Enhanced Particle PHD Filtering for Multiple Human Tracking

by

Pengming Feng

A doctoral thesis submitted in partial fulfilment of the requirements for the award of the degree of Doctor of Philosophy (PhD), from Newcastle University.

September 2016

Communications, Sensors, Signal and Information Processing Research Group (ComS^{2}IP),
School of Electrical and Electronic Engineering,
Newcastle University,
Newcastle upon Tyne, UK, NE1 7RU.

CERTIFICATE OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this thesis, that the original work is my own except as specified in acknowledgements or in footnotes, and that neither the thesis nor the original work contained therein has been submitted to this or any other institution for a degree.

..... (Signed)

..... (candidate)

I dedicate this thesis to my loving family.

Abstract

Video-based single human tracking has found wide application but multiple human tracking is more challenging and enhanced processing techniques are required to estimate the positions and number of targets in each frame. In this thesis, the particle probability hypothesis density (PHD) filter is therefore the focus due to its ability to estimate both localization and cardinality information related to multiple human targets. To improve the tracking performance of the particle PHD filter, a number of enhancements are proposed.

The Student's-t distribution is employed within the state and measurement models of the PHD filter to replace the Gaussian distribution because of its heavier tails, and thereby better predict particles with larger amplitudes. Moreover, the variational Bayesian approach is utilized to estimate the relationship between the measurement noise covariance matrix and the state model, and a joint multi-dimensioned Student's-t distribution is exploited.

In order to obtain more observable measurements, a backward retrodiction step is employed to increase the measurement set, building upon the concept of a smoothing algorithm. To make further improvement, an adaptive step is used to combine the forward filtering and backward retrodiction filtering operations through the similarities of measurements achieved over discrete time. As such, the errors in the delayed measurements generated by false alarms and environment noise are avoided.

In the final work, information describing human behaviour is employed

to aid particle sampling in the prediction step of the particle PHD filter, which is captured in a social force model. A novel social force model is proposed based on the exponential function. Furthermore, a Markov Chain Monte Carlo (MCMC) step is utilized to resample the predicted particles, and the acceptance ratio is calculated by the results from the social force model to achieve more robust prediction. Then, a one class support vector machine (OCSVM) is applied in the measurement model of the PHD filter, trained on human features, to mitigate noise from the environment and to achieve better tracking performance.

The proposed improvements of the particle PHD filters are evaluated with benchmark datasets such as the CAVIAR, PETS2009 and TUD datasets and assessed with quantitative and global evaluation measures, and are compared with state-of-the-art techniques to confirm the improvement of multiple human tracking performance.

Contents

Statement of Originality

The contributions of this thesis are mainly on the improvement of the particle probability hypothesis density (PHD) filter for multiple human tracking. The novelty of the contributions is supported by the following international journal and conference papers.

In Chapter 4, the Student's-t distribution is employed, which is firstly used to replace the Gaussian distribution in the prediction and the measurement models of traditional particle PHD filter; then it is applied as a joint distribution between the measurement noise covariance matrix and the state model, where the variational Bayesian approach is employed to estimate the best parameters for the joint distribution. These two contributions have been published in:

1. P. Feng, M. Yu, S. M. Naqvi, W. Wang and J. A. Chambers, 'A robust Student's-t distribution PHD filter with OCSVM updating for multiple human tracking', in *Proc. European Signal Processing Conference* (*EUSIPCO*), pp. 2396-2400, 2015.

2. P. Feng, W. Wang, S. M. Naqvi and J. A. Chambers, 'Variational Bayesian PHD filter with deep learning network updating for multiple human tracking', in *Proc. Sensor Signal Processing for Defence* (*SSPD*), pp. 1-5, 2015.

In Chapter 5, based upon the concept of the backward smoothing algorithm, the delayed measurements are employed to achieve more observation measurements; by considering the non-Gaussian and non-linear property of the particle PHD filter, the term 'Retrodiction PHD filter' is used instead of 'PHD smoother'. Moreover, an adaptive approach is proposed to combine the forward and retrodiction steps adaptively according to the similarities between the measurement set over discrete time, which helps to improve the tracking performance. The results of these two solutions are included in:

3. P. Feng, W. Wang, S. M. Naqvi and J. A. Chambers, 'Adaptive Retro-PHD filter for multiple human tracking', *IEEE Signal Processing Letters*, vol. 23, no. 11, pp. 1592-1596, 2016.

In Chapter 6, a novel exponential term-based social force model is employed to describe the human behaviour, which aids the particle sampling. Then a Markov Chain Monte Carlo (MCMC) resampling step is applied to obtain a more robust predicted particle set, where the acceptance ratio is calculated with the aid of the social force model. In the updating step of the particle PHD filter, an OCSVM classifier trained by human features is utilized to mitigate the noise from the environment, which further improves the tracking accuracy. The results of this scheme are presented in:

4. P. Feng, W. Wang, S. M. Naqvi, S. Dlay and J. A. Chambers, 'Social force model aided robust particle PHD filter for multiple human tracking', in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP*), pp. 4398-4402, 2016.

5. P. Feng, W. Wang, S. M. Naqvi, and J. A. Chambers, 'Social force model aided MCMC-OCSVM particle PHD filter for multiple human tracking', accepted subject to minor alterations in *IEEE Transactions on Multimedia*, 2016.

Acknowledgements

I am deeply and sincerely indebted to my supervisor Professor Jonathon Chambers for his consistent instruction, constant support and generous advice throughout the past three years, from Loughborough to Surrey and to Newcastle. I have benefited tremendously from his rare insight, his exceptional knowledge and his great enthusiasm with patient mentoring to students. It is difficult for me to accomplish this thesis without his tireless instruction and encouragement. It is my great privilege and exclusive honour to have been one of his research students. I wish that I will have more opportunities to work with him in the future.

I am also extremely thankful to Dr. Syed Mohsen Naqvi and Dr. Wenwu Wang, who helped me to enter my research field and kept giving me advice and encouraging me during my Phd period; due to their kindly help, I have learnt much from their instruction.

I would also like to give my appreciation to my friends and colleagues Miao Yu, Yangfeng Liang, Gaojie Chen, Waqas Rafique, Zeyu Fu, Zehan Liu and Yang Sun for their great help and providing a stable and cooperative environment during the past four years in the United Kingdom.

Lastly, but most importantly, I wish to express my deepest gratitude and love to my families, in particularly my loving parents. I really can not find appropriate words to represent my heartfelt thanks to them for their constant encouragement, prayers, attention and support in innumerable ways throughout my PhD and before. I would like to dedicate this thesis to my entire loving family.

Pengming Feng

September, 2016

List of Acronyms

EKF	Extended Kalman Filter
HOG	Histogram of Oriented Gradient
KF	Kalman Filter
KL	Kullback-Leibler
MCMC	Markov Chain Monte Carlo
MEE	Mean Euclidean Error
MHT	Multiple Hypothesis Tracking
MMSE	Minimum Mean Square Error
MOTP	Multiple Object Tracking Precision
MSE	Mean Square Error
OCSVM	One Class Support Vector Machine
OSPA	Optimal Sub-Pattern Assignment
PF	Particle Filter
PDF	Probability Density Function
PHD	Probability Hypothesis Density
Retro	Retrodiction

RFS	Random Finite Set
SD	Standard Deviation
SFM	Social Force Model
SIFT	Scale-Invariant Feature Transform
TLD	Tracking Learning Detection
VB	Variational Bayesian

List of Symbols

$\hat{\cdot}$	(Estimated value)
$\mathbf{f}$	State transition function
$\mathbf{F}$	Gaussian and linear state transition function
$\mathbf{h}$	Measurement transition function
$\mathbf{H}$	Gaussian and linear measurement transition function
$\mathbf{x}$	State vector
$\mathbf{z}$	Measurement vector
δ	Delta function
$\|\|.\|\|$	Euclidean distance
$(.)^T$	Transpose operator
μ	Mean value
p	Probability distribution
D	Probability density
N_{eff}	Effective sample size
$\sum$	Summation
Δt	Time interval

$\mathcal{N}$	Gaussian distribution
min	Minimum value
ϕ	Analogue of state transition probability of PHD filter
Υ	Intensity function of new target birth RFS
Γ	Gamma function
v	Degree of freedom parameter of Student's-t distribution
$\forall$	For all
$\in$	Belonging to
Ω	Importance density for the Student's-t distribution
N	Number of particles
M	Number of targets
KL	KL divergence
L	Number of iteration
E	Expectation
exp	Exponential
λ	Convex parameter
ℓ	Time lag

List of Figures

List of Tables

Chapter 1

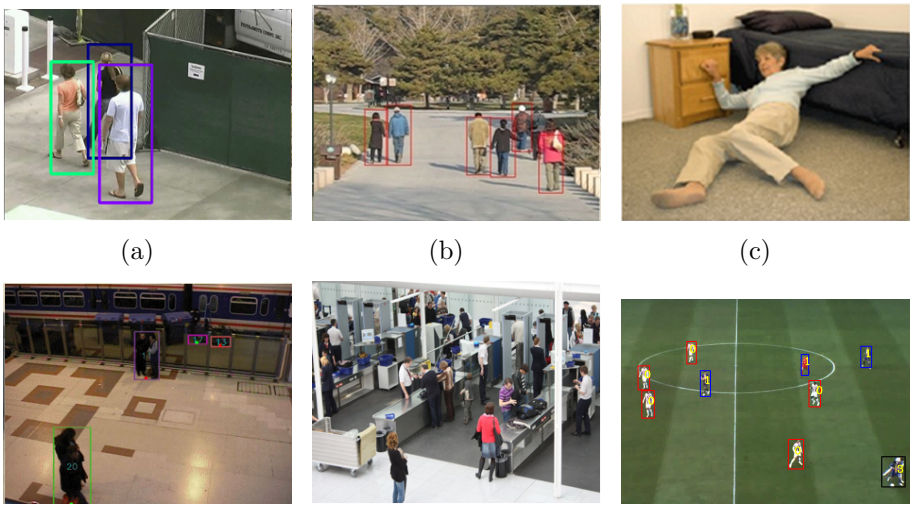
INTRODUCTION

1.1 Video tracking

"The process of estimating over time the location of one or more objects using a camera as a sensor is referred to as *video tracking*" - Emilio Maggio [3].

With the accelerated development of microelectronics and video analysis algorithms, capturing video and obtaining desired information from video frames is becoming increasingly easy. Such technology represents a step towards machines that have the ability to see and understand their environment in an intelligent way automatically. One of the fundamental challenges for intelligent machines to achieve this goal is to detect and track the objects of interest, namely perform video tracking. During the past decade, the creation of new algorithms and applications for video tracking systems has been accelerated by the rapid increase in computational power and the dramatic improvement in both quality and resolution of imaging sensors such as cameras, which have been applied in many areas, such as surveillance, security, medical imaging, robotics, human identification, human-machine interaction and assisted living in recent life [7, 8]. Some application contexts of video tracking are represented in Fig. 1.1.

All these example applications require tracking of multiple objects, which implies correctly detecting, identifying and finding the locations of desired targets with the help of some noisy measurements acquired through sensors. Besides the camera, other sensors have also been employed for track-



(a) (b) (c)

(d) (e) (f)

Figure 1.1. Example video frames for different video tracking application contexts, where (a) is tracking in an indoor environment; (b) is from an outdoor monitoring system; (c) is as in an assisted living system; (d) and (e) are for security in a train station and airport environment respectively and (f) is for competitive analysis in a football match.

ing applications. For example, depth sensors such as Kinect have been used for human-machine interaction, particularly in video applications [9]; wired or wireless sensors, for example, laser based tracking systems have been developed in the area of human surveillance and medical imaging applications [10, 11]; and audio signal based tracking can be used in human-robot interaction techniques to improve the accuracy of interaction systems [12]. However, in this thesis, tracking based upon video sequences is employed because of its universal applicability and the rich information content of video sequences, which can be used for tracking and identifying.

Most recent video-based multiple human tracking developments fall into one of three categories: achieving a more accurate dynamic model for prediction such as using an interaction model when predicting the position and velocity of a target [5, 6]; generating more stable recursive mathematical models such as unscented Kalman filters and Markov Chain Monte Carlo

(MCMC) particle filters; and searching for more accurate measurement models [13,14], for example, the tracking-learning-detection (TLD) [15] approach. The almost universally accepted mathematical framework used to describe multiple target tracking is that of filtering theory and, in particular, Bayesian filtering [16]. Instead of tracking the exact location of multiple targets, the Bayesian approach relies on estimating the probability distribution of the state, which contains the location, velocity and size information of the target. In particular, the posterior probability distribution is recursively predicted by the knowledge of the state model in the prediction stage, which describes the motion of a target as a prior distribution, after which, the predictions are corrected when a new observation becomes available, namely the updating stage [17]. In the case of video tracking, the pixels of a video frame are employed to help update the predictions.

By employing Bayesian filtering video tracking methods, the tracker is required to have the following capacities, namely to be able to:

- Estimate the location of each human target, which means the tracker should output the location of each target at each time frame.
- Update the dynamic model in order to predict the target states, which is used as the prior knowledge in the Bayesian filtering framework.
- Exploit the measurements of human targets in the processing, so that corrections to the prediction results can be performed.
- Determine the number of targets at each time frame as human targets may appear and disappear in the scenario; moreover, occlusions may also cause a varying number of targets.

To achieve the above requirements of a human tracking algorithm, particular consideration should be given to: the non-rigid nature of the human body and its non-linear movement and that when employing the Bayesian

filtering framework, the tracking performance depends on the accuracy of the measurement set, in particular, the environmental noise in the measurement can cause failure of the tracker. To give solutions for the above requirements, the aims and objectives of this thesis are listed in the next section.

In order to address the above challenges, particularly the problem of varying number of targets, the random finite set (RFS) [18] concept and the probability hypothesis density (PHD) filter which has been recently proposed [19] will be exploited. Both the cardinality of the targets and their states are estimated within the PHD filter, and thus it avoids the need for data association techniques as part of the multiple target tracking framework [20, 21]. Moreover, it mitigates the computational complexity issue which often occurs in other multiple target tracking approaches such as multiple hypothesis tracking (MHT) [3] since it simply utilizes the first-order moment of the multi-target posterior rather than the posterior itself [22].

Although the particle PHD filter has mainly been employed in multiple human tracking, there are several weaknesses within the particle PHD framework which can be improved. First of all, in the traditional PHD filter the prediction and updating model are assumed to be Gaussian distributed and are independent from each other, which can be improved by exploiting heavier-tailed distributions and combining the prediction model and updating model with a joint distribution. Secondly, by employing a backward step in the particle PHD filter, more observable measurements can be obtained which can improve the accuracy of the tracking system. Thirdly, the prediction model for multiple human tracking can be improved by utilizing human behaviour. Moreover, the measurement model can be made more robust to environment noise.

1.2 Aims and objectives

The overall aims of this thesis are to overcome the aforementioned weaknesses of the particle PHD filter and enhance its tracking performance. The particular objectives are:

- Objective 1: Improving tracking performance by replacing the Gaussian distribution in the traditional PHD filter by a Student's-t distribution

In Chapter 4, the Student's-t distribution is employed in the prediction and measurement models of the particle PHD filter, which can yield larger amplitude particles because of its heavier tails compared with the Gaussian distribution.

- Objective 2: Improving tracking performance by applying a joint distribution

In Chapter 4, the Student's-t distribution is also employed as a joint distribution between the covariance matrix of the measurement noise and the state model, where a variational Bayesian step is used to estimate the best solution for the measurement noise covariance.

- Objective 3: Improving tracking performance by applying the delayed measurement adaptively

In Chapter 5, a retrodiction step is used to describe the backward smoother-like step for the particle PHD filter, where the delayed measurement is employed in the particle PHD filter to obtain more observable measurements, hence the tracking performance is improved. Moreover, after the retrodiction step, the forward and backward results are combined adaptively by employing an adaptive parameter calculated by the similarity of the forward and backward measurements, so further improvement is obtained.

- Objective 4: Improving tracking performance by employing the human behaviour in the prediction model

In Chapter 6, the social force model is employed to describe the human behaviour in the tracking scenario, which is also used to calculate the likelihood for the MCMC resampling step, after which, the predicted particles are resampled by the MCMC, hence obtaining a more reliable prediction model for the particle PHD filter.

- Objective 5: Improving tracking performance by applying an OCSVM classifier in the updating step of the particle PHD filter

The measurement set can be affected by the environment noise in multiple human tracking, therefore, in Chapter 6, the OCSVM classifier is applied, which is trained by human features including histograms of colour and oriented gradient. In this case, the environment noise in the measurement set is mitigated and better tracking performance ensues.

1.3 Thesis outline

The outline of this thesis is listed as follows:

Chapter 2 includes a relevant literature review of visual multiple human tracking work, where different methods for multiple human tracking are given. Moreover, different challenges associated with multiple human tracking work are described, including varying number of targets, occlusions, environment noises and state estimation. Different tracking algorithms are discussed to address these challenges. Then multiple human tracking methods are categorized according to different aspects, where the application of different categories as well as their advantages and disadvantages are given.

Chapter 3 focuses on the background preliminaries of visual based multiple human tracking. In this thesis, the Bayesian filtering algorithm is used

as the fundamental tracking framework, which mainly contains prediction and updating processes. Based upon the concept of the Bayesian filtering algorithm, two commonly used filters, namely the Kalman filter and the particle filter are introduced in this chapter. In order to address the challenge of varying number of targets, the PHD filter is employed and is also described in detail in this chapter. At the end of the chapter, the datasets used for evaluation in this thesis and the evaluation measures are given.

Chapter 4 satisfies the first and second objectives of this thesis, the Student's-t distribution is firstly employed to replace the Gaussian distribution in the prediction and updating stages due to its heavier tails. In order to achieve further improvement, the Student's-t distribution is used as a joint distribution between the covariance matrix of measurement noise and the state model, where a variational Bayesian approach is used to estimate the best solution of the measurement noise, hence the tracking accuracy is improved.

Chapter 5 employs an adaptive Retro-PHD filter to improve the tracking performance. The smoothing algorithm is first explained, based on the concept of the backward smoothing method, then a backward method for a non-linear non-Gaussian particle PHD filter is proposed, namely the Retro-PHD filter, where the delayed measurement is employed to obtain more observable measurements. Moreover, after obtaining the retrodiction filtering results, an adaptive step is employed, where the adaptive weight is calculated by the similarity of the measurement achieved to correct the tracking results from both forward filtering and backward retrodiction steps, and hence can improve the tracking performance.

In Chapter 6, the tracking performance is improved by enhancing both the prediction and updating steps of the particle PHD filter. The social force model is employed to describe the human behaviour which aids the prediction of the state model; moreover, an MCMC resampling step is employed to

resample the predicted particles with the aid of results from the social force model, in this way, a more accurate particle set is obtained from prediction. In the updating step, an OCSVM classifier is employed to calculate the likelihood for the particles, which is trained by the histograms of colour and oriented gradient of human features to mitigate the environment noise from background subtraction. Evaluation comparisons confirm the improvement of the proposed tracking method over other state-of-the-art techniques.

Finally, conclusions are drawn, and future work is then discussed in Chapter 7.

Chapter 2

RELEVANT LITERATURE REVIEW

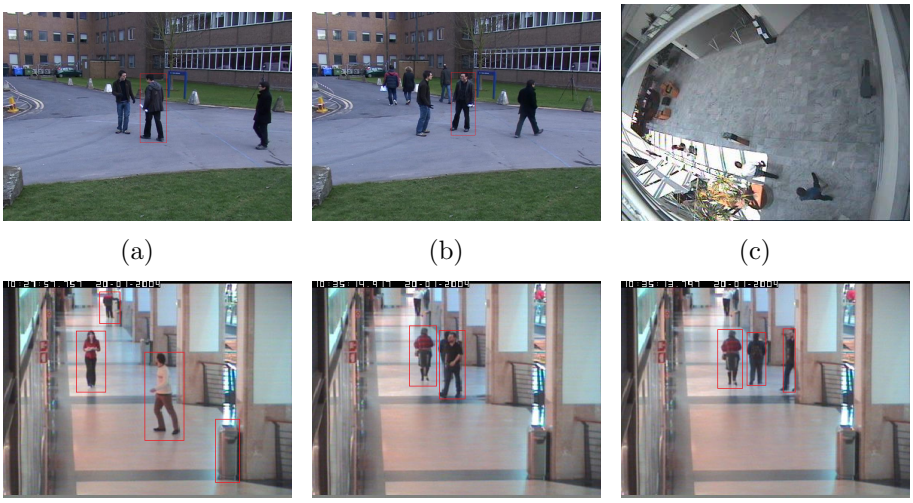
2.1 Introduction

In video-based multiple human tracking work, a video tracker estimates the target location by capturing target information in the form of sets of image pixels and modelling the relationship between the appearance of the target and its corresponding pixel values [3]. Besides the position of the targets, such relationship between the human targets and their image projections also depends on other factors, such as the velocity and size information of the targets, which makes multiple human tracking a difficult task. Different tracking methods have been proposed to give the solutions for multiple human tracking work in the literature. In this chapter, the challenges of video-based multiple human tracking work are discussed firstly, followed by different relevant solutions for each of them, then the components and different designs of video trackers are reviewed.

2.2 Challenges associated with multiple human tracking

To design a video-based multiple human tracker, the challenges are mainly related to four aspects [3]: the varying appearance of the human targets, which correspond to the changes of pose, velocity, ambient illumination and

colour information of targets; the similarity in appearance between the target and other objects and background, which commonly results in clutter and noise in the processing context; varying number of targets over time; and occlusions, which can be classified as partial occlusions and full occlusions. Fig. 2.1 represents some example challenges within video-based multiple human tracking work.



(a) (b) (c)

(d) (e) (f)

Figure 2.1. Example video frames for challenging cases in video-based multiple human tracking, where in (a) and (b) a target changed its pose and therefore the appearance is changed in the camera; the ambient illumination changes within the frame which cause the failure of detection for one target in (c); (d) is an example of the clutter challenge, where the bin in the frame shares a similar colour with the target and therefore distracts the tracker from the desired target of interest; (e) and (f) are examples for full and partial occlusion respectively, which may cause missed detections for the tracker. Figures (a) and (b) are sequences from the PETS2009 dataset [1] while others are from the CAVIAR dataset [2].

In the following subsections, the main challenges for video-based multiple human tracking work related to this thesis are studied, and their different relevant solutions are discussed.

2.2.1 Varying number of targets

Compared with the single target tracking work, multiple human tracking work has the particular challenge of varying number of targets, which can change with time, caused by new targets entering, or existing targets leaving the view of the video camera.

Many developments have been proposed recently to address the challenge of variable number of targets in video based multiple human tracking work. For instance, a reversible jump MCMC sampling technique is proposed in [23] to solve the varying number of targets problem, where the concept of Bayesian multiple-blob tracker [24] is employed to describe the set of identifiers of targets and hence provide the targets number. However, this method requires a strong assumption that the targets are restricted to enter and or leave from a very small region in the video frame. The method of combining the Bayesian filtering and clustering approaches is used in [25], which estimates the target number based on the number of detected clusters and produces multiple clusters per frame, but the accuracy of performance is limited since non-rigid human bodies may cause failure in detection. Based upon the concept of the random finite set (RFS) [18], the multiple hypothesis filter (MHT) [26] is proposed to track multiple targets with varying cardinality, with the basic idea to delay a decision regarding the assignment of an old trajectory to a new measurement, which is based upon the assumption that the targets generate independent measurements. The MHT method can also give a solution to data association, which is present in the video based multiple human tracking work framework, however, with the increasing number of targets, the computational complexity of the tracker grows exponentially, which is also a common problem for the tracking methods introduced above. In order to avoid this problem, based upon the concept of RFS, the PHD filter is proposed, which simply utilizes the first-order moment of the multi-target posterior distribution rather than the posterior itself [22] for multiple

target tracking, and is the focus of this thesis. Besides the PHD filter, the multi-Bernoulli filter [27] is also proposed for multiple human tracking, particularly to provide a solution for varying number of targets by employing the multi-Bernoulli recursion as a tractable approximation to the Bayes multiple target recursion under low clutter density scenarios [28, 29].

In this thesis, the PHD filtering algorithm is chosen because of its ability to cope with the challenge of varying number of targets in multiple human tracking, and also its relatively low computational complexity.

2.2.2 Occlusion

Besides the challenge of a varying number of targets, occlusion is also a potential problem for multiple human tracking, where human targets can be occluded by other targets or objects of no interest. Many researchers have given solutions to this challenge. One of the methods is to use multi-cameras in video-based tracking work as proposed in [30–32], however, exchanging of information is required between cameras which needs additional processing and results in the requirement of higher processing speed. Another method to overcome the occlusion challenge is to use 3-D calibration information, which is different from the 2-D tracking system which mainly depends on the appearance models of human targets. For example, in [33], a generative model is used to describe the shape, appearance and motion of the human body, and then use the 3-D articulated model is used for tracking. Apart from the above approaches, information from other sensor besides the video camera has also been considered for tracking. For instance, the audio localization methods [34–36] are widely used to aid video tracking because the audio generally remains unaffected during visual occlusions. However, the human targets are required to be speaking during the period when occlusion happens. Interaction models such as social force model [5, 6] are also employed to mitigate the influence from occlusion by introducing human

behaviour in the tracking systems. This solution works well when multiple targets do not occupy the same space, however, they fail to overcome the inter-target occlusions when targets crossover with other targets or objects.

