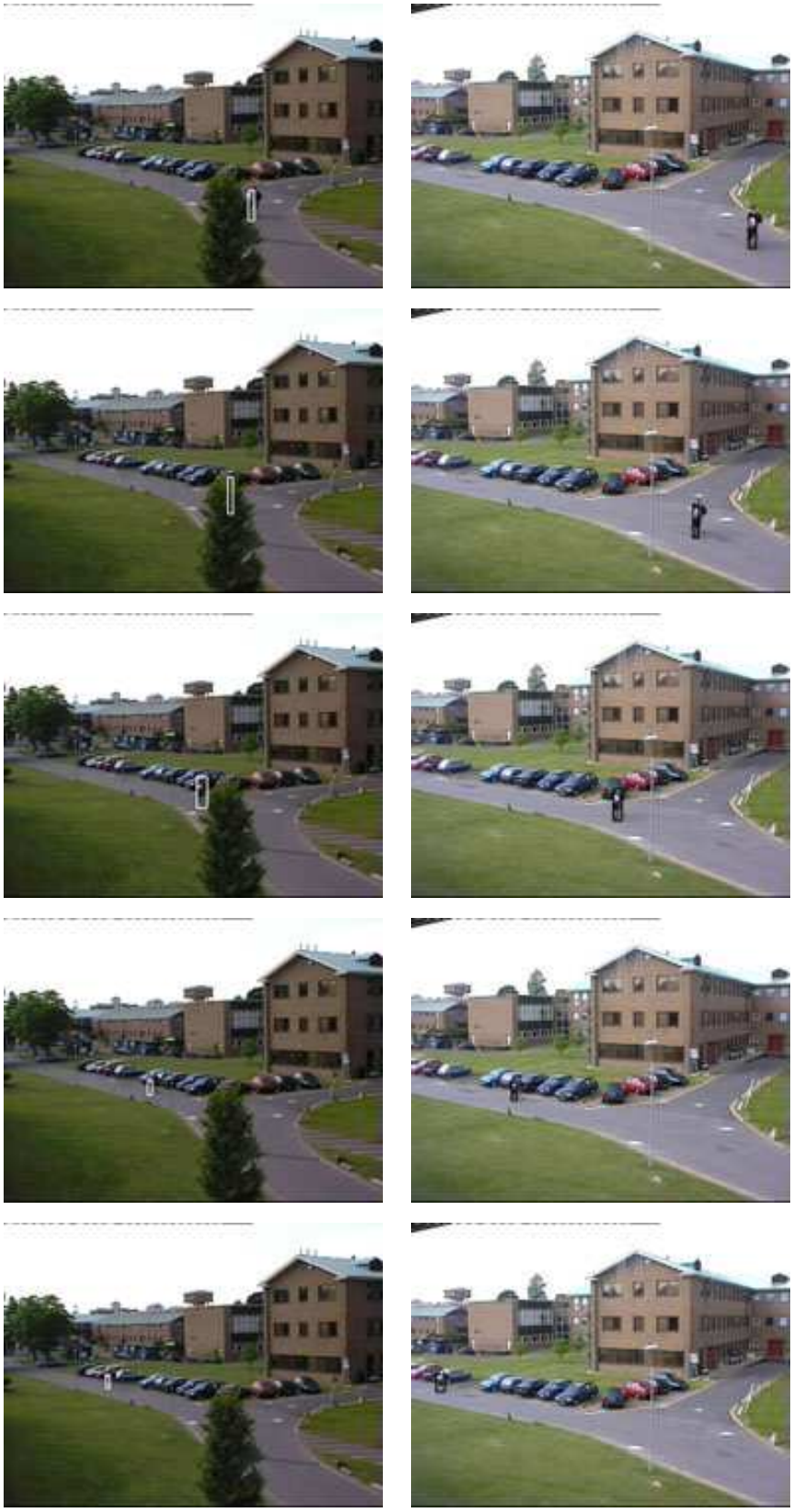
The condensation algorithm fails to track the person for frames 352 and 465 of the 1^{st} camera (Fig. 3.7(a)) but there is no problem in tracking the person for frames 352 and 465 of the 2^{nd} camera (3.7(b)). For the 1^{st} camera, the person is lost in frame 352 because no information is available to update the target's position. When the person reappears in frame 465, the condensation algorithm is not able to track him. The results show that the condensation algorithm is good for tracking using a single camera when there is no complete occlusion.

Our adaptive particle filter is able to track the person for frames 293, 352, 465, 651 and 837 of both cameras (Fig. 3.8). Some manually chosen correspondent points are used to obtain the nine parameters of the homography transformation. Although the target is occluded in the 1^{st} camera, our algorithm is still able to track the target in frame 352 using the information of the 2^{nd} camera. When the person reappears in frame 465, our algorithm is able to track him and continues to track him for the subsequent frames till the person moves out of the field of view. The results show that data fusion of multiple cameras can be used to solve the long-duration occlusion problem.



(a) Camera 1 (b) Camera 2

Figure 3.7: Tracking results using the condensation algorithm for frames 293, 352 and 465 superimposed on images obtained from two cameras.



(a) Camera 1 (b) camera 2

Figure 3.8: Tracking results using the adaptive particle filter for frames 293, 352, 465, 651 and 837 superimposed on images obtained from two cameras.

For the tracking task in Fig. 3.8, our data fusion method analyzes the quality of data of two cameras and dynamically adapts the mixed weights α_2 and α_3 (3.42) in our importance sampling (Fig. 3.9). At the beginning, most samples are drawn from the measurement of the 1^{st} camera. When a complete occlusion occurs in the 1^{st} camera at about frame 320, the samples drawn from the 2^{nd} camera increases. When a partial occlusion in the 2^{nd} camera occurs at about frame 420, less samples are drawn from the 2^{nd} camera.

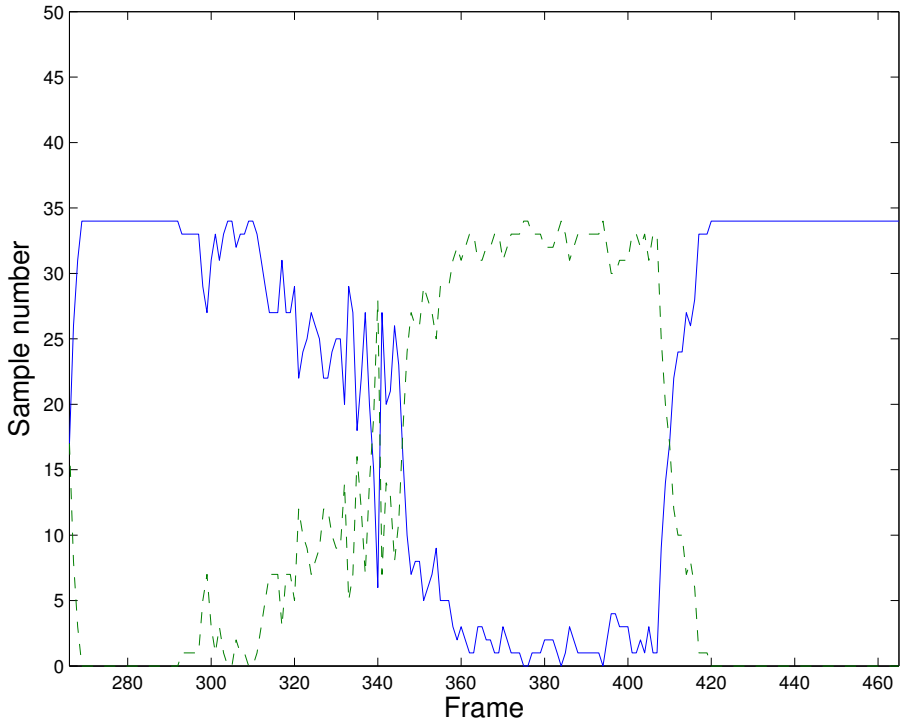
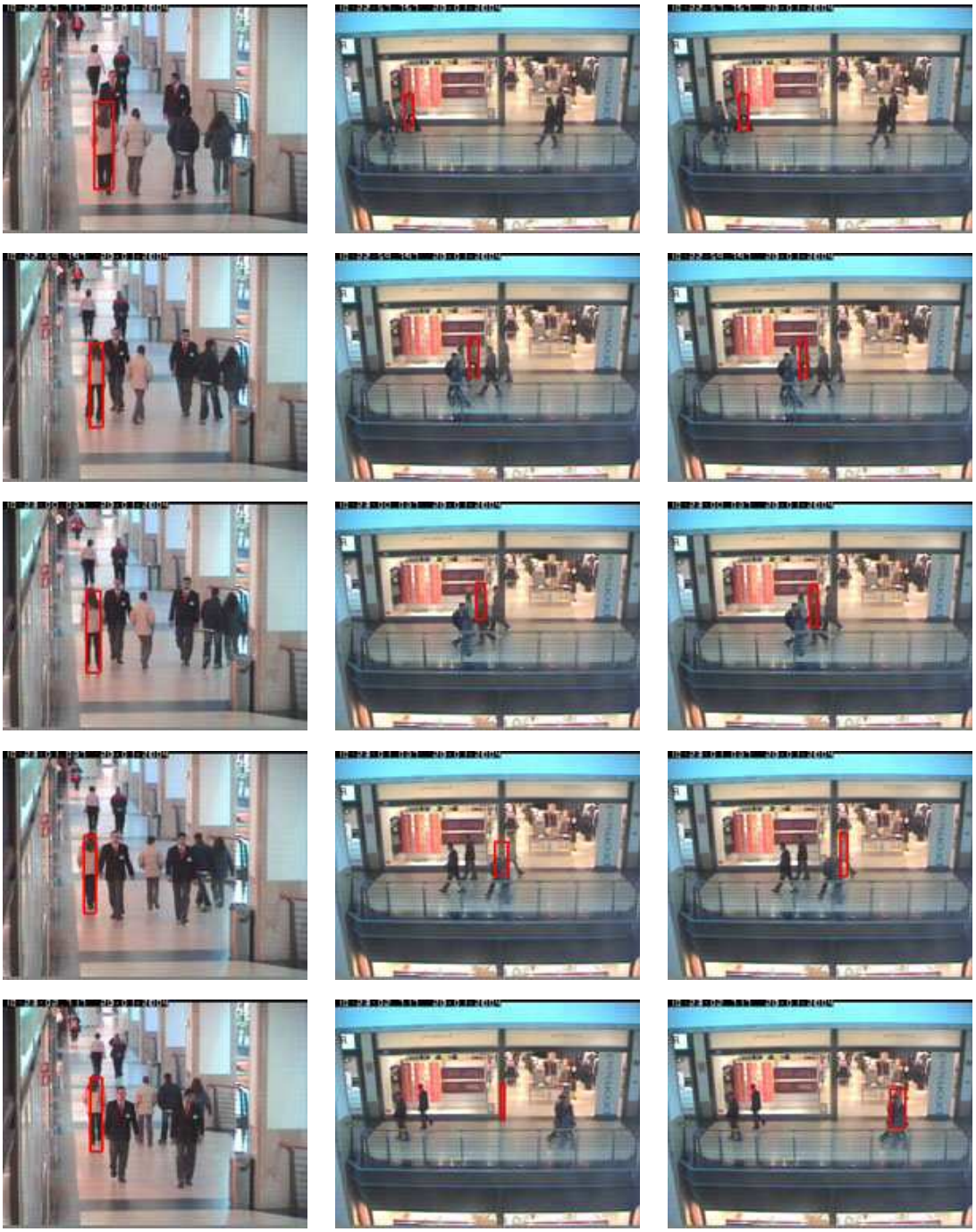


Figure 3.9: Dynamically allocated sample numbers during tracking. The solid line is the number of samples drawn from the 1^{st} camera. The dashed line is the number of samples drawn from the 2^{nd} camera.

We also tested the condensation algorithm and our algorithm using the most recent dataset of the European Commission Funded CAVIAR project [2] (Fig. 3.10). The condensation algorithm succeeds to track the person in frames 840, 891, 897, 922 and 964 of the 1^{st} camera (Fig. 3.10a); but it fails to track the same person in frames 999 and 1041 of the 2^{nd} camera (Fig. 3.10b) because the tracked person is confused with a pillar in the background. Our data fusion algorithm is able to track the target in frames 917, 968, 974, 999, and 1041 using information from both cameras till the target moves out of the overlapping fields of view of two cameras (Fig. 3.10c).



(a) (b) (c)

Figure 3.10: (a) Tracking results using the condensation algorithm for frames 840, 891, 897, 922 and 964 of the 1^{st} camera; (b) tracking results using condensation algorithm for frames 917, 968, 974, 999 and 1041 of the 2^{nd} camera; (c) tracking results using the adaptive particle filter for frames 917, 968, 974, 999 and 1041 of the 2^{nd} camera. Frame 840 of the 1^{st} camera and frame 917 of the 2^{nd} camera are at the same time.

3.6 Discussions

3.6.1 Target size

The minimum size of the target is mainly determined by the colour distribution of the target and the background. When the colour of the target differs from the background, a portion of the target is sufficient to track the target. In our experiments, the average size of the target is 22×56 pixels ($w \times h$) for Fig. 3.8(a) and 13×69 for Fig. 3.8(b), 18×92 for Fig. 3.10(a) and 10×47 for Fig. 3.10(c). These experiments show that the minimum size of a person can be 10×47 pixels.

3.6.2 Comparison with other multicamera tracking methods

Compared with other multicamera tracking methods such as [21, 97], our method is a data fusion method while [21, 97] are switching methods. In [21, 97], there is switching among cameras to choose one camera with the best view. For example, Cai and Aggarwal [21] selected a camera using three criteria: i) ability to track the object in the future; ii) robust spatial matching between cameras; and iii) ability to maintain objects over the most number of frames. Nummiaro *et al.* [97] selected the camera with the highest similarity for face's colour histogram. For the tracking task in Fig. 3.8, their methods always use information from the 2^{nd} camera for

tracking but not information from the 1^{st} camera because the 2^{nd} camera has the best view. Their methods do not recover the trajectory of the occluded object in the 1^{st} camera. Only one camera's information is used at every time instant. In contrast, our method produces the candidate positions from information of two cameras according to importance sampling. Next, we evaluate the weights of the candidate positions using the likelihoods of two cameras. For Fig. 3.8(a), our method tries to recover the position of the completely occluded object in the 1^{st} camera. Information of both cameras is always used at every time instant.

3.6.3 Adaptive mixed weights for importance sampling

We discuss here the influence of the mixed weights α_i on our algorithm (Fig. 3.4) using the effective sample size $\widehat{N_{eff}}$ (3.20). For the tracking task in Fig. 3.8, the average effective sample size of the adaptive algorithm is 33 samples (the average of the dashed line in Fig. 3.11) while the average effective sample size of the fixed algorithm is 8 samples (the average of the solid line in Fig. 3.11) among the total 50 samples during the 290 frames. The solid line is the effective sample size of the fixed importance sampling algorithm as (3.31). The dashed line is the effective sample size of the adaptive importance sampling algorithm as (3.42).

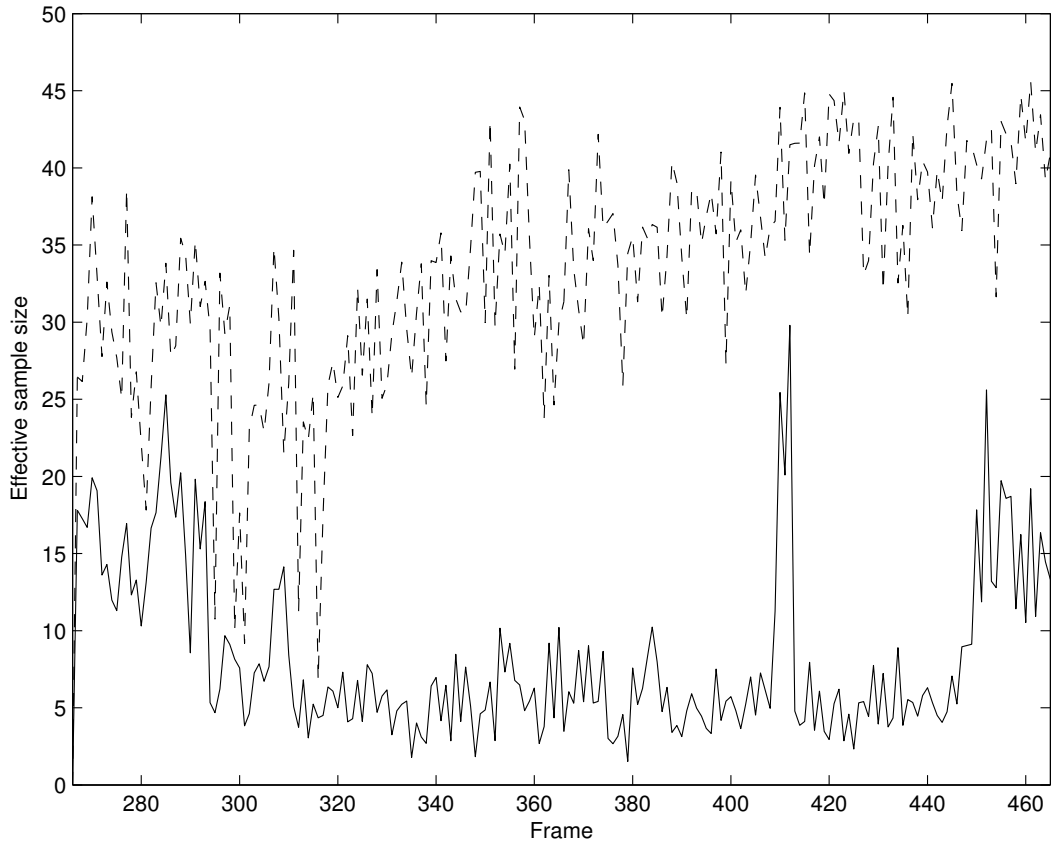


Figure 3.11: Comparison for the effective sample sizes. The solid line is the effective sample size of the fixed importance sampling (3.31) and the dashed line is the effective sample size of the adaptive importance sampling (3.42).

3.7 Summary

This chapter proposes a data fusion method based on an adaptive particle filter for visual tracking using multiple cameras with the overlapping fields of view. A theoretical framework based on the spatio-temporal recursive Bayesian filter is proposed for data fusion of multiple cameras. The spatio-temporal recursive Bayesian filter is formulated using an adaptive particle filter. The adaptive particle filter uses an adaptive importance sampling method to fuse information from multiple cameras. The algorithm is able to automatically recover the location of an occluded target while the mean shift algorithm and the condensation algorithm experience difficulties when tracking an occluded target. Therefore, information fusion of data from multiple cameras can solve the problem of occlusion.

Chapter 4

The PHD filter for visual tracking

4.1 Introduction

Tracking multiple targets remains a challenge [105]. Tracking problems are usually modelled as a dynamic system [8, 9, 10] whose order is fixed when there is the fixed number of targets. However, the problem becomes challenging when the number of targets is unknown and variable because the state or observation dimensions is time-varying under this situation. The following works are some attempts to meet this challenge. Reid proposed *multiple hypothesis tracking* (MHT) algorithm which enumerates multiple track-to-measurement association hypotheses during a period

till one hypothesis can be verified [107]. The problem of MHT is the potential combinatorial explosion in the number of hypotheses. Miller *et al.* generated the conditional mean estimates of an unknown number of targets and target types via *jump-diffusion* process [89]. Musicki *et al.* proposed *integrated probabilistic data association* (IPDA) [95] as a recursive formula for both data association and probability of target existence. Vermaak *et al.* presented the *existence joint probabilistic data association filter* (E-JPDAF) to track a variable number of targets [122]. E-JPDAF associates with each target a binary existence variable that indicates whether the correspondent target is active or not and assumes that a large and fixed target number (including both active and inactive targets) is known in advance. Green proposed a *reversible jump Markov chain Monte Carlo* (RJMCMC) approach [52] to generate samples with different dimensions by "jump" operations in a Markov chain. Khan *et al.* used this method to track a variable number of interacting ants [71]. Smith *et al.* used RJMCMC to track varying numbers of interacting people [114]. To simplify the sampling procedure for "jump", Ref. [71] and [114] assume only one target dead or birth at every time. Mori and Chong gave a *point process* formalism for multitarget tracking problems [93].