2.2.3 Clutter

Clutter can also influence the tracking accuracy, which is caused by the similarities of objects in the background with the desired targets. The most commonly used method to address this problem by employing a classifier to distinguish the clutter and the targets. For example, A dictionary learning method for multiple human tracking is proposed in [37], where the dictionary is pre-trained by human features. Then it is used for the likelihood calculation for the desired targets; this method requires prior knowledge about the human targets. Another widely used method is to use different features to describe the human targets. For instance, besides the colour and gradient information, human poses can also be employed to better discriminate human targets from background information [38]; moreover, by employing an online classifier [39], the observation measurement can be updated overtime, hence the tracking system can further overcome the clutter problem with a more accurate measurement set, thus the improved tracking performance ensues.

In this section, the main challenges as well as the developments to address them has been discussed. In the next section, necessary components required to design a video tracker, which can also give solutions to the above challenges are introduced.

2.3 Main components and relevant solutions

As introduced in [3], the main goal to design a video tracker is to model the relationship between the appearance model and its corresponding pixel

values, in particular to address the challenges discussed in Section 2.2. Most video trackers contain the main logical components as: feature extraction, target representation, localisation and track management [3], which can be shown as Fig. 2.2.

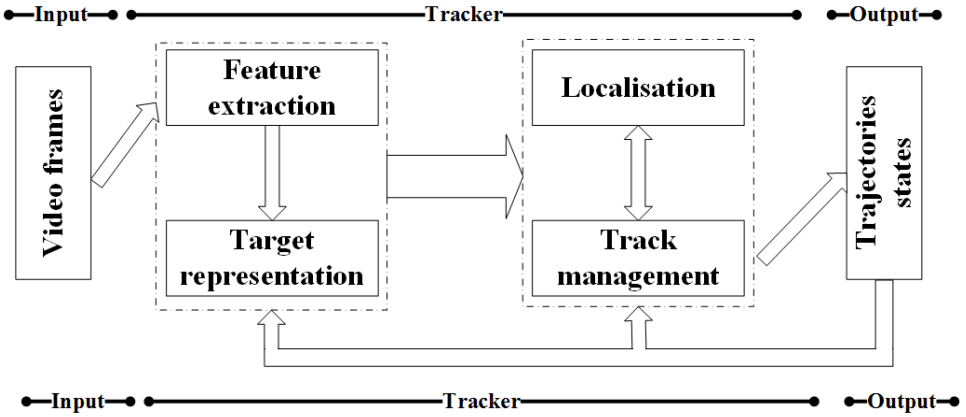


Figure 2.2. Common frame chart for video tracking, which shows the main logical components of video tracking systems [3].

In this section, following the main components described in Fig. 2.2, different developments to achieve the required components are given and discussed.

2.3.1 Feature extraction

Human targets from video sequences can be easily captured by human beings, however, for a machine, it is much more difficult to achieve this goal. Hence extracting relevant features for targets is important in video-based human tracking work, which can be used as observation in the measurement model. Many approaches have been developed for feature extraction. For example, low-level features such as colour [40] or gradient information are commonly used as human features, which can carry important information about the appearance of objects of interest. This method can be easily achieved by simple processing of the obtained video frames from the camera, however, the accuracy is limited since it can be affected by clutter and noise

from the background. Mid-level features such as interest points and edges have also been widely used as human features, such as the Scale-invariant Feature Transform (SIFT) [41] method. It can be used as a robust feature because it is detectable under changes in image scale, noise and illumination. Besides the low and mid level features, the feature extraction step can also be achieved by the detection of change, object classification and motion classification. For instance, the popular deep learning methods have been considered to be employed in the video-based tracking framework [42, 43], where with a hierarchical learning method is used to train the classification system, the natural progression structures trained from low level to high level of human features [44]. However, the deep learning methods require a large number of training datasets and the computational complexity is high due to the training of multiple layers in deep learning networks.

In this thesis, the main target is to improve the tracking algorithm to improve the performance in terms of tracking accuracy both in terms of localization error and determining the cardinality of targets, whilst reducing the computational complexity.

2.3.2 Target representation

The representation of targets gives the characteristics of the target to be used by the tracker, where the information of appearance and shape of targets can be included.

As mentioned in [3], the challenge of target representation is the trade-off between the invariance and the accuracy of description of human targets, so a robust representation method should be descriptive enough to have the ability to distinguish clutter and false alarms, while allowing a degree of flexibility to handle the change of pose, position, environment and partial occlusions.

For example, to represent human targets with shape information, three

kinds of models can be employed: basic models, articulated models and deformable models [3]. Basic models such as point approximation [45], area approximation [46] and volume approximation [47] use single points, bounding area and 3-D shape to represent human targets respectively; in particular, when applying the spatial volume approximation method, the challenge of occlusion can be handled with the estimation of the relative position with respect to the camera by giving camera calibration information. Besides the basic models, articulated models and deformable models are commonly employed [3], where the articulated models use topological connections and motion constraints of rigid models to represent targets, and deformable models employ fluid, contours and point distribution to represent targets and can solve the problem of limitation of prior information in the above two models.

Apart from the shape information, appearance representation methods can also be employed to describe the targets in human tracking [3], they are models of the expected projection of the target appearance onto the video frames. By employing the appearance model, a likelihood function is usually formulated to calculate the similarity of the appearance model and the particular target states. The advantage of using this likelihood function based method is its smoothness, which allows small variations in the target position and size.

2.3.3 Track management and localisation

Another important step in video based human tracking is track managing and localisation, which is the link between the target states and their measurement models. Methods to achieve this step can be separated into three categories: firstly, the trackers which rely on the detection of targets in every video frame, for example, methods described in [48, 49]; secondly, the trackers which use a state transition model to predict the targets states, then when the latest measurements are received, the predictions are updated and

the tracking results are obtained, as in the algorithms in [4, 23, 50]; and the third category is to use a basic detection technique to obtain the locations of targets [25]. Methods in the first category use detectors to generate initial tracking estimations, then targets over different frames are linked by employing further strategies such as an appearance model; however, these methods may fail in the situations when an occlusion happens; one way to overcome this challenge is to employ large amounts of training data to train the detector, which makes the tracker slow. The second category builds on the prediction and updating of target states, where the interaction models and robust state transition models can be employed within the tracking framework to obtain more robust tacking performance. Methods in the third category only depend on the results from detection, such as background subtraction, which can give solutions to close interactions, however, the tracker will fail because of clutter and occlusions.

The work in this thesis depends on the Bayesian filtering based tracking algorithm, which mainly falls into the second category. In the Bayesian filtering framework, the unknown posterior probability density function is estimated recursively over time using incoming measurements and a mathematical processing model. When compared with other tracking methods such as the tracking learning and detection (TLD) approach [15] and mean shift tracking method, the Bayesian tracking method attains advantages by employing the prior knowledge of the targets and recursively updating the posterior of the target distribution. When employed in tracking, Bayesian filters mainly contains two parts: the state model which is used for dynamic prediction and the measurement model, which is employed for making correction for target states [51]. Two fundamental tracking algorithms based upon the Bayesian filtering algorithm are the Kalman filter and particle filter; the Kalman filter has best performance when the models are Gaussian and linear. However, when the state and measurement models are non-Gaussian

and nonlinear, the extended Kalman filter [51] and the unscented Kalman filter can be employed to solve the problem caused by the non-Gaussian and nonlinearity. Moreover, the particle filter was also designed to give a solution for this problem [52]. However, although the Kalman filter and particle filter can perform well for single target tracking, however, they are not always able to associate measurements with particular targets for multiple human tracking. As introduced in Section 2.2.1, researchers have proposed many tracking methods for multiple target tracking, for example multiple hypothesis tracking (MHT) and joint probability data association (JPDA) tracking, but the computational complexity grows exponentially with increase in target number; in this case, the PHD filter, where only the first moment of the posterior probability density function rather than the whole posterior is employed, is chosen to be the basic tracking method in this thesis. Moreover, in order to cope with the non-Gaussian and nonlinear measurement model, the particle filtering based PHD filter is employed.

Improvement in the performance of Bayesian based tracking is mainly obtained from two aspects: enhancing from the state model or the measurement model. For the state model, approaches such as the dynamic appearance model, kinematic model or Langavine model [3] can be employed to describe the evolution of the states of human targets more accurately. Moreover, the interaction of human targets can also be employed to enhance the state model, for example, the social force model [5], which considers the interactions between human targets. Besides the state model, improvement from the measurement model can also improve tracking performance. In human tracking work, the main target of the measurement model is to represent the target from the prior information within the state model. As introduced in Section 2.3.2, different levels of features including ones from deep belief networks can be employed to aid the accuracy of the measurement model.

2.4 Summary

In this chapter, the challenges associated with video-based multiple human tracking along with the main components of the video tracking framework were presented and discussed. Firstly, different algorithms which can be employed to cope with the challenges as well as their advantages and disadvantages were studied. In the second part of this chapter, the main components to build the video tracker were presented, then different methods to obtain each component were briefly summarized. From the large number of algorithms in the literature reviewed in this chapter, the requirements of a robust video-based multiple human tracking can be summarised as to

- handle the varying number of targets and estimate the target number over time;
- cope with clutter and false alarms caused by the similarities of objects from background;
- give solutions to the failures caused by occlusions.

Therefore, the focus of this thesis is to achieve these abilities and achieve the above requirements.

In the next chapter, the background preliminaries of the tracking techniques which will be used in the thesis will be presented.

Chapter 3

BACKGROUND PRELIMINARIES

3.1 Introduction

In this chapter, the background material related to the problem of multiple human tracking investigated in this thesis will be introduced. Many researchers have proposed multiple human tracking methods in different areas, such as computer vision, video signal processing and in wearable sensor technology [3]. In this thesis, the target is to achieve a more robust multiple human tracking system by exploiting video signal processing. Section 3.2 describes the Bayesian approach for state estimation type tracking algorithms. Based on the fundamental Bayesian approach, two basic tracking methods, the Kalman filter and particle filter are given in Section 3.3. In Section 3.4 a more robust particle filtering algorithm, which is called the Markov chain Monte Carlo (MCMC) particle filter is explained. In Section 3.5, based upon the random finite set (RFS) concept, the particle PHD filter is explained. At the end of this chapter, evaluation sequences from three different datasets as well as the performance evaluation criteria employed in this thesis for making comparison between different tracking algorithms are introduced.

3.2 Bayesian estimation approach for tracking

In multiple human tracking, the primary objective is to estimate the states of targets. In this thesis, the Bayesian approach is employed to estimate the states of targets. The state for one target in video frame k is defined as $\mathbf{x}_k$, which includes its location $\{x, y\}$, its component velocities $\{v_x, v_y\}$ and its size $\{h, w\}$. The goal is to track the state of every target during the monitoring interval k, which is achieved with the aid of the measurement $\mathbf{z}_k$ obtained at a sensor. In this thesis, a video camera is used as the sensor for multiple human tracking. In this case, the measurement $\mathbf{z}_k$ is composed of features such as the gradient and colour histogram information captured from the frames measured by the video camera after background subtraction. In every video frame k, target state $\mathbf{x}_k$ is tracked with the aid of the measurement $\mathbf{z}_k$.

In the Bayesian filtering tracking model, the state estimation problem corresponds to estimating the probability distribution $p(\mathbf{x}_k|\mathbf{z}_k)$ with the aid of current and past measurement $\{\mathbf{z}_k, \mathbf{z}_{1:k-1}\}$ sequentially, where $\mathbf{z}_{1:k}$ denotes the measurements from frame 1 to k. The estimation includes two main stages: prediction and updating. In the prediction step, the target state $\hat{\mathbf{x}}_k$ is estimated with the prior knowledge of the previous state estimate $\mathbf{x}_{k-1}$ and the transition function $\mathbf{f}$, which can be described as [16, 53]

$$\hat{\mathbf{x}}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{w}_k) \tag{3.2.1}$$

where $\hat{\cdot}$ denote the estimated state value from the prediction and $\mathbf{w}_k$ is the prediction noise vector. After predicting the new state, the estimated states are updated with the aid of the measurement $\mathbf{z}_k$, which can be formulated as

$$\mathbf{z}_k = \mathbf{h}(\hat{\mathbf{x}}_\mathbf{k}, \mathbf{v}_k) \tag{3.2.2}$$

where $\mathbf{h}$ is the measurement transition function and $\mathbf{v}_k$ is the measurement noise vector. By assuming that the initial PDF $p(\mathbf{x}_0|\mathbf{z}_1) \equiv p(\mathbf{x}_0)$, then in the prediction stage of the Bayesian filtering framework, model (3.2.1) is invoked to obtain the prior PDF of the state at time k via the Chapman-Kolmogorov equation [16]

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})d\mathbf{x}_{k-1} \tag{3.2.3}$$

Then when a measurement $\mathbf{z}_k$ is available, the update step of the tracking system, namely calculating the posterior PDF, is obtain via Bayes' rule [16]

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1})}{\int p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1})d\mathbf{x}_k} \tag{3.2.4}$$

where $p(\mathbf{z}_k|\mathbf{x}_k)$ is the likelihood function, and from (3.2.4), the measurement $\mathbf{z}_k$ is used to modify the prior density to obtain the required posterior density of the current state.

A graphical representation of the Bayesian filtering model is described in Fig. 3.1 wherein the state is evolving from left to right.

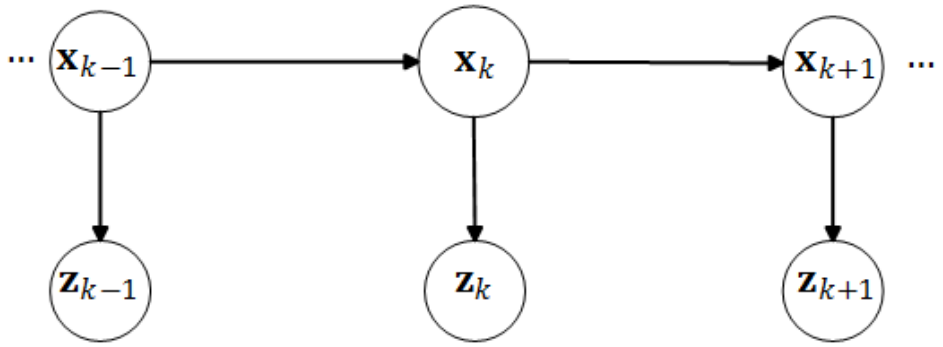


Figure 3.1. Graphical representation of the state and measurement evaluation within Bayesian filtering.

In the following section, based upon the concept of Bayesian filtering, two fundamental tracking methods, the Kalman filter and particle filter will be introduced.

3.3 Kalman and particle filtering

In the following two sections, Kalman and particle filtering are introduced.

3.3.1 Kalman filtering

The Kalman filter has been widely used by many researchers because of its optimal minimize mean square error (MMSE) performance within Gaussian and linear environments. Based upon the Bayesian filtering framework, the Kalman filter involves five main steps: prediction, calculating the minimum prediction mean square error (MSE) covariance matrix, calculating the Kalman gain matrix, correction and updating the MSE covariance matrix. The state and measurement models are assumed to be Gaussian and linear with the state and measurement transition matrices $\mathbf{F}$ and $\mathbf{H}$ respectively. By assuming that the noise vectors $\mathbf{v}_k$ and $\mathbf{w}_k$ in models (3.2.1) and (3.2.2) are Gaussian and are distributed as $\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{P}_k)$ and $\mathbf{w}_k \sim \mathcal{N}(0, \mathbf{R}_k)$, the distribution of the state can be defined as $\mathcal{N}(\mathbf{x}_k; \mathbf{m}_k, \mathbf{P}_k)$, which is a Gaussian density with argument $\mathbf{x}_k$, mean $\mathbf{m}_k$ and covariance $\mathbf{P}_k$, the Kalman filtering algorithm, which can be derived from the state and measurement models of the Bayesian filtering algorithm, can be described by the following recursive relationships [51]:

$$p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) = \mathcal{N}(\mathbf{x}_{k-1}; \mathbf{m}_{k-1}, \mathbf{P}_{k-1}) \tag{3.3.1}$$

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \mathcal{N}(\hat{\mathbf{x}}_k; \hat{\mathbf{m}}_k, \hat{\mathbf{P}}_k) \tag{3.3.2}$$

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \mathcal{N}(\mathbf{x}_k; \mathbf{m}_k, \mathbf{P}_k) \tag{3.3.3}$$

where

$$\hat{\mathbf{m}}_k = \mathbf{F}\mathbf{m}_{k-1} \tag{3.3.4}$$

$$\hat{\mathbf{P}}_k = \mathbf{Q}_{k-1} + \mathbf{F}\mathbf{P}_{k-1}\mathbf{F}^T \tag{3.3.5}$$

$$\mathbf{m}_k = \hat{\mathbf{m}}_k + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{m}}_k) \tag{3.3.6}$$

$$\mathbf{P}_k = \hat{\mathbf{P}}_k - \mathbf{K}_k\mathbf{H}\hat{\mathbf{P}}_k \tag{3.3.7}$$

and the Kalman Gain matrix $\mathbf{K}_k$ at time k is calculated as [51]

$$\mathbf{K}_k = \hat{\mathbf{P}}_k\mathbf{H}^T(\mathbf{H}\hat{\mathbf{P}}_k\mathbf{H}^T + \mathbf{R}_k)^{-1} \tag{3.3.8}$$

A graphical representation of the Kalman filtering model is given as Fig. 3.2 [51] The Kalman filter is widely employed in tracking work because it is

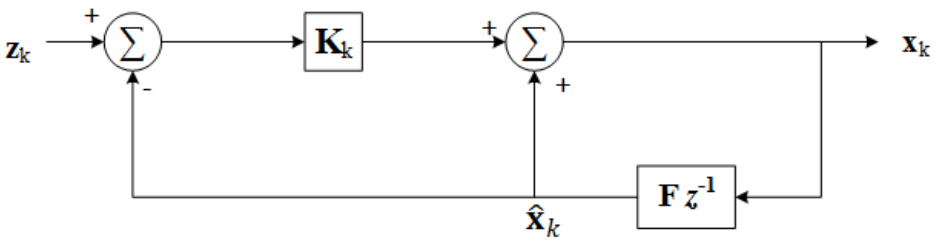


Figure 3.2. Graphical representation of Kalman filtering

optimal in the MMSE sense in the linear and Gaussian environment, moreover, with a non-linear model, the extended Kalman filter has also been proposed to use a local linearization of the equations as a sufficient description of the nonlinearity [51]. However, in a non-linear and non-Gaussian environment, (3.3.1) to (3.3.8) can not be derived accurately because of the approximation errors during the Jacobian step accumulating over sample times, hence in this thesis the particle filter is used as a fundamental filtering algorithm.

3.3.2 Particle filtering

Based on the concept of the sequential Monte Carlo algorithm, the key idea of the particle filter is to represent the required posterior density function $p(\mathbf{x}_k|\mathbf{z}_k)$ by a set of random samples $\{\mathbf{x}_k^i, w_k^i\}_{i=1}^N$ with associated weights

and to compute estimates based on these samples and weights, where N is the number of particles employed to represent the target states [16].

A discrete weighted sample set is employed to approximate the true posterior PDF $p(\mathbf{x}_k|\mathbf{z}_k)$ [16], where the weights are chosen using the principle of importance sampling [52]. Suppose $p(\mathbf{x}_k^i) \propto \pi(\mathbf{x}_k^i)$ as a probability density from which it is difficult to draw samples but for which $\pi(\mathbf{x}_k^i)$ can be evaluated. Then let particle samples $\mathbf{x}_k^i \sim q(\mathbf{x}_k), i = 1, ..., N$, be samples that are easily generated from a proposed distribution $q(.)$, which is called the importance density, then a weighted approximation $p(\cdot)$ to the density $q(\cdot)$ can be given by:

$$p(\mathbf{x}_k) \approx \sum_{i=1}^{N} w_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i) \tag{3.3.9}$$

where the weights of the samples can be described as:

$$w_k^i \propto \frac{\pi(\mathbf{x}_k^i)}{q(\mathbf{x}_k^i)} \tag{3.3.10}$$

which are the normalized weights of the particles. In this way, when the particle $\mathbf{x}_k^i$ is drawn from the importance density $q(\mathbf{x}_k|\mathbf{z}_{1:k})$, the weights of the particle can be calculated as:

$$w_k^i \propto \frac{p(\mathbf{x}_k^i|\mathbf{z}_{1:k})}{q(\mathbf{x}_k^i|\mathbf{z}_{1:k})} \tag{3.3.11}$$

When returning to the sequential case, at each iteration, the target is to sample an approximation to $p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$ and then to approximate $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ with a new set of samples, so that the importance density can be factorized as [52]

$$q(\mathbf{x}_k|\mathbf{z}_{1:k}) \triangleq q(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{z}_{1:k}) q(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) \tag{3.3.12}$$

then samples $\mathbf{x}_k^i \sim q(\mathbf{x}_k|\mathbf{z}_{1:k})$ can be obtained by augmenting each of the existing samples $\mathbf{x}_{k-1}^i \sim q(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$ with the new state $\mathbf{x}_k^i$.

By employing the Bayesian filtering framework (3.2.3) and (3.2.4), the posterior pdf $p(\mathbf{x}_k|\mathbf{z}_k)$ can be calculated as [16]

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{z}_{1:k}) &= \frac{p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{z}_{1:k-1})p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})} \\ &= \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})} \times p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) \\ &\propto p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{z}_{k-1}). \end{aligned} \tag{3.3.13}$$

When combining equations (3.3.12) and (3.3.13) above into (3.3.11), the relationship between the weight of the current particle and its previous value can be calculated as:

$$w_k^i \propto \ w_{k-1}^i \frac{p(\mathbf{z}_k|\mathbf{x}_k^i)p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i, \mathbf{z}_k)} \tag{3.3.14}$$

and the posterior filtered density $p(\mathbf{x}_k|\mathbf{z}_k)$ can be approximated as:

$$p(\mathbf{x}_k|\mathbf{z}_k) \approx \sum_{i=1}^{N} w_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i) \tag{3.3.15}$$

There are many methods to draw samples $\mathbf{x}_k^i$ from the importance density $q(\cdot)$ [52], here two such choices are described in detail.

Optimal choice

For the optimal choice, the importance density is equal to the posterior density, which can be described as [52]:

$$q(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{z}_k)_{opt} = p(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{z}_k) = \frac{p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{x}_{k-1}^i)p(\mathbf{x}_k|\mathbf{x}_{k-1}^i)}{p(\mathbf{z}_k|\mathbf{x}_{k-1}^i)} \tag{3.3.16}$$

in this case, the weight of the particle can be obtained as below:

$$\begin{aligned} w_k^i &\propto \ w_{k-1}^i p(\mathbf{z}_k|\mathbf{x}_{k-1}^i) \\ &= w_{k-1}^i \int p(\mathbf{z}_k|\mathbf{x}_k^i)p(\mathbf{x}_k|\mathbf{x}_{k-1}^i)d\mathbf{x}_k^i \end{aligned} \tag{3.3.17}$$

Using a Gaussian model to represent the importance density $q(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{z}_k)$ in this importance choice, as in the Kalman filter [54], two models are exploited, one is the state model and the other is the measurement model:

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{v}_k \tag{3.3.18}$$

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{w}_k \tag{3.3.19}$$

where (3.3.18) represents the state evolution model of the particle filter and (3.3.19) represents the measurement model. In this case, the optimal importance density and posterior density are Gaussian, that is:

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{z}_k) = N(\mathbf{x}_k; \mathbf{a}_k, \mathbf{\Sigma}_k) \tag{3.3.20}$$

$$p(\mathbf{z}_k|\mathbf{x}_{k-1}) = N(\mathbf{z}_k; \mathbf{b}_k, \mathbf{S}_k) \tag{3.3.21}$$

where $\mathbf{a}_k$ denotes the mean of the particles and $\mathbf{\Sigma}_k$ denotes the covariances matrix of the noise; $\mathbf{b}_k$ denotes the mean of the measurement particles and $\mathbf{S}_k$ denotes the covariance matrix of the measurement noise [55]

$$\mathbf{a}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{\Sigma}_k\mathbf{H}_k^T\mathbf{R}_k^{-1}(\mathbf{z}_k - \mathbf{b}_k) \tag{3.3.22}$$

$$\mathbf{\Sigma}_k = \mathbf{Q}_k - \mathbf{Q}_k\mathbf{H}_k^T\mathbf{S}_k^{-1}\mathbf{H}_k\mathbf{Q}_k \tag{3.3.23}$$

$$\mathbf{S}_k = \mathbf{H}_k\mathbf{Q}_k\mathbf{H}_k^T + \mathbf{R}_k \tag{3.3.24}$$

$$\mathbf{b}_k = \mathbf{H}_k\mathbf{F}\mathbf{x}_{k-1} \tag{3.3.25}$$

However, for many models, such analysis equations are not possible because of their Gaussian and linear requirement. Another option was proposed in [52], which is known as the suboptimal importance density selection for particle filtering.

Suboptimal choice

In this thesis, the suboptimal choice for the importance density is used, where the prior distribution of particle states is chosen to be the importance density, which can be described as follows [52]:

$$q(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{z}_k)_{sub-opt} = p(\mathbf{x}_k|\mathbf{x}_{k-1}^i) \tag{3.3.26}$$

so the posterior density is simply:

$$p(\mathbf{z}_k|\mathbf{x}_{k-1}) = \mathcal{N}(\mathbf{x}_k; \mathbf{f}\mathbf{x}_{k-1}, \mathbf{Q}_{k-1}) \tag{3.3.27}$$

and the weights can be calculated as:

$$w_k^i \propto \ w_{k-1}^i p(\mathbf{z}_k|\mathbf{x}_k^i) \tag{3.3.28}$$

By solving the above equations, the particle set $\{\mathbf{x}_k^i\}_{i=1}^N$ associated with their weights $\{w_k^i\}_{i=1}^N$ at time k can be obtained, moreover, in this way, the choice can avoid the requirement that the state and measurement be Gaussian and linear, hence it has much wider potential application.

Resampling

By employing the particle filtering algorithm described above, one common problem is the degeneracy phenomenon [16], where after a few iterations, all but one particle will have negligible weight but the variance of the importance weights can only increase over time, thus causing the problem of degeneracy. One popular measure of degeneracy of the algorithm is the effective sample size N_{eff}, which can be calculated as [52]

$$N_{eff} = \frac{1}{\sum_{i=1}^{N}(w_k^i)^2} \tag{3.3.29}$$

A smaller value of N_{eff} indicates a larger degeneracy; when all the particles have uniform weights, N_{eff} should be equal to the number of particles N. The solution for this problem is known as resampling [52].

Suppose the importance sampling approximation of the target distribution $p(\mathbf{x}_k|\mathbf{z}_k)$ is represented as $\hat{p}(\mathbf{x}_k|\mathbf{z}_k)$, which is based on the weighted samples from the importance density $q(\mathbf{x}_k|\mathbf{z}_k)$, the resampling process eliminates the samples with low weights and concentrates on the samples with high weights. The resampling process generates N samples from the approximated probability distribution $\hat{p}(\mathbf{x}_k|\mathbf{z}_k)$ and assigns equal weights $1/N$ to every particle. The pseudocode is shown in Algorithm 3.1 [52]:

Algorithm 3.1 Resampling Algorithm for particle filtering

Input: $\{\mathbf{x}_k^i, w_k^i\}_{i=1}^N$.
Output: $\{\mathbf{x}_k^{j^*}, w_k^j, i^j\}_{j=1}^N$.

Initialize the cumulative sum of weights (CSW): $c_1 = w_k^1$
for $i = 2 : N$ **do**
    Construct CSW: $c_i = c_{i-1} + w_k^i$
end for
Start at the bottom of the CSW: $i = 1$
Draw a starting point from a uniform distribution: $\mu_1 \sim \mathcal{U}[0, N^{-1}]$
for j $= 1 : N$ **do**
    Move along the CSW: $\mu_j = \mu_1 + N^{-1}(j-1)$
    while $\mu_j > c_i$ **do**
        $*i = i + 1$
    end while
    Assign sample: $\mathbf{x}_k^{j^*} = \mathbf{x}_k^i$
    Assign weight: $w_k^j = N^{-1}$
    Assign parent: $i^j = i$
end for

In a summary, the pseudocode for the whole algorithm for the particle filter is shown in Algorithm 3.2 [52], where the particles are sampled from the prediction stage, then after calculating the likelihood of the particle, the updating stage is employed to update and obtain the target state at the current time. After the above steps, the resampling step is employed to avoid the problem of degeneracy.

Algorithm 3.2 Pseudocode for Particle Filtering

Input: $\{\mathbf{x}_{k-1}^i, w_{k-1}^i\}_{i=1}^N$.
Output: $\{\mathbf{x}_k^i, w_k^i\}_{i=1}^N$.

1: **for** $i = 1 : N$ **do**
2: Draw particle $\mathbf{x}_k^i$ with importance density function.
3: Calculate $\hat{w}_k^i$ with (3.3.14).
4: **end for**
5: **for** $i = 1 : N$ **do**
6: Normalize: $w_k^i = w_k^i / \sum_{i=1}^{N} w_k^i$
7: Resample as Algorithm 3.1.
8: **end for**

3.4 MCMC particle filtering

In the traditional particle filter, importance sampling is used in the sample selection step. Another approach is to apply a Markov Chain Monte Carlo (MCMC) approach in the sampling step, which is called the MCMC particle filter [23]. By building a Markov chain, the MCMC particle filter explores the posterior distribution, and thus can obtain more accurate tracking results.

As described in [56], during the MCMC resampling, a particle $\mathbf{x}_k^i$ is propagated to a new state $\mathbf{x}_k^{i*}$ based on the following model

$$\mathbf{x}_k^{i*} = \mathbf{x}_k^i + \mathbf{q} \tag{3.4.1}$$

where $\mathbf{q}$ denotes a zero-mean Gaussian noise vector. From the Metropolis-Hastings acceptance probability [52], the acceptance ratio is calculated as

$$\alpha = min\left\{1, \frac{p(\mathbf{z}_k|\mathbf{x}_k^{i*})p(\mathbf{x}_k^{i*}|\mathbf{x}_{k-1})q(\mathbf{x}_k^i|\mathbf{x}_k^{i*})}{p(\mathbf{z}_k|\mathbf{x}_k^i)p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^m)q(\mathbf{x}_k^{i*}|_k^i)}\right\}. \tag{3.4.2}$$

Since in this thesis, $q(\cdot|\mathbf{x}_k^i)$ is symmetric in its arguments, that is:

$$q(\mathbf{x}_k^{i*}|\mathbf{x}_k^i) = q(\mathbf{x}_k^i|\mathbf{x}_k^{i*}) \tag{3.4.3}$$

the acceptance ratio α can be calculated as:

$$\alpha = min\left\{1, \frac{p(\mathbf{z}_k|\mathbf{x}_k^{i*})p(\mathbf{x}_k^{i*}|\mathbf{x}_{k-1})}{p(\mathbf{z}_k|\mathbf{x}_k^{i})p(\mathbf{x}_k^{i}|\mathbf{x}_{k-1})}\right\}. \tag{3.4.4}$$

The state to be preserved is determined by drawing a point j from a uniform distribution. If $j < \alpha$ then the new state $\mathbf{x}_k^{i*}$ is retained, otherwise it is rejected.

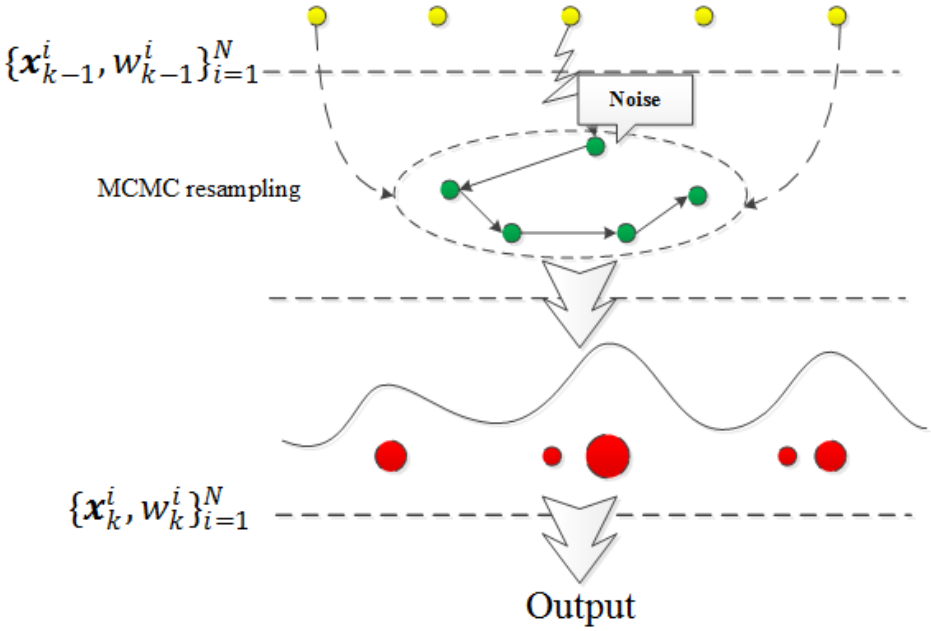


Figure 3.3. Graphical representation of the MCMC particle filtering.

Fig. 3.3 shows the steps of the MCMC particle filtering algorithm, where a particle $\hat{\mathbf{x}}_k^i$ predicted from the state model is chosen as the initial value of the Markov chain. Then a posterior PDF is obtained in the form of weighted samples [23]. After calculating the acceptance ratio α for each particle and discarding the burn-in samples [23], more robust particles are obtained by the MCMC particle filter, hence more accurate tracking results can be obtained. The example pseudocode of this MCMC particle filter for target m is then summarised as Algorithm 3.3, where the inputs are the predicted particles for target m and the output is the posterior distribution, and B denotes the

number of burn-in period particles.

Algorithm 3.3 MCMC particle filtering

Input: $\{\mathbf{x}_{k-1}^i, w_{k-1}^i\}_{i=1}^{i=N}$
Output: $\{\mathbf{x}_k^i, w_k^i\}_{i=1}^{i=N}$

1: Initialize the Markov chain by the predicted particles from the state transition function using the states of the target at $k-1$.
2: **for** $i = 1{:}N + B$ **do**
3: Propagate $\mathbf{x}_k^{i*}$ from $\mathbf{x}_k^i$ with (3.4.1).
4: Compute α with (3.4.4).
5: Draw a point j from a uniform distribution.
6: **if** $j < \alpha$ **then**
7: retain the new state: $\mathbf{x}_k^i = \mathbf{x}_k^{i*}$.
8: **else** reject the new state.
9: **end if**
10: **end for**
11: Discard the first B particles of the iterations.

The previous sections introduced the fundamental method for multiple target tracking, however, video based multiple human tracking often involves the problem of varying number of targets. To address this challenge, the particle PHD filter is employed in this thesis.

3.5 Particle PHD filtering

In multiple target tracking problems, the common problems are variable number of targets and occlusion; the probability hypothesis density (PHD) filter was therefore proposed to overcome the problem of a variable number of targets. In this section, the random finite set (RFS) approach for multiple targets is described firstly, and then, the first order moment of an RFS, known as the PHD, is propagated, namely the intensity, instead of the posterior density, to yield the particle PHD filter.

3.5.1 Random finite set

An RFS provides a principled solution to the problem of extension of the uncertainty modelling to the cardinality of the state set $\mathbf{X}_k$, which contains

the state of all M_k targets at time k, where $\mathbf{X}_k = \{\mathbf{x}_k^m, m = 1, ..., M_k\}$, and the observation set $\mathbf{Z}_k$ [57], which contains all the observations obtained from the video camera at time k. An RFS is a set where [3]:

- The elements are random stochastic processes.
- The set cardinality is also a stochastic process.

Compared with existing solutions such as multiple hypothesis tracking (MHT) [3], the RFS method presents a more effective way to deal with the birth of new targets, clutter, missing detection and spatial noise, it can integrate spatial and temporal filtering in a single Bayesian framework. As it is described in [3], let Ξ_k be the RFS associated with the multi-target state

$$\Xi_k = \mathbf{S}_k(\mathbf{X}_{k-1}) \bigcup \mathbf{B}_k(\mathbf{X}_{k-1}) \bigcup \mathbf{\Gamma}_k \tag{3.5.1}$$

where $\mathbf{S}_k(\mathbf{X}_{k-1})$ denotes the RFS of survived targets, $\mathbf{B}_k(\mathbf{X}_{k-1})$ denotes the targets spawned from the previous set of targets $\mathbf{X}_{k-1}$ and $\mathbf{\Gamma}_k$ is the RFS of the new-born targets. By generalizing the single target recursive Bayes filter, the multiple prediction and update recursive equations for the RFS can be written as

$$p(\mathbf{X}_k|\mathbf{Z}_{k-1}) = \int p(\mathbf{X}_k|\mathbf{X}_{k-1}) p(\mathbf{X}_{k-1}|\mathbf{Z}_{k-1}) \mu(d\mathbf{X}_{k-1}) \tag{3.5.2}$$

$$p(\mathbf{X}_k|\mathbf{Z}_k) = \frac{p(\mathbf{Z}_k|\mathbf{X}_k) p(\mathbf{X}_k|\mathbf{Z}_{k-1})}{\int p(\mathbf{Z}_k|\mathbf{X}_k) p(\mathbf{X}_k|\mathbf{Z}_{k-1}) \mu(d\mathbf{X}_k)} \tag{3.5.3}$$

where $p(\mathbf{X}_k|\mathbf{Z}_k)$ is the multi-target posterior density at time k and $p(\mathbf{Z}_k|\mathbf{X}_k)$ is the multi-target likelihood, μ is an appropriate dominating measure on the RFS Ξ [58]. In order to limit computational cost and memory usage, a hypothesis pruning mechanism can be applied [3]. In the next section a better solution to this problem known as the PHD filter is described.

3.5.2 Probabilistic hypothesis density filter

Although Monte Carlo methods can compute the recursion of (3.5.2) and (3.5.3), as the dimensionality of the multiple target tracking state $\mathbf{X}_k$ increases with the number of targets in the scene, the number of particles required grows exponentially with the scene [22]. For this reason, an approximation is necessary to make the problem computationally tractable. In this case, instead of a posterior density, the first-order moment of the multi-target posterior can be propagated. The resulting filter is known as the PHD filter, which is based on the following three assumptions [3]

- The targets evolve and generate measurements independently.
- The clutter RFS, Υ_k, is Poisson-distributed.
- The predicted multi-target RFS is Poisson-distributed.

Denoting $D(\cdot)$ as the PHD at time k associated with the multi-target posterior density $p(\mathbf{X}_k|\mathbf{Z}_k)$, the Bayesian interactive prediction and update $D(x)$ is known as the PHD filter [59].

The PHD prediction step is defined as:

$$D(\mathbf{X}_k|\mathbf{Z}_{k-1}) = \int \phi(\mathbf{x}_k|\mathbf{X}_{k-1}) D(\mathbf{X}_{k-1}|\mathbf{Z}_{k-1}) d(\xi) + \Upsilon_k \tag{3.5.4}$$

where Υ_k is the intensity function of the new target birth RFS, $\phi(\cdot)$ is the analogue of the state transition probability in the single target case

$$\phi(\mathbf{x}_k|\mathbf{X}_{k-1}) = e(\mathbf{x}_k|\mathbf{X}_{k-1}) + \beta(\mathbf{x}_k|\mathbf{X}_{k-1}) \tag{3.5.5}$$

in which $e(\cdot)$ is the probability that the target still exists at time k and $\beta(\cdot)$ is the intensity of the RFS that a target is spawned from the state $\mathbf{X}_{k-1}$.

The PHD update step is defined as [60]:

$$D(\mathbf{X}_k|\mathbf{Z}_k) = \left[p_M(\mathbf{x}_k) + \sum_{\mathbf{z}_k \in \mathbf{Z}_k} \frac{\psi_{k,\mathbf{z}_k}(\mathbf{x}_k)}{\kappa_k + \langle \psi_{k,\mathbf{z}_k}, D(\mathbf{X}_k|\mathbf{Z}_{k-1}) \rangle} \right] D(\mathbf{X}_k|\mathbf{Z}_{k-1}) \tag{3.5.6}$$

where p_M is the missing detection probability, $\psi_{k,\mathbf{z}_k}(\mathbf{x}_k) = (1-p_M)g_k(\mathbf{z}_k|\mathbf{x}_k)$ is the single-target likelihood defining the probability that $\mathbf{z}_k$ is generated by a target with state $\mathbf{x}_k$, κ_k is the clutter intensity the inner product $\langle f, g \rangle = \int f(x)g(x)dx$ [3].

3.5.3 SMC probabilistic hypothesis density filter

There are numerical solutions for the integrals in (3.5.4) and (3.5.6), one of which is a solution obtained using a sequential Monte Carlo method that approximates the PHD with a set of weighted random samples, which is called the particle PHD filter. Where the PDF and PHD of the states of M_k targets are described by a set of weighted particles [16] $\{x_k^i, w_k^i\}_{i=1}^{M_k \times N}$, where M_k is the target number at time k. By employing the concept of the particle filter described in Section 3.3, and by utilizing N particles to represent the state of each target m, the prediction and updating steps of the particle PHD filer can be described as:

1. Prediction: Particles $\hat{\mathbf{x}}_k^i$ are drawn from the predicted particle set as (3.3.27) and fed into (3.5.4), which is the prediction model of the particle PHD filter, which is described as [60]

$$D(\mathbf{X}_k|\mathbf{Z}_{k-1}) = \int \phi(\hat{\mathbf{x}}_k^i|\mathbf{X}_{k-1}) D(\mathbf{X}_{k-1}|\mathbf{Z}_{k-1}) \delta\hat{\mathbf{x}}_k^i + \Upsilon_k(\mathbf{X}_k) \tag{3.5.7}$$

where

$$\phi(\hat{\mathbf{x}}_k^i|\mathbf{X}_{k-1}) = e(\hat{\mathbf{x}}_k^i|\mathbf{X}_{k-1}) + \beta(\hat{\mathbf{x}}_k^i|\mathbf{X}_{k-1}) \tag{3.5.8}$$

When exploiting the PHD filter with the particle filter, the PHD of the

states is represented by the weights of the particles, which include the survived particles and new-born particles. Assuming at time k, J_k particles are sampled for the new-born targets, the initial weights assigned to the new born particles, which are employed to represent the new-born targets, are calculated as

$$\hat{w}_k^i = \frac{1}{J_k}, \quad i = M_{k-1} \times N + 1, ..., M_{k-1} \times N + J_k \tag{3.5.9}$$

then the weights are fed into (3.5.7). With this method, a particle set is obtained, which includes particles for both survived targets and new born targets

$$\{\hat{\mathbf{x}}_k^i, \hat{w}_k^i\}_{i=1}^{M_{k-1} \times N + J_k} \tag{3.5.10}$$

where i is the index of the i^{th} particle. The weights obtained from the prediction step are given as

$$\hat{w}_k^i = \begin{cases} \phi(\hat{\mathbf{x}}_k^i) w_{k-1}^i & i = M_{k-1} \times N \\ \\ \frac{\Upsilon_k}{J_k} & i = M_{k-1} \times N + 1, ..., M_{k-1} \times N + J_k \end{cases} \tag{3.5.11}$$

In this way, the predicted PHD $D(\mathbf{X}_k|\mathbf{Z}_{k-1})$ at time k for target states $\mathbf{X}_k$ is obtained based on the weights of the particles.

2. Measurement update: In the particle PHD filter, the PHD $D(\cdot)$ is represented by the weights of particles. Once the new set of observations is available, by substituting the approximation of $D(\mathbf{X}_k|\mathbf{Z}_{k-1})$ into (3.5.6), the weights of each particle are updated based upon the receipt of the measurement $\mathbf{Z}_k$ as [60]

$$w_k^i = \left[p_M(\hat{\mathbf{x}}_k^i) + \sum_{\forall \mathbf{z}_k \in \mathbf{Z}_k} \frac{\psi_{k,\mathbf{z}_k}(\hat{\mathbf{x}}_k^i)}{\kappa_k + C_k(\mathbf{z}_k)} \right] \hat{w}_k^i \tag{3.5.12}$$

where

$$C_k(\mathbf{z}_k) = \sum_{i=1}^{M_{k-1}\times N+J_k} \psi_{k,\mathbf{z}_k}(\hat{\mathbf{x}}_k^i)\hat{w}_k^i \tag{3.5.13}$$

Then the number of targets is calculated by the sum of all the weights for particles as follows [60]

$$\hat{M}_k = \sum_{i=1}^{M_{k-1}\times N+J_k} w_k^i \tag{3.5.14}$$

$$M_k = int(\hat{M}_k) \tag{3.5.15}$$

where $int(\cdot)$ denotes the integer nearest to $\hat{M}_k$.

Algorithm 3.4 Adapted particle PHD filter

Input: $\{\mathbf{x}_{k-1}^m\}_{m=1}^{m=M_{k-1}}$.
Output: $\{\mathbf{x}_k^m\}_{m=1}^{m=M_k}$ with M_k targets.

Generate $\{x_{k-1}^i, w_{k-1}^i\}_{i=1}^{M_k\times N}$ from $\{\mathbf{x}_{k-1}^m\}_{m=1}^{m=M_{k-1}}$ and feed into (3.5.7).
Select new-born particles.
Obtain particle weights as (3.5.11).
for $i = 1 : M_{k-1} \times N + J_k$ **do**
    Calculate $g(\mathbf{z}_k|\hat{\mathbf{x}}_k^i)$.
    Update particle weights with (3.5.12).
end for; % Achieve particle set $\{\hat{\mathbf{x}}_k^i, \hat{w}_k^i\}_{i=1}^{M_{k-1}\times N+J_k}$ with updated weight.
Calculate M_k by (3.5.14) and (3.5.15).
Normalize w_k^i.
Initialize the cumulative probability $c_1 = 0$
Update $c_i = c_{i-1} + \hat{w}_k^i$, $i = 2, ..., M_{k-1} \times N + J_k$.
Draw a starting point $\mu_1 \sim [0, (M_k \times N)^{-1}]$
for $j = 1, ..., M_k \times N$ **do**
    $\mu_j = \mu_1 + (j-1)/(M_k \times N)$
    while $\mu_j > c_i$ **do**
        $i = i + 1$
    end while
    $\mathbf{x}_k^j = \hat{\mathbf{x}}_k^i$
    $w_k^j = N^{-1}$
end for
Clustering $\{\mathbf{x}_k^i, \frac{1}{N}\}_{i=1}^{i=M_k\times N}$, calculate (3.3.15) and output $\{\mathbf{x}_k^m\}_{m=1}^{m=M_k}$

At each iteration k, J_k new particles are added to the old $M_{k-1}\times N$ parti-

cles for the new born targets. To limit the growth of the number of particles, and to avoid the problem of degeneracy, a resampling step is performed after the update step.

The algorithm for the adapted particle PHD filter with a resampling step at each time k is described as Algorithm 3.4 [4], where the input $\{\mathbf{x}_{k-1}^{m}\}_{m=1}^{m=M_{k-1}}$ represents the survived targets from the previous time $k-1$ and the output $\{\mathbf{x}_{k}^{m}\}_{m=1}^{m=M_{k}}$ denotes the tracking results in the form of the states of the targets.

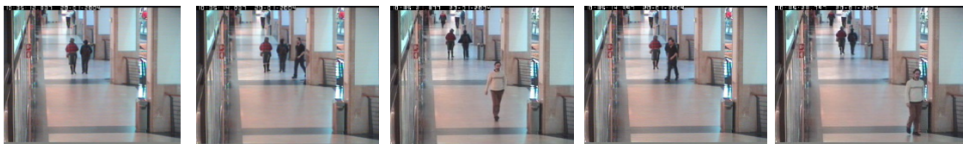
The above method underpins the traditional particle PHD filter for multiple human tracking. In the next section, evaluation datasets as well as evaluation measures employed in this thesis will be introduced.

3.6 Evaluation dataset

In order to evaluate the performance of the proposed system for multiple human tracking, particularly to handle the situation of varying number of targets, close interactions and occlusions, sequences from three different publicly available video datasets are chosen: one from the PETS2009 dataset [1] where 3-6 human targets are walking in an outdoor campus environment, one sequence from the CAVIAR dataset [2] where 1-5 human targets are walking in a shopping mall environment and one from the TUD dataset [42] where 5-7 human targets are walking in an outdoor-shopping mall environment. Some of the selected samples from the three dataset are shown in Fig. 3.4

3.7 Performance evaluation methods

Several measures are employed to examine the performance of the proposed particle PHD filter and compare the results from the related algorithms, including the Euclidean error in each frame, the optimal subpattern assignment



(a) Frames from the CAVIAR dataset

(b) Frames from the PETS2009 dataset

(c) Frames from the TUD dataset

Figure 3.4. Selected frames from the three selected sequences from three different datasets, i.e. (a) is from the 'EnterExitCrossingPaths1cor' sequence from the CAVIAR dataset, (b) is from the 'PETS09_View001_S2_L1' sequence from the PETS2009 dataset and (c) is from 'TUD_Stadtmitte' sequence from the TUD dataset.

(OSPA) [61] [62], and the multiple object tracking precision (MOTP) [63]. For a sequence with ℓ frames, assuming at time k, the tracking system gives the tracking results $\mathbf{O}_k = \{\mathbf{o}_k^1, ..., \mathbf{o}_k^n\}$ with n targets while $\mathbf{Y} = \{\mathbf{y}_k^1, ..., \mathbf{y}_k^m\}$ is the ground truth information with m targets. These measures are defined below. In addition, the computational complexity can also been considered in evaluations.

Mean and standard deviation of Euclidean error on each frame

The localization error for each target in terms of mean and variance can be used as a performance metric to evaluate the accuracy and stability of our proposed tracking system. The mean of Euclidean errors (MEE) at frame number k is denoted by

$$MEE_k = \frac{1}{n} \sum_{i=1}^{n} \|\mathbf{o}_k^i - \mathbf{y}_k^i\| \tag{3.7.1}$$

where $\|\cdot\|$ denotes Euclidean distance and its standard deviation (SD) is given by

$$SD_k = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\|\mathbf{o}_k^i - \mathbf{y}_k^i\| - MEE_k)^2} \tag{3.7.2}$$

OSPA

In multiple human tracking, the accuracy not only depends on the error between the estimated position and the ground truth information of the targets in the scenario, but also the missed detections and false alarms. Dominic et al. proposed a metric to evaluate the tracking system by error from both distance and the number of targets [64] which is used by Ristic et al. [61] for evaluating multiple human tracking algorithms. As described in [64], given the set of tracking results $\mathbf{O}_k$ and the ground truth information $\mathbf{Y}_k$, the distance between $\mathbf{O}_k$ and $\mathbf{Y}_k^m$, $d_k^c(\mathbf{o}_k^n, \mathbf{y}_k^m) := min(c, d(\mathbf{o}_k^n, \mathbf{y}_k^m))$ with cut off at $c > 0$ and $1 \leq p \leq \infty$, is calculated as [61]

$$d_{k,p}^c(\mathbf{O}_k, \mathbf{Y}_k) := \left(\frac{1}{n}\left(\min_{\pi\in\prod_n}\sum_{i=1}^{m} d^c(\mathbf{o}_k^i, \mathbf{y}_k^{\pi(i)})^p + c^p(n-m)\right)\right)^{\frac{1}{p}} \tag{3.7.3}$$

for $m \leq n$, and $d_{k,p}^c(\mathbf{O}, \mathbf{Y}) = d_p^c(\mathbf{Y}, \mathbf{O})$ for $m > n$. The function d_p^c is named as the OSPA metric of order p with cut-off c. In this thesis, $c = 20$ and $p = 2$ are used in evaluations as are commonly used by other researchers. Based on the OSPA metric, a new evaluation measure for multiple target tracking has been recently proposed, named optimal subpattern assignment for multiple target tracking (OSPAMT) [62], however, in this thesis OSPA and the following MOTP measure, which is also employed in [65], are sufficient for comparative evaluation.