The *FInite Set STatistics* (FISST) proposed by Mahler is the first systematic treatment of multisensor multitarget tracking. It contributes to a unified framework of data fusion [49, 83]. The problem of FISST is its computational complexity when dealing with multiple sensors and multiple targets. To reduce the complexity,

Mahler devised the *Probability Hypothesis Density* (PHD) filter as an approximation of multitarget filter [85]. There are two implemented methods for the PHD filter. One is particle filter implemented by Zajic [131], Sidenbladh [112] and Vo *et al.* [125]. Johansen *et al.* [66] and Clark and Bell [28] demonstrated the convergence property of the particle PHD filter respectively, which show that the empirical representation of the PHD converges to the true PHD. The other is Gaussian mixture proposed by Vo and Ma [124]. Clark and Vo [27] proved the convergence property of the Gaussian mixture PHD filter.

The particle PHD filter differs from the other particle filters. There has been much work on tracking multiple targets using particle filters. These works can mainly be divided into two categories: 1) one particle filter with the joint state space for multiple targets [60, 64, 72]; 2) one mixed particle filter, where each component (mode or cluster) is modelled with one individual particle filter that forms part of the mixture [121, 98]. The disadvantage of the 1st approaches is that it is difficult to find an efficient importance sampling function when the target number is large and the dimension of the joint state space is high. The 2nd approaches usually use some heuristic methods to determine the target number first and derive the states of targets. For example, the boosted particle filter [98] adds, deletes, and merges targets according to the overlapping regions between the targets detected by Adaboost algorithm and the existing targets (from the authors' programs [3]). The particle PHD filter is similar with the second approach but the particle PHD

filter has an important property that the integral of the PHD over a region in a state space is the expected number of targets within this region. The PHD filter can automatically determine the target number by this property, which differs from the other multitarget particle filters.

There have been some applications of FISST and PHD. Sidenbladh tracked vehicles in terrain using the FISST particle filtering [113]. Tobias and Lanterman [118] applied the particle PHD filter for radar tracking problem. Clark and Bell [29] used the particle PHD filter in tracking in sonar images. Ikoma *et al.* filtered trajectories of feature points in images using the particle PHD filter [61]. Haworth *et al.* presented a system to detect and track metallic objects concealed on people in sequences of millimeter-wave images [55]. Clark *et al.* developed the Gaussian mixture PHD multitarget tracker [25] and demonstrated it on forward-looking sonar data [28].

Some applications in business intelligence such as customer statistics only care about the number of people or groups near a store and do not need the identification information of them. The PHD filter is suitable for these scenarios. In this chapter, object detection is combined with the probability hypothesis density filter to automatically track an unknown and variable number of people or groups in image sequences without human intervention. The procedure is outlined in Fig. 4.1.

Figure 4.1: PHD visual tracking implementation.

The PHD filter is implemented by 2 methods: both particle filter and Gaussian mixture. A key issue for the particle PHD filter is the design of importance function. Most of previous works on importance function [51, 41] only care about the fixed number of targets, whereas the PHD filter is to deal with the variable number of targets. At the same time, the previous particle PHD filters [112, 125] use the dynamic model of system as importance function. But this choice of importance function does not consider the current measurements and may be inefficient. Moreover, the current measurements for the PHD filter are not a single measurement but a random measurement set. Therefore, how to design importance function of the particle PHD filter to incorporate the current measurement set remains a challenge.

Assume that the tracked targets consist of two classes: survival targets and spontaneous birth targets. We propose importance functions and weight functions of particle filter for survival targets and spontaneous birth targets. The importance function for survival targets is an theoretical extension of the optimal importance function of linear Gaussian model. For this extension we provide a mathematical

proof under some assumptions. Whereas the importance function for spontaneous birth targets is a Gaussian mixture with means being the centroids of new detected foreground blobs. This is a data-driven method for particle PHD filter.

We also propose a scene-driven method to initialize the Gaussian mixture PHD filter and to model the appearance/birth of new objects. This filter combines the data-driven method (detection) with the model-driven method (the PHD filter) and the scene-driven method (prior knowledge).

Our results show that both the particle PHD filter and the Gaussian mixture PHD filter could track the variable number of people or groups and their positions when people or groups appear, merge, split, and disappear in the field of view of a camera.

4.2 Detecting foreground people

Detection methods for visual tracking include background subtraction with a mixture of Gaussian as background model [115] and statistical background modelling [79]. In our work, we use the statistical background modelling which incorporates spectral, spatial, and temporal features to characterize the background appearance. Background is divided into 2 classes: static background and dynamic background. The color $c = [R, G, B]^T$, the gradient $e = [g_x, g_y]^T$ are selected as features of static

background while the color co-occurrence $cc = [c_t^T, c_{t-1}^T]^T$ ($c_t^T = [R_t, G_t, B_t]^T$) is selected as features of dynamic background.

The principal feature representation of background is constructed as follows. Let v_i be the feature vector sorted in descending order with respect to the probability $p_s(v_i|b)$ that is v_i being observed as a background at the pixel $s = (x, y)$. Then there would be a small integer $N(v)$, a high percentage value M_1 and a low percentage value M_2 such that

$$\sum_{i=1}^{N(v)} p_s(v_i|b) > M_1 \quad \text{and} \quad \sum_{i=1}^{N(v)} p_s(v_i|f) < M_2 \tag{4.1}$$

where $p_s(v_i|f)$ is the probability of v_i being observed as a foreground at the pixel s. The $N(v)$ feature vectors are defined as the principal features of the background at the pixel s. A table of statistics of principal features is established as follows:

$$T_v(s) = \begin{cases} p_v^t(b) \\ \\ \{S_v^t(i)\} \quad i = 1, \cdots, M(v)) \end{cases} \tag{4.2}$$

where $p_v^t(b)$ is the learned prior probability of the pixel s belonging to the background based on the vector v and $\{S_v^t(i)\}$ is the statistics of the $M(v)$ most frequent feature vectors and defined as follows:

$$S_v^t(i) = \begin{cases} p_{v_i}^t = P_s(v_i) \\ \\ p_{v_i|b}^t = P_s(V_i|b) \\ \\ v_i = (v_{i1}, \cdots, v_{iD(v)})^T \end{cases} \tag{4.3}$$

where $P_s(v_i)$ is the prior probability of the feature vector v_i being observed as the position s and $D(v)$ is the dimension of the feature vector v.

For gradual background changes, the table $T_c(s)$ ($v = c$, e, or cc) is updated at each time by:

$$p_v^{t+1}(b) = (1-\alpha)p_v^t(b) + \alpha L_b^t \tag{4.4}$$

$$p_{v_i}^{t+1} = (1-\alpha)p_{v_i}^t + \alpha L_{v_i}^t \tag{4.5}$$

$$p_{v_i|b}^{t+1} = (1-\alpha)p_{v_i|b}^t + \alpha(L_b^t L_{v_i}^t) \tag{4.6}$$

where α is a learning rate, $L_b^t = 1$ if s is classified as a background point at time t in the final segmentation, otherwise, $L_b^t = 0$. $L_{v_i}^t = 1$ if the ith vector of the table $T_v(s)$ matches the input vector v, otherwise, $L_{v_i}^t = 0$. For "once-off" background changes, the learning operation is:

$$p_v^{t+1}(b) = 1 - p_v^t(b) \tag{4.7}$$

$$p_{v_i}^{t+1} = p_{v_i}^t \tag{4.8}$$

$$p_{v_i|b}^{t+1} = \frac{p_{v_i}^t - p_v^t(b)p_{v_i|b}^t}{p_v^{t+1}(b)} \tag{4.9}$$

for $i = 1, \cdots, N(v)$.

The foreground object detection consists of 4 stages: change detection, change classification, background maintenance, and foreground segmentation. The background subtraction and the temporal (or interframe) difference are used for change detection. Their results are used for classifying each pixel to static point or dynamic

point. For background maintenance, principal feature of static point is updated as (4.4)-(4.6) while principal feature of dynamic point is updated as (4.7)-(4.9). The morphological operation is applied to the foreground blobs and small regions are removed to reduce noise. The centroids of remaining foreground blobs are chosen as the measurements and are input to the following PHD filter.

4.3 Tracking model

The linear Gaussian dynamic model with the constant velocity (pp. 273, [12]) is used:

$$x_{t+1} = Fx_t + u_t \tag{4.10}$$

where the state of a target x_t consists of its position and velocity

$$x_t = \begin{bmatrix} x_t & \dot{x}_t & y_t & \dot{y}_t \end{bmatrix}^T \tag{4.11}$$

T is the transpose, $[x_t, y_t]^T$ is the position and $[\dot{x}_t, \dot{y}_t]^T$ is the velocity at time t, the state-transition matrix is

$$F = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.12}$$

the system noise $u_t = [u_{t,1}, u_{t,2}, u_{t,3}, u_{t,4}]^T$ is mutually independent zero-mean Gaussian white noise with covariance $\Sigma_u = \sigma_u^2 I_4$, and I_n is $n \times n$ identify matrix. Only

position measurements are available and the measurement model is

$$y_t = Hx_t + v_t \tag{4.13}$$

the measurement matrix is

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{4.14}$$

the measurement noise $v_t = [v_{t,1}, v_{t,2}]^T$ is mutually independent zero-mean Gaussian white noise with covariance $\Sigma_v = \sigma_v^2 I_2$.

4.4 Finite set statistics

The finite set statistics contributes to a unified framework of multisensor multitarget tracking and data fusion [49, 83, 84, 86]. There are a number of direct mathematical parallels between single-sensor single-target statistics and multisensor multitarget statistics. The parallels is summarized in Table 4.1.

In this section the major elements of FISST are introduced. The problem of accurately modelling multitarget state spaces and multisensor multitarget measurement spaces is described in section 4.4.1. Belief-mass functions, set integrals, and set derivatives are introduced in section 4.4.2; and their application to multisensor multitarget formal Bayesian modelling in section 4.4.3. The multisource multitarget Bayesian filter is described in section 4.4.4. Probability generating functionals and their functional derivatives are introduced in section 4.4.5.

Table 4.1: Single-target versus multi-target statistics

Random Vector $\mathbf{y}$	**Random Finite Set** Ψ	**Random Finite Set** Ψ
observation vector y	observation set Y	
sensor model $y_t = h(x_t, v_t)$	multitarget sensor model $\Sigma_t = E(X_t) \bigcup C(X_t)$	
motion model $x_{t+1} = f(x_t, u_t)$	multitarget sensor model $\Xi_{t+1\|t} = D_t(X) \bigcup B_t(X)$	
probability mass function $p_{\mathbf{y}}(S) = \Pr(y \in S)$	belief mass function $\beta_\Psi(S) = \Pr(\Psi \cap S)$	probability generating functional (p.g.fl.) $G_\Psi[h]$
Radon-Nikodym derivative $\frac{dp_{\mathbf{y}}}{dy}$	set derivative $\frac{\delta \beta_\Psi}{\delta Y}(S)$	functional derivative $\frac{\delta G_\Psi}{\delta Y}[h]$
density function $f_{\mathbf{y}}(y) = \frac{dp_{\mathbf{y}}}{dy}$	multitarget density $f_\Psi(Y) = \frac{\delta \beta_\Psi}{\delta Y}(\emptyset)$	multitarget density $f_\Psi(Y) = \frac{\delta G_\Psi}{\delta Y}[0]$
Lebegue integral $\int_S f_{\mathbf{y}}(y)dy = p_{\mathbf{y}}(S)$	set integral $\int_S f_\Psi(Y)\delta Y = \beta_\Psi(S)$	set integral $\int_S h^Y f_\Psi(Y)\delta Y = G_\Psi[h]$
expected value $\bar{y} = \int y f_{\mathbf{y}}(y)dy$	probability hypothesis density $D_\Psi(y) = \int f_\Psi(\{y\} \cup Y)\delta Y$	probability hypothesis density $D_\Psi(y) = \frac{\delta G_\Psi}{\delta y}[1]$
likelihood function $f_t(y\|x)$	multitarget likelihood $f_t(Y\|X)$	
Markov density $f_{t+1\|t}(x\|x')$	multitarget Markov density $f_{t+1\|t}(X\|X')$	
posterior density $f_{t\|t}(x\|y_{1:t})$	multitarget posterior $f_{t\|t}(X\|Y_{1:t})$	

4.4.1 Random state sets and random measurement sets

The complete description of the state of a multitarget system requires a *unified state representation*: a finite set of the form $X = \{x_1, \ldots, x_n\}$ where n is the number of targets and $x_1, \ldots, x_n$ are the state vectors of the individual targets (in general, x is assumed to include a discrete identity/label state variable). This description must include the possibility $n = 0$, i.e., no target is present, in which case $X = \emptyset$. Such a unified representation accounts for the fact that n is variable and that targets have no physically inherent order. Thus $\{x_1, x_2\} = \{x_2, x_1\}$ is a single unified state model of two targets with state vectors x_1, x_2. On the other hand, vectors (x_1, x_2) and (x_2, x_1) do not correctly represent the physical multitarget state since they do so redundantly and cannot model its inherent permutation symmetry.

In a careful Bayesian approach the unknown state must be a random quantity. Consequently, the unknown state set at time step t must be a randomly varying finite set $\Xi_{t|t}$. One cannot define a random variable of any kind without, typically, first defining a topology on the space of targets to be randomized and then defining random elements of that space in terms of the Borel subsets [82]. The space of state sets is topologized using the Mathéron "hit-or-miss" topology [49]. Once this is done, the probability law of a finite random state-set Ξ is its probability-mass function (a.k.a. probability measure) $p_\Xi(O) = \Pr(\Xi \in O)$ where O is any Borel-measurable subset of the Mathéron topology.

Similar considerations apply to observations. A *unified observation representation* is a finite set of the form $Y = \{y_1, \ldots, y_m\}$ where m is the number of observations and $y_1, \ldots, y_m$ are observation vectors generated by all sensors from all targets (in general, y is assumed to include a discrete sensor tag describing the originating sensor). When no observations have been collected, $Y = \emptyset$.

4.4.2 Belief-mass functions and multitarget integro-differential calculus

Let Ψ denote a random finite subset of some space Y (e.g., the space of target states or the space of measurements from any sensor). The statistical behavior of Ψ is described by its probability-mass function (a.k.a. probability measure) $\Pr(\Psi \in O)$. For engineering purposes it is inconvenient to deal with Borel sets O which are continuously infinite sets whose elements are finite sets. The *Choquet-Mathéron theorem* (pp. 96, [49]) states that the additive probability measure $p_\Psi(O) = \Pr(\Psi \in O)$ is equivalent to the *non-additive* measure (a.k.a. "capacity" or "Choquet functional")

$$\pi_\Psi(S) = \Pr(\Psi \cap S \neq \emptyset)$$

where S is a subset of ordinary single-target state space. Therefore, $p_\Psi(O)$ is also equivalent to

$$\beta_\Phi(S) = 1 - \pi_\Phi(S^c) = 1 - \Pr(\Phi \cap S^c \neq \emptyset) = \Pr(\Phi \subseteq S)$$

For engineering purposes $p_\Psi(O)$ is replaced by $\beta_\Phi(S)$. By analogy with $p_\Psi(O)$ $\beta_\Phi(S)$ is called the *belief-mass function* (a.k.a. *belief measure*) of the random finite set Ψ.

In single target problems the density function $f_Y(y)$ of $p_Y(S)$ is defined as follows:

$$p_Y(S) = \int_S f_Y(y)dy \tag{4.15}$$

in which case $f_Y(y)$ is called the *Radon-Nikodym derivative* of $p_Y(S)$.

In multitarget engineering a multitarget density function $f_\Phi(Y)$ of $\beta_\Phi(S)$ is defined by analogy with (4.15)

$$p_\Phi(S) = \int_S f_\Phi(Y)\delta Y \tag{4.16}$$

This equation does not make sense unless the indicated integral is defined firstly. Let $f(Y)$ be any real-valued function of a finite set variable Y which has the following property. For each $n \geq 0$, use the convention $f(\{y_1, \ldots, y_n\}) = 0$ whenever $y_i = y_j$ for some $i \neq j$, and also assume that $\int f(\{y_1, \ldots, y_n\})dy_1 \cdots dy_n$ is finite and has no units of measurement. Then the *set integral* of $f(Y)$ in a region S is defined as

$$\int_S f(Y)\delta Y = f(\emptyset) + \sum_{n=1}^{\infty} \frac{1}{n!} \int_{S^n} f(\{y_1, \ldots, y_n\})dy_1 \cdots dy_n \tag{4.17}$$

Given any belief-mass function $\beta_\Phi(S)$, how to construct its corresponding density function $f_\Phi(Y)$ so that (4.16) is satisfied? This requires the inverse operation of the set integral, the *set derivative*. For arbitrary functions $F(S)$ of a finite set

variable S and for $y_1, \ldots, y_n$ distinct, it is defined by

$$\frac{\delta F}{\delta y}(S) = \lim_{v(E_y)\to 0} \frac{F(S \cup E_y) - F(S)}{v(E_y)} \tag{4.18}$$

$$\frac{\delta \beta}{\delta Y}(S) = \frac{\delta^n \beta}{\delta y_n \cdots \delta y_1}(S) = \frac{\delta}{\delta y_n} \frac{\delta^{n-1} \beta}{\delta y_{n-1} \cdots \delta y_1}(S) \tag{4.19}$$

where E_y is a small neighborhood of y and $v(S)$ is the hypervolume of set S.

The set derivative is the continuous variable analog of the Möbius transform of Dempster-Shafer theory (pp. 149, [49]). It can be computed using "turn the crank" rules such as the following (pp. 31-32, [83]):

- *Sum Rule*:

$$\frac{\delta}{\delta Y}(\alpha_1 \beta_1(S) + \alpha_2 \beta_2(S)) = \alpha_1 \frac{\beta_1}{\delta Y}(S) + \alpha_2 \frac{\beta_2}{\delta Y}(S) \tag{4.20}$$

- *Product Rule*:

$$\frac{\delta}{\delta Y}(\beta_1(S)\beta_2(S)) = \sum_{W \subseteq Y} \frac{\delta \beta_1}{\delta W}(S) \frac{\delta \beta_2}{\delta (Y - W)}(S) \tag{4.21}$$

- *Chain Rule*:

$$\frac{\delta}{\delta y} f(\beta_1(S), \ldots, \beta_n(S)) = \sum_{i=1}^{n} \frac{\partial f}{\partial x_i}(\beta_1(S), \ldots, \beta_n(S)) \frac{\beta_i}{\delta y}(S) \tag{4.22}$$

- *Constant Rule*: If $Y \neq \emptyset$ and K is a constant, then

$$\frac{\delta}{\delta Y} K = 0 \tag{4.23}$$

- *Power Rule*: If $p(S)$ is a probability mass function with density function $f_p(y)$, then

$$\frac{\delta}{\delta Y}p(S)^n = \begin{cases} \frac{n!}{(n-k)!}p(S)^{n-k}f_p(y_1)\cdots f_p(y_k) & \text{if} \quad k \leq n \\ \\ 0 & \text{if} \quad k > n \end{cases} \tag{4.24}$$

Given these it can be shown

$$\beta_\Psi(S) = \int_S \frac{\delta\beta_\psi}{\delta Y}(\emptyset)\delta Y \tag{4.25}$$

that is,

$$f_\Psi(Y) = \frac{\delta\beta_\psi}{\delta Y}(\emptyset) \tag{4.26}$$

the multiobject density function of $\beta_\Psi(S)$.