MOTP

The MOTP [63] is the total error in the estimated position for matched object-hypotheses pairs over all frames, averaged by the total number of

matches made. It shows the ability of the tracker to estimate the precise object positions, which can be calculated as

$$MOTP_k = \frac{\sum_{i=1,k=1}^{i=n,k=\ell} error_k^i}{\sum_{k=1}^{\ell} c_k} \tag{3.7.4}$$

where $error_k^i$ denotes the Euclidean error for each target i at time k and c_k is the total number of matched targets at time k.

3.8 Summary

In this chapter, the fundamental algorithms for multiple tracking with Bayesian filters were explained, where four solutions: the Kalman filter, the particle filter, the MCMC particle filter and the PHD filter are introduced. The Kalman filter is optimal for linear and Gaussian systems, however, when the models are nonlinear and non-Gaussian, such as in human tracking from video, particle filtering algorithm is widely employed. These two algorithms give solutions in the Bayesian tracking framework, however, they fail to address the problem of varying number of targets in multiple human tracking work, so the particle PHD filter is employed in this thesis. At the end of this chapter, the performance evaluation measures as well as the evaluation sequences were given to evaluate the tracking performance. In the next chapter, the Student's-t and variational Bayesian methods will be proposed to improve the tracking performance of the particle PHD filter.

Chapter 4

PARTICLE PHD FILTER WITH STUDENT'S-T DISTRIBUTION AND VARIATIONAL BAYESIAN APPROACH FOR MULTIPLE HUMAN TRACKING

4.1 Introduction

As described in Chapter 3, the particle PHD filter is proposed for multiple human tracking, particularly to address the problem of varying number of targets. Within the Bayesian filtering framework [16], the noise vectors $\mathbf{v}_k$ and $\mathbf{w}_k$ representing model uncertainty are particularly important in the prediction and updating models. In the traditional PHD filter, the two noise vectors are assumed to be independent within each model and are represented as Gaussian distributions. However, the performance of the PHD filter can be affected by estimation errors which are related to the

noise vectors. The Student's-t distribution, which is also a member of the family of continuous probability distributions, has been widely employed by researchers in many areas such as source separation [66], filtering [67] and tracking [68] in order to achieve more robust performance due to its heavier tails than a Gaussian distribution. In this chapter, the Student's-t distribution is firstly employed in the prediction and updating steps of the particle PHD filter, which is introduced in Section 4.2. Moreover, by considering the relationship between the measurement noise vector $\mathbf{w}_k$ and the prediction model, which is

$$\hat{\mathbf{x}}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{w}_k) \tag{4.1.1}$$

the Student's-t distribution is employed as a joint distribution between $\mathbf{w}_k$ and the predicted state vector (4.1.1) with a variational Bayesian approach, which will be introduced in Section 4.3 as another approach in detail. This chapter is targeted at satisfying the first two objectives of this thesis, which are the Student's-t distribution particle PHD filter published in [69] and the variational Bayesian particle PHD filter for multiple human tracking published in [70].

The performance of the two proposed tracking algorithms is evaluated with real video sequences from the CAVIAR and PETS2009 datasets, and compared with the traditional particle PHD filter for multiple human tracking. Tracking and comparison results are shown at the end of this chapter, which confirm the improved performance from the two proposed trackers by reducing the OSPA value.

4.2 Particle PHD filter with Student's-t distribution for multiple human tracking

In this section, the univariate Student's-t distribution is firstly introduced and compared with the univariate Gaussian distribution, then it is employed to represent the prior and posterior distributions instead of the Gaussian distribution for the particle PHD filter. In multiple human tracking work, the state and measurement models contain multiple variables, such as location, velocity and size of target, so the differentiation and integration of functions involve multiple variables, therefore rather than scalar, multivariate distributions are employed in this work.

4.2.1 Student's-t distribution

Developed by William Sealy Gosset under the pseudonym Student [71], the Student's-t distribution is known as a super Gaussian distribution, which has heavier tails than the Gaussian distribution, hence more robust prediction of the target states can generally be achieved. In particular, the statistical dependence modelled by a multivariate Student's-t distribution can be an advantage when sampling particles.

The univariate Student's-t distribution can be exploited as the marginal density function, which can be formed as [72]:

$$p(x) = \frac{\mathbf{\Gamma}(\frac{v+1}{2})}{\sqrt{v\pi}\mathbf{\Gamma}(\frac{v}{2})}\left(1+\frac{x^2}{v}\right)^{-\frac{v+1}{2}} \tag{4.2.1}$$

where v is the degree of freedom parameter and $\mathbf{\Gamma}$ is the gamma function, which can be calculated as

$$\Gamma(v) = (v-1)! \tag{4.2.2}$$

From the probability density function, it can be deduced that the degree

of freedom parameter v can tune the variance and leptokurtic nature of the distribution, with decreasing v, the tails of the distribution becomes heavier. Fig. 4.1 shows the comparison of the density for the univariate Student's-t distribution for increasing values of v compared with the normal distribution with zero mean and unit variance.

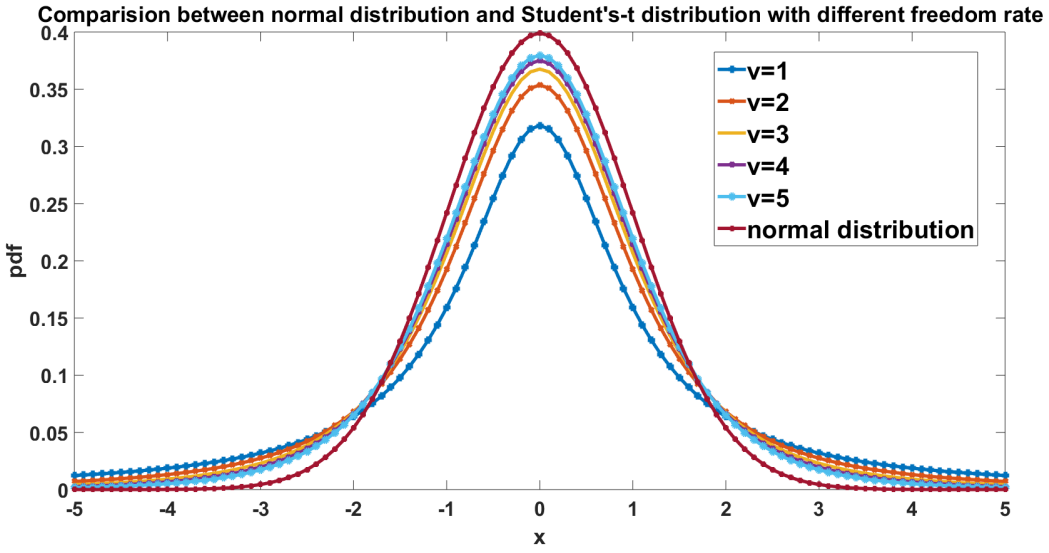


Figure 4.1. Comparison of univariate Gaussian distribution and univariate Student's-t distribution with different degree of freedom rates.

From the comparison, it is clear that, when compared with the normal distribution, the distribution for the Student's-t distribution is lower and wider, which can potentially improve robustness by increasing the probability in sampling particles of larger amplitude and thereby providing more variable states when applied in the Bayesian tracking system. Due to the advantages from the Student's-t distribution, in this chapter, it is employed for the prediction stage for the particle PHD filter. However, since the states and measurements within the human tracking framework contains multiple variables, the multivariate Student's-t distribution is employed instead of the univariate one. Given the $d \times 1$ dimension variable vector $\mathbf{x}$, the pdf of

the multivariate Student's-t distribution is given as

$$p(\mathbf{x}) = \frac{1}{\Gamma(\frac{v}{2})v^{d/2}\pi^{d/2}\|\mathbf{\Sigma}\|^{1/2}} \frac{\Gamma[(v+d)/2]}{[1+\frac{1}{v}(\mathbf{x}-\mu)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)]^{(v+d)/2}} \tag{4.2.3}$$

where μ and $\mathbf{\Sigma}$ are the location vector and scale matrix for the multivariate Student's-t distribution.

4.2.2 Student's-t distribution particle PHD filter for multiple human tracking

Two fundamental actions for the particle PHD filter are prediction and updating. As introduced in Section 3.5, the basic principle of importance sampling, which is used for PHD prediction is to represent a PHD $D(\mathbf{X}_k|\mathbf{Z}_k)$ by a set of particles associated with their weights, namely $\{\mathbf{x}_k^i, w_k^i\}_{i=1}^N$, where $\mathbf{X}_k$ and $\mathbf{Z}_k$ contain the states for all targets $\mathbf{x}_k$ and all measurements $\mathbf{z}_k$ at time k respectively. The Student's-t distribution based particle PHD filter is designed to sample and update the particles in the models generated by the Student's-t distribution. Assuming the particle set at time $k-1$ $\{\mathbf{x}_{k-1}^i\}_{i=1}^N$ is sampled from the pdf $p(\mathbf{X}_{k-1}|\mathbf{Z}_{k-1})$, based on the Bayesian filtering method, the Student's-t distribution based particle PHD filter mainly contains two steps [73]:

Prediction: Draw particle $\mathbf{x}_{k-1}^i$ from $D(\mathbf{X}_{k-1}|\mathbf{Z}_{k-1})$ and feed it into the prediction model of particle PHD filter:

$$D(\mathbf{X}_k|\mathbf{Z}_{k-1}) = \int \phi(\hat{\mathbf{x}}_k^i|\mathbf{X}_{k-1})D(\mathbf{X}_{k-1}|\mathbf{Z}_{k-1})\delta\hat{\mathbf{x}}_k^i + \Upsilon_k(\mathbf{X}_k) \tag{4.2.4}$$

to obtain particles at time k. If the distribution of $\mathbf{X}_{k-1}$ has a Student's-t distribution or stays close to it, then the predicted PHD $D(\mathbf{X}_k|\mathbf{Z}_k)$ can be approximated as the first moment of a Student's-t distribution. Thus the

prediction model can be calculated by

$$\begin{aligned} D(\mathbf{X}_k|\mathbf{Z}_{k-1}) &= \int \phi(\hat{\mathbf{x}}_k^i|\mathbf{X}_{k-1}) D(\mathbf{X}_{k-1}|\mathbf{Z}_{k-1}) \delta\hat{\mathbf{x}}_k^i + \Upsilon_k(\mathbf{X}_k) \\ &\approx t(\mathbf{X}_k|\boldsymbol{\Omega}_{k-1}) \end{aligned} \tag{4.2.5}$$

where $t(\cdot)$ is the predicted PHD and $\boldsymbol{\Omega}_{k-1}$ is the importance density for the Student's-t distribution of the PHD filter. By exploiting the Student's-t distribution for prediction, larger amplitude particles are more likely to be sampled due to its heavier tails compared with the Gaussian distribution, which makes the prediction more widely distributed. After obtaining the new-born particles, the predicted particle set can be achieved as

$$\hat{w}_k^i = \begin{cases} \phi(\hat{\mathbf{x}}_k^i) w_{k-1}^i & i = M_{k-1} \times N \\ \\ \frac{\Upsilon_k}{J_k} & i = M_{k-1} \times N + 1, ..., M_{k-1} \times N + J_k \end{cases} \tag{4.2.6}$$

which has been described in Section 3.5.3.

Updating: Upon the receipt of the measurement set $\mathbf{Z}_k$, the likelihood of each prior sample $g(\mathbf{z}_k|\hat{\mathbf{x}}_k^i)$, can be evaluated with the probability density function of the Student's-t distribution (4.2.3) [73]. Feeding the likelihood into the particle PHD updating model, the PHD at time k can be calculated as:

$$\begin{aligned} D(\mathbf{X}_k|\mathbf{Z}_k) &= \int \left[p_M(\hat{\mathbf{x}}_k^i) + \sum_{\forall \mathbf{z}_k \in \mathbf{Z}_k} \frac{\psi_{k,\mathbf{z}_k}(\hat{\mathbf{X}}_k^i)}{\kappa_k + C_k(\mathbf{z}_k)} \right] D(\hat{\mathbf{x}}_k^i|\mathbf{Z}_{k-1}) \delta\hat{\mathbf{x}}_k^i \\ &= t(\mathbf{X}_k|\boldsymbol{\Omega}_k) \end{aligned} \tag{4.2.7}$$

Equations (4.2.5) and (4.2.7) form the basis of the proposed robust Student's-t distribution particle PHD filter, and after obtaining the PHD at time k, Algorithm 3.4 introduced in Chapter 3 can be employed to calculate the target position and number. The simulation results will be shown in Sec-

tion 4.4, which show the improvement from the Student's-t distribution; in order to find another approach based upon the Student's-t distribution, the Student's-t distribution is employed as a joint distribution between the Gaussian distribution and the inverse Gamma distribution, and the variational Bayesian approach is employed to estimate the parameters within the tracking system, which is introduced in the next section.

4.3 Variational Bayesian approach for particle PHD filter for multiple human tracking

In the above section, the Student's-t distribution was employed to represent the prior and posterior distributions of the Bayesian filtering framework, which improved the robustness of the particle PHD filter. In this section, the variational Bayesian approach is employed to estimate the parameters within the Bayesian filtering framework, where the Student's-t distribution is utilized as a joint distribution between the state model and the measurement noise covariance.

The traditional particle PHD filter often assumes a priori knowledge of the measurement and dynamic model parameters, including the noise statistics. However, such knowledge is not always available in practical applications [74]. Recently, the variational Bayesian approximation technique proposed in [75] has been introduced to estimate the target states in the linear Gaussian scenario with unknown measurement noise variance. The main idea of the variational Bayesian approach is that the joint posterior of the target state and measurement noise covariance can be approximated by a factored free form or a fixed form distribution [76]. In this work, the variational Bayesian technique is introduced into the framework of the particle PHD filter for multiple human tracking to jointly estimate the posterior of multi human target states and measurement noise covariance by using a

factored free form Student's-t distribution.

4.3.1 Optimal Bayesian filtering

The state model and measurement models of the Bayesian filtering are given as

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{w}_k \tag{4.3.1}$$

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \tag{4.3.2}$$

where $\mathbf{F}$ and $\mathbf{H}$ are state and measurement transition functions respectively, and $\mathbf{w}_k$, $\mathbf{v}_k$ are the state and measurement noise vectors, with covariance matrices $\mathbf{P}_k$ and $\mathbf{R}_k$ respectively. By assuming that the state model (3.2.1) and measurement model (3.2.2) are independent, taking the single target state vector $\mathbf{x}_k$ and measurement vector $\mathbf{z}_k$ as example, the relationship between the covariance matrix $\mathbf{R}_k$ for measurement noise with the state model can be described as

$$p(\mathbf{x}_k, \mathbf{R}_k|\mathbf{x}_{k-1}, \mathbf{R}_{k-1}) = p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{R}_k|\mathbf{R}_{k-1}) \tag{4.3.3}$$

In this work, the goal of Bayesian optimal filtering of the above model is to compute the posterior of the joint distribution $p(\mathbf{x}_k, \mathbf{R}_k|\mathbf{z}_k)$. As described in [74], the filtering problem consists of the following steps [77]:

Initialization: Start the recursion with the given initial prior distribution as $p(\mathbf{x}_0, \mathbf{R}_0)$.

Prediction: The predictive distribution of the state vector $\mathbf{x}_k$ and measurement noise covariance matrix $\mathbf{R}_k$ is given by the Chapman-Kolmogorov equation:

$$\begin{aligned} p(\mathbf{x}_k, \mathbf{R}_k|\mathbf{z}_{k-1}) = \int & p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{R}_k|\mathbf{R}_{k-1})\times \\ & p(\mathbf{x}_{k-1}, \mathbf{R}_{k-1}|\mathbf{z}_{k-1})d\mathbf{x}_{k-1}d\mathbf{R}_{k-1} \end{aligned} \tag{4.3.4}$$

Updating: When the measurement vector $\mathbf{z}_k$ is available, the predictive distribution (4.3.4) is updated to a posterior distribution with the Bayes rule [75]:

$$p(\mathbf{x}_k, \mathbf{R}_k|\mathbf{z}_k) \propto \frac{p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{R}_k)p(\mathbf{x}_k, \mathbf{R}_k|\mathbf{z}_{k-1})}{\int p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{R}_k)p(\mathbf{x}_k, \mathbf{R}_k|\mathbf{z}_{k-1})d\mathbf{x}_k d\mathbf{R}_k} \tag{4.3.5}$$

where $p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{R}_k)$ is the likelihood function. The integrations in the general solution are usually not analytically tractable due to the involved integrations [74]; in the following section, the recursion steps are solved by using a variational Baysian approximation step for the posterior update, which is accompanied by the suitable dynamics for the measurement noise covariance matrix [75].

4.3.2 Variational approximation

Given that the inverse-Gamma distribution is the conjugate prior distribution for the variance of a Gaussian distribution [77], a product of inverse-Gamma distributions can be adopted to approximate the posterior distribution for the measurement noise matrix $\mathbf{R}_k$ and the state model, in this case, assuming the posterior distribution at time $k-1$ can be represented by

$$\begin{aligned} p(\mathbf{x}_{k-1}, \mathbf{R}_{k-1}|\mathbf{z}_{k-1}) =& \mathcal{N}(\mathbf{x}_{k-1}, \mu_{k-1}, \mathbf{P}_{k-1}) \\ & \times \prod_{i=1}^{d} IG(\sigma^2_{k-1,i}|\alpha_{k-1,i}, \beta_{k-1,i}) \end{aligned} \tag{4.3.6}$$

where d is the dimension of the measurement noise vector and $IG(\sigma^2_{k-1,i}|\alpha_{k-1,i}, \beta_{k-1,i})$ denotes the inverse-Gamma distribution, which has the degrees of freedom parameter $\alpha_{k-1,i}$ and the scalar parameter $\beta_{k-1,i}$.

Because the dynamic models of the state and measurement noise variance are independent, the joint predictive distribution remains a factored form [78] of a Gaussian distribution for predicted state vector $\mathbf{x}_k$ with mean μ_k

and covariance matrix $\mathbf{P}_k$. It is assumed that the dynamic model for the parameters in the inverse Gamma distribution $p(\sigma_{k,i}^2|\sigma_{k-1,i}^2)$ are of such a form that each $\sigma_{k,i}^2$ will always results in an inverse Gamma distribution with parameters $\alpha_{k,i}$ and $\beta_{k,i}$ in the variational Bayesian prediction step. So the joint distribution can be predicted as

$$\begin{aligned} p(\mathbf{x}_k, \mathbf{R}_k|\mathbf{z}_k) &= p(\mathbf{x}_k|\mathbf{z}_{k-1})p(\mathbf{R}_k|\mathbf{z}_{k-1}) \\ &= N(\mathbf{x}_k, \mu_k, \mathbf{P}_k) \prod_{i=1}^{d} IG(\sigma_{k,i}^2|\alpha_{k,i}, \beta_{k,i}) \end{aligned} \tag{4.3.7}$$

where in the joint posterior distribution, the state and measurement noise covariance will be coupled with the likelihood function $p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{R}_k)$, which makes the exact posterior intractable.

The next step is to derive an analytical expression for the posterior distribution within the update equation; in order to make the computation tractable, an approximation of the posterior distribution is formed [76]. The standard VB approach is employed and a free form factored approximate distribution $p(\mathbf{x}_k, \mathbf{R}_k|\mathbf{z}_k)$ can be approximated as [75]

$$p(\mathbf{x}_k, \mathbf{R}_k|\mathbf{z}_k) \approx Q_{\mathbf{x}}(\mathbf{x}_k)Q_{\mathbf{R}}(\mathbf{R}_k) \tag{4.3.8}$$

where $Q_{\mathbf{x}}(\mathbf{x}_k)$ and $Q_{\mathbf{R}}(\mathbf{R}_k)$ are respectively a Gaussian distribution and inverse Gamma distribution as follows:

$$Q_{\mathbf{x}}(\mathbf{x}_k) = \mathcal{N}(\mathbf{x}_k, \mu_k, \mathbf{P}_k) \tag{4.3.9}$$

$$Q_{\mathbf{R}}(\mathbf{R}_k) = \prod_{i=1}^{d} IG(\sigma_{k,i}^2|\alpha_{k,i}, \beta_{k,i}) \tag{4.3.10}$$

Then the approximate posterior densities can be determined by minimizing the Kullback-Leibler(KL) divergence between the separable approximation

and the true posterior density, which is expressed as

$$\begin{aligned} &KL\{Q_{\mathbf{x}}(\mathbf{x}_k)Q_{\mathbf{R}}(\mathbf{R}_k)||p(\mathbf{x}_k,\mathbf{R}_k|\mathbf{z}_k)\} = \\ &\int Q_{\mathbf{x}}(\mathbf{x}_k)Q_{\mathbf{R}}(\mathbf{R}_k)\log\frac{Q_{\mathbf{x}}(\mathbf{x}_k)Q_{\mathbf{R}}(\mathbf{R}_k)}{p(\mathbf{x}_k,\mathbf{R}_k|\mathbf{z}_k)}d\mathbf{x}_k d\mathbf{R}_k \end{aligned} \tag{4.3.11}$$

In order to minimize the KL-divergence, methods from calculus of variations [74] are employed. Using alternating optimisation, the probability densities $Q_{\mathbf{x}}(\mathbf{x}_k)$ and $Q_{\mathbf{R}}(\mathbf{R}_k)$ are calculated in turn, while keeping the other fixed, yielding:

$$Q_{\mathbf{x}}(\mathbf{x}_k) \propto E\left\{\int \log p(\mathbf{z}_k,\mathbf{x}_k,\mathbf{R}_k|\mathbf{z}_{k-1})Q_{\mathbf{R}}(\mathbf{R}_k)d\mathbf{R}_k\right\} \tag{4.3.12}$$

$$Q_{\mathbf{R}}(\mathbf{R}_k) \propto E\left\{\int \log p(\mathbf{z}_k,\mathbf{x}_k,\mathbf{R}_k|\mathbf{z}_{k-1})Q_{\mathbf{x}}(\mathbf{x}_k)d\mathbf{x}_k\right\} \tag{4.3.13}$$

Since the two equations are coupled, they cannot be solved directly, however, computing the expectation of the first equation yields the following equation [74]

$$\begin{aligned} &\int \log p(\mathbf{z}_k,\mathbf{x}_k,\mathbf{R}_k|\mathbf{z}_{k-1})Q_{\mathbf{R}}(\mathbf{R}_k)d\mathbf{R}_k = \\ &-0.5(\mathbf{z}_k-\mathbf{H}_k\mathbf{x}_k)^T\langle\mathbf{R}_k^{-1}\rangle_{\mathbf{R}}(\mathbf{z}_k-\mathbf{H}_k\mathbf{x}_k) \\ &-0.5(\mathbf{x}_k-\mathbf{F}_k\mathbf{x}_{k-1})^T(\mathbf{P}_k^{-1})(\mathbf{x}_k-\mathbf{F}_k\mathbf{x}_{k-1})+C_1 \end{aligned} \tag{4.3.14}$$

where $\langle\cdot\rangle_{\mathbf{R}} = \int(\cdot)Q_{\mathbf{R}}(\mathbf{R}_k)d\mathbf{R}_k$ denotes the expected value with respect to the approximation distribution $Q_{\mathbf{R}}(\mathbf{R}_k)$ as (4.3.10) and C_1 denotes the terms independent of $\mathbf{x}_k$.