4.4.3 Multisensor multitarget Bayesian modelling

Belief-mass functions and their set derivatives provide the means for generalizing formal Bayesian modelling to multisensor multitarget problems. Under FISST, the motion model of multiple targets can be modelled as:

$$\Xi_{t+1|t} = D_t(X)\bigcup B_t(X) \tag{4.27}$$

where $D_t(X)$ models presumed target motion and the persistence/disappearance of existing targets while $B_t(X)$ models the appearance of new targets. The measurement model of sensors can be modelled as

$$\Sigma_t = E_t(X)\bigcup C_t(X) \tag{4.28}$$

where $E_t(X)$ models the self-noise of sensors and their detection probabilities while $C_t(X)$ models false alarms and clutter. Then their corresponding belief-mass functions are constructed as follows:

$$\beta_{t+1|t}(T|X) = \Pr(\Xi_{t+1|t} \subseteq T|X) \tag{4.29}$$

$$\beta_t(S|X) = \Pr(\Sigma_t \subseteq S|X) \tag{4.30}$$

Finally, from (4.26) we can explicitly construct general, implementation-independent formulas for the true multitarget likelihood function and the true multitarget Markov density as follows:

$$f_{t+1|t}(X|W) = \frac{\delta \beta_{t+1|t}}{\delta X}(\emptyset|W) \tag{4.31}$$

$$f_t(Y|X) = \frac{\delta \beta_t}{\delta Y}(\emptyset|X) \tag{4.32}$$

These multitarget density functions contain the same information as their respective belief-mass functions; and therefore the same information as the models used to construct those belief-mass functions.

4.4.4 Unified fusion of multisource-multitarget information

With these preliminaries in place the single-sensor, single-target Bayesian filter of (2.4)-(2.6) may be generalized to multisource multitarget problems. They become,

respectively,

$$f_{t+1|t}(X|Y^t) = \int f_{t+1|t}(X|W) f_{t|t}(W|Y^t) \delta W \tag{4.33}$$

$$f_{t+1|t+1}(X|Y^{t+1}) \propto f_{t+1}(Y_{t+1}|X) f_{t+1|t}(X|Y^t) \tag{4.34}$$

$$f_{t+1}(Y_{t+1}|Y^t) = \int f_{t+1}(Y_{t+1}|X) f_{t+1|t}(X|Y^t) \delta X \tag{4.35}$$

Here $f_{t|t}(X|Y^t)$ is the multitarget posterior distribution; $Y^t = \{Y_1, \ldots, Y_t\}$ is the time sequence of multisource measurement sets; and the integrals are set integrals. The multitarget posterior distribution has the form

$f_{t|t}(\emptyset|Y^t)$: no targets present

$f_{t|t}(\{x_1\}|Y^t)$: one target with state x_1

$f_{t|t}(\{x_1, x_2\}|Y^t)$: two targets with states x_1, x_2

.....

$f_{t|t}(\{x_1, \ldots, x_n\}|Y^t)$: n targets with states $x_1, \ldots, x_n$

.....

4.4.5 Probability generating functionals and functional derivatives

It is easy to extend the concept of a belief-mass function to fuzzy sets by rewriting (4.16) as

$$\begin{aligned}\beta_\Phi(S) &= \int_S f_\Phi(Y)\delta Y = \int 1_S^Y f_\Phi(Y)\delta Y \\ &= \sum_{n=0}^{\infty} \frac{1}{n!} \int 1_S(y_1)\cdots 1_S(y_n) f_\Phi(\{y_1,\ldots,y_n\}) dy_1 \ldots dy_n\end{aligned} \tag{4.36}$$

where $1_S(y)$ is defined by $1_S(y) = 1$ if $y \in S$ and $1_S(y) = 0$ otherwise; and where

$$1_S^Y = \prod_{y \in Y} 1_S(y)$$

Now, let $\mu(y)$ be the membership function for a fuzzy set. Then (4.36) is generalized as

$$\begin{aligned}G_\Phi[\mu] &= \int \mu^Y f_\Phi(Y)\delta Y \\ &= \sum_{n=0}^{\infty} \frac{1}{n!} \int \mu(y_1)\cdots\mu(y_n) f_\Phi(\{y_1,\ldots,y_n\}) dy_1 \ldots dy_n\end{aligned} \tag{4.37}$$

where

$$\mu^Y = \prod_{y \in Y} \mu(y) \tag{4.38}$$

In the point process literature $G_\Phi[\mu]$ is known as the *probability generating functional* (p.g.fl.) of Φ (pp. 141, 220, [34]). Note that $G_\Phi[1_S] = \beta_\Phi(S)$, so that p.g.fl.'s do indeed generalize belief-mass functions.

The p.g.fl. $G_\Phi[\mu]$ is, like the multitarget density $f_\Phi(X)$ and the belief-mass function

$\beta_\Phi(S)$, a fundamental descriptor of the statistics of Φ. But it is often more useful than $f_\Phi(X)$ or $\beta_\Phi(S)$ because it results in much simpler formulas.

The set derivative of a belief-mass function can be generalized to functional derivatives of p.g.fl.'s. Recall that the *gradient derivative* (a.k.a. directional or Frechét derivative) of a real-valued function $G(x)$ in the direction of a vector w is

$$\frac{\partial G}{\partial w}(x) = \lim_{\varepsilon \to 0} \frac{G(x + \varepsilon \cdot w) - G(x)}{\varepsilon} \tag{4.39}$$

where for each x the function $x \to \frac{\partial G}{\partial w}(x)$ is linear and continuous; and so

$$\frac{\partial G}{\partial w}(x) = w_1 \frac{\partial G}{\partial w_1}(x) + \ldots + w_n \frac{\partial G}{\partial w_n}(x)$$

for all $w = (w_1, \ldots, w_N)$, where the derivatives on the right are ordinary partial derivatives. Likewise, the gradient derivative of a p.g.fl. $G[h]$ in the direction of the function g is

$$\frac{\partial G}{\partial g}(x) = \lim_{\varepsilon \to 0} \frac{G[h + \varepsilon \cdot g] - G[h]}{\varepsilon} \tag{4.40}$$

where for each h the functional $g \to \frac{\partial G}{\partial g}(h)$ is linear and continuous. In physics, gradient derivatives with $g = \delta_x$ are called "functional derivatives" (pp. 140-141, [110]). Using the simplified version of this physics notation employed in FISST, the *functional derivatives* of a p.g.fl. $G[h]$ is defined as:

$$\frac{\delta^0 G}{\delta x^0}[h] = G[h], \quad \frac{\delta G}{\delta x}[h] = \frac{\partial G}{\partial \delta_x}[h] \tag{4.41}$$

$$\frac{\delta^n G}{\delta x_1 \cdots \delta x_n}[h] = \frac{\partial^n G}{\partial \delta_{x_1} \cdots \partial \delta_{x_n}}[h] \tag{4.42}$$

It can be shown (p. 1162 of [85]) that the set derivative of $\beta_{\Xi}(S)$ is a functional derivative of $G_{\Xi}[\mu]$

$$\frac{\delta\beta_{\Xi}}{\delta x}(S) = \frac{\partial G_{\Xi}}{\partial \delta_x}[1_S] \tag{4.43}$$

with $g = \delta_x$ and $h = 1_S$. Likewise for the iterated derivatives:

$$\frac{\delta\beta_{\Xi}}{\delta X}(S) = \frac{\delta^n\beta_{\Xi}}{\delta x_1 \cdots \delta x_n}(S) = \frac{\partial^n G_{\Xi}}{\partial \delta_{x_1} \cdots \partial \delta_{x_n}}[1_S] \tag{4.44}$$

for $X = \{x_1, \cdots, x_n\}$ with $x_1, \cdots, x_n$ are distinct. So for $X = \{x_1, \cdots, x_n\}$, the multitarget probability distribution of a random state set Ξ is:

$$f_{\xi}(X) = \frac{\delta^n\beta_{\Xi}}{\delta x_1 \cdots \delta x_n}(\emptyset) = \frac{\partial^n G_{\Xi}}{\partial \delta_{x_1} \cdots \partial \delta_{x_n}}[0] \tag{4.45}$$

4.5 Probability hypothesis density

Mahler devised the *Probability Hypothesis Density* filter as an approximation of multitarget filter in FISST [85]. The 1st moment of a RFS is the analogue of the expectation of a random vector. In the point process literatures [34, 116], a finite subset X can also be equivalently represented by the counting measure N_X defined by $N_X = \sum_{x \in X} 1_S(x) = |X \bigcap S|$, where S is a measurement subset, $1_S(x)$ is the indicator function of S defined by $1_S(x) = 1$ if $x \in S$ and $1_S(x) = 0$ otherwise, and the notation $|A|$ denotes the number of elements in A. Consequently, the random finite set Ξ can also be represented by a random counting measure N_{Ξ} defined by $N_{\Xi} = |\Xi \bigcap S|$.

Using the random counting measure representation, the 1st moment or *intensity measure* $V_{\Xi}(S)$ of a RFS Ξ is defined as follows:

$$V_{\Xi}(S) \equiv E[N_{\Xi}(S)] = \int (\sum_{x \in X} 1_S(x)) P_{\Xi}(dX) \tag{4.46}$$

for each measurable set S. The intensity measure $V_{\Xi}(x)$ over a region S gives the expected number of elements of Ξ that are in S.

The density of the intensity measure V_{Ξ} w.r.t. the Lebegue measure:

$$D_{\Xi} = \frac{dV_{\Xi}}{d\lambda} \tag{4.47}$$

is called the *intensity function* or *Probability Hypothesis Density* (PHD) [85]. The integral of the PHD D_{Ξ} over a region S in a state space $\int_S D_{\Xi}(x)\lambda(dx) = E|\Xi \bigcap S|$ is the expected number of targets within this region. Consequently, the peaks of PHD D_{Ξ} are points the highest local concentration of expected number of targets and can be used to generate estimates for the states of targets Ξ.

The generalized FISST calculus provides the foundation for a systematic procedure for devising computational approximation strategies. This procedure has been used, for example, to derive the predictor and corrector equations for the PHD filter in [85]. Generally speaking this procedure consists of the following steps:

1. Rewrite the multitarget predictor integral, (4.33), in p.g.fl. form:

$$G_{t+1|t}[h] = \int G_{t+1|t}[h|X] f_{t|t}(X|Y^t) \delta X \tag{4.48}$$

where

$$G_{t+1|t}[h|X] = \int h^Y \cdot G_{t+1|t}(Y|X)\delta Y \tag{4.49}$$

where h^Y is as defined in (4.38).

2. Given a multitarget Markov density based on a specific multitarget motion model as in (4.27), derive a formula of the form $G_{t+1|t}[h] = Gt|t[\Phi[h]]$ for some functional transformation $h \to \Phi[h]$. This formula can then be used to derive approximate prediction equations, e.g., for the predicted first-order multitarget moment

$$D_{t+1|t}(x|Y^t) = \frac{\delta G_{t+1|t}}{\delta x}[1] \tag{4.50}$$

3. Rewrite the numerator of multitarget Bayes' rule, (4.34), as a p.g.fl.:

$$F_{t+1}[g,h] = \int h^X \cdot G_{t+1}[g|X] f_{t+1|t}(X|Y^t)\delta X \tag{4.51}$$

where

$$G_{t+1}[g|X] = \int g^Y \cdot f_{t+1}(Y|X)\delta Y \tag{4.52}$$

4. Rewrite the multitarget Bayes rule, (4.34), in terms of p.g.fl.s and their functional derivatives:

$$G_{t+1|t}[h] = \frac{\frac{\delta F_{t+1}}{\delta Y_{m+1}}[0,h]}{\frac{\delta F_{t+1}}{\delta Y_{m+1}}[0,1]} \tag{4.53}$$

5. Assume that the predicted p.g.fl. $G_{t+1|t}[h]$ has a suitably simplified form such as

$$G_{t+1|t}[h] = \exp(-\lambda + \lambda \int h(x)s(x)dx) \tag{4.54}$$

(the Poisson approximation) or

$$G_{t+1|t}[h] = \prod_{j=1}^{n}(1 - q_j + q_j \int h(x) f_j(x) dx)) \tag{4.55}$$

(the multi-hypothesis correlator approximation).

6. Using a multitarget likelihood function constructed from a specific measurement model as in (4.28), derive the updated first-order moment (the PHD):

$$D_{t+1|t+1}(x|Y^{t+1}) = \frac{\delta G_{t+1|t+1}}{\delta x}[1] \tag{4.56}$$

7. Suppose that some objective function for use in sensor management is given, such as the posterior expected number of targets

$$N_{t+1|t+1} = \frac{\partial}{\partial y} G_{t+1|t+1}[e^y] = \int |X| \cdot f_{t+1|t+1}(X|Y^{t+1}) \delta X \tag{4.57}$$

Use the approximations of Step 5 to derive approximate formulas for the objective function.

Let $D_{t|t}$ denote the probability hypothesis density associated with the multi-target posterior $p_{t|t}(X|Y^t)$ at time t. The PHD filter consists of two steps: prediction and update. The PHD prediction equation is:

$$D_{t+1|t}(x) = b_{t+1|t}(x) + \int (p_S(w) f_{t+1|t}(x|w) + b_{t+1|t}(x|w)) D_{t|t}(w) dw \tag{4.58}$$

where $b_{t+1|t}(x)$ denotes the intensity function of the spontaneous birth RFS, $b_{t+1|t}(x|w)$ denotes the intensity function of the RFS of targets spawned from the previous

state w, $p_S(w)$ is the probability that the target still exits at time $t+1$ given it has previous state w, and $f_{t+1|t}(x|w)$ is the transition probability density of individual targets. The PHD update equation is:

$$D_{t+1|t+1}(x) \cong F_{t+1}(Y_{t+1}|x)D_{t+1|t}(x) \tag{4.59}$$

$$F_{t+1}(Y|x) = 1 - p_D(x) + \sum_{y \in Y_{t+1}} \frac{p_D(x)p_{t+1}(y|x)}{\lambda c(y) + D_{t+1|t}[p_D(x)p_{t+1}(y|x)]} \tag{4.60}$$

where $p_D(x)$ is the probability of detection, $p_{t+1}(y|x)$ is the likelihood of individual target, λ is the average number of clutter points per scan, $c(y)$ is the probability distribution of each clutter point, and $D_{t+1|t}[h] = \int h(x_{t+1})D(x_{t+1}|Y^t)dx_{t+1}$.

4.6 Particle PHD filter

In this section we introduce the basic particle PHD filter implemented using the sequential Monte Carlo method. We assume that there are no spawned targets in the prediction stage and all targets at time $t+1$ consist of two classes: survival targets and spontaneous birth targets.

Let L_t denote the particle number at time t, J_t denote the new particle number for the spontaneous birth targets at time t, and w denote a particle's weight. The basic particle PHD filter is as follows:

At time $t \geq 0$, let $\{x_t^{(i)}, w_t^{(i)}\}_{i=1}^{L_t}$ denote a particle approximation of the PHD.

1. *Detection*

   Detecting the foreground objects using background subtraction. The centriods of all foreground blobs are the measurement set Y_{t+1} at time $t+1$.

2. *Prediction*

   - For the survival targets, the importance function is the dynamic model (4.10). Therefore, for $i = 1, ..., L_t$, generate a sample $\tilde{x}_{t+1}^{(i)}$ using (4.10) and compute the predicted weights

   $$\hat{w}_{t+1}^{(i)} = w_t^{(i)}$$

   - For the spontaneous birth targets, we propose a uniform distribution on the whole image region as the importance function because we assume that we have no prior knowledge about new-birth objects:

   $$b(x_{t+1}) \sim U[1, width] \times U[1, height] \tag{4.61}$$

   where $width$ and $height$ are the size of the image and $U[c, d]$ is a uniform distribution function on the interval $[c, d]$. Therefor, for $i = L_t+1, ..., L_t+J_t$, sample $\tilde{x}_{t+1}^{(i)}$ using (4.61) and compute the predicted weights

   $$\hat{w}_{t+1}^{(i)} = 1/J_{t+1}$$

3. *Update*

   - For each target, the centroid of its foreground blob is used as the measurement to update the PHD filter. We propose the likelihood function as follows:

$$p(y|x_{t+1}) = \frac{1}{2\pi|\Sigma_v|^{1/2}} \exp[-\frac{1}{2}(y - x_{t+1})^T \Sigma_v^{-1}(y - x_{t+1})] \tag{4.62}$$

     where Σ_v is the covariance matrix of the measurement noise.

   - For each $y \in Y_{t+1}$, use the likelihood and compute

$$C_{t+1}(y) = \sum_{i=1}^{L_t+J_{t+1}} p_D(\tilde{x}_{t+1}^{(i)}) p(y|\tilde{x}_{t+1}^{(i)}) \hat{w}_{t+1}^{(i)}$$

   - For $i = 1, ..., L_t + J_{t+1}$, update weights

$$\tilde{w}_{t+1}^{(i)} = [1 - p_D(\tilde{x}_{t+1}^{(i)}) + \sum_{y \in Y_{t+1}} \frac{p_D(\tilde{x}_{t+1}^{(i)}) p(y|\tilde{x}_{t+1}^{(i)})}{\lambda c(y) + C_{t+1}(y)}] \hat{w}_{t+1}^{(i)}$$

4. *Resampling*

   - Compute the target number at time $t + 1$

$$\hat{N}_{t+1} = \sum_{i+1}^{L_t+J_{t+1}} \tilde{w}_{t+1}^{(i)}$$

   - Initialize the cumulative probability $c_1 = 0$,

     $c_i = c_{i-1} + \tilde{w}_{t+1}^{(i)} / \hat{N}_{t+1}$, $i = 2, ..., L_t + J_{t+1}$.

   - Draw a starting point $u_1 \sim U[0, L_{t+1}^{-1}]$.

- For $j = 1, ..., L_{t+1}$,

$$u_j = u_1 + L_{t+1}^{-1}(j-1)$$

While $u_j > c_i$, $i = i + 1$. End while.

$$x_{t+1}^{(j)} = \tilde{x}_{t+1}^{(i)}$$

$$w_{t+1}^{(j)} = L_{t+1}^{-1}$$

- Rescale (multiply) the weights by $\hat{N}_{t+1}$ to get

$$\{x_{t+1}^{(i)}, \hat{N}_{t+1}/L_{t+1}\}_{i=1}^{L_{t+1}}$$

5. *State extraction*

   Do k-means clustering for particles $\{x_{t+1}^{(i)}\}_{i=1}^{L_{t+1}}$ with the cluster number $k = \text{round}(\hat{N}_{t+1})$ and round(N) is the integer nearest to N. The means of clusters are used as the state estimation of targets.

4.7 Data-driven particle PHD filter

We proposed a data-driven method for the particle PHD filter in this section. The "data-driven" means that the current measurement set is used to design the importance function of the particle PHD filter. The design of importance function is a key issue for particle PHD filter. Most of previous works on importance function

only care about the fixed number of targets, whereas the PHD filter is to deal with the variable number of targets. Moreover, the current measurements for the PHD filter are not a single measurement but a random measurement set. To meet this challenge, we have modelled the targets into two categories: survival objects and spontaneous birth objects. For survival objects, the importance function is an theoretical extension of the optimal importance function of the linear Gaussian model. Whereas for spontaneous birth objects, the importance function is a Gaussian mixture with means being the centroids of new detected foreground blobs.