Similarly, the second expectation can be computed as follows [75]

$$\begin{aligned} \int \log p(\mathbf{z}_k,\mathbf{x}_k,\mathbf{R}_k|\mathbf{z}_{k-1})Q_{\mathbf{x}}(\mathbf{x}_k)d\mathbf{x}_k = &-\sum_{i=1}^{d}(\frac{3}{2}+\alpha_{k,i})\ln(\sigma_{k,i}^2) \\ &-\sum_{i=1}^{d}\frac{\beta_{k,i}}{\sigma_{k,i}^2}-\frac{1}{2}\sum_{i=1}^{d}\frac{\langle(\mathbf{z}_k-\mathbf{H}_k\mathbf{x}_k)_i^2\rangle_{\mathbf{x}}}{\sigma_{k,i}^2}+C_2 \end{aligned} \tag{4.3.15}$$

where $\langle\cdot\rangle_{\mathbf{x}} = \int(\cdot)Q_{\mathbf{x}}(\mathbf{x}_k)d\mathbf{x}_k$ denotes the expected value with respect to the approximation distribution $Q_{\mathbf{x}}(\mathbf{x}_k)$ as (4.3.9), C_2 denotes the terms independent of $\mathbf{R}_k$. Given the parameters $\mu_k, \mathbf{P}_k, \alpha_{k,i}$ and $\beta_{k,i}$ follow the Gaussian and inverse Gamma distribution as model (4.3.9) and (4.3.10) respectively, they are considered to be the solutions to the following coupled set of equations [75]:

$$\begin{aligned}
\mu_k &= \mu_{k-1} + \hat{\mathbf{P}}_k\mathbf{H}_k^T(\mathbf{H}^T\hat{\mathbf{P}}_k\mathbf{H}^T + \hat{\mathbf{R}}_k)^{-1}(\mathbf{z}_k - \mathbf{H}\hat{\mu}_k) \\
\mathbf{P}_k &= \hat{\mathbf{P}}_k - \hat{\mathbf{P}}_k\mathbf{H}^T(\mathbf{H}^T\hat{\mathbf{P}}_k\mathbf{H}^T + \hat{\mathbf{R}}_k)^{-1}\mathbf{H}^T\hat{\mathbf{P}}_k \\
\alpha_{k,i} &= \hat{\alpha}_{k-1,i} + \frac{1}{2} \\
\beta_{k,i} &= \hat{\beta}_{k-1,i} + \frac{1}{2}[(\mathbf{z}_k - \mathbf{H}\hat{\mu}_k)_i^2 + (\mathbf{H}\mathbf{P}_k\mathbf{H}^T)_{ii}]
\end{aligned} \tag{4.3.16}$$

where $(\cdot)_i$ denotes the i^{th} dimension of the vector and the estimated covariance matrix $\hat{\mathbf{R}}_k$ is

$$\hat{\mathbf{R}}_k = diag\left\{\frac{\beta_{k,1}}{\alpha_{k,1}}, ..., \frac{\beta_{k,d}}{\alpha_{k,d}}\right\} \tag{4.3.17}$$

where ' $\hat{\cdot}$ ' denotes the estimate of the parameters. Following (4.3.16) and (4.3.17), the process of variational Bayes measurement parameter updating can be described as in [78], which includes prediction and updating steps:

Prediction: Compute the parameters of the predicted distribution as follows:

$$\begin{aligned}
\hat{\alpha}_{k,i} &= \rho_i\alpha_{k-1,i} \\
\hat{\beta}_{k,i} &= \rho_i\beta_{k-1,i} \\
\hat{\mu}_k &= \mathbf{F}\hat{\mu}_{k-1} \\
\hat{\mathbf{P}}_k &= \mathbf{F}\hat{\mathbf{P}}_{k-1}\mathbf{F}^T + \mathbf{Q}_{k-1}
\end{aligned} \tag{4.3.18}$$

where $\rho_i \in (0, 1]$ is a scalar used for the extension of the noise fluctuations.

Updating: In the updating step, a fixed point iteration method is employed to achieve the best solution of the equations given in (4.3.16). First

set the initial parameters following (4.3.19)

$$\begin{aligned} \mu_k^0 &= \hat{\mu}_k \\ \mathbf{P}_k^0 &= \hat{\mathbf{P}}_k \\ \alpha_{k,i}^0 &= \frac{1}{2} + \hat{\alpha}_{k,i} \\ \beta_{k,i}^0 &= \hat{\beta}_{k,i} \end{aligned} \tag{4.3.19}$$

for $i = 1, ..., d$. Then use the fixed-point iteration to achieve the solution of (4.3.16) and (4.3.17) for ℓ steps and find the best solution as $\beta_{k,i}^{\ell}, m_k^{\ell}$ and $P_k = P_k^{\ell}$, then set $\beta_{k,i} = \beta_{k,i}^{\ell}, m_k = m_k^{\ell}$ and $P_k = P_k^{\ell}$, after obtaining the optimal solution for the equations, the parameters are updated within the measurement model of the particle filter which thereby helps calculate the weights of each particle.

In this way, the noise vector for the measurement model is estimated with the aid of the variational approach. Then it can be fed into the particle PHD framework and more accurate tracking results should ensue.

4.3.3 Variational Bayesian PHD recursion

The above subsection introduced the variational Bayesian approach for parameter estimation, in this subsection, the above approach is introduced into the PHD filter for multiple human tracking with unknown measurement noise variance, which is called VB-PHD.

In the prediction step of the VB-PHD filter, each survived target is assumed to follow a linear Gaussian model, and the birth intensity Υ_k in (4.2.4) is comprised of an inverse Gamma distribution and a Gaussian distribution, so coupled with $\Upsilon_k(\mathbf{X}_k, \mathbf{R})$, the joint predicted PHD at time k can be given

as

$$D(\mathbf{X}_k, \mathbf{R}_k|\mathbf{Z}_{k-1}) = \int \phi(\hat{\mathbf{x}}_k^i, \mathbf{R}_k|\mathbf{X}_{k-1}, \mathbf{R}_{k-1}) D(\mathbf{X}_{k-1}, \mathbf{R}_{k-1}|\mathbf{Z}_{k-1}) \delta\hat{\mathbf{x}}_k^i + \Upsilon_k(\mathbf{X}_k, \mathbf{R}_k) \tag{4.3.20}$$

and the parameters for the joint distribution are predicted following (4.3.18). After obtaining the predicted particle set associated with weights, the variational Bayesian approach is employed to estimate the measurement noise covariance matrix $\mathbf{R}_k$. By utilizing the fixed point iteration to find the best solution for the coupled equations (4.3.16), the measurement noise with covariance matrix $\mathbf{R}_k$ is employed to calculate the likelihood $g(\mathbf{z}_k|\hat{\mathbf{x}}_k^i)$, which is employed to calculate the weights for particles following the traditional particle PHD filter updating stage:

$$w_k^i = \left[p_M(\hat{\mathbf{x}}_k^i) + \sum_{\forall \mathbf{z}_k \in \mathbf{Z}_k} \frac{(1 - p_M(\hat{\mathbf{x}}_k^i) g(\mathbf{z}_k|\hat{\mathbf{x}}_k^i))}{\kappa_k + C_k(\mathbf{z}_k)} \right] \hat{w}_k^i \tag{4.3.21}$$

then the states and number of targets can be obtained as the traditional particle PHD filter.

In summary, the VB-PHD filter can be described by the pseudocode given in Algorithm 4.5.

The simulation results are presented in Section 4.4, which shows the outcome from the variational Bayesian approach.

4.4 Simulation experiments

The performance of the proposed algorithms is analyzed with the aid of two sequences from the CAVIAR and PETS2009 datasets, both sequences are recorded at 25 frames per second with an image size of 320×240 pixels. The Student's-t distribution and variational Bayesian based particle PHD

Algorithm 4.5 Variational Bayesian PHD filter

Input: $\{\mathbf{x}_{k-1}^m\}_{m=1}^{m=M_{k-1}}$.
Output: $\{\mathbf{x}_k^m\}_{m=1}^{m=M_k}$ with M_k targets.
1: Generate $\{x_{k-1}^i, w_{k-1}^i\}_{i=1}^{M_k \times N}$ from $\{\mathbf{x}_{k-1}^m\}_{m=1}^{m=M_{k-1}}$ and feed into (4.3.20).
2: Sample new-born particles following the Gaussian-inverse Gamma joint distribution.
3: Variational Bayesian parameters prediction following (4.3.18).
4: **for** $i = 1 : M_{k-1} \times N + J_k$ **do**
5: Given $\{\mathbf{P}_k^0, \alpha_k^0, \beta_k^0\}$ as the initialization parameters for variational Bayesian step.
6: Using a fixed point iteration to solve the coupled equations (4.3.16) and calculate $\mathbf{R}_k$.
7: Calculate likelihood for particle $g(\mathbf{z}_k|\hat{\mathbf{x}}_k^i)$ from measurement model by predicted $\mathbf{R}_k$.
8: Update particle weights with (4.3.21).
9: **end for**;
10: Calculate $M_k = \sum_{i=1}^{M_{k-1}\times N+J_k} w_k^i$.
11: Particle resampling.
12: Clustering $\{\mathbf{x}_k^i, \frac{1}{N}\}_{i=1}^{i=M_k\times N}$ and output $\{\mathbf{x}_k^m\}_{m=1}^{m=M_k}$

filters are compared with the traditional particle PHD filter for multiple human tracking proposed in [4]. The proposed method is specifically for tracking multiple human targets with appearing, disappearing and occlusion randomly in the scenario; both methods are tested to track the position of the human targets. All the parameters are chosen empirically to yield the best results.

4.4.1 State model

Based upon the concept of the Bayesian filtering framework, the states of the survived particles are estimated from the state model, which is described as function $e(\cdot)$ in the particle PHD filter. A rectangular region which contains the whole body of the human target is used to represent the area of the target. The pixel in the center of the rectangular is considered as the center of the human body. For the survived particles, the transition state model used

to test the proposed system is given as (4.3.1) and the transition function $\mathbf{F}$ is given as

$$\mathbf{F} = \begin{pmatrix} 1 & 0 & \Delta t & 0 & 0 & 0 \\ 0 & 1 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.4.1}$$

where Δt is the time interval between frame k and $k+1$ which is set as 1 in the simulations, the zero-mean noise vector $\mathbf{w}_k$ for prediction in the state model has covariance structure $cov\{\mathbf{w}_0\} = Diag\{25, 25, 16, 16, 4, 4\}$ and for $\mathbf{v}_0$ $cov\{\mathbf{v}_k\} = Diag\{25, 25\}$. The missed detection probability $p_M = 0.01$, the survival probability $e = 0.99$, the new born intensity $\Upsilon = 0.1$ and clutter intensity $\kappa = 0.01$, which are selected empirically to fit the situations in the video frames.

4.4.2 Likelihood model

In the Bayesian filtering concept, the likelihood model is important since it helps to calculate the weights for the particles. In this work, the likelihood model is obtained from background subtraction.

The codebook method is employed for background subtraction, which is robust to capture structural background motion over a long period of time under limited memory. In this method, samples at each pixel are clustered into the set of codewords based on a colour distortion metric together with brightness bounds. Not all pixels are represented with the same number of codewords. The background is encoded on a pixel-by-pixel basis. Background/foreground detection involves testing the difference of the current image from the background model with respect to colour and brightness differences. If an incoming pixel satisfies two conditions, it is classified as

background: first, the colour distortion to a codeword is less than the detection threshold; second, its brightness lies within the brightness range of that codeword. Otherwise, it is classified as foreground [79].

Some background subtraction results are shown in Fig. 4.2.

Figure 4.2. Examples of background subtraction results from the 'EnterExitCrossingPaths1cor' sequence of the CAVIAR dataset. The left, middle and right figures correspond to frame 188, 67 and 345 respectively. The green part in the left figure shows the occlusion of two human targets, which may cause missed detection; the red part in the middle figure shows the appearance of another human target and the yellow part in the right figure shows the disappearance of a human target. Noise components can be observed in the background subtraction results, including patches of salt and pepper noise, which may cause false alarms in the multiple human tracking.

The results can be used to select the new-born targets and build up an RFS for the measurement set. The center of each block $\mathbf{c}_k = [c_{k,x}, c_{k,y}]^T$ which contains the localization information, can be employed as one part of the measurement [4], so the likelihood for each particle based upon one measurement $\mathbf{z}_k$ from the foreground position RFS $g(\mathbf{c}_k|\hat{\mathbf{x}}_k^i)$ can be calculated as

$$g(\mathbf{z}_k|\hat{\mathbf{x}}_k^i) = e^{-\frac{(\mathbf{p}_k^i - \mathbf{c}_k)^T(\mathbf{p}_k^i - \mathbf{c}_k)}{\sigma_R^2}} \tag{4.4.2}$$

which shows the distance between the state of the particles and the foreground information, where $\mathbf{p}_k^i = [p_{k,x}^i, p_{k,y}^i]^T$ denotes the position of the targets taken from the particle $\tilde{\mathbf{x}}_k^i$ and σ_R is the standard deviation of the measurement model in the Bayesian filtering model.

4.4.3 Student's-t distribution based particle PHD filter results

The Student's-t distribution aided particle PHD filter is evaluated by two sequences, one from the CAVIAR dataset and one from the PETS2009 dataset, which contain human targets appearing, occluding and disappearing in the scenario. To make comparison, the number of particles are set to be 1000 and the OSPA measure is used for evaluation.

In order to select the most suitable value for the degrees freedom parameters v, as an example, the mean of the OSPA measure as a function of the freedom rate value are calculated by employing a sequence from the CAVIAR dataset, where $v = 1, ..., 5$ are set for comparison, which is shown in Table 4.1.

Table 4.1. Degrees of freedom parameter comparisons for Student's-t distribution based particle PHD filters

	$v = 1$	$v = 2$	$v = 3$	$v = 4$	$v = 5$
OSPA	24.31	23.48	18.97	20.58	22.27

The comparison shows that the value of degrees of freedom parameter chosen as 3 tends to provide a good compromise.

A performance comparison is presented between the proposed algorithm and the traditional particle PHD filter proposed in [4]. Both the algorithms are tested on the same video sequences to track the human targets. The OSPA measure comparison for the two sequences are shown in Fig. 4.3 and Fig. 4.4. It is clear from the results that the particle PHD filter with Student's-t distribution successfully tracks the targets which results in a smaller OSPA as compared to the traditional particle PHD filter.

To make clearer comparison, the mean of the OSPA value for both algorithms are compared, and the improvement from the Student's-t distribution is shown in Table 4.2, where PHD denotes the results from the traditional particle PHD filter and T-PHD denote the results from the Student's-t dis-

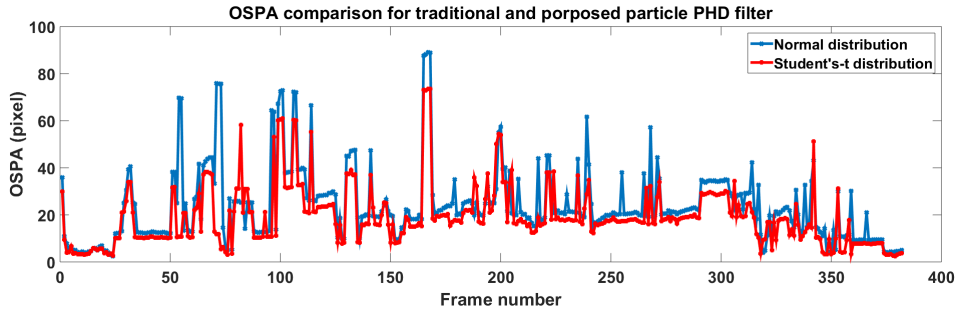


Figure 4.3. Comparison of OSPA measure between the proposed and traditional particle PHD filter with sequence from CAVIAR dataset, where the blue line denotes the OSPA value from the traditional particle PHD filter proposed in [4] and the red line denotes the OSPA value from the proposed Student's-t distribution based particle PHD filter.

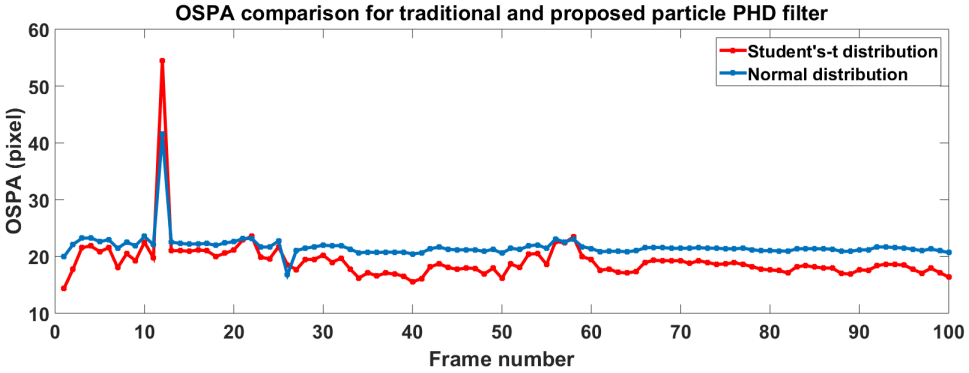


Figure 4.4. Comparison of OSPA measure between the proposed and traditional particle PHD filter with sequence from PETS2009 dataset, where the blue line denotes the OSPA value from the traditional particle PHD filter proposed in [4] and the red line denotes the OSPA value from the proposed Student's-t distribution based particle PHD filter.

tribution based particle PHD filter.

The comparisons show the improvement from the proposed algorithm, where for the sequences from the CAVIAR and PETS2009 datasets, the mean of the OSPA value reduced by 4.72 and 2.54 pixels respectively. The improvement is due to the heavier tails from the Student's-t distribution when compared with the Gaussian distribution, hence more robust predic-

Table 4.2. Comparison between the traditional and the proposed Student's-t distribution based particle PHD filter with sequences from CAVIAR and PETS2009 dataset

	CAVIAR	PETS2009
PHD	23.69	21.72
T-PHD	18.97	19.18
Improvement	19.92%	11.70%

tion for the target states are estimated, thus can obtain more accurate tracking results.

4.4.4 Variational Bayesian based particle PHD filter results

To evaluate the proposed variational Bayesian particle PHD filter, the same sequences used for evaluating the T-PHD filter are employed. Besides the parameters used for the traditional particle PHD filters, in the VB-PHD filter, other parameters are set as $\alpha_0 = 1, \beta_0 = 1$, the forgetting factor ρ is set to be 0.9 and the iteration step number $\ell = 20$. The OSPA comparisons are shown as Fig. 4.5, Fig. 4.6 and Table 4.3.

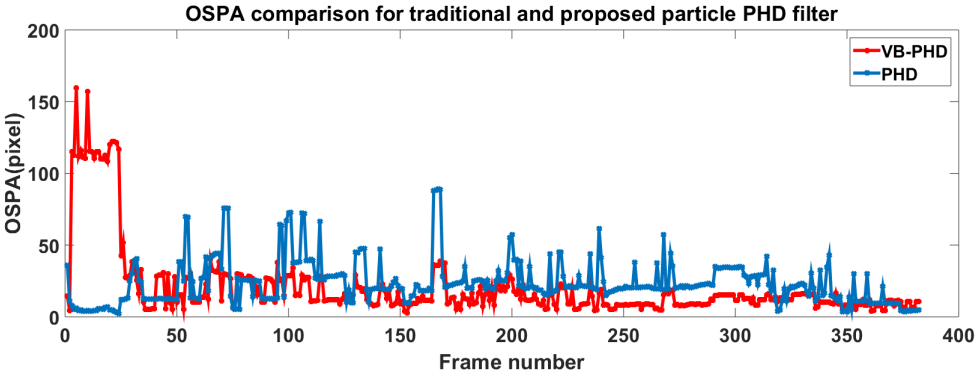


Figure 4.5. Comparison of OSPA measure between the proposed and traditional particle PHD filter with sequence from CAVIAR dataset, where the blue line denotes the OSPA value from the traditional particle PHD filter proposed in [4] and the red line denotes the OSPA value from the proposed variational Bayesian based particle PHD filter.

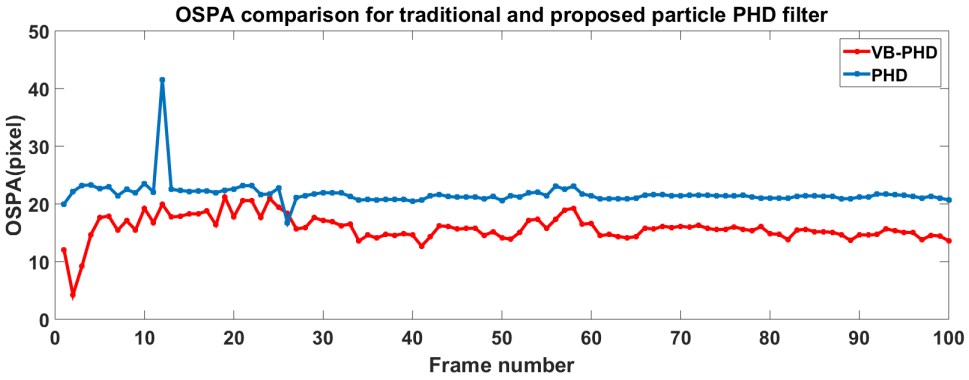


Figure 4.6. Comparison of OSPA measure between the proposed and traditional particle PHD filter with sequence from PETS2009 dataset, where the blue line denotes the OSPA value from the traditional particle PHD filter proposed in [4] and the red line denotes the OSPA value from the proposed variational Bayesian based particle PHD filter.

Table 4.3. Comparison between the traditional and the proposed Student's-t distribution based particle PHD filter with sequences from CAVIAR and PETS2009 dataset

	CAVIAR	PETS2009
PHD	23.69	21.72
VB-PHD	20.74	15.82
Improvement	12.45%	27.16%

Fig. 4.5, Fig. 4.6 and Table 4.3 shows that the proposed variational Bayesian particle PHD filter performs better than the traditional particle PHD filter, where for the sequences from the CAVIAR and PETS2009 datasets, the mean of the OSPA value reduced by 2.95 and 5.9 pixels respectively. The higher OSPA value at the first frames of the CAVIAR dataset is because when employing the variational Bayesian step, the parameters are estimated recursively. Since the OSPA values are calculated from the aspect of both error from location and error from the number of target, the proposed algorithm can be deduced to be more accurate with a lower OSPA value.

4.5 Summary

Two improved particle PHD algorithms were proposed in this chapter. The Student's-t distribution was firstly employed to replace the Gaussian distribution in the state and updating stages of the particle PHD filter to improve the robustness of the tracking system; the tracking results were compared with the results from the traditional particle PHD filter by sequences from CAVIAR and PETS2009 datasets. It was shown that the proposed algorithm provided better performance because of the heavier tails of the Student's-t distribution. Then the Student's-t distribution was exploited as a joint distribution of the measurement noise covariance and the state model, where the variational Bayesian approach was employed for parameters estimation. The OSPA evaluation also showed the improvement of the tracking performance, which is because more accurate parameters in the measurement model were employed. Both proposed methods helped to achieve better tracking results and reduced tracking error in both localization and cardinality aspects.

However, although the proposed tracking methods can obtain better tracking performance compared with the traditional one, they have limitations. In the measurement model, only the results from the background subtraction are employed because of the variational Bayesian step, which is because of the requirement of linear and Gaussian models. In human tracking work, human features can be employed as the measurement to calculate the likelihood of particles. Another limitation of the proposed algorithms is that only the forward filtering concept was employed. The next chapter proposes a new tracking framework where the forward and retrodiction steps are combined with an adaptive step to improve the tracking performance.

Chapter 5

ADAPTIVE RETRO-PHD FILTER FOR MULTIPLE HUMAN TRACKING

5.1 Introduction

In the previous chapters, the particle PHD filter for multiple human tracking has been introduced to address the problem of a variable number of targets. Chapter 3 provided the fundamental preliminary knowledge underlying Bayesian filtering algorithms. In Chapter 4, the Student's-t distribution was employed to obtain a more robust prediction and posterior distribution, and it was employed as a joint distribution of the measurement noise and state model, where the variational Bayesian approach was utilized to estimate the measurement noise variance. However, as mentioned in [80], the performance of the PHD filter depends on the current measurement set, so in the case of a low number of observable target measurements, the performance is limited. In order to achieve a more accurate measurement set, the delayed measurements have been considered by recent researchers, and relates to a smoothing algorithm. This chapter focuses on the third objective of this thesis, which is the adaptive Retro-PHD filter for multiple human tracking accepted by IEEE Signal Processing Letters.

When employing the smoothing algorithm, the backward estimation is employed with the aid of the delayed measurement [81]; in this way, more observable measurements are obtained, hence the accuracy of the tracking system can be potentially improved. The smoothing algorithm has been employed in many scenarios, such as the Kalman filter [82], particle filter [83] and PHD filter [84]. However, in [82], Psiaki has pointed out that, when using the smoothing algorithm for a nonlinear filtering system, it is necessary to consider a batch of data and minimize the aggregated error. As a consequence, when one attempts the backward processing with embedded approximations such as in PHD smoothing with a nonlinear model, the use of the term smoother should be avoided. In this case, the term retrodiction is adopted to represent the backward filtering process, namely the Retro-PHD in this thesis.

Adaptive filters [85, 86] on the other hand are a widely used signal processing technique for their exploitation of recursion in tracking [11]. Following the idea of the combination of adaptive filters proposed in [85], in this thesis, a new method for the Retro-PHD filter is proposed by using an adaptive recursion step, in which the measurements from both forward and backward processing are employed for target state estimation. In the adaptive step, forward and backward measurements are utilized to calculate the adaptive weights, which are then used to enhance the tracking results.

In this chapter, the concept of the backward smoothing algorithm is firstly introduced in Section 5.2; based upon the concept of backward filtering in the smoother, the Retro-PHD is proposed in Section 5.3. By combining the forward filtering algorithm with the backward retrodiction step by an adaptive weight, the adaptive retrodiction particle PHD filter is obtained, which will be described in Section 5.4. Evaluation and comparisons will be shown at the end of this chapter in Section 5.5, which show the improvement from the proposed tracking algorithm.

5.2 Smoothing algorithms

In previous chapters, the Bayesian filtering algorithms can be observed to provide an approximation of the distribution $p(\mathbf{x}_k|\mathbf{z}_k)$, where $\mathbf{x}_k$ is the single target state and $\mathbf{z}_k$ denotes the single measurement. However, the results from the forward filtering algorithms can be improved by extending the observable measurement; in order to address this challenge, delayed measurements are considered, which will be utilized in the backward stage, namely a smoothing type algorithm.

Given the current time k, and t denotes the time for the target state to be obtained. If $t > k$, the system is called prediction; the system is called filtering when $t = k$; and it is called smoothing in the case of $t < k$ [81]. There are three smoothing methods used widely by recent researchers: fixed-interval smoothing, fixed-point smoothing and fixed-lag smoothing. By using the fixed-interval smoothing algorithm, the density $p(\mathbf{x}_k|\mathbf{z}_k)$ is found at all time indices $t = 1, ..., k$, and is employed commonly in off-line systems; the fixed-point smoothing algorithm is concerned with the density at a fixed time t when k varies; when the fixed-lag smoothing algorithm is employed, the density at time $t = k - \ell$ is concerned, where ℓ is the fixed time lag. In this work, the fixed-lag smoothing algorithm is utilized to estimate the target states, which contains two main steps: forward filtering and backward smoothing.

The prediction and measurement updating models of the Bayesian model are given as

$$p_{k|k-1}(\mathbf{X}_k|\mathbf{Z}_{1:k-1}) = \int p_{k|k-1}(\mathbf{X}_k|\mathbf{X}_{k-1}) p_{k-1|k-1}(\mathbf{X}_{k-1}|\mathbf{Z}_{1:k-1}) d\mathbf{X}_{k-1} \tag{5.2.1}$$

and

$$p_{k|k}(\mathbf{X}_k|\mathbf{Z}_{1:k}) = \frac{p_k(\mathbf{Z}_k|\mathbf{X}_k) p_{k|k-1}(\mathbf{X}_k|\mathbf{Z}_{1:k-1})}{\int p_k(\mathbf{Z}_k|\mathbf{X}_k) p_{k|k-1}(\mathbf{X}_k|\mathbf{Z}_{1:k-1}) d\mathbf{X}_k} \tag{5.2.2}$$

where $\mathbf{X}_k$ denotes the multi-target states at time k, $\mathbf{Z}_{1:k}$ denotes the collected measurement up to time k; $p_k(\mathbf{Z}_k|\mathbf{X}_k)$ denotes the multi-target likelihood function and $p_{k|k-1}(\mathbf{X}_k|\mathbf{X}_{k-1})$ denotes the transition model from multi-target states at time $k-1$ to time k [81].

Similar to the forward Bayesian filtering algorithm, the formula for the backward smoothing step at time t where $t < k$ can be given as [81, 87]:

$$p_{t|k}(\mathbf{X}_t|\mathbf{Z}_{1:k}) = p_{t|t}(\mathbf{X}_t|\mathbf{Z}_{1:t}) \times \int \frac{p_{t+1|k}(\mathbf{X}_{t+1}|\mathbf{Z}_{1:k})p_{t+1|t}(\mathbf{X}_{t+1}|\mathbf{X}_t)}{p_{t+1|t}(\mathbf{X}_{t+1}|\mathbf{Z}_{1:t})} d\mathbf{X}_{t+1}$$
$$t = k-1, ..., k-\ell \tag{5.2.3}$$

where $p_{t|t}(\mathbf{X}_t|\mathbf{Z}_{1:t})$ is the density for multi-target state at time t from the forward filtering step [87]. In a Gaussian and linear model, the backward model (5.2.3) can be easily achieved, however, as mentioned in [82] when the model is non-linear and non-Gaussian, a computationally tractable approximation is necessary; in addition, within the smoothing algorithms, a batch of data should be considered to minimize the aggregated error over the batch at each time to mitigate the approximation error. In [82], Psiaki gives a solution for the smoothing algorithm with a non-linear model, namely the backward smoothing extended Kalman filter (BSEK). In the BSEK, given the Bayesian state and measurement model as

$$\mathbf{x}_{k|k-1} = \mathbf{f}(\mathbf{x}_{k-1|k-1}, \mathbf{w}_k) \tag{5.2.4}$$

where $(\cdot)_{k|k-1}$ denotes the estimated state value at time k from time $k-1$ and $\mathbf{w}_k$ is the prediction noise vector with covariance matrix $\mathbf{P}_k$. And the measurement model

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_{k|k-1}, \mathbf{v}_k) \tag{5.2.5}$$

where $\mathbf{h}$ is the measurement transition function and $\mathbf{v}_k$ is the measurement

noise vector with covariance matrix $\mathbf{R}_k$. Then the cost function for the BSEK can be calculated as [82]

$$J = \frac{1}{2}\sum_{t=0}^{k-1}\{\mathbf{w}_t^T\mathbf{Q}_t^{-1}\mathbf{w}_t+[\mathbf{z}_{t+1}-\mathbf{h}(\mathbf{x}_{t+1|t})]^T\mathbf{R}_{t+1}^{-1}[\mathbf{z}_{t+1}-\mathbf{h}(\mathbf{x}_{t+1|t})]\} + \frac{1}{2}(\mathbf{x}_0-\hat{\mathbf{x}}_0)^T\mathbf{P}_0^{-1}(\mathbf{x}_0-\hat{\mathbf{x}}_0) \tag{5.2.6}$$

where $\mathbf{Q}_t$ is the covariance matrix for the process noise, which is modeled as discrete time Gaussian white noise with zero mean; and the starting point of the filtering is $t = 0$ with a posterior state estimate $\hat{\mathbf{x}}_0$ and the posterior state estimation error covariance matrix $\mathbf{P}_0$ [82].

In this case, the target of the BSEKF is to find $\mathbf{x}_t$ given $\mathbf{x}_{1:k-1}$ and $\mathbf{w}_{1:k-1}$ by minimizing the cost function J. However, as it is shown in the function, the model is required to be Gaussian, moreover, in BSEKF, a batch filter is required to find the best solution for minimizing the cost function. Since the above challenges are difficult to be addressed in the particle PHD filter, the backward stage in the backward PHD filtering algorithm cannot be named as a smoother; in this case, instead of the term of smoothing algorithm, the term retrodiction is employed in this thesis, where the integration step is approximated by an importance sampling step following the concept of the particle filter and is employed in the PHD backward recursion, namely the retrodiction particle PHD filter (Retro-PHD).

5.3 Retrodiction Particle PHD filter

Following the concept of the smoothing algorithm, the tracking performance can be improved by employing a backward processing. A retrodiction step is also utilized in the PHD framework to use the delayed measurement, hence improving the accuracy of the tracking system.

5.3.1 Forward particle PHD filtering

In the particle Retro-PHD algorithm, the forward particle PHD filtering step is firstly employed, as introduced in Chapter 3, the main steps of the particle filtering are prediction and updating. In the prediction stage, the particle set contains the survived particles and the new-born particles, and the density PHD of the targets at time $k-1$ is

$$D_{k-1|k-1}(\mathbf{X}_{k-1}|\mathbf{Z}_{k-1}) = \sum_{i=1}^{L_{k-1}} w^i_{k-1|k-1}\delta(\mathbf{x}^i_{k-1|k-1}) \tag{5.3.1}$$

where L_{k-1} is the particle number at time $k-1$, which is also used as the number of survived particle number at time k in the prediction step. In this case, the forward PHD prediction stage can be given as

$$D_{k|k-1}(\mathbf{X}_k|\mathbf{Z}_{k-1}) = \sum_{i=1}^{L_{k-1}} w^i_{e|k-1}\delta(\mathbf{x}^i_{k|k-1}) + \sum_{i=1}^{J_k} w^i_{\Upsilon_k|k-1}\delta(\mathbf{x}^i_{\Upsilon_k|k}) \tag{5.3.2}$$

where e denotes the probability of the particles survived at time k and Υ_k denotes the probability for the new-born particles from the target state at time k. In this way, the predicted particle set with associated weights $\{\mathbf{x}^i_{k|k-1}, w^i_{k|k-1}\}_{i=1}^{L_{k-1}+J_k}$ can be calculated as

$$w^i_{k|k-1} = \begin{cases} e(\mathbf{x}^i_{k|k-1}|\mathbf{X}_k)w^i_{k-1} & i = 1, ..., L_{k-1} \\ \\ \frac{\Upsilon_k}{J_k} & i = L_{k-1}+1, ..., L_{k-1}+J_k \end{cases} \tag{5.3.3}$$

where J_k denotes the particle number used to represent the new born targets. The details have been described in Section 3.5 of Chapter 3.

After obtaining the predicted particle set, as introduced in Section 3.5, the PHD updating step is employed based upon the receipt of the measure-

ment set $\mathbf{Z}_k$ as:

$$w_{k|k}^i = \left[p_M(\mathbf{x}_{k|k-1}^i) + \sum_{\forall \mathbf{z}_k \in \mathbf{Z}_k} \frac{\psi_{k,\mathbf{z}_k}(\mathbf{x}_{k|k-1}^i)}{\kappa_k + C_k(\mathbf{z}_k)} \right] w_{k|k-1}^i \tag{5.3.4}$$

where

$$C_k(\mathbf{z}_k) = \sum_{i=1}^{L_{k-1}+J_k} \psi_{k,\mathbf{z}_k}(\mathbf{x}_{k|k-1}^i) w_{k|k-1}^i \tag{5.3.5}$$

and $\psi_{k,\mathbf{z}_k}(\mathbf{x}_{k|k-1}^i) = (1 - p_M(\mathbf{x}_{k|k-1}^i))g(\mathbf{z}_k|\mathbf{x}_{k|k-1}^i)$ is the single target likelihood. In this work, since the adaptive step will be employed with the aid of information from the human features, the likelihood of each particle is calculated by histograms of colour and oriented gradient features of human targets [69], which will be introduced in Section 5.5 later. By assuming that the noise on the colour and oriented gradient likelihood function is Gaussian

$$\begin{aligned} g(\mathbf{z}_k|\mathbf{x}_{k|k-1}^i) &\sim \mathcal{N}(\mathbf{z}_k; 0, \sigma_g^2) \\ &= \frac{1}{\sqrt{2\pi\sigma_g^2}} exp\left(-\frac{\{G(\mathbf{x}_{k|k-1}^i)\}^2}{2\sigma_g^2} \right) \end{aligned} \tag{5.3.6}$$

where σ_g^2 is the variance of the noise for the colour and gradient likelihood and $G(\mathbf{x}_{k|k-1}^i)$ is the colour similarities calculated as the Bhattacharyya distance between the reference measurement and the histogram of colour and oriented gradient $\mathbf{o}(\cdot)$ extracted from the rectangular area centered around the particle location, which can be calculated as

$$G(\mathbf{x}_{k|k-1}^i) = \sqrt{1 - \mathbf{o}(\mathbf{x}_{k|k-1}^i)^T \mathbf{z}_k} \tag{5.3.7}$$

After the updating step of the particle PHD filter, the number of targets is calculated by the sum of all the weights for particles as [60]

$$\tilde{M}_k = int\left(\sum_{i=1}^{L_{k-1}+J_k} w_{k|k}^i \right) \tag{5.3.8}$$

where $int(\cdot)$ takes the nearest integer. After calculating the number of targets, a resampling step is employed as described in [3] in order to limit the number of particles, which has been introduced in detail in Algorithm 3.4 in Chapter 3, thereby avoiding the number of particles growing exponentially and obtaining the resampled particle set $\{\tilde{\mathbf{x}}_k^i, \tilde{w}_k^i\}_{i=1}^{L_k}$. Then the tracking results $\{\tilde{\mathbf{x}}_k^m\}_{m=1}^{m=\tilde{M}_k}$ are obtained from the particle PHD filter, where $(\tilde{\cdot})$ denotes the states from the forward PHD filtering algorithm.

5.3.2 Particle Retro-PHD filtering

The Retro-PHD filter is employed to use more measurements beyond the current time by processing information from later stages in an approximate manner, and can potentially achieve more accurate tracking results. Similar to forward particle PHD filtering, the retrodiction step is also generalized by the RFS [59, 81]. By employing the concept of the fixed-lag smoothing algorithm, the backward retrodiction algorithm is concerned with the density at time $t = k-\ell$, where ℓ is the time lag. When employing the particle Retro-PHD filter, the retrodicted particle weights at time t are evaluated using the backward iterations using filter outputs $\{\tilde{\mathbf{x}}_k^i, \tilde{w}_k^i\}_{i=1}^{L_k}$, for $t = k-\ell, ..., k$. The particle weights from the backward retrodiction stage are computed as derived in [81]:

$$\hat{w}_{t|k}^i = \tilde{w}_{t|t}^i \left[e(\tilde{\mathbf{x}}_{t|t}^i) \sum_{q=1}^{L_t} \frac{w_{t+1|k}^q f_{t+1|t}(\tilde{\mathbf{x}}_{t+1|t+1}^q | \tilde{\mathbf{x}}_{t|t}^i)}{\mu_{t+1|t}^q} + (1 - e(\tilde{\mathbf{x}}_t^i)) \right] \quad (5.3.9)$$

where

$$\mu_{t+1|t}^q = \Upsilon_{t+1}(\tilde{\mathbf{x}}_{t+1|t+1}^q) + \sum_{r=1}^{L_t} \tilde{w}_{t|t}^r \times \{ e(\tilde{\mathbf{x}}_{t+1|t+1}^r) f_{t+1|t}(\tilde{\mathbf{x}}_{t+1|t+1}^q | \tilde{\mathbf{x}}_{t|t}^r) \} \quad (5.3.10)$$

and the conversion function $f_{t|t-1}(\cdot)$ is given as:

$$f_{t|t-1}(\tilde{\mathbf{x}}_t^i|\tilde{\mathbf{x}}_{t-1}^i) = \frac{exp\left(-\frac{(\tilde{\mathbf{x}}_t^i-\mathbf{F}(\tilde{\mathbf{x}}_{t-1}^i))^T(\tilde{\mathbf{x}}_t^i-\mathbf{F}(\tilde{\mathbf{x}}_{t-1}^i))}{2\sigma_f^2}\right)}{\sqrt{2\pi\sigma_f^2}} \tag{5.3.11}$$

where σ_f^2 is the variance of the conversion function and $\mathbf{F}(\cdot)$ is the state transformation matrix. After obtaining the particle set $\{\hat{\mathbf{x}}_t^i, \hat{w}_t^i\}_{i=1}^{i=N}$ from the particle Retro-PHD filter, the number and states of the human targets are obtained in the same way as in the forward particle PHD filter and in order to mitigate the effects of particle depletion, a resampling step is employed as described in the following subsection [84].

5.3.3 Particle Retro-PHD resampling

In order to address the challenge of particle depletion, a resampling step is employed after the retrodiction stage. Assuming the target density PHD from the retrodiction step at time t is given as

$$D_{t|k}(\mathbf{X}_t|\mathbf{Z}_{1:k}) = \sum_{i=1}^{L_{t|k}} w_{t|k}^i \delta(\mathbf{x}_{t|k}^i) \tag{5.3.12}$$

where $w_{t|k}^i$ contains the weights for the particles from both target PHD and the particles from missed detection. By employing the traditional resampling algorithms as described in Algorithm 3.1 introduced in Chapter 3, particles are resampled proportionally, so the particles with lower weights are rarely selected for resampling, which may incur losses in tracking in the backward retrodiction stage. To address this problem, a different resampling method is utilized for the retrodiction PHD as proposed in [84], where the particles from the target PHD and missed detection are resampled separately. Assuming $L_{t|k}$ particles are obtained from the backward stage of the Retro-PHD, which includes particles from both the missed detected parti-

cle set $\{\mathbf{x}^i_{m,t|k}, w^i_{m,t|k}\}_{i=1}^{L_{m,t|k}}$ and the particle set obtained from target PHD $\{\mathbf{x}^i_{d,t|k}, w^i_{d,t|k}\}_{i=1}^{L_{d,t|k}}$, where $L_{m,t|k}$ denotes the particle number for the missed detection and $L_{d,t|k}$ denotes the number of particles from the target PHD; in this way, the resampled approximation of the target density PHD can be described as

$$D_{t|k}(\mathbf{X}_t|\mathbf{Z}_{1:k}) = w_{d,t|k} \sum_{i=1}^{L_{d,t|k}} \delta(\mathbf{x}^i_{d,t|k}) + w_{m,t|k} \sum_{i=1}^{L_{m,t|k}} \delta(\mathbf{x}^i_{m,t|k}) \tag{5.3.13}$$

where the resampled weights for particles from target PHD $w_{d,t|k}$ and missed detection weights $w_{m,t|k}$ are given as

$$w_{d,t|k} = \sum_{i=1}^{L_{d,t|k}} w^i_{d,t|k}/L_{d,t|k} \quad w_{m,t|k} = \sum_{i=1}^{L_{m,t|k}} w^i_{m,t|k}/L_{m,t|k} \tag{5.3.14}$$

After the resampling step, the tracking results from the particle Retro-PHD filter are represented as $\{\hat{\mathbf{x}}^m_k\}_{m=1}^{m=\hat{M}_k}$. The comparison between the forward particle PHD filter and the Retro-PHD filtering algorithm will be shown in Section 5.5, which shows the improvement from the Retro-PHD.

When evaluating the Retro-PHD algorithm, for multiple human tracking, although the approach can improve the tracking results over the PHD filter, its performance deteriorates with an increasing number of human targets appearing and disappearing in the monitored area. In such a scenario, more false measurements for delayed estimation are introduced by false alarm or missed detection. Moreover, the measurement noise can also reduce the accuracy of the backward Retro-PHD. To address these challenges, an adaptive solution is employed, where the forward and retrodiction results are combined adaptively to improve the tracking performance.

5.4 Adaptive solution for particle Retro-PHD filter

From the above steps, results from both the forward and backward filtering processes are obtained, in which the forward measurements are utilized in the filtering algorithm and backward measurements are utilized in the retrodiction algorithm to estimate the target states, which are represented with the black lines and the blue lines in the graphical representation in Fig. 5.1 respectively. However, as mentioned in Section 5.3.2, the accuracy of the

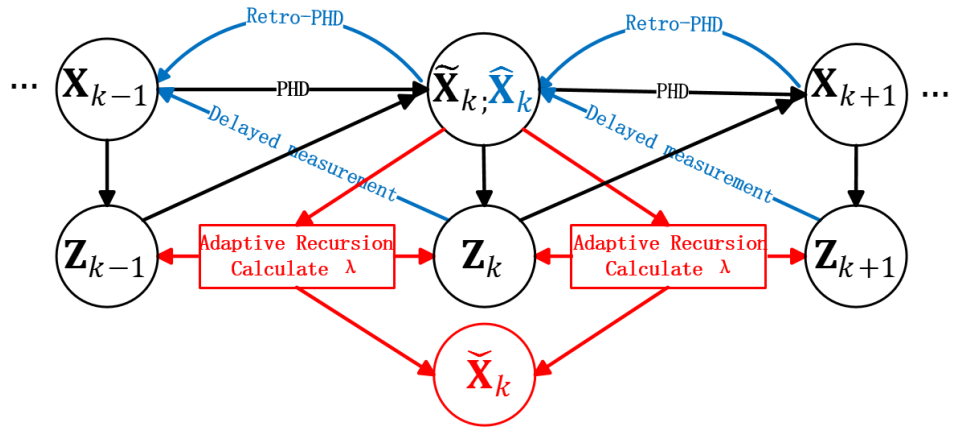


Figure 5.1. Graphical comparison between PHD filtering, Retro-PHD filtering and the proposed adaptive Retro-PHD filtering algorithms, where the black lines denote the PHD filtering algorithm, the blue lines denote the Retro-PHD filtering algorithm and the red lines denote the proposed adaptive retrodiction step.

backward state estimation from the Retro-PHD filter is limited because of the limitation of the accuracy of the delayed measurements. When the number of targets changes and the environmental noise increases, the delayed measurements are easily influenced by missed detection and false alarms, which will cause false measurements, and hence reduce the accuracy of the Retro-PHD filter. To address this issue, an adaptive step is designed for combining the forward and retrodiction state estimation. As shown in the red lines of Fig. 5.1, an adaptive scalar parameter λ is employed to weight

the results given from the forward filtering and retrodiction filtering algorithms, which is calculated by the similarity of the measurement set over discrete time samples. Assuming the measurement set at times $k-1$, k, $k+1$ are $\mathbf{Z}_{k-1}$, $\mathbf{Z}_k$ and $\mathbf{Z}_{k+1}$ respectively, which are generated by the target states being tracked, and the measurement RFS is extracted from the RFS of the tracking results, which includes M_k targets. Adaptive parameters $\lambda_o,\ o \in \{filtering,\ retrodiction\}$ are calculated as:

$$\lambda_{filtering} = \frac{\sum_{i=1}^{M_k}\sum_{r=1}^{M_{k-1}} exp\left(-\frac{(\mathbf{z}_k^i-\mathbf{z}_{k-1}^r)^T(\mathbf{z}_k^i-\mathbf{z}_{k-1}^r)}{2\sigma_\lambda^2}\right)}{M_{k-1}} \tag{5.4.1}$$

$$\lambda_{retrodiction} = \frac{\sum_{i=1}^{M_k}\sum_{r=1}^{M_{k+1}} exp\left(-\frac{(\mathbf{z}_k^i-\mathbf{z}_{k+1}^r)^T(\mathbf{z}_k^i-\mathbf{z}_{k+1}^r)}{2\sigma_\lambda^2}\right)}{M_{k+1}} \tag{5.4.2}$$

and by normalizing $\lambda_{filtering}$ and $\lambda_{retrodiction}$, the weight value for the forward and backward measurements is given as:

$$\lambda = \frac{\lambda_{filtering}}{\lambda_{filtering}+\lambda_{retrodiction}}. \tag{5.4.3}$$

which gives convex weights to the results from both tracking and filtering. Thus the tracking position from the adaptive step is found by using a convex combination of results from both filtering and retrodiction as:

$$\check{\mathbf{x}}_k^m = \begin{cases} \tilde{\mathbf{x}}_k^m \quad if \quad target \quad m \quad disappears \quad at \quad k+1 \\ \\ \lambda\tilde{\mathbf{x}}_k^m + (1-\lambda)\hat{\mathbf{x}}_k^m \quad otherwise \end{cases} \tag{5.4.4}$$

where $\check{(\cdot)}$ denotes the results from the adaptive recursion retrodiction step. The convex combination is used because it is a simple and intuitive way for fusing the information and provides flexibility to automatically control the contribution of the forward and backward information adaptively. In this way, the filtering results are employed to make corrections for the results

from the Retro-PHD filter based on the similarity between forward and delayed measurements, which reduces the probability of the false measurements caused by missed detection and false alarms.

In summary, for time $k > 1$, the adaptive particle Retro-PHD filter can be described as Algorithm 5.6.

Algorithm 5.6 Adaptive Retro-PHD filter

Input: $\{\mathbf{x}_{k-1}^i, w_{k-1}^i\}_{i=1}^{i=N}$.
Output: $\{\check{\mathbf{x}}_{k-1}^m\}_{m=1}^{m=M_{k-1}}$

Forward Filtering
Select new-born particles from background subtraction.
Particle prediction by (5.3.2).
Obtain prediction weights by (5.3.3).
for $i = 1 : L_{k-1} + J_k$ **do**
    Calculate $g(\mathbf{z}_k|\tilde{\mathbf{x}}_k^i)$ by (5.3.6).
    Update particle weights with (5.3.4).
end for
Calculate target number by (5.3.8).
Resample updated particles and discard J_k particles.
Data association for survived and new-born particles.
Clustering with K-means and obtain $\{\tilde{\mathbf{x}}_k^m\}_{m=1}^{m=M_k}$.
Backward Retrodiction
for $i = 1 : L_t$ **do**
    if $\mathbf{x}_t^i \in$ *survived particles* **then**
        Calculate $f(\cdot)$ following (5.3.11).
    end if
    Calculate retrodiction weight with (5.3.9).
end for
Clustering with K-means and obtain $\{\hat{\mathbf{x}}_t^m\}_{m=1}^{m=M_t}$.
Adaptive Recursion
Obtain the measurement set $\mathbf{Z}_k$.
for $m = 1 : M_t$ **do**
    Calculate filtering and retrodiction weight λ by (5.4.3).
    Make correction for tracking position with λ by (5.4.4).
end for
Clustering with K-means and obtain $\{\check{\mathbf{x}}_t^m\}_{m=1}^{m=M_t}$.

The above steps give the adaptive solution for the particle Retro-PHD. Simulation results will be given in Section 5.5, where the comparison will confirm the improvement of performance from the adaptive step.

5.5 Simulation

In order to evaluate the performance of the proposed adaptive particle Retro-PHD filter, sequences from the CAVIAR and PETS2009 datasets are employed, where in the CAVIAR dataset, 3-5 human targets are walking in a shopping mall environment and in the PETS2009 dataset, 3-6 human targets are walking in a campus environment, and these include human target occlusion, appearing and disappearing randomly in the scene. In this work, 1000 particles are employed to represent targets in the CAVIAR dataset and 1500 particles are employed in the PETS2009 dataset; 200 particles are employed for new-born targets in each frame.

5.5.1 State model and measurement model

Following previous experience, the zero-mean noise vector $\mathbf{w}_k$ for prediction in the state model has covariance structure $cov\{\mathbf{w}_k\} = diag\{25,$ $25, 16, 16, 4, 4\}$ and for $\mathbf{v}_k$, $cov\{\mathbf{v}_k\} = diag\{25, 25\}$. The missed detection probability $p_M = 0.01$, the survival probability $e = 0.99$, the new born intensity is given as $\Upsilon = 0.1$ and clutter intensity $\kappa = 0.01$. The variance of the conversion function σ_f^2 and λ function σ_λ^2 are set empirically to be 25 and 25 respectively. In order to reduce the computational complexity, the time lag ℓ is set to be 1. The state transformation matrix $\mathbf{F}$ is given as

$$\mathbf{F} = \begin{pmatrix} 1 & 0 & \Delta t & 0 & 0 & 0 \\ 0 & 1 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{5.5.1}$$

where Δt is the time interval between time k and $k-1$.

As introduced in Section 5.3, the measurement contains the histograms of colour and oriented gradient features of human targets, which is employed in the likelihood calculation of each particle.

The histogram of oriented gradient feature is employed in this thesis because: 1) Capture edge or gradient structure is characteristic of local shape; 2) It is relatively invariant to local geometric and photometric transformations; 3) The spatial and orientation sampling densities can be tuned for application. As proposed in [88], the histogram of oriented gradient (HOG) was designed to describe the human feature in video frame by gradient information. As introduced in [88], the basic idea of HOG is to describe the local object appearance and shape by the distribution of local intensity gradients of edge directions. The main steps for calculating the HOG are given as:

- Colour normalisation;
- Computing the gradients;
- Orientation binning, where each pixel votes for an orientation according to the closest bin in the range;
- Gradients normalization;
- Collect HOG over detection window.