The sequential importance sampling is described in section 4.7.1. The optimal importance function of the linear Gaussian model is introduced in section 4.7.2. The importance function for survival targets is proposed in section 4.7.3. The importance function for spontaneous birth targets is presented in section 4.7.4. The data-driven PHD filter is summarized in section 4.7.5.

4.7.1 Sequential importance sampling

Let $q(x_{0:t+1}|y_{1:t+1})$ be the importance function of particle filter and it can be factored into

$$q(x_{0:t+1}|y_{1:t+1}) = q(x_0)\prod_{k=1}^{t} q(x_{k+1}|x_{0:k}, y_{1:k+1}) \tag{4.63}$$

then sequential importance sampling filter is

For times $t = 0, 1, 2, \cdots$

- For $i = 1, \cdots, N$, sample $x_{t+1}^{(i)}$ $q(x_{t+1}|x_{0:t}, y_{1:t+1})$ and set $x_{0:t+1}^{(i)} = (x_{0:t}^{(i)}, x_{t+1}^{(i)})$.
- For $i = 1, \cdots, N$, evaluate the importance weights up to a normalizing constant:

$$w_{t+1}^{(i)} = w_t^{(i)} \frac{p(y_{t+1}|x_{t+1}^{(i)}) p(x_{t+1}^{(i)}|x_t^{(i)})}{q(x_{k+1}|x_{0:k}, y_{1:k+1})} \tag{4.64}$$

- For $i = 1, \cdots, N$, normalize the importance weights:

$$\tilde{w}_{t+1}^{(i)} = \frac{w_{t+1}^{(i)}}{\sum_{j=1}^{N} w_{t+1}^{(j)}} \tag{4.65}$$

4.7.2 Optimal importance function

Lemma 4.1. *The optimal importance sampling function $q(x_{t+1}|x_{0:t}^{(i)}, y_{1:t+1})$ which minimises the variance of the importance weight $w_{t+1}^{(i)}$ is $p(x_{t+1}|x_t^{(i)}, y_{t+1})$ conditional upon $x_{0:t}^{(i)}$ and $y_{1:t+1}$. The correspondent weight function is $w_{t+1}^{(i)} = w_t^{(i)} p(y_{t+1}|x_t^{(i)})$.*

Proof. Straightforward calculations using yield

$$\begin{aligned}
&var_{q(x_{t+1}|x_{0:t}^{(i)}, y_{1:t+1})}(w_{t+1}^{(i)}) \\
&= E_{q(x_{t+1}|x_{0:t}^{(i)}, y_{1:t+1})}[(w_{t+1}^{(i)})^2] - [E_{q(x_{t+1}|x_{0:t}^{(i)}, y_{1:t+1})}(w_{t+1}^{(i)})]^2 \\
&= \int (w_{t+1}^{(i)})^2 q(x_{t+1}|x_{0:t}^{(i)}, y_{1:t+1}) dx_{t+1} - [\int w_{t+1}^{(i)} q(x_{t+1}|x_{0:t}^{(i)}, y_{1:t+1}) dx_{t+1}]^2 \\
&= (w_t^{(i)})^2 [\int \frac{[p(y_{t+1}|x_{t+1}^{(i)}) p(x_{t+1}^{(i)}|x_t^{(i)})]^2}{q(x_{t+1}|x_{0:t}^{(i)}, y_{1:t+1})} dx_{t+1} - p^2(y_{t+1}|x_t^{(i)})]
\end{aligned} \tag{4.66}$$

When $q(x_{t+1}|x_{0:t}^{(i)}, y_{1:t+1}) = p(x_{t+1}|x_t^{(i)}, y_{t+1})$, the above variance is zero because

$$\begin{aligned} p(x_{t+1}|x_t^{(i)}, y_{t+1}) &= \frac{p(y_{t+1}|x_{t+1}, x_t^{(i)})p(x_{t+1}|x_t^{(i)})}{p(y_{t+1}|x_t^{(i)})} \\ &= \frac{p(y_{t+1}|x_{t+1})p(x_{t+1}|x_t^{(i)})}{p(y_{t+1}|x_t^{(i)})} \end{aligned} \tag{4.67}$$

and the weight (4.64) is $w_{t+1}^{(i)} = w_t^{(i)} p(y_{t+1}|x_t^{(i)})$ □

Lemma 4.2. *For the linear Gaussian model (4.10) and (4.13), the conditional distributions* $p(x_{t+1}|x_t^{(i)}, y_{t+1})$ *and* $p(y_{t+1}|x_t^{(i)})$ *are*

$$p(x_{t+1}|x_t^{(i)}, y_{t+1}) \sim N(m_{t+1}, \Sigma) \tag{4.68}$$

$$\Sigma^{-1} = \Sigma_u^{-1} + H^T \Sigma_v^{-1} H \tag{4.69}$$

$$m_{t+1} = \Sigma(\Sigma_u^{-1} F x_t^{(i)} + H^T \Sigma_v^{-1} y_{t+1}) \tag{4.70}$$

$$p(y_{t+1}|x_t^{(i)}) \sim N(HFx_t^{(i)}, \Sigma_v + H\Sigma_u H^T) \tag{4.71}$$

Proof. From (4.10), it can be obtained that

$$x_{t+1} \sim N(Fx_t^{(i)}, \Sigma_u) \tag{4.72}$$

and from (4.13), it can be obtained that

$$x_{t+1} \sim N(H^{-1}y, (H^{-1})^T \Sigma_v H^{-1}) \tag{4.73}$$

where H^{-1} is the pseudo inverse of H. To combine (4.72) and (4.73), the linear estimator of x_{t+1} is

$$x_{t+1} = (1-a)(Fx_t^{(i)} + u_t) + a(H^{-1}y - H^{-1}v_{t+1}) \tag{4.74}$$

where a is the hybrid parameter. Thus,

$$E(x_{t+1}) = (1-a)Fx_t^{(i)} + aH^{-1}y \tag{4.75}$$

$$\text{var}(x_{t+1}) = E(x_{t+1}^2) - E^2(x_{t+1}) = (1-a)^2\Sigma_u + a^2H^{-1}\Sigma_v(H^{-1})^T \tag{4.76}$$

$$\frac{\partial \text{var}(x_{t+1})}{\partial a} = 2(a-1)\Sigma_u + 2aH^{-1}\Sigma_v(H^{-1})^T = 0 \tag{4.77}$$

then we get

$$a = \frac{\Sigma_u}{\Sigma_u + H^{-1}\Sigma_v(H^{-1})^T} \tag{4.78}$$

Using (4.78) in (4.76) we obtain

$$\Sigma = \text{var}(x_{t+1}) = \frac{\Sigma_u H^{-1}\Sigma_v(H^{-1})^T}{\Sigma_u + H^{-1}\Sigma_v(H^{-1})^T} \tag{4.79}$$

so (4.69) becomes

$$\Sigma^{-1} = \Sigma_u^{-1} + [H^{-1}\Sigma_v(H^{-1})^T]^{-1} = \Sigma_u^{-1} + H^T\Sigma_v^{-1}H \tag{4.80}$$

Using (4.78) in (4.75), we obtain

$$m_{t+1} = E(x_{t+1}) = \frac{H^{-1}\Sigma_v(H^{-1})^T}{\Sigma_u + H^{-1}\Sigma_v(H^{-1})^T}Fx_t^{(i)} + \frac{\Sigma_u}{\Sigma_u + H^{-1}\Sigma_v(H^{-1})^T}H^{-1}y \tag{4.81}$$

Then (4.70) and (4.68) are obtained.

Using (4.10) in (4.13), we get

$$y_{t+1} = HFx_t^{(i)} + Hu_t + v_{t+1} \tag{4.82}$$

then,

$$E(y_{t+1}) = HFx_t^{(i)} \tag{4.83}$$

$$\mathrm{var}(y_{t+1}) = E(y_{t+1}^T y_{t+1}) - E^2(y_{t+1}) = \Sigma_v + H\Sigma_u H^T \tag{4.84}$$

Using (4.83) and (4.84), we obtain (4.71). □

4.7.3 Importance function for survival targets

For our tracking task, the measurement at time $t+1$ is not a single measurement y_{t+1} but a measurement set Y_{t+1}. The goal of this subsection is to derive the analytical expressions for importance function and weight function of particle filter for measurement sets.

Several measurements may be available at each time. Each measurement may be generated by survival targets or spontaneous birth targets. Taking into account measurements of spontaneous birth targets in the update of survival targets may dramatically decrease the quality of the estimate of survival targets. To solve the problem of distinguishing measurements of survival targets from spontaneous birth targets, the validation gating technology (pp. 166, [12]) is introduced to filter the measurements and obtain a validation measurement set of each particle for survival targets near its predicted position as follows:

$$\widetilde{Y}_{t+1}^{(i)} = \{y_{t+1,k} : (y_{t+1,k} - HFx_t^{(i)})^T \Sigma_v^{-1} (y_{t+1,k} - HFx_t^{(i)}) \leq U\} \tag{4.85}$$

where U is the gating threshold, $HFx_t^{(i)}$ is the predicted measurement for the particle $x_t^{(i)}$, and $y_{t+1,k}$ is the kth measurement of the set Y_{t+1}. The measurement

set of survival targets is defined as the union of all survival measurement sets:

$$\widetilde{Y}_{t+1} = \bigcup_{i=1}^{N} \widetilde{Y}_{t+1}^{(i)} \tag{4.86}$$

and the residual measurement set is defined as:

$$\overline{Y}_{t+1} = Y_{t+1} - \widetilde{Y}_{t+1} \tag{4.87}$$

We give an example to illustrate the gating technology in Fig. 4.2.

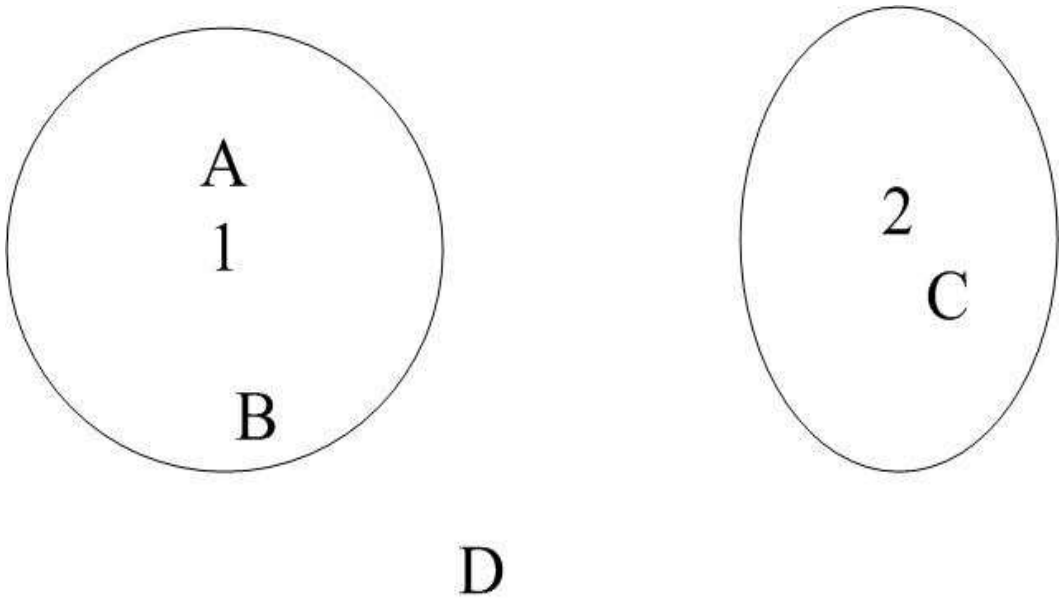


Figure 4.2: The two circles are the predicted gate regions of particle 1 and 2. A, B, C, and D are four measurements. A and B are in the gate region of the 1st particle, i.e., $\widetilde{Y}_{t+1}^{(1)} = \{A, B\}$. C is in the gate region of the 2nd particle, i.e., $\widetilde{Y}_{t+1}^{(2)} = \{C\}$. D is the residual measurement, i.e., $\overline{Y}_{t+1} = \{D\}$.

Let $y_{t+1,j}$ be the measurement which is nearest to the predicted measurement of

the particle $x_t^{(i)}$ of spontaneous birth targets in the residual measurement set, i.e.,

$$j = \underset{k}{\operatorname{argmin}}\{|y_{t+1,k} - Hx_{t+1}^{(i)}|\}, \quad y_{t+1,k} \in \overline{Y}_{t+1} \tag{4.88}$$

We make an assumption for survival targets in our tracking scenario:

Assumption 4.1. For survival targets, the measurements of each target must be within its validation measurement set (4.2). Thus, the conditional distribution $p(x_{t+1}|x_t^{(i)}, Y_{t+1})$ can be approximated by another conditional distribution $p(x_{t+1}|x_t^{(i)}, \widetilde{Y}_{t+1}^{(i)})$, i.e.,

$$p(x_{t+1}|x_t^{(i)}, Y_{t+1}) \cong p(x_{t+1}|x_t^{(i)}, \widetilde{Y}_{t+1}^{(i)}) \tag{4.89}$$

The likelihood function can be approximated as follows:

$$p(Y_{t+1}|x_t^{(i)}) \cong p(\widetilde{Y}_{t+1}^{(i)}|x_t^{(i)}) \tag{4.90}$$

and given the state, the measurements are conditionally independent from each other:

$$p(\widetilde{Y}_{t+1}^{(i)}|x_t^{(i)}) = \prod_{y \in \widetilde{Y}_{t+1}^{(i)}} p(y|x_t^{(i)}) \tag{4.91}$$

From the above assumption, we propose importance functions and weight functions of survival targets.

Proposition 4.3. *The optimal importance function for each survival target is:*

$$p(x_{t+1}|x_t^{(i)}, Y_{t+1}) \sim N(m_{t+1}^{(i)}, \Sigma^{(i)}) \tag{4.92}$$

$$(\Sigma^{(i)})^{-1} = \Sigma_u^{-1} + |\widetilde{Y}_{t+1}^{(i)}| H^T \Sigma_v^{-1} H \tag{4.93}$$

$$m_{t+1}^{(i)} = \Sigma^{(i)}(\Sigma_u^{-1} F x_t^{(i)} + H^T \Sigma_v^{-1} \sum_{y \in \widetilde{Y}_{t+1}^{(i)}} y) \tag{4.94}$$

and the weight function is:

$$w_{t+1}^{(i)} \propto w_t^{(i)} \exp\{-\frac{1}{2} \sum_{y \in \widetilde{Y}_{t+1}^{(i)}} [(y - HFx_t^{(i)})^T (\Sigma_v + H\Sigma_u H^T)^{-1} (y - HFx_t^{(i)})]\} \tag{4.95}$$

I present two proofs for this proposition.

Proof. Let $M_i = |\widetilde{Y}_{t+1}^{(i)}|$ and $\widetilde{Y}_{t+1}^{(i)} = \{y_1^{(i)}, \cdots, y_{M_i}^{(i)}\}$. From Lemma 4.1, we can obtain that the optimal importance function is

$$p(x_{t+1} | x_t^{(i)}, Y_{t+1}) \tag{4.96}$$

Using (4.89), we obtain

$$p(x_{t+1} | x_t^{(i)}, Y_{t+1}) \cong p(x_{t+1} | x_t^{(i)}, \widetilde{Y}_{t+1}^{(i)})$$

From (4.10) and (4.13), we get

$$x_{t+1} \sim N(Fx_t^{(i)}, \Sigma_u) \tag{4.97}$$

$$x_{t+1} \sim N(H^{-1} y_1, (H^{-1})^T \Sigma_v H^{-1}) \tag{4.98}$$

$$x_{t+1} \sim N(H^{-1} y_2, (H^{-1})^T \Sigma_v H^{-1}) \tag{4.99}$$

$$\cdots\cdots$$

$$x_{t+1} \sim N(H^{-1} y_{M_i}, (H^{-1})^T \Sigma_v H^{-1}) \tag{4.100}$$

As for (4.97) and (4.98), we use Lemma 4.2 (4.68)-(4.70) and can obtain

$$p(x_{t+1}|x_t^{(i)}, y_1^{(i)}) \sim N(m_{t+1,1}, \Sigma_1) \tag{4.101}$$

$$\Sigma_1^{-1} = \Sigma_u^{-1} + H^T \Sigma_v^{-1} H \tag{4.102}$$

$$m_{t+1,1} = \Sigma_1(\Sigma_u^{-1} F x_t^{(i)} + H^T \Sigma_v^{-1} y_1^{(i)}) \tag{4.103}$$

For (4.101) and (4.99), we use Lemma 2 (4.68)-(4.70) again and obtain

$$p(x_{t+1}|x_t^{(i)}, y_1^{(i)}, y_2^{(i)}) \sim N(m_{t+1,2}, \Sigma_2) \tag{4.104}$$

$$\Sigma_2^{-1} = \Sigma_1^{-1} + H^T \Sigma_v^{-1} H = \Sigma_u^{-1} + 2H^T \Sigma_v^{-1} H \tag{4.105}$$

$$\begin{aligned} m_{t+1,2} &= \Sigma_2(\Sigma_1^{-1} m_{t+1,1} + H^T \Sigma_v^{-1} y_2^{(i)}) \\ &= \Sigma_2[\Sigma_1^{-1}\Sigma_1(\Sigma_u^{-1} F x_t^{(i)} + H^T \Sigma_v^{-1} y_1^{(i)}) + H^T \Sigma_v^{-1} y_2^{(i)}] \\ &= \Sigma_2[\Sigma_u^{-1} F x_t^{(i)} + H^T \Sigma_v^{-1}(y_1^{(i)} + y_2^{(i)})] \end{aligned} \tag{4.106}$$

Repeat this process from $y_1^{(i)}$ to $y_{M_i}^{(i)}$, we obtain

$$(\Sigma^{(i)})^{-1} = \Sigma_{M_i}^{-1} = \Sigma_{M_i-1}^{-1} + H^T \Sigma_v^{-1} H = \Sigma_u^{-1} + M_i H^T \Sigma_v^{-1} H \tag{4.107}$$

$$\begin{aligned} m_{t+1}^{(i)} = m_{t+1,M_i} &= \Sigma^{(i)}(\Sigma_{M_i-1}^{-1} m_{t+1,M_i-1} + H^T \Sigma_v^{-1} y_{M_i}^{(i)}) \\ &= \Sigma^{(i)}(\Sigma_u^{-1} F x_t^{(i)} + H^T \Sigma_v^{-1} \sum_{j=1}^{M_i} y_j^{(i)}) \end{aligned} \tag{4.108}$$

i.e., (4.92)-(4.94).

From Lemma 4.1, the weight function is

$$w_{t+1}^{(i)} = w_t^{(i)} p(Y_{t+1}|x_t^{(i)}) \tag{4.109}$$

using Assumption 4.1 (4.90) and (4.91), we obtain

$$w_{t+1}^{(i)} = w_t^{(i)} \prod_{y \in \widetilde{Y}_{t+1}^{(i)}} p(y|x_t^{(i)}) \tag{4.110}$$

using Lemma 4.2 (4.71) in (4.110), we obtain (4.95). □

The following is the second proof method.

Proof. From (4.97) to (4.100), the linear estimator of x_{t+1} is

$$x_{t+1} = b_0(Fx_t^{(i)} + u_t) + \sum_{j=1}^{M_i} b_j H^{-1}(y_j^{(i)} - v_{t+1,j}) \tag{4.111}$$

where $\{b_j\}, j = 0, \cdots, M_i$ are the hybrid parameters and $\sum_{j=0}^{M_i} b_j = 1$. We assume that all measurements are the same contribution for the linear estimator (4.111), thus, $b_1 = b_2 = \cdots = b_{M_i} = b$ and $b_0 = 1 - M_i b$, then (4.111) becomes

$$x_{t+1} = (1 - M_i b)(Fx_t^{(i)} + u_t) + bH^{-1} \sum_{j=1}^{M_i} (y_j^{(i)} - v_{t+1,j}) \tag{4.112}$$

The mean of the linear estimator (4.111) is:

$$E(x_{t+1}) = (1 - M_i b)Fx_t^{(i)} + bH^{-1} \sum_{j=1}^{M_i} y_j^{(i)} \tag{4.113}$$

and the variance of the linear estimator (4.111) is

$$\text{var}(x_{t+1}) = E(x_{t+1}^2) - E^2(x_{t+1}) = (1 - M_i b)^2 \Sigma_u + M_i b^2 H^{-1} \Sigma_v (H^{-1})^T \tag{4.114}$$

$$\frac{\partial \text{var}(x_{t+1})}{\partial b} = 2M_i(M_i b - 1)\Sigma_u + 2M_i b H^{-1} \Sigma_v (H^{-1})^T = 0 \tag{4.115}$$

then we get

$$b = \frac{\Sigma_u}{M_i \Sigma_u + H^{-1} \Sigma_v (H^{-1})^T} \tag{4.116}$$

Using (4.116) in (4.114), we obtain

$$\mathrm{var}(x_{t+1}) = \frac{H^{-1}\Sigma_v(H^{-1})^T\Sigma_u}{M_i\Sigma_u + H^{-1}\Sigma_v(H^{-1})^T} \tag{4.117}$$

$$\begin{aligned}(\Sigma^{(i)})^{-1} &= (\mathrm{var}(x_{t+1}))^{-1} \\ &= \Sigma_u^{-1} + M_i(H^{-1}\Sigma_v(H^{-1})^T)^{-1} \\ &= \Sigma_u^{-1} + M_i H^T\Sigma_v^{-1}H\end{aligned} \tag{4.118}$$

i.e., (4.93). Using (4.116) in (4.113), we obtain

$$\begin{aligned}m_{t+1}^{(i)} &= \frac{H^{-1}\Sigma_v(H^{-1})^T F x_t^{(i)} + \Sigma_u H^{-1}\sum_{j=1}^{M_i} y_j^{(i)}}{M_i\Sigma_u + H^{-1}\Sigma_v(H^{-1})^T} \\ &= \Sigma^{(i)}(\Sigma_u^{-1}Fx_t^{(i)} + H^T\Sigma_v^{-1}\sum_{j=1}^{M_i} y_j^{(i)})\end{aligned} \tag{4.119}$$

i.e.,(4.94). □

4.7.4 Importance function for spontaneous birth targets

We make an assumption for spontaneous birth targets in our tracking scenario:

Assumption 4.2. For spontaneous birth targets, each target can generate at most one measurement and the measurement is nearest to the predicted measurement of its particle in the residual measurement set as (4.87). Thus, the likelihood function of target $p(Y_{t+1}|x_{t+1}^{(i)})$ can be approximated by the individual likelihood $p(y_{t+1,j}|x_{t+1}^{(i)})$, i.e.,

$$p(Y_{t+1}|x_{t+1}^{(i)}) \cong p(y_{t+1,j}|x_{t+1}^{(i)}) \tag{4.120}$$

Proposition 4.4. *For spontaneous birth targets, the importance function is a Gaussian mixture with its means being the measurements in the residual measurements (4.87), i.e.,*

$$r(x_{t+1}) \sim \frac{1}{\overline{Y}_{t+1}} \sum_{y \in \overline{Y}_{t+1}} N(x_{t+1}; H^{-1}y, H^{-1}\Sigma_v(H^{-1})^T) \tag{4.121}$$

where y is a measurement in the residual measurement sets $\overline{Y}_{t+1}$. The weight function is:

$$p(Y_{t+1}|x_{t+1}^{(i)}) \propto \exp[-\frac{1}{2}(y_{t+1,j} - Hx_{t+1}^{(i)})^T\Sigma_v^{-1}(y_{t+1,j} - Hx_{t+1}^{(i)})] \tag{4.122}$$

The goal of the importance function $r(x_{t+1})$ of the spontaneous birth targets is to generate the particles near the residual measurements. The Gaussian mixture (4.121) is a suitable candidate because it may concentrate the samples in the region of high probability. From the measurement model (4.13) and Assumption 4.2 (4.120), we obtain the weight (4.122). When there is only a residual measurement, this importance function becomes a Gaussian distribution. When there are several residual measurements, this importance function becomes a Gaussian mixture.

4.7.5 Data-driven particle PHD filter

Let L_t denote the particle number at time t, J_t denote the new particle number for the spontaneous birth targets at time t, and w denote a particle's weight. The data-driven particle PHD filter is summarized as follows:

At time $t \geq 0$, let $\{x_t^{(i)}, w_t^{(i)}\}_{i=1}^{L_t}$ denote a particle approximation of the PHD.

1. *Detection*

   Detecting the foreground objects using background substraction with statistical background modelling. The centriods of all foreground blobs are the measurement set Y_{t+1} at time $t+1$.

2. *Prediction*

   - Generate samples for survival targets

     For $i = 1, ..., L_t$,

     (a) generate a measurement set $\widetilde{Y}_{t+1}^{(i)}$ near the predicted position of each particle $x_t^{(i)}$ as (4.2),

     (b) compute a Gaussian distribution (4.92) for each particle using $\widetilde{Y}_{t+1}^{(i)}$,

     (c) generate a sample $\tilde{x}_{t+1}^{(i)}$ from each Gaussian distribution,

     (d) compute the predicted weights as (4.95)

   - Generate samples for spontaneous birth targets

     (a) Generate a residual measurement set $\overline{Y}_{t+1}$ as (4.86) and (4.87),

     (b) generate a Gaussian sum distribution (4.121) using (4.87),

(c) sample $\tilde{x}_{t+1}^{(i)}$ from the Gaussian sum (4.121) distribution for $i = L_t + 1, ..., L_t + J_t$,

(d) compute the predicted weights based on (4.122) as follows:

$$w_{t+1}^{(i)} \propto \frac{1}{J_{t+1}} \exp[-\frac{1}{2}(y_{t+1,j} - Hx_{t+1}^{(i)})^T \Sigma_v^{-1}(y_{t+1,j} - Hx_{t+1}^{(i)})] \tag{4.123}$$

3. *Update*

For each $y \in Y_{t+1}$, use the likelihood and compute

$$C_{t+1}(y) = \sum_{i=1}^{L_t+J_{t+1}} p_D(\tilde{x}_{t+1}^{(i)}) p(y|\tilde{x}_{t+1}^{(i)}) w_{t+1}^{(i)}$$

For $i = 1, ..., L_t + J_{t+1}$, update weights

$$\tilde{w}_{t+1}^{(i)} = [1 - p_D(\tilde{x}_{t+1}^{(i)}) + \sum_{y \in Y_{t+1}} \frac{p_D(\tilde{x}_{t+1}^{(i)}) p(y|\tilde{x}_{t+1}^{(i)})}{\lambda c(y) + C_{t+1}(y)}] w_{t+1}^{(i)}$$

4. *Resampling*

Compute the target number at time $t + 1$

$$\hat{N}_{t+1} = \sum_{i=1}^{L_t+J_{t+1}} \tilde{w}_{t+1}^{(i)}$$

Initialize the cumulative probability $c_1 = 0$, $c_i = c_{i-1} + \tilde{w}_{t=1}^{(i)} / \hat{N}_{t+1}$, $i = 2, ..., L_t + J_{t+1}$.

Draw a starting point $u_1 \sim U[0, L_{t+1}^{-1}]$.

For $j = 1, ..., L_{t+1}$,

$$u_j = u_1 + L_{t+1}^{-1}(j - 1)$$

While $u_j > c_i$, $i = i + 1$. End while.

$$x_{t+1}^{(j)} = \tilde{x}_{t+1}^{(i)}$$

$$w_{t+1}^{(j)} = L_{t+1}^{-1}$$

Rescale (multiply) the weights by $\hat{N}_{t+1}$ to get $\{x_{t+1}^{(i)}, \hat{N}_{t+1}/L_{t+1}\}_{i=1}^{L_{t+1}}$

5. *State extraction*

   Do k-means clustering for particles $\{x_{t+1}^{(i)}\}_{i=1}^{L_{t+1}}$ with the cluster number $k = \text{round}(\hat{N}_{t+1})$ and round(N) is the integer nearest to N. The means of clusters are used as the state estimation of targets.

4.8 Gaussian mixture PHD filter

The basic Gaussian mixture PHD filter is introduced in section 4.8.1. We propose a scene-driven methods for the GMPHD filter in section 4.8.2.

4.8.1 Basic Gaussian mixture PHD filter

The GMPHD filter is initialized in Step 1 and iterates through Steps 2 to 6.

1. *Initialization*

Initialize the algorithm with the weighted sum of J_0 Gaussians,

$$D_{0|0} = \sum_{i=1}^{J_0} w_0^{(i)} N(x; m_0^{(i)}, P_0^{(i)}) \tag{4.124}$$

where $N(x; m, P)$ is a Gaussian distribution with the mean m and the variance P. The sum of weights,

$$\sum_{i=1}^{J_0} w_0^{(i)} = \hat{T}_0 \tag{4.125}$$

is the expected number of objects at the beginning.

2. *Prediction*

The prediction density at time $t+1$ is

$$D_{t+1|t}(x) = b_{t+1}(x) + D_{S,t+1|t}(x) \tag{4.126}$$

The intensity of the spontaneous birth objects is

$$b_{t+1}(x) = \sum_{i=1}^{J_b} w_{b,t+1}^{(i)} N(x; m_{b,t+1}^{(i)}, P_{b,t+1}^{(i)}) \tag{4.127}$$

The intensity of the survival objects is

$$D_{S,t+1|t}(x) = p_S \sum_{i=1}^{J_t} w_t^{(i)} N(x; m_{s,t+1}^{(i)}, P_{s,t+1}^{(i)}) \tag{4.128}$$

$$m_{s,t+1|t}^{(i)} = F m_t^{(i)} \tag{4.129}$$

$$P_{s,t+1|t}^{(i)} = \Sigma_u + F P_t^{(i)} F^T \tag{4.130}$$

3. *Update*

When the measurements $Y_{t+1} = \{y_{t+1,1}, ..., y_{t+1,|Y_{t+1}|}\}$ at time $t+1$ are available, the posterior intensity is computed as follows:

$$D_{t+1|t+1}(x) = (1 - p_D)D_{t+1|t}(x) + \sum_{y \in Y_{t+1}} \sum_{i=1}^{J_{t+1|t}} w_{t+1}^{(i)}(y) N(x; m_{t+1|t+1}^{(i)}, P_{t+1|t+1}^{(i)}) \tag{4.131}$$

$$w_{t+1}^{(i)}(y) = \frac{p_D w_{t+1|t}^{(i)} N(y; Hm_{t+1|t}^{(i)}, \Sigma_v + HP_{t+1|t}^{(i)} H^T)}{\lambda c(y) + \sum_{j=1}^{J_{t+1|t}} p_D w_{t+1|t}^{(j)} N(y; Hm_{t+1|t}^{(i)}, \Sigma_v + HP_{t+1|t}^{(i)} H^T)} \tag{4.132}$$

$$m_{t+1|t+1}^{(i)}(y) = m_{t+1|t}^{(i)} + K_{t+1}^{(i)}(y - Hm_{t+1|t}^{(i)}) \tag{4.133}$$

$$P_{t+1|t+1}^{(i)} = [I - K_{t+1}^{(i)} H] P_{t+1|t}^{(i)} \tag{4.134}$$

$$K_{t+1}^{(i)} = P_{t+1|t}^{(i)} H^T (HP_{t+1|t}^{(i)} H^T + \Sigma_v)^{-1} \tag{4.135}$$

4. *Pruning*

   In the pruning stage, the Gaussian components with low weights are eliminated. Let the weights $w_{t+1}^{(1)}, \cdots, w_{t+1}^{(N_P)}$ be those which are below the eliminated threshold, and the intensity after pruning is

$$\overline{D}_{t+1|t+1} = \frac{\sum_{l=1}^{J_{t+1}} w_{t+1}^{(l)}}{\sum_{j=N_P+1}^{J_{t+1}} w_{t+1}^{(j)}} \sum_{i=N_P+1}^{J_{t+1}} w_{t+1}^{(i)} N(x; m_{t+1}^{(i)}, P_{t+1}^{(i)}) \tag{4.136}$$

5. *Merging*

   In the merging stage, Gaussian components whose distance between the means falls within a threshold U are merged. For example, if the means of components i and j satisfies

$$(m_{t+1}^{(i)} - m_{t+1}^{(j)})^T (P_{t+1}^{(i)})^{-1} (m_{t+1}^{(i)} - m_{t+1}^{(j)}) \leq U \tag{4.137}$$

these components are merged into a single Gaussian.

Given $\{w_{t+1}^{(i)}, m_{t+1}^{(i)}, P_{t+1}^{(i)}\}_{i=N_P+1}^{J_{t+1}}$, a merging threshold U, and a maximum allowable number of Gaussian terms J_{max}, the merging procedure is as follows:

Set l=0, and $I = \{i = 1, \cdots, J_{t+1} | w_{t+1}^{(i)} > \tau\}$

Repeat

$$l = l + 1$$

$$j = \underset{i \in I}{\operatorname{argmax}}\, w_{t+1}^{(i)}$$

$$L = \{i \in I | (m_{t+1}^{(i)} - m_{t+1}^{(j)})^T (P_{t+1}^{(i)})^{-1} (m_{t+1}^{(i)} - m_{t+1}^{(j)}) \leq U\}$$

$$\tilde{w}_{t+1}^{(l)} = \sum_{i \in L} w_{t+1}^{(i)}$$

$$\tilde{m}_{t+1}^{(l)} = \frac{1}{\tilde{w}_{t+1}^{(l)}} \sum_{i \in L} w_{t+1}^{(i)} m_{t+1}^{(i)}$$

$$\tilde{P}_{t+1}^{(l)} = \frac{1}{\tilde{w}_{t+1}^{(l)}} \sum_{i \in L} w_{t+1}^{(i)} [P_{t+1}^{(i)} + (\tilde{m}_{t+1}^{(l)} - m_{t+1}^{(i)})(\tilde{m}_{t+1}^{(l)} - m_{t+1}^{(i)})^T]$$

$$I = I \backslash L$$

Until $I = \phi$

If $l > J_{max}$, replace $\{\tilde{w}_{t+1}^{(i)}, \tilde{m}_{t+1}^{(i)}, \tilde{P}_{t+1}^{(i)}\}_{i=1}^{l}$ by the J_{max} Gaussians with largest weights.

Output $\{\tilde{w}_{t+1}^{(i)}, \tilde{m}_{t+1}^{(i)}, \tilde{P}_{t+1}^{(i)}\}_{i=1}^{l}$ as the merged Gaussian components.

6. *State estimation*

The states of objects are determined from the posterior intensity by taking the components whose weights are above a specific threshold, which represents the expectation of the object. For example, if the weight is greater than 0.5, the expectation of an object which falls within the region defined by component i is than 0.5. The state set estimates at time $t+1$ is

$$\hat{X}_{t+1} = \{m_{t+1}^{(i)} : w_{t+1}^{(i)} > 0.5\} \tag{4.138}$$

4.8.2 Scene-driven method for new-birth objects

We found that new objects can only enter the field of view of the camera at 3 positions, i.e., position A, B, and C in Fig. 4.3.

We use this prior scene knowledge in tracking. For the initialization of Gaussian mixture (4.124) and the model of new-birth objects (4.127), we model them with 3-components Gaussian mixture whose means are the locations of position A, B, and C as follows:

$$\frac{1}{3}\sum_{i=A,B,C}[N(x_{t+1}; H^{-1}z_i, H^{-1}\Sigma_v(H^{-1})^T)] \tag{4.139}$$

where z_i is the position of the ith entry point in the filed of view of camera.

Figure 4.3: Scene-driven method. A, B, and C are the positions where new objects may appear.

4.9 Results

4.9.1 Particle PHD filter

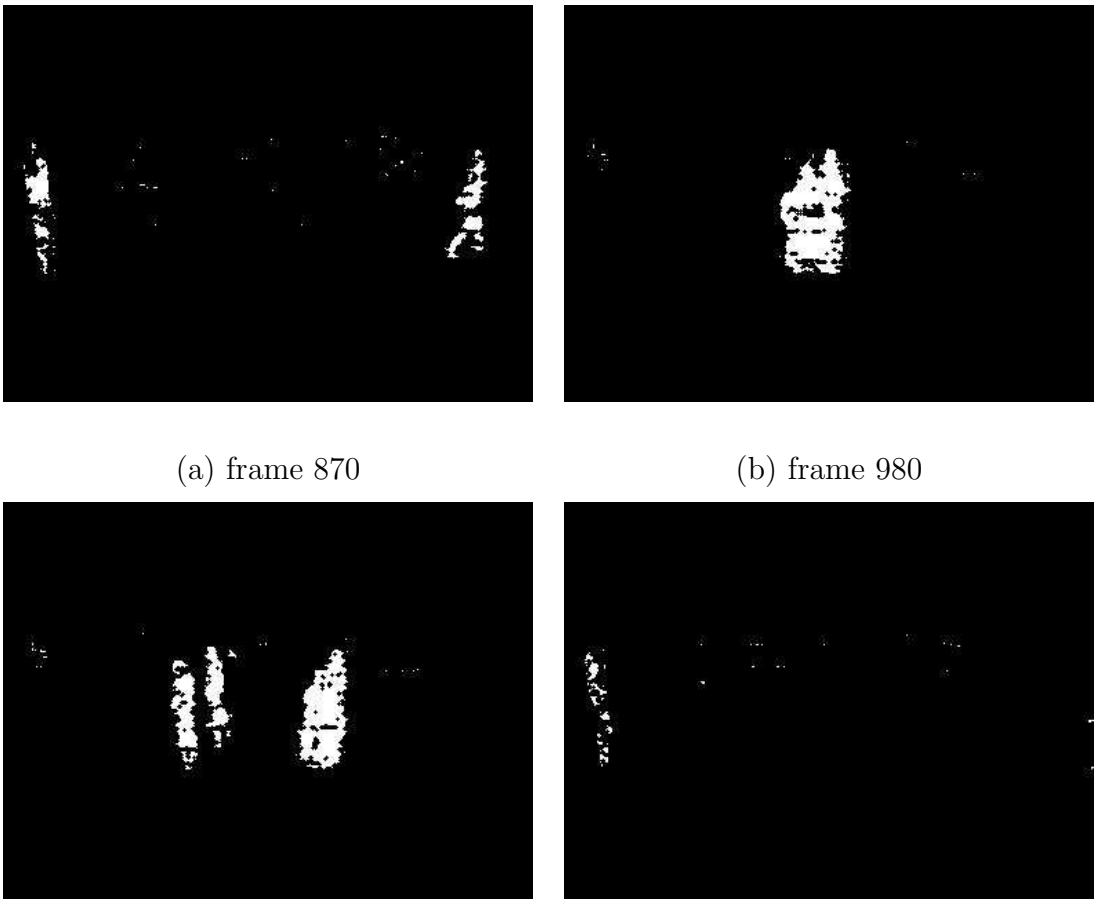
We test our method using the dataset of the European Commission Funded CAVIAR project [2]. The parameters used in experiments are summarized in Table 4.2.