In this thesis, the bin number of HOG windows per bound box is chosen as 3 and the number of histogram bins is set as 9.

Besides HOG, the colour histogram is also employed in this work, since the colour of the human targets and background are different, it can be used to classify the human target from the image. As described in [89], scaled versions of red (R), green (G) and blue (B) colours are used to obtain the colour histogram for human targets, where R-G and G-R are employed to represent the chrominance information and the luminance information is

represented by the R+G+B. In this work, $16 \times 16 \times 16$ histogram bins are used to model the colour information.

At the initial frame of the video sequence, the reference measurement $\mathbf{s}$ includes the histogram of colour and oriented gradient. When a particle $\mathbf{x}^i_{k|k-1}$ is obtained, the Bhattacharyya distance $G(\mathbf{x}^i_{k|k-1})$ between the reference measurement and the image patch extracted from the states described in the predicted particle as

$$G(\mathbf{x}^i_{k|k-1}) = \sqrt{1 - \mathbf{o}(\mathbf{x}^i_{k|k-1})^T \mathbf{z}_k} \tag{5.5.2}$$

and the likelihood model is given as

$$\begin{aligned} g(\mathbf{z}_k|\mathbf{x}^i_{k|k-1}) &\sim \mathcal{N}(\mathbf{z}_k; 0, \sigma_g^2) \\ &= \frac{1}{\sqrt{2\pi\sigma_g^2}} exp\left(-\frac{\{G(\mathbf{x}^i_{k|k-1})\}^2}{2\sigma_g^2}\right) \end{aligned} \tag{5.5.3}$$

where σ_g^2 is the variance of the noise, which is given as 25 empirically.

The above settings give the state and measurement models of the adaptive Retro-PHD algorithm. In the next subsection, the simulation as well as comparison between PHD filtering, Retro-PHD filtering and adaptive Retro-PHD are given.

5.5.2 Retro-PHD results

To evaluate the Retro-PHD algorithm, sequences from the CAVIAR and PETS2009 datasets are employed, where the mean of Euclidean error (MEE) between tracking results and ground truth and OSPA are utilized for comparison. The MEE comparison results are shown in Table 5.1 while the comparison for OSPA are shown in Fig. 5.2 and 5.3.

The comparisons show the improvement from the Retro-PHD over the traditional PHD filtering algorithm. However, although Table 5.1 shows the

Table 5.1. MEE comparison between the forward PHD filtering and the Retro-PHD filtering algorithms with sequences from CAVIAR and PETS2009 dataset.

	CAVIAR	PETS2009
PHD	34.85	43.08
Retro-PHD	28.43	36.49
Improvement	18.42%	15.30%

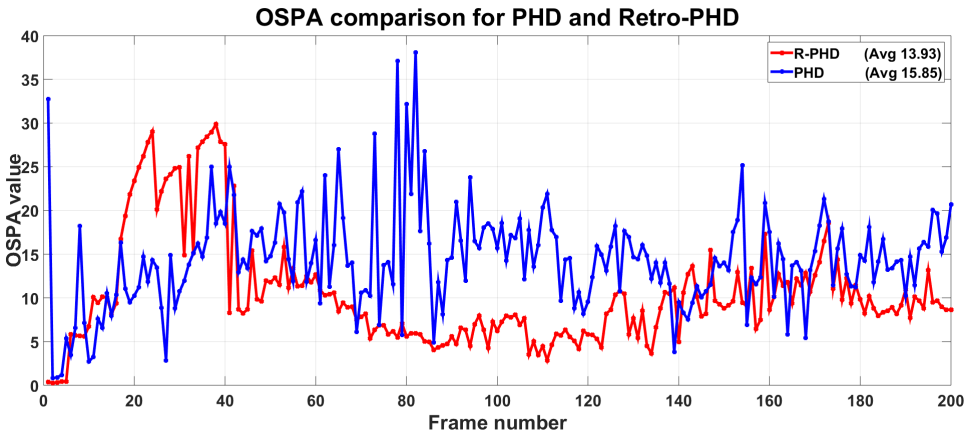


Figure 5.2. Comparison of OSPA measure between the traditional particle PHD and Retro-PHD algorithms with sequence from CAVIAR dataset, where the blue line denotes the OSPA value from the traditional particle PHD filter proposed in [4] and the red line denotes the OSPA value from the Retro-PHD.

improvement by comparing the MEE in each frame, when evaluating by OSPA measure, the improvement from the Retro-PHD filtering algorithm is limited, whereas for the CAVIAR dataset shown in Fig. 5.2, the mean of the OSPA value reduces from 15.95 to 13.93 and for the PETS2009 dataset shown in Fig. 5.3, the mean value reduces from 13.69 to 13.66. The reason for this poor improvement is that the accuracy of the delayed measurement is limited because of the missed detection and false alarms caused by measurement noise and variable number of targets. To make further comparison, sequence 'OneLeaveShopReenter2cor' is also employed to evaluate the Retro-

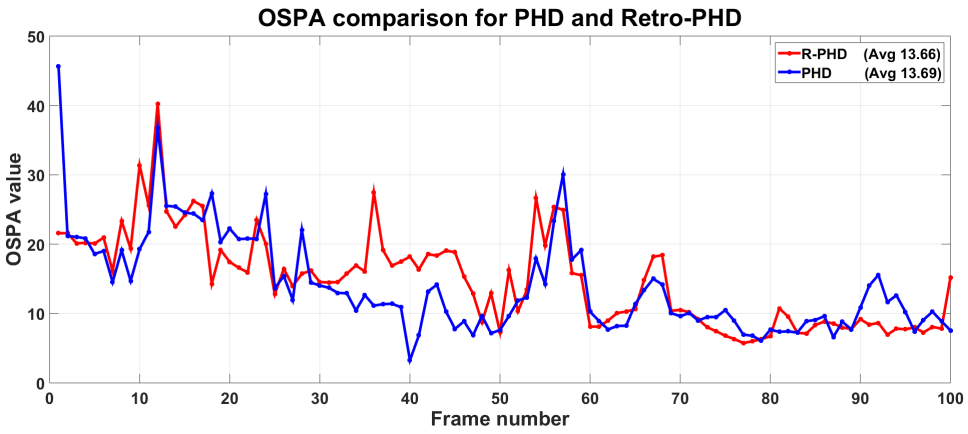


Figure 5.3. Comparison of OSPA measure between the traditional particle PHD and Retro-PHD algorithms with sequence from PETS2009 dataset, where the blue line denotes the OSPA value from the traditional particle PHD filter proposed in [4] and the red line denotes the OSPA value from the Retro-PHD.

PHD filtering algorithm, for which the OSPA evaluation result is shown as Fig. 5.4, where the average of the OSPA value reduces from 19.59 to 18.39, which shows the improvement from the backward retrodiction step of the Retro-PHD filtering algorithm.

In order to mitigate the influence caused by the delayed measurement, the adaptive step is employed with the aid of the similarity of measurement, and the simulations given in the next subsection confirms the improvement from the proposed algorithm.

5.5.3 Adaptive Retro-PHD results

In order to obtain further improvement, as introduced in Section 5.4, the adaptive step is employed after the retrodiction step of Retro-PHD filtering. To evaluate the proposed adaptive Retro-PHD, the same sequences used for evaluating the Retro-PHD filtering algorithm are employed. As introduced in Section 5.4, human features including histograms of colour and

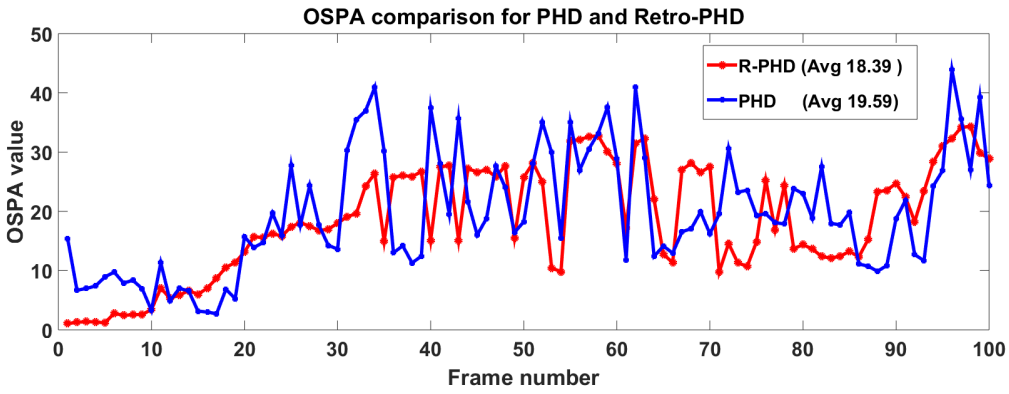


Figure 5.4. Comparison of OSPA measure between the traditional particle PHD and Retro-PHD algorithms with sequence 'OneLeaveShopReenter2cor' from CAVIAR dataset, where the blue line denotes the OSPA value from the traditional particle PHD filter proposed in [4] and the red line denotes the OSPA value from the Retro-PHD.

oriented gradient are employed to calculate the adaptive parameter λ. The improvement shown by MEE evaluation results are given in Table 5.2 where Retro-PHD denotes the results from the Retro-PHD filtering algorithm and A-PHD denotes the results from the proposed adaptive Retro-PHD filtering method. In order to evaluate the proposed method from both localization and cardinality aspects, the OSPA measure is employed, which are shown in Fig. 5.5 and Fig. 5.6. The traditional PHD filter and the Retro-PHD are used as baselines for comparison, which show the improvement from the adaptive step.

Table 5.2. MEE comparisons for PHD, Retro-PHD and Adaptive Retro-PHD.

	CAVIAR			PETS2009		
	PHD	Retro-PHD	A-PHD	PHD	Retro-PHD	A-PHD
ME (pixel)	34.85	28.43	25.71	43.08	36.49	34.95
Improvement (%)	-	18.42%	26.23%	-	15.30%	18.87%

From the comparison, it is clear that for the CAVIAR dataset, the av-

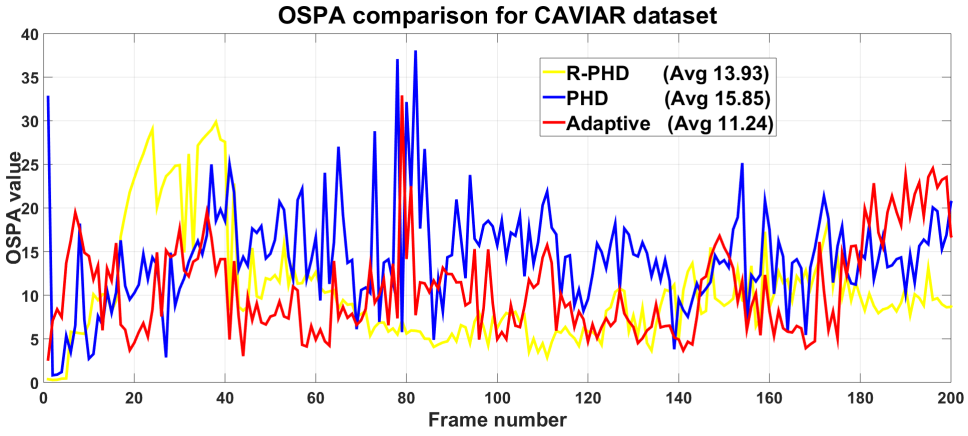


Figure 5.5. Comparison of OSPA measure between the traditional particle PHD and Retro-PHD algorithms with sequence from CAVIAR dataset, where the blue line denotes the OSPA value from the traditional particle PHD filter proposed in [4] and the red line denotes the OSPA value from the Retro-PHD.

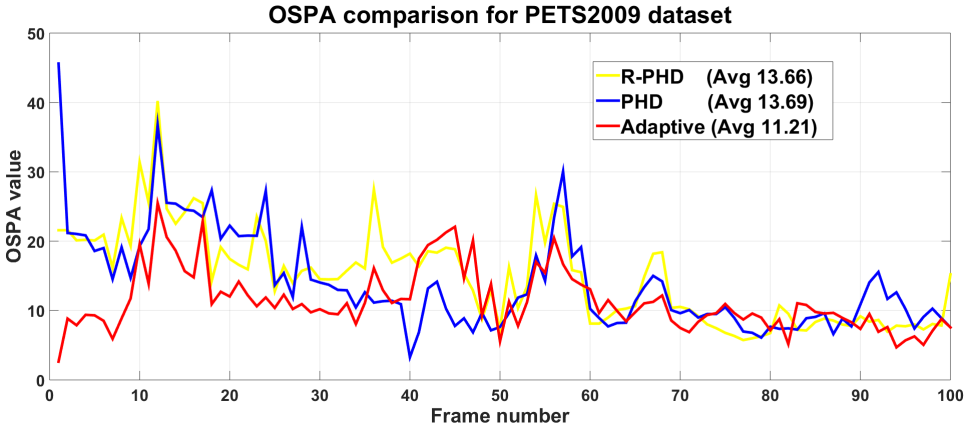


Figure 5.6. Comparison of OSPA measure between the traditional particle PHD and Retro-PHD algorithms with sequence from PETS2009 dataset, where the blue line denotes the OSPA value from the traditional particle PHD filter proposed in [4] and the red line denotes the OSPA value from the Retro-PHD.

erage value of OSPA is reduced by 4.61 and 2.69 when compared with the PHD and Retro-PHD, respectively. For the PETS2009 dataset, the average

value of OSPA is reduced by 2.48 and 2.45 respectively. The peak value around frame 80 of the CAVIAR dataset from the adaptive Retro-PHD is caused by the change of the number of targets and occlusion. To make further comparison, sequence 'OneLeaveShopReenter2cor' is also employed to evaluate the Retro-PHD filtering algorithm, for which the OSPA evaluation result is shown as Fig. 5.7, where the red line denotes the OSPA value from the proposed adaptive Retro-PHD filtering algorithm and also confirms the improvement over the PHD and Retro-PHD filtering algorithms.

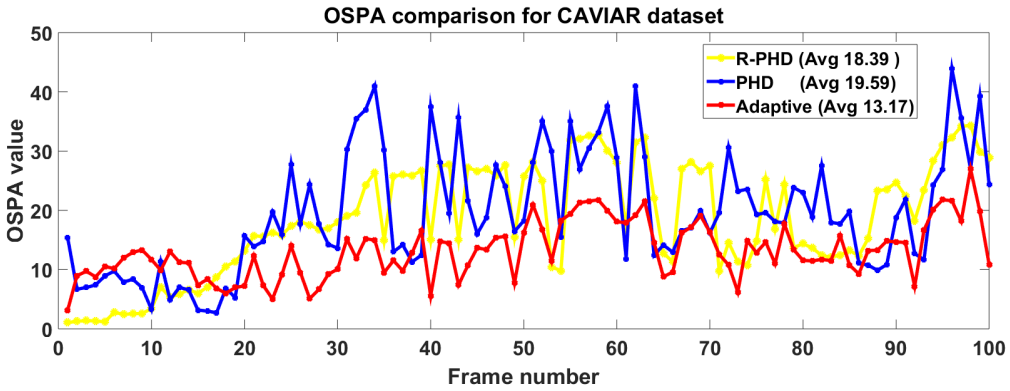


Figure 5.7. Comparison of OSPA measure between the traditional particle PHD and Retro-PHD algorithms with 'OneLeaveShopReenter2cor' sequence from CAVIAR dataset, where the blue line denotes the OSPA value from the traditional particle PHD filter proposed in [4] and the red line denotes the OSPA value from the Retro-PHD.

When employing the adaptive recursion step, the weights for the measurements from filtering and retrodiction are given based upon the observation extracted from the state, hence can address the challenges caused by false measurements caused by time-varying number of targets, missed detections and false alarms. Because of this step, the tracking system becomes more accurate as verified by these experiments.

5.6 Summary

In this chapter, the backward retrodiction approach was employed for multiple human tracking based upon the concept of smoothing algorithm. The fundamental knowledge of the smoother was introduced firstly at the beginning of this chapter, however, since the recently proposed PHD-smoother lacks the processing of employing a batch of data at each step to mitigate the approximation error, which is necessary in nonlinear model, the term of retrodiction PHD filtering was used to replace the term of PHD smoother. By employing the backward retrodiction step, delayed measurement set was employed to obtain more accurate tracking results, the improvement from the Retro-PHD was confirmed by the comparison results shown in Section 5.5.2. Moreover, since in multiple human tracking work, the accuracy could be influenced by the limitation of the accuracy of delayed measurement, an adaptive step was proposed to combine the forward and retrodiction step with a convex function, where the convex parameter was achieved by calculating the similarity of the measurement set over discrete samples. The evaluation results given in Section 5.5.3 showed the improvement when compared with the forward PHD Retro-PHD algorithms.

In the next chapter, to obtain further improvement, contributions from both prediction and updating models are considered. In the prediction stage, human behaviour is considered to describe the transition function, namely the social force model. An MCMC resampling approach is also employed to aid in the prediction of more robust particles for the PHD filter. Moveover, an OCSVM classifier which is trained by features from both colour and oriented gradient histogram is utilized to mitigate measurement noise in background subtraction, thereby further reducing the probability of false alarms and hence improving the performance of the PHD filter.

Chapter 6

SOCIAL FORCE MODEL BASED MCMC-OCSVM PARTICLE PHD FILTER FOR MULTIPLE HUMAN TRACKING

6.1 Introduction

In this chapter, improvements to the particle PHD filter are made both in terms of the prediction step and updating step. In the prediction step of the particle PHD filter, a novel exponential term based social force model is employed to describe the human behaviour, which aids the estimating of the state of the human target. Moreover, to further improve the prediction accuracy, the predicted particles are resampled by an MCMC step, in which the acceptance ratio is obtained with the aid of the results from the social force model. In the updating step, the background subtraction approach is firstly used as the measurement; however, since the noise from the background subtraction may cause false detection, a one-class support vector machine (OCSVM) is employed to mitigate the noise, which is trained by

human features including histograms of colour and oriented gradient. This chapter focuses on the fourth and fifth objectives of this thesis, which relate to the social force model aided robust particle PHD filter for multiple human tracking published in [90] and the social force model based MCMC-OCSVM particle PHD filter for multiple human tracking submitted to IEEE Transactions on Multimedia [91].

In this Chapter, to make the notations in the later sections clearer, the particle PHD filter framework is briefly described at the beginning, which has been described in detail in Chapter 3. Then the social force model is firstly introduced in Section 6.3; after introducing the traditional social force model, an exponential-term social force model is proposed and is employed to calculate the likelihood in the MCMC resampling step as detailed in Section 6.4. After obtaining a more robust predicted particle set, the OCSVM is trained and used to calculate the likelihood model in the updating step of the particle PHD filter, which is explained in Section 6.5. The above contributions are then added in the particle PHD filter framework and introduced as pseudocode in Section 6.6. At the end of this chapter, simulations and comparisons as well as the failure cases of the proposed tracking system are given in Section 6.7, which confirm the improvement from the proposed social force model based MCMC-OCSVM particle PHD filter.

6.2 Adapted particle PHD filter

Assuming the target set $\{\mathbf{x}_k^m\}_{m=1}^{m=M_k}$ includes the states of all the human targets, where $\mathbf{x}_k^m = [p_{k,x}^m, p_{k,y}^m, v_{k,x}^m, v_{k,y}^m, h_k^m, w_k^m]^T \in \mathbb{R}^6$ denotes the state of the m-th target at discrete time k, including the 2D position $(p_{k,x}^m, p_{k,y}^m)$, velocity $(v_{k,x}^m, v_{k,y}^m)$, height and width of targets h_k^m, w_k^m; where $(\cdot)^T$ denotes the transpose operator and subscripts x, y are the horizontal and vertical coordinates of the target; M_k is the number of targets at time k. Denote

the measurement set at time k as $\mathbf{Z}_k$, which includes $\mathbf{z}_k$ for each target and $\mathbf{X}_k = \{\mathbf{x}_k^{m,i}, m = 1, ..., M_k, i = 1, ..., N\}$, which denotes all the particles utilized to describe the states of all human targets at time k, where N is the number of particles employed to describe the state of a target; in this chapter, the notation with superscript $(\cdot)^{m,i}$ denotes a particle with index i employed to describe the state of the target with index m. Given a set of targets with states at time $k-1$, $\{\mathbf{x}_{k-1}^m\}_{m=1}^{m=M_{k-1}}$, the set of predicted particles and the associated weights from the state model at time k is given by [16]

$$\{\tilde{\mathbf{x}}_k^{m,i}, w_{k-1}^{m,i}\}_{m=1,i=1}^{m=M_k,i=N} \tag{6.2.1}$$

As has been introduced in Chapter 3 in detail, the prediction and updating step for the particle PHD filter can be described as follows:

1. Prediction: Particles $\tilde{\mathbf{x}}_k^{m,i}$ are drawn from the predicted particle set and fed into the prediction model of the particle PHD filter, which is described as [60]

$$D(\mathbf{X}_k|\mathbf{Z}_{k-1}) = \int \phi(\tilde{\mathbf{x}}_k^{m,i}|\mathbf{X}_{k-1}) D(\mathbf{X}_{k-1}|\mathbf{Z}_{k-1}) \delta\tilde{\mathbf{x}}_k^{m,i} + \Upsilon_k(\mathbf{X}_{k-1}) \tag{6.2.2}$$

where Υ_k is the intensity function of the new target birth RFS, $\phi(\tilde{\mathbf{x}}_k^{m,i})$ is the analogue of the state transition probability in the single target case which is calculated from

$$\phi(\tilde{\mathbf{x}}_k^{m,i}|\mathbf{X}_{k-1}) = e(\tilde{\mathbf{x}}_k^{m,i}|\mathbf{X}_{k-1}) + \beta(\tilde{\mathbf{x}}_k^{m,i}|\mathbf{X}_{k-1}) \tag{6.2.3}$$

in which $e(\cdot)$ is the probability that the target still exists at time k and $\beta(\cdot)$ is the intensity of the RFS for spawned targets. When exploiting the PHD filter with the particle filter, the PHD of states is represented by the weights of the particles, which include the survived particles and new-born particles.

In the traditional particle PHD filter, the particles employed to describe

the new-born targets are selected randomly in the scene; however, in human tracking, the new-born targets can be obtained by employing a background subtraction step. With this method, the particle set, which includes particles for both survived targets and new born targets can be obtained

$$\{\tilde{\mathbf{x}}_k^i, \tilde{w}_{k|k-1}^i\}_{i=1}^{i=(M_{k-1}+J_k)\times N} \tag{6.2.4}$$

where i is the index of the i-th particle and J_k is the number of new blocks from the background subtraction, which is assumed to be the number of new born targets in the prediction step. The weights obtained from the prediction step are given as

$$\tilde{w}_{k|k-1}^i = \begin{cases} \phi(\tilde{\mathbf{x}}_k^i) w_{k-1}^i & i = 1, ..., M_{k-1} \times N \\ \\ \frac{\Upsilon_k}{J_k \times N} & i = M_{k-1} \times N + 1, ..., (M_{k-1} + J_k) \times N \end{cases} \tag{6.2.5}$$

In this way, the predicted PHD $D(\mathbf{X}_k|\mathbf{Z}_{k-1})$ at time k for target states $\mathbf{X}_k$ is obtained based on the weights of the particles.

2. Measurement update: The updating step of the particle PHD filter is defined as [60]: Once the new set of observations is available, the weights of each particle are updated based upon the receipt of the measurement $\mathbf{Z}_k$ as [60]

$$\tilde{w}_k^i = \left[p_M(\tilde{\mathbf{x}}_k^i) + \sum_{\forall \mathbf{z}_k \in \mathbf{Z}_k} \frac{\psi_{k,\mathbf{z}_k}(\tilde{\mathbf{x}}_k^i)}{\kappa_k + \sum_{i=1}^{(M_{k-1}+J_k)\times N} \psi_{k,\mathbf{z}_k}(\tilde{\mathbf{x}}_k^i) \tilde{w}_{k|k-1}^i} \right] \tilde{w}_{k|k-1}^i \tag{6.2.6}$$

Then the number of targets is calculated by the sum of all the weights for particles as follows [60]

$$\tilde{M}_k = \sum_{i=1}^{(M_{k-1}+J_k)\times N} \tilde{w}_k^i \tag{6.2.7}$$

$$M_k = int(\tilde{M}_k) \tag{6.2.8}$$

where $int(\cdot)$ takes the integer nearest to $\tilde{M}_k$. The above steps constitute the main steps of the particle PHD filter, and the details have been given in Chapter 3.

6.3 Social force model

The interaction information is exploited by a model constraining the motion between the targets in this thesis. Within existing interaction models, the social force model (SFM) [5] is considered due to its ability in handling the interaction between human targets as well as their typical behaviour. Several researchers have used social force models to predict the states of humans based on their behaviour [5,6]. Within the social force model, the behaviour of human targets is modelled via energy potentials which are adjusted by other targets and obstacles through repulsive forces [92].

Given a current set of target states $\{\mathbf{x}_k^m\}_{m=1}^{m=M_k}$ based on the position, velocity and walking behaviour of each target including their destination and avoiding collision with others [6], it is assumed in a social force model that every human target knows its current position and velocity, as well as its destination. In addition, it has social force with other targets if they are closer in distance than a pre-defined threshold. It is also often assumed that each target will predict the movement of other targets via a constant velocity model. Thus, the position information $\mathbf{p}_k^m = [p_{k,x}^m,\, p_{k,y}^m]^T$ and the velocity information $\mathbf{v}_k^m = [v_{k,x}^m,\, v_{k,y}^m]^T$, from the state of target $\mathbf{x}_k^m$ at time k, can be used to represent the social force between the targets. The social force model for target m is calculated between target m and all other targets. For example, the social force between targets m and n ($n \neq m$) is calculated based upon the following parameters: the distance and angular displacement between m and n: $d_k^m(n)$ and $A_k^m(n)$; the change of velocity compared

with target m: U_k^m and the cosine of the angle between the velocity and destination path of target m: W_k^m [5].