Video *OneStopMoveEnter1front* has 1588 frames. There are two human groups

Table 4.2: Parameter list of the particle PHD filter

σ_u: standard deviation of state noise	3
σ_v: standard deviation of measurement noise	3
ρ: particle number per target	50
J_t particle number for spontaneous birth targets	50
P_D: detection probability	0.99
λ: average number of clutter points per frame	0.01
c: probability distribution of each clutter point	$(352 * 288)^{-1}$
P_S: probability that the target exits	0.95

appearing, merging, splitting, or disappearing in the field of view of the camera. The detection results using background subtraction [128] are shown in Fig. 4.4. Fig. 4.5 shows 4 video frames with white circles indicating the tracking results: the centroids of human groups. The particle number used for spontaneous birth targets is 50 as shown in Table 4.2. Because the PHD filter explicitly models the processes of birth, survival, death of targets and false alarms of clutter, as shown by our experimental results, the particle PHD filter is able to track the variable number of human groups and their positions.



(a) frame 870 (b) frame 980

(c) frame 1010 (d) frame 1110

Figure 4.4: Detection results of adaptive background subtraction for frames 870, 980, 1010, and 1110 of video *OneStopMoveEnter1front*.

(a) two groups appeared (frame 870)

(b) two groups merged (frame 980)

(c) two groups split (frame 1010)

(d) one group disappeared (frame 1110)

Figure 4.5: Tracking results of the particle PHD filter for frames 870, 980, 1010, and 1110 of video *OneStopMoveEnter1front*. The two human groups appear, merge, split, and disappear in the field of view of the camera. The white circles are the centroids of human groups.

Fig. 4.6 provides the tracking results for the first 1000 frames of video *OneStop-MoveEnter1front*. The correct frame number is 744 out of the 1000 frames. The errors mainly come from two factors: i) the inaccuracy of measurements; ii) the importance sampling for new-birth targets does not generate samples near the birth target's position.

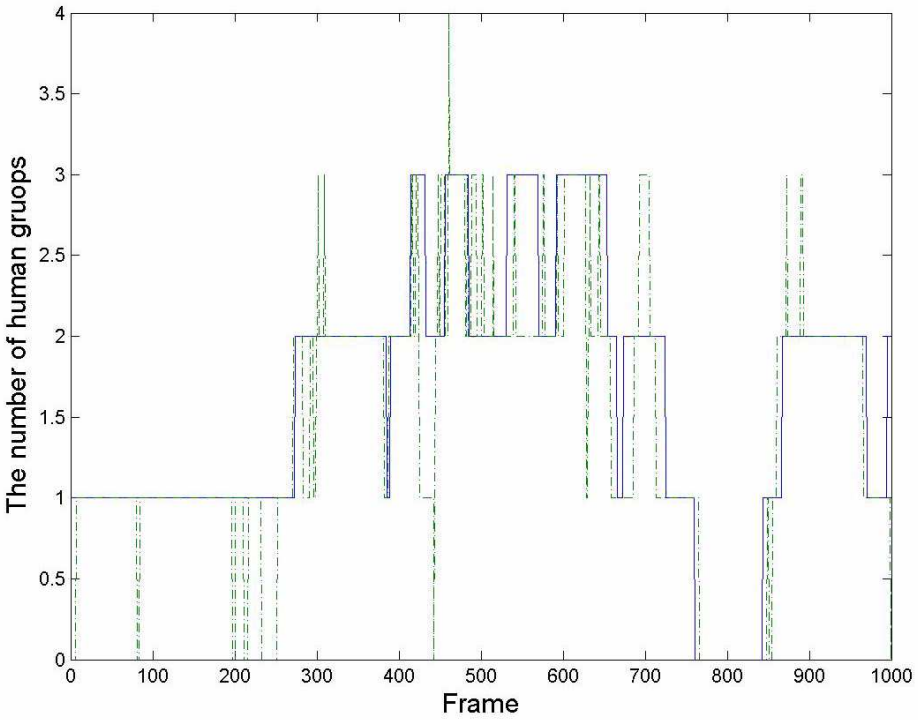


Figure 4.6: Tracking result of the particle PHD filter for the number of targets. The solid line is the ground truth number of people or groups. The dashed line is the tracking result of the PHD filter.

The results confirm that the probability hypothesis density filter can track the

variable number of targets and their positions. This property of the PHD filter may be suitable for multisensor multitarget tracking under complex environments. The results can be explained by the fact that the PHD filter explicitly models the processes of birth, survival, death of targets and false alarms of clutter. This is consistent with the earlier results of Vo *et al.* [125] and Sidenbladh [112]. It is worth noting that the PHD filter differs from the traditional visual tracking methods. The traditional visual tracking methods rely on only detection results to determine the birth or death of targets. Therefore, they are data-driven methods. On the other hand, the PHD filter explicitly models the birth, survival, or death of targets in its dynamics. Therefore, the PHD filter is a model-driven method for tracking.

4.9.2 Data-driven PHD filter

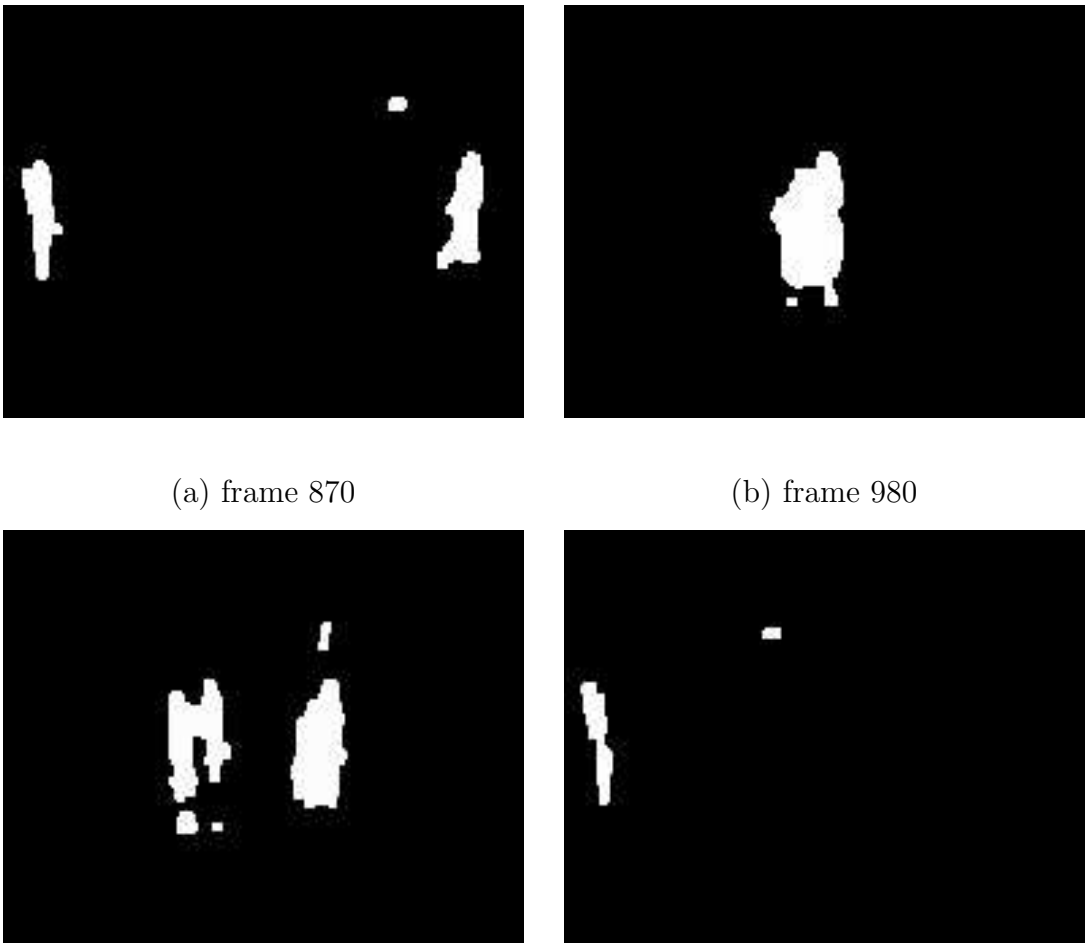
The data-driven particle PHD filter is tested using the dataset of the CAVIAR project [2]. Some results of video *OneStopMoveEnter1front* using the statistical background modelling described in Section 4.2 are shown in Fig. 4.7 .

Fig. 4.8 shows four video frames with white squares being the centroids of people or groups. As shown by the experimental results, the data-driven particle PHD filter is able to track a variable number of objects because the PHD filter explicitly models the processes of birth, survival, death of targets and false alarms of clutter.

It is noted that our method considers the left 2 people in Fig. 4.8c as a human group. The reason is that the detection algorithm detects the two close targets into one foreground objects as show in Fig. 4.7c.

Video *Meet_Split_3rdGuy* has three people in the field of view of the camera. The detection results using statistical background modelling are shown in Fig. 4.9.

The data-driven particle PHD filter is then applied for these detection results and Fig. 4.10 shows 4 frames with white squares being the centroids of people or groups. The particle number used for spontaneous birth targets is 50 as shown in Table 4.2. When a person at the bottom of Fig. 4.10a appears in the field of view of the camera, the Gaussian mixture importance function (4.121) quickly generates samples for the new-birth person and locate his position. When the left 2 people in Fig. 4.10b merge into a group, the data-driven particle PHD filter tracks the centroid of the group. When the left 2 people split, the data-driven particle PHD filter tracks the positions of the 2 people (Fig. 4.10c). When a person at the bottom of Fig. 4.10d moves out of the field of view of the camera, the data-driven particle PHD filter detects the death of the existing target.



(a) frame 870 (b) frame 980

(c) frame 1010 (d) frame 1110

Figure 4.7: Detection results for video *OneStopMoveEnter1front*.

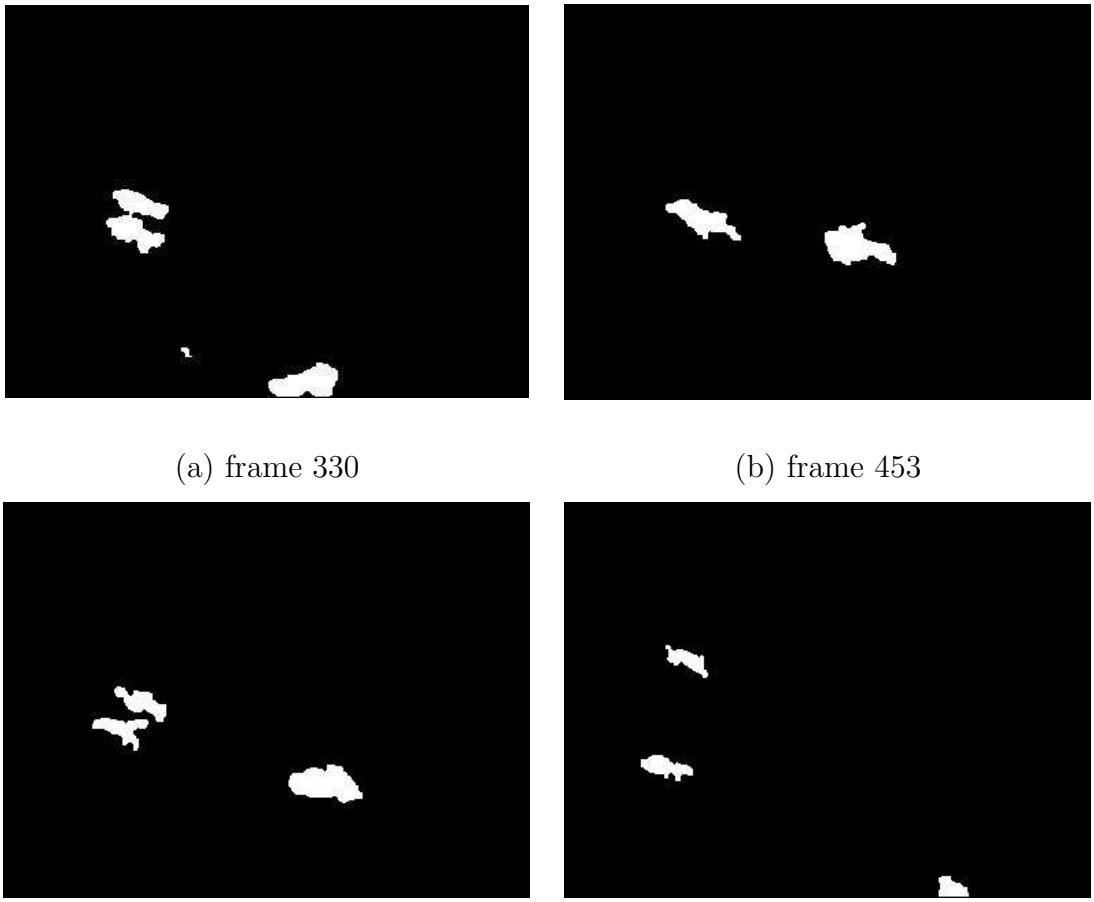
(a) two groups appeared (frame 870)

(b) two groups merged (frame 980)

(c) two groups split (frame 1010)

(d) one group disappeared (frame 1110)

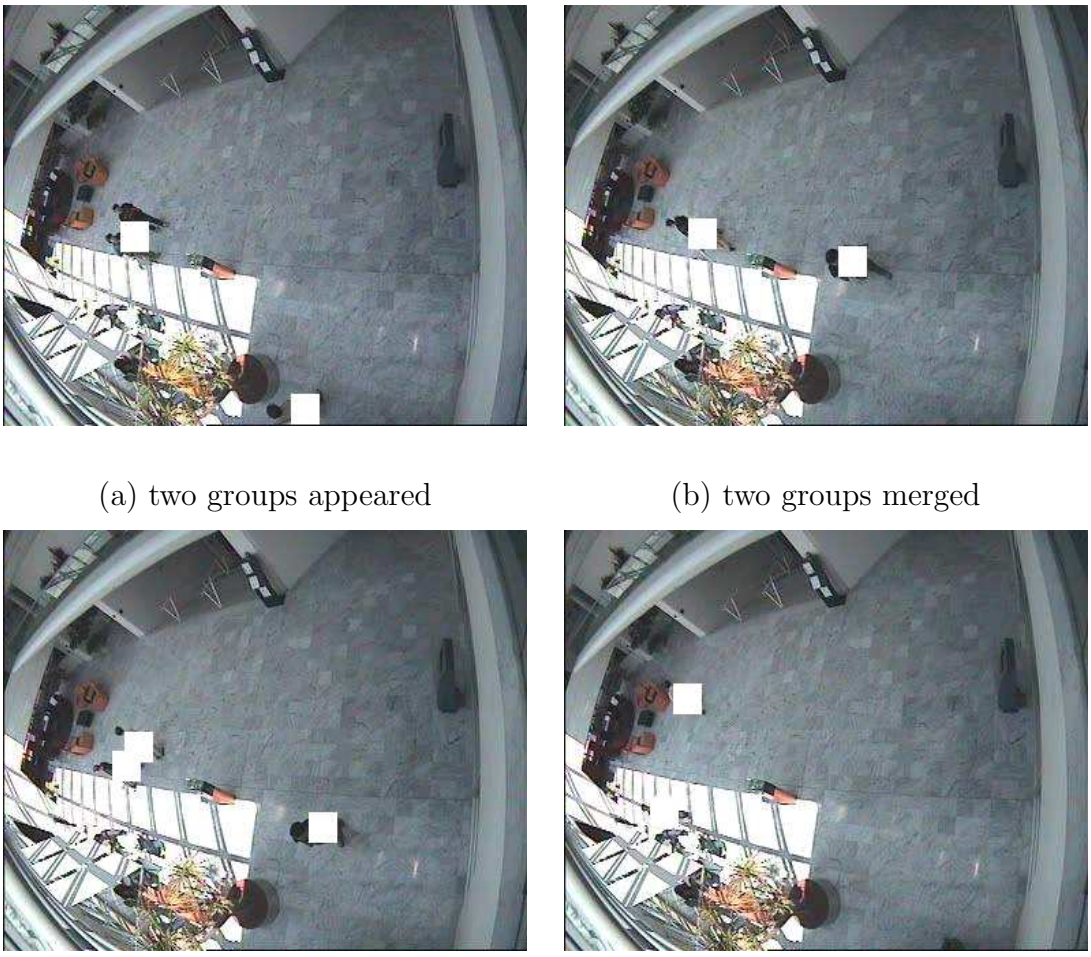
Figure 4.8: Tracking results of the data-driven particle PHD filter for frames 870, 980, 1010, and 1110 of video *OneStopMoveEnter1front*. The white squares are the centroids of people or groups.



(a) frame 330 (b) frame 453

(c) frame 469 (d) frame 517

Figure 4.9: Detection results of statistical background modelling for frames 330, 453, 469, and 517 of video *Meet_Split_3rdGuy*.



(a) two groups appeared

(b) two groups merged

(c) two groups split

(d) one group disappeared

Figure 4.10: Tracking results of the data-driven particle PHD filter for frames 330, 453, 469, and 517 of video *Meet_Split_3rdGuy*. The white squares are the centroids of people or groups.

We compare the data-driven particle PHD filter in section 4.7 with the particle PHD filter in section 4.6. Fig. 4.11 shows the tracking results using the particle PHD filter.

(a) frame 334 (b) frame 335

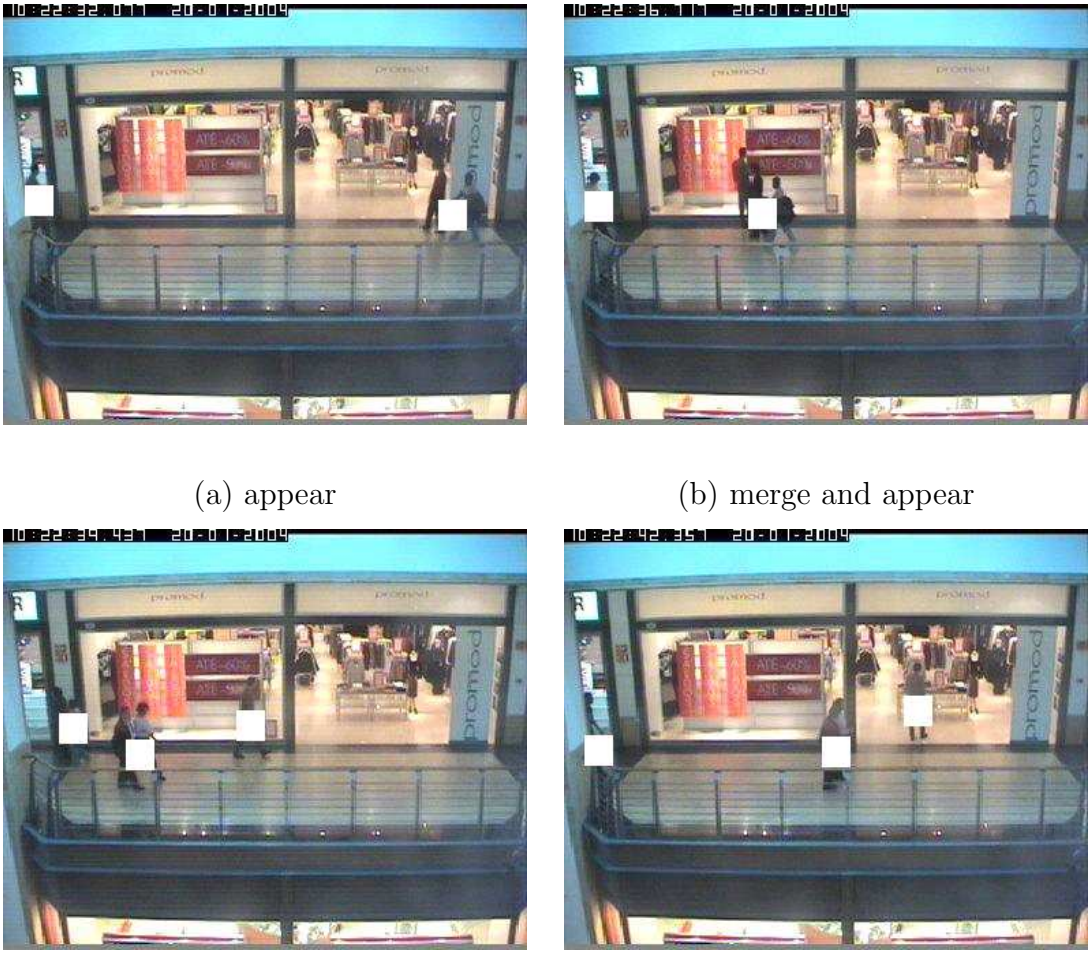
Figure 4.11: Tracking results of the particle PHD filter for video *Meet_Split_3rdGuy*. The white squares are the centroids of objects.

The particle number used for spontaneous birth targets is 50 as shown in Table 4.2. The Gaussian mixture importance function (4.121) could track the new-birth target at frame 330 whereas the uniform importance function (4.61) started tracking the new targets at frame 335. Increasing the particle number for spontaneous birth targets should be able to speed finding new targets at the cost of increasing computational load. The Gaussian mixture importance function uses the data-driven information to concentrate samples on high-probability regions where new targets may appear. On contrast, the uniform importance function must randomly search

the whole image to verify the new target's appearing. Therefore, the Gaussian mixture importance function can track new birth objects faster than the uniform importance function.

4.9.3 Gaussian mixture PHD filter

The GMPHD filter is tested using the CAVIAR dataset [2]. Fig. 4.12 shows 4 frames (frame 275, 391, 459, and 484) of video *OneStopMoveEnter1front* with white squares being the tracking results.



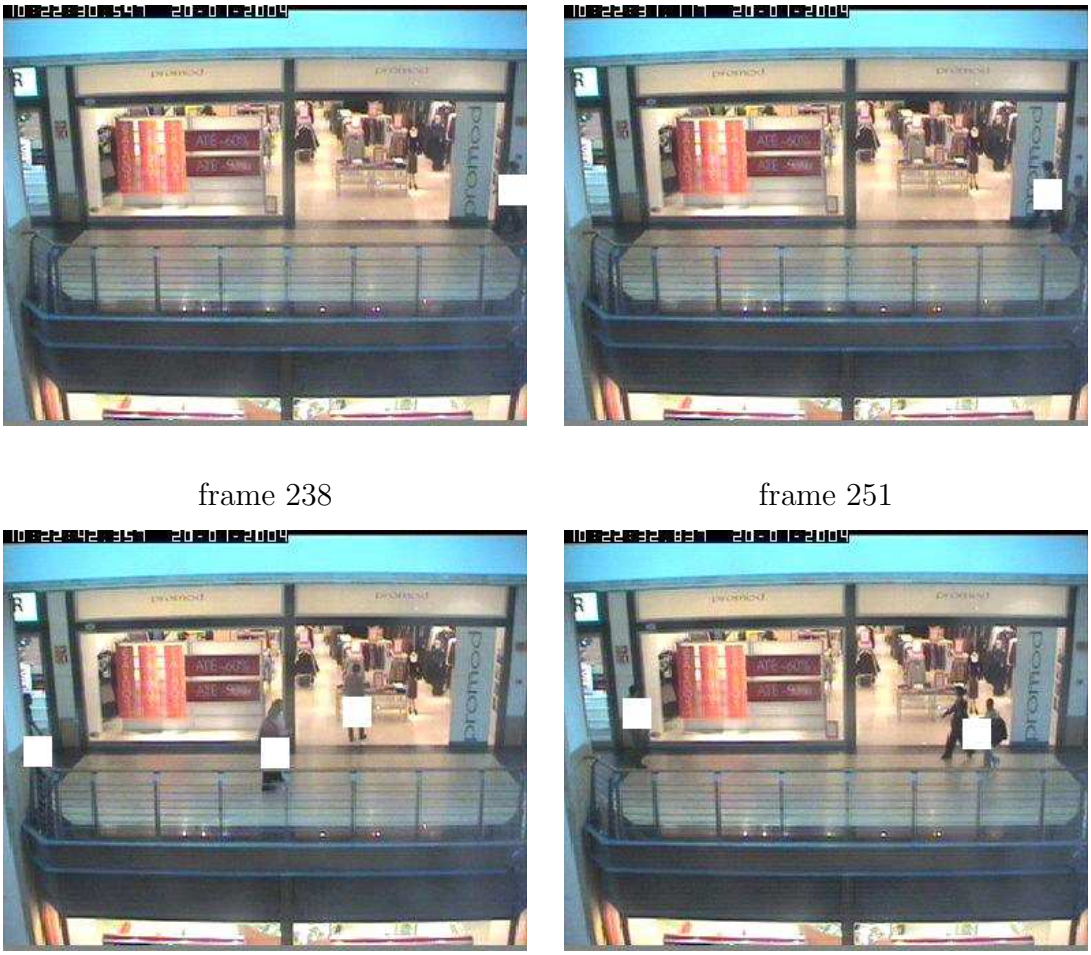
(a) appear (b) merge and appear

(c) split (d) disappear

Figure 4.12: Tracking result of the GMPHD filter for video *OneStopMoveEnter1front*. The white squares are tracking results.

Because the PHD filter explicitly models the processes of birth, survival, death of targets and false alarms of clutter, as shown by the experimental results, this method is able to track the variable number of people or groups. It is noted that our method considered the 2 people on the right in Fig. 4.12a as a group. The explanation for this is that the detection algorithm detects the two close targets into one foreground object.

We compare the scene-driven GMPHD filter (section 4.8) with the particle PHD filter (section 4.6). The scene-driven GMPHD filter can track the birth of new objects faster than the particle PHD filter. Fig. 4.13 shows the first frame when the new targets are tracked. The white squares in Fig. 4.13a are the results of the GMPHD filter and the white squares in Fig. 4.13b are the results of the particle PHD filter. Because the particle PHD filter uses a uniform distribution as the proposal density of particle filter for new-birth objects and the sample number of particle filter is limited in practice, it is possible that it does not generate samples near the positions of new birth objects. While the GMPHD filter uses the prior scene knowledge and is able to track the new-birth objects quickly.



frame 238 frame 251

frame 274 frame 294

(a) GMPHD (b) particle PHD

Figure 4.13: Comparison of the GMPHD filter and the particle PHD filter for new-birth objects. The first row is the results for a person appearing at position C and the second row is the results for a person appearing at position A.

Fig. 4.14 provides the estimates of the target number of the GMPHD filter for video *OneStopMoveEnter1front*. The correct frame number is 1148 out of the 1588 frames.

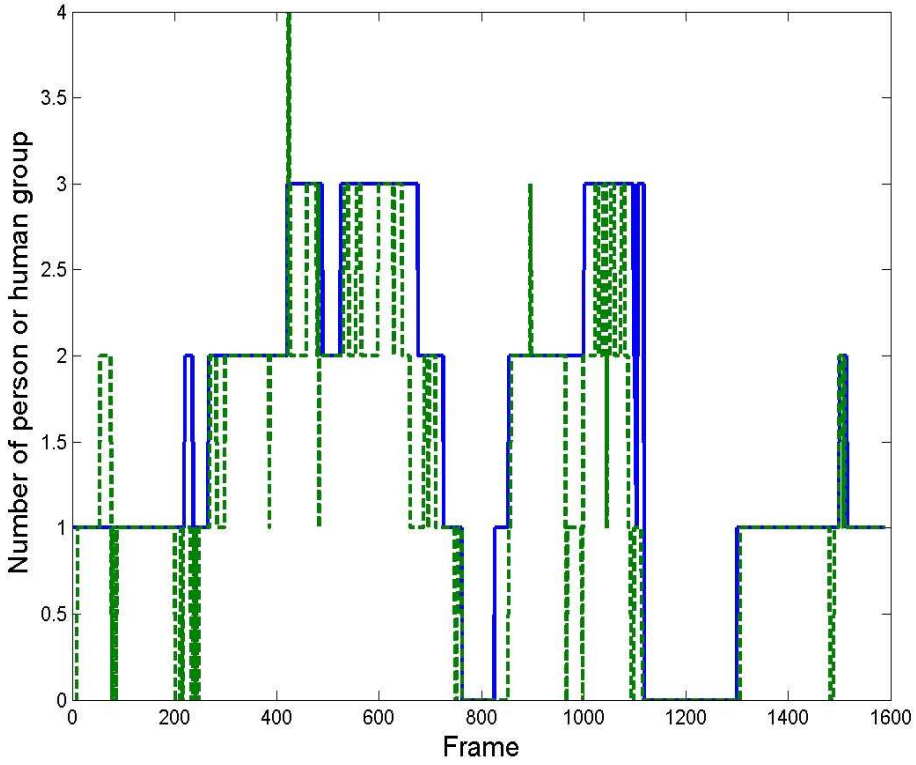


Figure 4.14: Absolute error in estimates of target number. The solid line is the ground truth of the number of targets. The dashed line is the tracked target numbers of the GMPHD filter.

For the estimated positions, the Wasserstein distance [58] is used as a metric to measure the performance because it defines a metric for multitarget distance which

penalizes when the estimated number of targets is incorrect. The above figure of Fig. 4.15 is the Wasserstein distance between the estimated positions of the GMPHD filter and the ground-truth positions. While the below figure of Fig. 4.15 is the Wasserstein distance between the positions of detected targets and the ground-truth positions. The results show that the tracking errors mainly come from the inaccuracy of measurements.

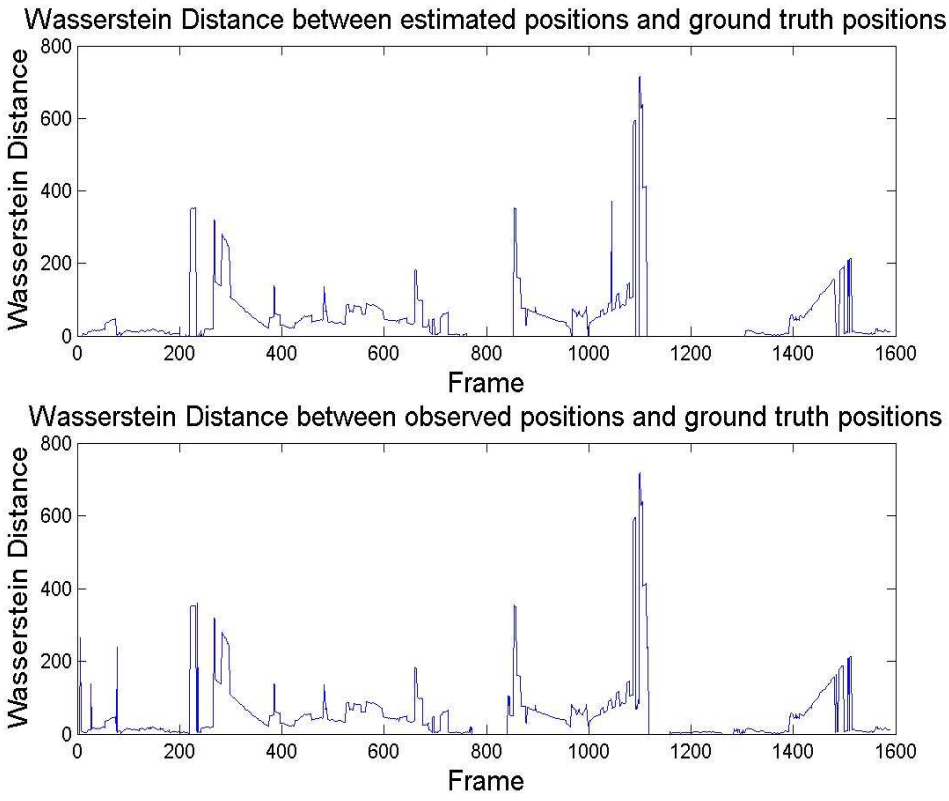


Figure 4.15: Wasserstein distance.

Table 4.3: Comparison between the GMPHD filter and the particle PHD filter

	GMPHD	Particle PHD
Rate of frame with the correct number of targets	72.3 %	74.4 %
Average Wasserstein distance per frame	49.4153	72.4567

The comparison between the GMPHD filter and the particle PHD filter is summarized in Table 4.3, which are based the statistical results of 1588 frames. The particle PHD filter used 50 particles for each object.

4.10 Discussion

The results confirm that both the particle probability hypothesis density filter and the Gaussian mixture probability hypothesis density filter can track a variable number of targets and derive their positions. This property of the PHD filter may be suitable for multisensor multitarget tracking under complex environments. The results can be explained by the fact that the PHD filter uses samples and the GMPHD filter used Gaussian components to explicitly model the processes of birth, survival, death of targets, missed detection, and false alarms of clutter. This is consistent with the earlier results of [125], [113] and [124].

When the target number is time-varying, the tracking algorithm usually determines

the target number firstly, and then derives the states of targets. It is worth noting that the PHD filter differs from the traditional multi-target tracking methods in determining the target number. The traditional multi-target tracking methods rely on only detection results of sensor to determine the numbers of targets and are data-driven methods. For example, the boosted particle filter [98] adds, deletes, and merges targets according to the overlapping regions between the targets detected by Adaboost algorithm and the existing targets (from the authors' programs [3]). Reversible jump Markov chain Monte Carlo (RJMCMC) methods [72], [114] uses "hypothesize and test" approach to determine the target number. For example, [72] restricted proposals of RJMCMC to add or remove a single target and [114] defined a global observation model to evaluate the configurations of variable number of targets. Whereas the PHD filter automatically determines the target number by using the integral of PHD over the field of view (the sum of weights of all particles in particle filter based implementation and the sum of weights of all Gaussian components in Gaussian mixture based implementation). This method can track spontaneous birth and death of multiple targets in one frame. Moreover, the PHD filter explicitly models the birth, survival, or death of targets in its dynamics and also explicitly models the missed detection and the false alarms by clutter environment. Therefore, the PHD filter is a model-driven method. Our contribution is: i) to combine the traditional visual tracking method and the PHD filter according to the importance sampling of particle filter. This data-driven

particle PHD filter automatically determines the target number in the tracking region and improves the tracking performance of the PHD filter; ii) to combine the data-driven method (detection) with the model-driven method (GMPHD) and the scene-driven method (prior knowledge).

The detection and filtering was carried out in two separate phases in our experiments using an Intel 1.86GHz CPU PC. Detection is achieved at a rate of 3 frames per second for 352×288 images while the data-driven particle PHD filtering is achieved at a rate of 15 frames per second. The computational complexity of the particle PHD filter at time $t+1$ is $O((L_t + J_{t+1})|Y_{t+1}|)$. As we can see here, the processing time is linearly proportional to the number of particles L_t at time t, the number of particles for the spontaneous birth targets J_{t+1} at time $t+1$, and the number of measurements $|Y_{t+1}|$ at time $t+1$.

4.11 Summary

In this chapter, the probability hypothesis density filter is applied to a visual tracking problem. Foreground objects are detected using the statistical background modelling, and a variable number of people or human groups are tracked using the PHD filter implemented by both sequential Monte Carlo method and Gaussian mixture. We present a data-driven particle PHD filter and propose two importance functions and weight functions for it. We also introduce a scene-driven Gaussian

mixture PHD filter. The result shows both methods are able to track a variable number of targets and derive their positions in image sequences.

Chapter 5

Conclusion and future work

Target tracking is the core of the systems that perform functions such as surveillance or guidance. For multi-sensor multi-target tracking, the recursive state-space Bayesian filter provides a framework to fuse the spatial and temporal information. However, many issues in multisensor-multitarget tracking, especially the information fusion of multiple cameras and tracking time-varying number of targets, remain as very challenging problems. This thesis introduced two Bayesian filtering methods, namely, particle filter and the probability hypothesis density filter, to solve these two challenges and demonstrated their use in real visual tracking scenarios.

The first contribution of this thesis is our proposal for a data fusion method based on an adaptive mixed particle filter for visual tracking using multiple cameras

with the overlapping fields of view. A theoretical framework based on the spatio-temporal recursive Bayesian filtering was presented for data fusion of multiple cameras. The spatio-temporal recursive Bayesian filtering was formulated using an adaptive mixed particle filter. The particle filter uses the mixed importance sampling strategy to fuse spatial information from multiple cameras and temporal information of dynamic system. The particle filter is adaptive in sense that it automatically ranks data from multiple cameras and assigns weights according to quality of the data in the fusion process. The adaptive mixed particle filter can automatically recover the location of an occluded target while the previous methods (e.g. the mean shift algorithm [32] and the condensation algorithm [62]) experience difficulties.

The second contribution of this thesis is the ability to apply the probability hypothesis density (PHD) filter to a visual tracking problem. Foreground objects were detected using the statistical background modelling, and a variable number of people or groups were tracked using the PHD filter, which was implemented using two methods: both particle filter and Gaussian mixture. For the particle PHD filter, two importance functions and corresponding weight functions were proposed for survival targets and spontaneous-birth targets, respectively. The importance function for survival targets theoretically extends the optimal importance function of the linear Gaussian model from single-measurement case to measurement-set (multi-measurement) case. This is a data-driven importance sampling method.

The importance function for spontaneous-birth targets is also a data-driven method which uses the current measurements in the sampling process of the particle PHD filter. For the Gaussian mixture PHD filter, a scene-driven method which incorporates the prior knowledge of scene into the PHD filter was presented. The results demonstrated that these PHD filters are able to track a variable number of people or groups in image sequences and might be used in tracking a variable number of targets under complex environments.

In this work, we extended two Bayesian filtering methods, the particle filter and the probability hypothesis density filter, to real visual tracking scenarios. There remains a number of topics which invite further investigation.

- Tracking an unknown number of targets using multiple cameras is very important in video surveillance applications, and so far there are no suitable solutions for this class of problems. Combining the adaptive particle filter for information fusion of multiple camera and the PHD filter for tracking unknown number of targets can provide a promising solution for this class of problems.

- For very crowded scenes, the labels of objects may switch during occlusion. For example, the soccer players may slow down, cease motion, and occlude each other when they congregate and celebrate a goal. Deriving the contextual three-dimension information could be helpful for resolving this situation.

By this way the prior knowledge of scenes is integrated into the Bayesian filtering framework for a more robust tracking system.

- The combination of information fusion of multiple sensors and tracking variable number of targets may also be extended to other application fields such as radar tracking, sonar tracking, or infrared tracking. In these tracking scenarios, data association may be incorporated into this Bayesian filtering framework to track both positions and identities of targets.

Publications

1. Ya-Dong Wang, Jian-Kang Wu, and Ashraf A. Kassim, "Adaptive particle filter for data fusion of multiple cameras," Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology, volume 49, number 3, December 2007, pages 363-376

2. Ya-Dong Wang, Jian-Kang Wu, Weimin Huang, and Ashraf A. Kassim, "Gaussian mixture probability hypothesis density for visual people tracking," the 10th International Conference on Information Fusion, Quebec City, Canada, 2007

3. Ya-Dong Wang, Jian-Kang Wu, Ashraf A. Kassim, and Wei-Min Huang, "Tracking a variable number of human groups in video using probability hypothesis density," the 18th International Conference on Pattern Recognition, Hong Kong, 2006

4. Ya-Dong Wang, Jian-Kang Wu, and Ashraf A. Kassim, "Particle filter for visual tracking using multiple cameras," the 9th IAPR Conference on Machine

Vision Applications, Tsukuba Science City, Japan, 2005

5. Ya-Dong Wang, Jian-Kang Wu, and Ashraf A. Kassim, "Multiple cameras tracking using particle filtering," IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, Breckenridge, Colorado, 2005

6. Ya-Dong Wang, Jian-Kang Wu, Ashraf A. Kassim, and Wei-Min Huang, "Data-driven probability hypothesis density for visual tracking," IEEE Transactions on Circuits and Systems for Video Technology, under review

7. Ya-Dong Wang, Ashraf A. Kassim, and Jian-Kang Wu, "Adaptive mixture importance sampling for particle filter," Circuit, System, and Signal Processing, under review

Bibliography

[1] ftp://pets2001.cs.rdg.ac.uk. Dataset of Second IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS).

[2] http://groups.inf.ed.ac.uk/vision/caviar/caviardata1/. Dataset of European Commission funded Context Aware Vision using Image-based Active Recognition (CAVIAR) project.

[3] http://www.cs.ubc.ca/∼okumak/research.html. Programs for boosted particle filter.

[4] J.K. Aggarwal and Q. Cai. Human motion analysis: A review. *Computer Vision and Image Understanding*, 73(3):428–440, March 1999.

[5] B.D.O. Anderson and J.B. Moore. *Optimal filtering*. Prentice-Hall, NJ, 1979.

[6] C. Andriew, N. de Freitas, A. Doucet, and M.I. Jordon. An introduction to MCMC for machine learning. *Machine Learning*, 50:5–43, 2003.

[7] E. Arnaud, E. Memin, and B. Cernuschi-Frias. Conditional filters for image sequence based tracking - application to point tracking. *IEEE Transaction on Image Processing*, 14(1):63–79, January 2005.

[8] Y. Bar-Shalom, editor. *Multitarget-Multisensor Tracking: Advanced Applications*. Artech House, 1990.

[9] Y. Bar-Shalom, editor. *Multitarget-Multisensor Tracking: Applications and Advances*. Artech House, 1992.

[10] Y. Bar-Shalom and W. Dale Blair, editors. *Multitarget-Multisensor Tracking: Applications and Advances*. Artech House, 2000.

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

[12] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation: Algorithms and Software for Information Extraction*. J. Wiley and Sons, 2001.

[13] Y. Bar-Shalom and E. Tse. Tracking in a cluttered environment with probabilistic data association. *Automatica*, 11:451–460, September 1975.

[14] J.L. Barron, D.J. Fleet, and S.S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.

[15] J. Black and T. Ellis. Multiple camera image tracking. In *Performance Evaluation of Tracking and Surveillance (PETS 2001), with CVPR 2001*, pages 315–320, Kauai, Hawaii, December 2001.

[16] S. Blackman and R. Popoli. *Design and analysis of modern tracking systems*. Artech House, 1999.

[17] S.S. Blackman. Multiple hypothesis tracking for multiple target tracking. *IEEE Aerospace and Electronic Systems Magazine*, 19(1):518, January 2004.

[18] A. Blake. Visual tracking: a short research roadmap. In O. Faugeras, Y. Chen, and N. Paragios, editors, *In Mathematical Models of Computer Vision: The Handbook*. Springer, 2005.

[19] A. Blake and M. Isard. *Active Contours*. Springer, 2000.

[20] H.A.P. Blom and Y. Bar-Shalom. The interacting multiple model algorithm for systems with Markovian switching coefficients. *IEEE Transactions on Automatic Control*, 33, Augest 1988.

[21] Q. Cai and J.K. Aggarwal. Tracking human motion in structured environments using a distributed camera system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1241–1247, November 1999.

[22] J. Carpenter, P. Clifford, and P. Fearnhead. Improved particle filter for nonlinear problems. *Proc. Inst. Elect. Eng., Radar, Sonar, Navig.*, 146(1):2–7, February 1999.

[23] C. Cedras and M. Shah. Motion based recognition: A survey. *Image and Vision Computing*, 13(2):129–155, March 1995.

[24] T-H. Chang and S. Gong. Tracking multiple people with a multi-camera system. In *Proceedings of IEEE Workshop on Multi-Object Tracking, with ICCV 2001*, Vancouver, B.C., Canada, 2001.

[25] D. Clark, K. Panta, and B.-N. Vo. The GM-PHD filter multi-target tracker. In *Proc. Int. Conf. Information Fusion*, Florence, July 2006.

[26] D. Clark and B. Vo. Convergence results of the particle PHD filter. *IEEE Trans. Signal Processing*, 54(7):2652–2661, July 2006.

[27] D. Clark and B. Vo. Convergence analysis of the Gaussian mixture probability hypothesis density filter. *IEEE Trans. Signal Processing*, 55(4):1204–1212, 2007.

[28] D. Clark, B.-N. Vo, and J. Bell. GM-PHD filter multi-target tracking in sonar images. In *SPIE Defense and Security Symposium*, Orlando, Florida, 2006.

[29] D.E. Clark and J. Bell. Bayesian multiple target tracking in forward scan sonar images using the PHD filter. *IEE Proc. Radar, Sonar, Navig.*, 152(5):327–334, 2005.

[30] R. Collins, A. Lipton, and T. Kanade. Special issue on video surveillance and monitoring. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):745–746, August 2000.

[31] R. Collins, A. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, and O. Hasegawa. A system for video surveillance and monitoring. Technical Report CMU-RI-TR-00-12, May 2000. Robotics Institute, Carnegie Mellon University.

[32] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Proceeding of IEEE Conference Computer Vision and Pattern Recognition*, pages 142–149, 2000.

[33] I.J. Cox. An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(2):138–150, Febrary 1996.

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

[35] F. Daum. Nonlinear filters: beyond the Kalman filter. *IEEE Aerospace and Electronic Systems Magazine*, 20(8):57–69, Augest 2005.

[36] S. Deb, M. Yeddanapudy, K. Pattipati, and Y. Bar-Shalom. A generalized S-D assignment algorithm for multisensor-multitarget state estimation. *IEEE Aerospace and Electronic Systems Magazine*, 33(2):523–538, April 1997.

[37] P.M. Djuric and S.J. Godsill. Special issue on Monte Carlo methods for statistical siganl processing. *IEEE Transactions on Siganl processing*, 50(2):173–173, February 2002.

[38] A. Doucet. On sequential simulation-based methods for Bayesian filtering. Technical Report CUED/F-INFENG/TR 310, 1998. Cambridge University.

[39] A. Doucet, N. de Freitas, and N.J. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Series Statistics for Engineering and Information Science. New York: Springer-Verlag, May 2001.

[40] A. Doucet, N. de Freitas, K. Murphy, and S. Russell. Rao-blackwellised particle filtering for dynamic bayesian networks. In *Uncertainty in Artificial Intelligence (UAI)*, 2000.

[41] A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.

[42] D.J. Fleet and A.D. Jepson. Computation of component image velocity from local phase information. *International Journal of Computer Vision*, 5:77–104, 1990.

[43] T.E. Fortmann, Y. Bar-Shalom, and M. Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE Journal of Oceanic Engineering*, 8:173–184, 1983.

[44] D.M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73(1):82–98, January 1999.

[45] A. Gelb, editor. *Applied optimal estimation*. MIT Press, 1974.

[46] W.R. Gilks and C. Berzuini. Following a moving target Monte Carlo inference for dynamic bayesian models. *J. R. Statist. Soc. B*, 63:127–146, 2001.

[47] W.R. Gilks, S. Richardson, and D.J. Spiegelhalter, editors. *Markov chain Monte Carlo in practice*. Chapman and Hall, 1996.

[48] S.J. Godsill and J. Vermaak. Models and algorithms for tracking using trans-dimensional sequential Monte Carlo. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2004.

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

[50] N.J. Gordon. A hybrid bootstrap filter for target tracking in clutter. *IEEE Transactions on Aerospace and Electronic Systems*, 33:353–358, April 1997.

[51] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-gaussian Bayesian state estimation. *Proceedings IEE. F*, 140(2):107–113, 1993.

[52] P.J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

[53] P.J. Green. Trans-dimensional Markov chain Monte Carlo. In P.J. Green, N.L. Hjort, and S. Richardson, editors, *Highly Structured Stochastic Systems*, Oxford Statistical Science Series (0-19-961199-8). Oxford University press, 2003.

[54] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P-J. Nordlund. Particle filters for positioning, navigation and tracking. *IEEE Transaction on Signal Processing*, 50(2):425–437, February 2002.

[55] C. D. Haworth, Y. De Saint-Pern, and et al. D. Clark. Detection and tracking of multiple metallic objects in millimetre-wave images. *International Journal of Computer Vision*, 71(2):183–196, Feburary 2007.

[56] S. Haykin and N. de Freitas. Special issue on sequential state estimation. *Proceedings of the IEEE*, 92(3):399–400, March 2004.

[57] S.M. Hermanr. *A particle filtering approach to joint passive radar tracking and target classification*. PhD thesis, Department of Electrical Engineering, University of Illinois at Urbana-Champaign, 2002.

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

[59] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

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

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

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

[63] M. Isard and A. Blake. Icondensation: Unifying low-level and high-level tracking in a stochastic framework. In *Proceedings of 5th European Conference Computer Vision*, volume 1, pages 893–908, 1998.

[64] M. Isard and J. MacCormick. BraMBLe: A Bayesian multiple-blob tracker. In *Proc. Int. Conf. Computer Vision*, volume 2, pages 34–41, 2001.

[65] A.H. Jazwinski. *Stochastic Processes and Filtering Theory*. New York: Academic, 1970.

[66] A.M. Johansen, S.S. Singh, A. Doucet, and B.-N. Vo. Convergence of the SMC implementation of the PHD filteter. *Methology and Computing in Applied Probability*, 8(2):265–291, June 2006.

[67] S.J. Julier and J.K. Uhlmann. A new extension of the Kalman filter to nonlinear systems. In *Proc. AeroSense: 11th Int. Symp. Aerospace/Defense Sensing, Simulation and Controls*, pages 182–193, 1997.

[68] R.E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[69] R. Karlsson. *Simulation Based Methods for Target Tracking*. PhD thesis, Department of Electrical Engineering, Linkoping University, Sweden, 2002.

[70] V. Kettnaker and R. Zabih. Bayesian multi-camera sureveillance. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 253–259, 1999.

[71] S. Khan and M. Shah. Consistent labeling of tracked objects in multiple cameras with overlapping fields of view. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1355–1360, October 2003.

[72] Z. Khan, T. Balch, and F. Dellaert. MCMC-based particle filtering for tracking a variable number of interacting targets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1805–1819, Nobember 2005.

[73] T. Kirubarajan, Y. Bar-Shalom, K.R. Pattipati, and I. Kadar. Ground target tracking with topography-based variable structure IMM estimator. *IEEE Transactions on Aerospace and Electronic Systems*, 36(1):26–46, January 2000.

[74] G. Kitagawa. Monte Carlo filter and smoother for non-gaussian nonlinear state space models. *J. Comput. Graph. Statist.*, 5(1):1C–25, 1996.

[75] A. Kong and W.H. Wong J.S. Liu. Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association*, 89(425):278–288, March 1994.

[76] J.H. Kotecha and P.M. Djurid. Gaussian particle filtering. *IEEE Transactions on Signal Processing*, 51(10):2592–2601, October 2003.

[77] J.H. Kotecha and P.M. Djurid. Gaussian sum particle filtering. *IEEE Transactions on Signal Processing*, 51(10):2602–2612, October 2003.

[78] J. Krumm, S. Harris, B. Myers, B. Brummit, M. Hale, and S. Shafer. Multi-camera multi-person tracking for easy living. In *Third IEEE Workshop on Visual Surveillance*, 2000.

[79] L. Li, W. Huang, I. Y.-H. Gu, and Q. Tian. Statistical modeling of complex backgrounds for foreground object detection. *IEEE Transactions on Image Processing*, 13(11):1459–1472, 2004.

[80] J.S. Liu and R. Chen. Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93:1032–1044, 1998.

[81] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[82] R. Mahler. Bayesian vs. "plain-vanilla bayesian" multitarget statistics. In I. Kadar, editor, *Sign. Proc., Sensor Fusion, and Targ. Recog XIII, SPIE*, volume 5429, pages 1–12, Bellingham WA, 2004.

[83] R.P.S. Mahler. An introduction to multisource-multitarget statistics and its applications. Technical monograph, Regan MN, 2000. Lockheed Martin.

[84] R.P.S. Mahler. Random set theory for target tracking and identification. In D.L. Hall and J. Llinas, editors, *Handbook of Multisensor Data Fusion*. CRC Press, Boca Raton FL, 2002.

[85] R.P.S. Mahler. Multitarget Bayes filtering via first-order multitarget moments. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1152–1178, 2003.

[86] R.P.S. Mahler. Random sets: Unification and computation for information fusion—a retrospective assessment. In *Proceedings of Internation Conference on Information Fusion*, pages 1–20, 2004.

[87] A.-R. Mansouri. Region tracking via level set PDEs without motion computation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):947–961, July 2002.

[88] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan. Interacting multiple model methods in target tracking: A survey. *IEEE Transactions on Aerospace and Electronic Systems*, 34(1):103–123, January 1998.

[89] M.I. Miller, A. Srivastava, and U. Grenander. Conditional-mean estimation via jump-diffusion processes in multiple target tracking/recognition. *IEEE Transactions on Signal Processing*, 43:2678–2690, 1995.

[90] A. Mittal and L.S. Davis. M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene using region-based stereo. In *European Conference on Computer Vision (1)*, 2002.

[91] T.B. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding: CVIU*, 81(3):231–268, 2001.

[92] M.R. Morelande and S. Challa. Manoeuvring target tracking in clutter using particle filters. *IEEE Transactions on Aerospace and Electronic Systems*, 41(1):252–270, January 2005.

[93] S. Mori and C.-Y. Chong. Point process formalism for multiple target tracking. In *Procceedings of International Conference on Information Fusion*, pages 10–17, 2003.

[94] K.G. Murty. An algorithm for ranking all the assignments in order of increasing cost. *Operations Research*, 16:682–687, 1968.

[95] D. Musicki, R. Evans, and S. Stankovic. Integrated probabilistic data association. *IEEE Transaction Automatic Control*, 39(6):1237–1241, 1994.

[96] K. Nummiaro, E. Koller-Meier, and L. V. Gool. An adaptive color-based particle filter. *Image and Vision computing*, 21(1):99–110, 2003.

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

[98] K. Okuma, A. Taleghani, N. de Freitas, J.J. Little, and D.G. Lowe. A boosted particle filter: Multitarget detection and tracking. In *Proceedings of 8th European Conference Computer Vision*, pages 28–39, 2004.

[99] N. Paragios and R. Deriche. Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(3):266–280, March 2000.

[100] H. Pasula, S. Russell, M. Ostland, and Y. Ritov. Tracking many objects with many sensors. In *Proceedings of International Joint Conference Artificial Intelligence*, Stockholm, 1999.

[101] K.R. Pattipati, S. Deb, Y. Bar-Shalom, and Jr. R.B. Washburn. A new relaxation algorithm and passive sensor data association. *IEEE Transactions on Automatic Control*, 37(2):198–213, February 1992.

[102] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In *Proceedings of 7th European Conference Computer Vision*, pages 661 – 675, 2002.

[103] P. Pérez, J. Vermaak, and A. Blake. Data fusion for visual tracking with particles. *Proceedings of the IEEE*, 92(3):495–513, 2004.

[104] M. Pitt and N. Shephard. Filtering via simulation: Auxiliary particle filters. *J. Amer. Statist. Assoc.*, 94(446):590–599, 1999.

[105] G.W. Pulfold. Taxonomy of multiple target tracking methods. *IEE Proc. Radar, Sonar, Navig.*, 152(5):291–304, 2005.

[106] C. Regazzoni, V. Ramesh, and G.L. Foresti. Special issue on video communications, processing, and understanding for third generation surveillance systems. *Proceedings of the IEEE*, 89(10):1355–1367, October 2001.

[107] D.B. Reid. An algorithm for tracking multiple targets. *IEEE Transaction on Automatic Control*, 24(6):843–854, December 1979.

[108] B. Rosenhahn, T. Brox, and J. Weickert. Three-dimensional shape knowledge for joint image segmentation and pose tracking. *International Journal of Computer Vision*, 73(3):243–262, July 2007.

[109] Y. Rui and Y. Chen. Better proposal distributions: Object tracking using unscented particle filter. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 786–793, December 2001.

[110] L. Ryder. *Quantum Field Theory*. Cambridge University Press, Cambridge UK, 2nd edition, 1996.

[111] J. Shi and C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[112] H. Sidenbladh. Multi-target particle filtering for the probability hypothesis density. In *Proceedings of Internation Conference on Information Fusion*, pages 800–806, Cairns, Australia, 2003.

[113] H. Sidenbladh and S.-L. Wirkander. Tracking random sets of vehicles in terrain. In *IEEE Workshop on Multi-Object Tracking*, Madison, WI, USA, 2003.

[114] K. Smith, D. Gatica-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.

[115] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, June 1999.

[116] D. Stoyan, D. Kendall, and J. Mecke. *Stochastic Geometry and its Applications*. John Wiley and Sons, 1995.

[117] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust Monte Carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2):99–141, 2000.

[118] M. Tobias and A.D. Lanterman. Probability hypothesis density-based multitarget tracking with bistatic range and doppler. *IEE Proc. Radar, Sonar, Navig.*, 152(3):195–205, 2005.

[119] C. Tomasi and Takeo Kanade. Detection and tracking of point features. Technical report, April 1991. Carnegie Mellon University CMU-CS-91-132.

[120] R. van der Merwe, A. Doucet, N. de Freitas, and E. Wan. The unscented particle filter. Technical Report CUED/F-INFENG/TR-380, Augest 2000. Cambridge University Engineering Department, Cambridge.

[121] J. Vermaak, A. Doucet, and P. Perez. Maintaining multi-modality through mixture tracking. In *Proceedings of 9th International Conference on Computer Vision*, 2003.

[122] J. Vermaak, S. Maskell, and M. Briers. Tracking a variablenumber of targets using the existence joint probabilistic data association filter. Technical Report CUED/F-INFENG/TR.514, January 2005. Cambridge University Engineering Department, Cambridge.

[123] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

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

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

[126] L. Wang, W. Hu, and T. Tan. Recent developments in human motion analysis. *Pattern Recognition*, 36:585–601, 2002.

[127] G. Welch and G. Bishop. http://www.cs.unc.edu/ welch/kalman/. Some tutorials, references, and research on the Kalman filter.

[128] C. Wren, A. Azarbayejani, T. Darell, and A. Pentland. Pfinder: real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, July 1997.

[129] M. Yeddanapudi, Y. Bar-Shalom, and K.R. Pattipati. IMM estimation for multitarget-multisensor air traffic surveillance. *Proceedings of the IEEE*, 85(1):80–94, Januany 1997.

[130] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Computing Surveys*, 38(4), 2006.

[131] T. Zajic and R. Mahler. A particle-systems implementation of the PHD multitarget tracking filter. In *SPIE Signal Processing, Sensor Fusion and Target Recognition*, volume 5096, pages 291–299, 2003.

[132] D. Zhong and S.-F. Chang. Long-term moving object segmentation and tracking using spatiotemporal consistency. In *Proceedings of IEEE International Conference on Image Processing*, volume 2, pages 57–60, 2001.

[133] X.S. Zhou, D. Comaniciu, and A. Gupta. An information fusion framework for robust shape tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(1):115–129, January 2005.