The distance $d_k^m(n)$ can be calculated as [6]

$$d_k^m(n) = \|\mathbf{p}_k^m + t\mathbf{v}_k^m - \mathbf{p}_k^n - t\mathbf{v}_k^n\| \tag{6.3.1}$$

where $\|\cdot\|$ denotes the Euclidean norm and t is the time interval between frame $k-1$ and k. Since each target is assumed to intend to avoid collisions with other targets, the angular displacement between the velocity of the two targets is also considered as one of the important parameters for the social force model, which can be represented as factor $A_k^m(n)$ [5]

$$A_k^m(n) = 1 + \frac{(\mathbf{v}_k^m)^T\mathbf{v}_k^n}{\|\mathbf{v}_k^m\|\|\mathbf{v}_k^n\|} \tag{6.3.2}$$

In the social force model, each target m is assumed to walk towards a destination $\mathbf{p}_o^m = [p_{o,x}^m, p_{o,y}^m]^T$, and in doing so tries to maintain a desired speed $\mathbf{u}^m = [u_x^m, u_y^m]^T$. These two components can be described as two energy functions U_k^m and W_k^m, which denote the change of velocity and cosine of the angle between the current velocity and destination path for target m respectively

$$U_k^m = \|(\mathbf{v}_k^m - \mathbf{u}^m)\| \tag{6.3.3}$$

$$W_k^m = \frac{(\mathbf{p}_o^m - \mathbf{p}_k^m)^T\mathbf{v}_k^m}{\|\mathbf{p}_o^m - \mathbf{p}_k^m\|\|\mathbf{v}_k^m\|} \tag{6.3.4}$$

where $\mathbf{v}_k^m$ denotes the velocity of target m at time k.

After calculating the above parameters, the overall social force for target m at time k can be written as [6]

$$S_k^m = \sum_{n \neq m} d_k^m(n) A_k^m(n) + \lambda_1 U_k^m + \lambda_2 W_k^m \tag{6.3.5}$$

where λ_1 and $\lambda_2 \in \mathbb{R}^+$ control the influence of the two regularizers. After

the social force is obtained for each target, it can be incorporated into the prediction step of the particle filter.

6.4 Exponential-term based social force model aided MCMC resampling

In order to achieve more robust predicted particle set, an exponential-term is employed to describe the social force model, and the output is used as the likelihood model to calculate the acceptance ratio in the MCMC resampling step.

6.4.1 Exponential-term based social force model

In this work, an exponential-term based energy function similar to that in [6] is exploited to describe the social force model for the likelihood calculation in the prediction stage of the MCMC-PHD filter. When a particle $\tilde{\mathbf{x}}_k^{m,i}$ is predicted to represent the state of target m, $\mathbf{x}_k^m$, at time k, its weight is predicted by the social force model representing interactions with other existing targets. Based upon (6.3.1), the distance between particle $\tilde{\mathbf{x}}_k^{m,i}$ from target m and the state $\mathbf{x}_k^n$ of target n can be used within an energy term

$$E_{k,d}^{m,i}(n) = e^{\frac{d_k^{m,i}(n)}{2\sigma_d^2}} \tag{6.4.1}$$

where σ_d controls the influence of the distance factor (denoted by subscript d) on the social force model. So the larger the distance between the predicted particle and the selected target, the higher the energy from the distance aspect, and $E_{k,d}^{m,i}(n)$ becomes minimum if the linear trajectories collide with each other. In this work, obstacles in the scenes are also considered; the states of which are considered as targets with velocity $\mathbf{v}_k^m = [0, 0]^T$ to calculate the social force model for each particle. Since the pedestrians will

change their speed and angular velocity in order to avoid collision with others, by employing (6.3.2), the angular displacement factor for the social force model can be represented as

$$E_{k,\phi}^{m,i}(n) = (A_k^{m,i}(n))^{\beta} \tag{6.4.2}$$

where β controls the influence from the direction of the velocity and the subscript ϕ is used to represent angular displacement. Based on (6.4.1) and (6.4.2), the influence of multiple subjects can now be modeled as a weighted product. For example where particle $\tilde{\mathbf{x}}_k^{m,i}$ is assigned an energy with respect to each target n $(n \neq m)$ depending on its current distance and angular displacement ϕ [6] becomes

$$E_k^{m,i}(n) = E_{k,d}^{m,i}(n) E_{k,\phi}^{m,i}(n) \tag{6.4.3}$$

Two energy functions which denote the change of velocity and cosine of the angle between current velocity and destination path for particle $\mathbf{x}_k^{m,i}$ respectively can also be represented:

$$E_{k,U}(m,i) = e^{-\frac{U_k^{m,i}}{2\sigma_v^2}} \tag{6.4.4}$$

$$E_{k,W}(m,i) = e^{-\frac{W_k^{m,i}}{2\sigma_D^2}} \tag{6.4.5}$$

where σ_v and σ_D control the influence of changing the velocity and destination on the social force of the target respectively.

To represent the state of target m, the overall interaction energy for particle $\mathbf{x}_k^{m,i}$ is predicted as

$$S_k^{m,i} = \prod_{n \neq m} E_k^{m,i}(n) E_{k,U}(m,i) E_{k,W}(m,i) \tag{6.4.6}$$

where the calculation of $S_k^{m,i}$ is different from those in [5] and [6] where a

sum function instead of a product function was used.

The above equations can be used as the social force weight functions for establishing a posterior distribution within the prediction stage of the particle PHD filter.

By calculating the social force from other targets, the estimated weight for prediction $s_k^{m,i}$ can be obtained by normalizing $S_k^{m,i}$. After calculating the social force, the particle set with their social force model results is obtained, which is used to derive the likelihood for particles in the MCMC resampling step described in the next subsection, and thereby achieve more accurate prediction for particles.

In order to make a clear comparison between the social force model proposed in this work and two existing social force models, the proposed social force model is compared with the two early proposed social force models proposed in [5] and [6], where their similarities, differences, advantages and disadvantages are considered: In all these approaches, interaction forces between the targets are used to simulate the dynamic model of pedestrians; distance and angle between the targets, change of velocity and destination of individual targets are considered in the models; and the models are designed to use domain knowledge and thereby improve the performance of multiple human tracking. The major differences are that in [5] a sum function is used to combine the components of the social force model, but in [6] an exponential-term model is used and, a summation form is used to combine the model parameters. The approach instead uses a product function as described by (6.4.6) and the influence of each model parameter is controlled by the variance terms in the exponential models. Moreover, a threshold is introduced to avoid calculating social forces when two targets are a large distance apart. The model in [5] has the advantage that it is simple, but the model in [6] is more flexible offering better tracking accuracy. Equation (6.3.5) also allows the influence from different model components to be more

easily matched to different environments. The model in this work offers further improvement in accuracy in more complicated environments and these are demonstrated in Table 6.4 in the simulation section. Finally, in terms of disadvantages, the approach in [5] has much less flexibility for use in different environments than the proposed approach and that in [6].

6.4.2 Social force model based MCMC resampling

In the traditional particle filter, an importance function is used in the sample selection step [23, 52], however, the MCMC resampling step replaces the importance sampling step by building a Markov chain which exploits the posterior distribution [52], and thereby improves diversity among particles. In this work, the MCMC resampling step is employed to improve the accuracy of the prior distribution, where the social force model is utilized to replace the likelihood function in the traditional MCMC particle filter [93].

As described in [56], during the MCMC resampling, a particle $\tilde{\mathbf{x}}_k^{m,i}$ is propagated to a new state $\tilde{\mathbf{x}}_k^{m,i*}$ based on the following model

$$\tilde{\mathbf{x}}_k^{m,i*} = \tilde{\mathbf{x}}_k^{m,i} + \mathbf{q} \tag{6.4.7}$$

where $\mathbf{q}$ denotes a zero-mean Gaussian noise vector. From the Metropolis-Hastings acceptance probability [52], the acceptance ratio is calculated as

$$\alpha = min\left\{1, \frac{p(\mathbf{z}_k|\tilde{\mathbf{x}}_k^{m,i*})p(\tilde{\mathbf{x}}_k^{m,i*}|\mathbf{x}_{k-1}^m)q(\tilde{\mathbf{x}}_k^{m,i}|\tilde{\mathbf{x}}_k^{m,i*})}{p(\mathbf{z}_k|\tilde{\mathbf{x}}_k^{m,i})p(\tilde{\mathbf{x}}_k^{m,i}|\mathbf{x}_{k-1}^m)q(\tilde{\mathbf{x}}_k^{m,i*}|\tilde{\mathbf{x}}_k^{m,i})}\right\}. \tag{6.4.8}$$

Since in this work, $q(\cdot|\tilde{\mathbf{x}}_k^{m,i})$ is symmetric in its arguments, that is:

$$q(\tilde{\mathbf{x}}_k^{m,i*}|\tilde{\mathbf{x}}_k^{m,i}) = q(\tilde{\mathbf{x}}_k^{m,i}|\tilde{\mathbf{x}}_k^{m,i*}) \tag{6.4.9}$$

in this case, the acceptance ratio α can be calculated as:

$$\alpha = min\left\{1, \frac{p(\mathbf{z}_k|\tilde{\mathbf{x}}_k^{m,i*})p(\tilde{\mathbf{x}}_k^{m,i*}|\mathbf{x}_{k-1}^m)}{p(\mathbf{z}_k|\tilde{\mathbf{x}}_k^{m,i})p(\tilde{\mathbf{x}}_k^{m,i}|\mathbf{x}_{k-1}^m)}\right\}. \tag{6.4.10}$$

In this work, the likelihoods of particle state $p(\mathbf{z}_k|\cdot)$ are replaced by the results obtained from the social force model, thus

$$\alpha = min\left\{1, \frac{s_k^{m,i*}p(\tilde{\mathbf{x}}_k^{m,i*}|\mathbf{x}_{k-1}^m)}{s_k^{m,i}p(\tilde{\mathbf{x}}_k^{m,i}|\mathbf{x}_{k-1}^m)}\right\}. \tag{6.4.11}$$

The state to be preserved is determined by drawing a point j from a uniform distribution. If $j < \alpha$ then the new state $\mathbf{x}_k^{m,i*}$ is retained, otherwise it is rejected. In this way, the social force model is fed into the MCMC resampling step for achieving more robust prediction.

Fig. 6.1 shows the steps of the social force model aided MCMC resampling step with an example target m. As shown in Fig. 6.1, a particle $\tilde{\mathbf{x}}_k^{m,i}$

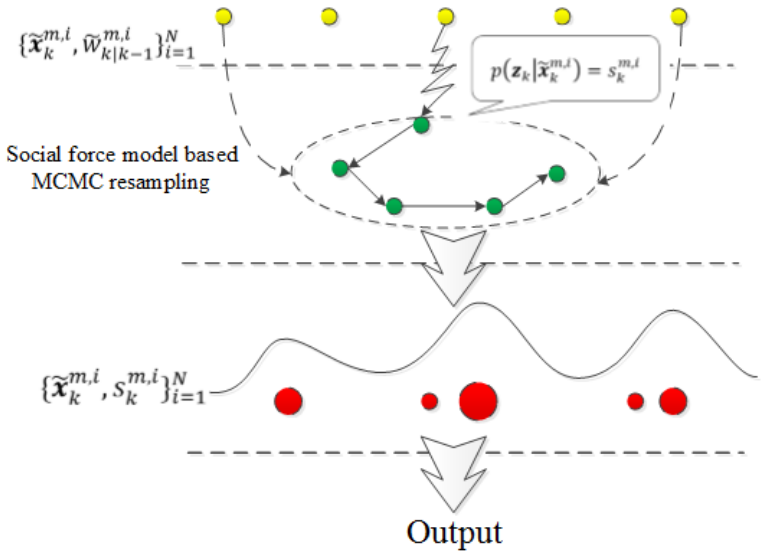


Figure 6.1. The basic operation of the proposed social force model-aided MCMC based particle filter.

predicted from the state model is chosen as the initial value of the Markov

chain. Then the posterior density is obtained in the form of weighted samples [93], where the predicted weight $\tilde{w}^i_{k|k-1}$ is obtained from the social force model described in Section 6.4.1. After calculating the acceptance ratio α for each particle and discarding the burn-in samples [23], the prior distribution $p(\mathbf{X}_k)$ can be obtained from the MCMC prediction, which can be utilized by the particle PHD updating step. The example pseudocode of this MCMC particle filter for target m is then summarised as Algorithm 6.7, where the inputs are the predicted particles for target m and the output is the posterior distribution from the prediction stage, and B denotes the number of burn-in period particles.

Algorithm 6.7 Social force model based MCMC resampling step (SFM-MCMC)

Input: Predicted particles for target m from state model $\{\tilde{\mathbf{x}}_k^{m,i}\}_{i=1}^{i=N}$
Output: Particles with predicted weights from the social force model aided MCMC resampling $\{\tilde{\mathbf{x}}_k^{m,i}, \tilde{w}_{k|k-1}^{m,i}\}_{i=1}^{i=N}$

Initialize the Markov chain by the predicted particles from the state transition function using the states of target at $k-1$.
for $i = 1{:}N+B$ **do**
    Propagate $\tilde{\mathbf{x}}_k^{m,i*}$ from $\tilde{\mathbf{x}}_k^{m,i}$ with (6.4.7) .
    Calculate $s_k^{m,i}$ and $s_k^{m,i*}$ for $\tilde{\mathbf{x}}_k^{m,i}$ and $\tilde{\mathbf{x}}_k^{m,i*}$ with (6.4.6).
    Compute α with (6.4.11).
    Draw a point j from a uniform distribution.
    if $j < \alpha$ **then**
        retain the new state: $\tilde{\mathbf{x}}_k^{m,i} = \tilde{\mathbf{x}}_k^{m,i*}$.
    else reject the new state.
    end if
end for
Discard the first B particles of the iterations.

6.5 One class support vector machine for likelihood calculation

Besides the state model, another important step is the measurement model for particle updating. In this work, two main steps are employed to obtain a robust measurement model: background subtraction and a one class support vector machine.

As introduced in Chapter 4, the codebook method is employed for background subtraction. The results can be used to select the new-born targets and build up an RFS for the measurement set. The center of each block $\mathbf{c}_k = [c_{k,x}, c_{k,y}]^T$ which contains the localization information, can be employed as one part of the measurement [4], so the likelihood for each particle based upon the foreground position $g_b(\mathbf{c}_k|\tilde{\mathbf{x}}_k^i)$ can be calculated as

$$g_b(\mathbf{c}_k|\tilde{\mathbf{x}}_k^i) = e^{-\frac{(\mathbf{p}_k^i - \mathbf{c}_k)^T(\mathbf{p}_k^i - \mathbf{c}_k)}{\sigma_R^2}} \tag{6.5.1}$$

which shows the distance between the state of the particles and the foreground information, where $\mathbf{p}_k^i = [p_{k,x}^i, p_{k,y}^i]^T$ denotes the position of the targets taken from the particle $\tilde{\mathbf{x}}_k^i$ and σ_R is the standard deviation of the measurement model in the Bayesian filtering model.

However, the raw background subtraction results generally contain many artifacts, which include small 'salt and pepper' terms and large noise patches caused by the problem of poor illumination and similar colour between the foreground and background. The noise patches may be regarded as new born targets in the prediction step of the PHD filter and cause the occurrence of false alarms. To address this issue, in this work, an OCSVM classifier [94] is used to distinguish the human targets from noise as described next.

The basic idea of the OCSVM is that given a data set drawn from an underlying probability distribution p, the OCSVM estimates a function f to describe its 'support region' (where a sample of p most likely comes from), where the corresponding values of the function f are larger than a particular threshold value [69].

To design the classifier, based on a training dataset, the following quadratic

optimisation problem needs to be solved:

$$\begin{aligned} \min_{\mathbf{w},\varsigma,\rho} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{\nu L}\sum_{i=1}^{L}\varsigma_i - \rho \\ \text{subject to} \quad & (\mathbf{w}^T\Phi(\tilde{\mathbf{x}}_k^i)) \geq \rho - \varsigma_i, \quad \xi_i \geq 0 \end{aligned} \tag{6.5.2}$$

where $\mathbf{w}$ is the normal vector, $\nu \in (0, 1]$, ρ is from the Lagrangian model of the SVM, which is set to be zero in this work and the nonzero slack variables $\varsigma = [\varsigma_1, ..., \varsigma_L]$ are introduced to allow for the possibility of outliers (the data points which are not drawn from the supporting region) and $\Phi(\cdot)$ is a nonlinear kernel function which maps the original data into a different space for better separation. For a test particle $\tilde{\mathbf{x}}_k^i$, the decision function for estimating whether it comes from the determined distribution is:

$$f(\tilde{\mathbf{x}}_k^i) = (\mathbf{w}^T\Phi(\tilde{\mathbf{x}}_k^i)) - \rho \tag{6.5.3}$$

In the application of multiple human tracking, the features from both colour and oriented gradient [88,95] of multiple human regions are employed for training the OCSVM classifier, which can be used to estimate the likelihood function value for each particle. Given a particle $\tilde{\mathbf{x}}_k^i$ at time instance k, the features from both the colour and oriented gradient histogram are extracted based upon the position, width and height information of $\tilde{\mathbf{x}}_k^i$ and the corresponding likelihood function, $\vartheta_k(\tilde{\mathbf{x}}_k^i)$ can be estimated as:

$$\vartheta_k(\tilde{\mathbf{x}}_k^i) = e^{(\varpi \cdot f(\tilde{\mathbf{x}}_k^i))} \tag{6.5.4}$$

where ϖ is a constant set for calculating the weights for the particles, thereby controlling the influence of the sub-likelihood from the OCSVM. Its value is chosen empirically in this work. In this way, the likelihood for each particle is obtained and these weights can then be taken as the input to the updating step of the PHD filter.

Thus, the likelihood model of the proposed particle PHD filter $g(\mathbf{z}_k|\tilde{\mathbf{x}}_k^i)$ is calculated based upon the results from both background subtraction and the OCSVM:

$$g(\mathbf{z}_k|\tilde{\mathbf{x}}_k^i) = \vartheta_k(\mathbf{x}_k^i) g_b(\mathbf{c}_k|\tilde{\mathbf{x}}_k^i) \tag{6.5.5}$$

By feeding (6.5.5) into (6.2.6), the weights of the particles are updated. Then the number of human targets and the particles are resampled as described in Section 6.2.

6.6 Summary of the social force model aided MCMC-OCSVM particle PHD filter

A summary of the proposed system is given in Algorithm 6.8, which is referred to as SFM-MCMC-OCSVM-PHD.

Algorithm 6.8 Social force model-aided MCMC-OCSVM particle PHD filter (SFM-MCMC-OCSVM-PHD)

Input: Video sequence with ℓ frames.
Output: $\{\mathbf{x}_k^m\}_{m=1}^{M_k}$ and M_k.

1: OCSVM classifier training.
2: Initialize targets states in the first frame $\{\mathbf{x}_1^m\}_{m=1}^{m=M_0}$.
3: **for** $k = 2{:}\ell$ **do**
4: Background subtraction to extract the measurement set $\mathbf{Z}_k$ for targets and the estimated positions of the new-born targets.
5: Predict particles for both survived targets and new born targets separately.
6: Calculate social force s_k^i for each particle.
7: SFM-MCMC resampling with Algorithm 6.7.
8: Calculate $g(\mathbf{z}_k|\tilde{\mathbf{x}}_k^i)$ by (6.5.1), (6.5.4) and (6.5.5).
9: Update the PHD weights with (6.2.6).
10: Calculate M_k by (6.2.8).
11: Particle resampling.
12: Output tracking results at time k, $\{\mathbf{x}_k^m\}_{m=1}^{M_k}$ and M_k.
13: **end for**

In the next section, the simulation results and comparisons with baseline methods will be given and the performance of the proposed SFM-MCMC-

OCSVM-PHD method will be shown.

6.7 Simulation

In this section, simulations are provided to examine the performance of the system and to compare with results from other recent methods.

6.7.1 Dataset selection and parameter setup

In order to evaluate the performance of the proposed system for multiple human tracking, particularly to handle the situation of varying number of targets, close interactions and occlusions, sequences from three different publicly available video datasets are chosen: one from the PETS2009 dataset [1] where 3-6 human targets are walking in an outdoor campus environment, one sequence from the CAVIAR dataset [2] where 1-5 human targets are walking in a shopping mall environment and one from the TUD dataset [42] where 5-7 human targets are walking in an outdoor-shopping mall environment. In order to make a more reliable evaluation, 17 more sequences from the CAVIAR dataset are also employed. All sequences are recorded at a resolution of 320×240 pixels at 25 frames/sec and each sequence contains around 200 frames, including human targets appearing, disappearing and occlusion in the scenario, and selected example frames are given in Fig. 6.2. 100 particles are employed for each target and for the MCMC step, 20 burn-in particles are used for each target in the MCMC resampling step. The set up of the remaining parameters is discussed in the following sections. The dynamic and measurement models which were used to predict and update the particles are described as

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \omega_k \tag{6.7.1}$$

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \tag{6.7.2}$$

where the state and measurement transformation matrices $\mathbf{F}$ and $\mathbf{H}$ are given as

$$\mathbf{F} = \begin{pmatrix} 1 & 0 & \Delta t & 0 & 0 & 0 \\ 0 & 1 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \mathbf{H} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}^T \tag{6.7.3}$$

where Δt is the time interval between frame k and $k+1$ which is set as 1 in the simulations, the zero-mean noise vector ω_k for prediction in the state model has covariance structure $cov\{\omega_k\} = Diag\{25, 25, 16, 16, 4, 4\}$ and for $\mathbf{v}_k$ $cov\{\mathbf{v}_k\} = Diag\{25, 25\}$. The missed detection probability $p_M = 0.01$, the survival probability $e = 0.99$, the new born intensity $\Upsilon = 0.1$ and clutter intensity $\kappa = 0.01$. The parameters for background subtraction, exponential-term based social force model and OCSVM classifier are selected empirically, which are shown as follows:

Parameters for background subtraction

For the background subtraction method described in Section 6.5, the parameters that need to be set include the shadow bound α_b, the highlight bound β_b and the colour detection threshold ε_b which for each sequence are given as Table 6.1, which were found empirically to yield best performance. For other parameters, the default values are used as those set in [79], for example, the colour sampling bandwidth is set to be 20 and the max negative run-length is set to be 60% for all these three datasets.

Parameters for exponential-term based social force model

The exponential-term based social force model introduced in Section 6.4 has many parameters, such as σ_d, β, σ_v, and σ_D, which control the influence

Table 6.1. Background subtraction parameters for each sequence

	α_b	β_b	ε_b
CAVIAR dataset	0.5	2	30
PETS 2009	0.7	1.5	20
TUD dataset	0.7	1.7	10

from distance, angular displacement, change of velocity and destination respectively. In this work, the parameters are selected based on pilot tests, a sequence from PETS2009 is employed to perform simulations with different values of the above four parameters and use the mean Euclidean error for each target position as the evaluation measure to select the best parameter set, which is shown in Table 6.2.

Table 6.2. Experimental values for parameters used in the social force model

σ_d	β	σ_v	σ_D	Mean of Euclidean error
1	1	1	1	5.15
1	1	1	2	5.11
1	1	1	4	5.33
1	1	1	8	5.12
1	2	1	2	5.15
1	4	1	2	5.14
1	8	1	2	5.33
1	4	2	2	5.53
1	4	4	2	5.21
1	4	8	2	5.41
2	4	4	2	5.12
4	4	4	2	5.05
8	4	4	2	5.22
16	4	4	2	5.09
32	4	4	2	5.18

From the experiment and comparison, it is found when the parameters are chosen as: $\sigma_d = 4$, $\beta = 4$, $\sigma_v = 4$, and $\sigma_D = 2$, the exponential-term based social force model performs the best, therefore these settings are

adopted for simulations in subsequent sections.

Parameters for OCSVM-classifier

In this work, to obtain the OCSVM classifier, the training dataset $\mathbf{S} = [\mathbf{s}_1, ..., \mathbf{s}_L]$ is employed, where each training data sample $\mathbf{s}$ is a 593×1 vector, containing human features extracted from the training frame, including 512 parameters from the colour histogram and 81 from oriented gradient histogram. The OCSVM classifier is trained by 82 sets of features extracted from different human targets. The influence of the OCSVM is controlled by parameter ϖ in (6.5.4), which is also chosen based on experiments. The OSPA results with respect to the different values of ϖ are shown as Table 6.3. From the comparison, $\varpi = 75$ is found to perform the best, hence is employed in later simulations.

Table 6.3. Parameters values used in OCSVM classifier and system evaluation

ϖ value	1	25	50	75	100
OSPA value	4.49	4.77	4.86	4.25	4.77

6.7.2 Evaluation of tracking results

In this section, the proposed exponential-term based SFM-MCMC is compared with the traditional SFM proposed in [5] and the S-SFM proposed by Pellegrini et al. in [6]. The proposed SFM-MCMC-OVSCM-PHD filter is compared with the traditional particle PHD filter in [4]. First, the comparison between the particle PHD filter and SFM-MCMC-PHD filter is made, followed by the comparison between the SFM-MCMC-PHD and SFM-MCMCOCSVM-PHD filters.

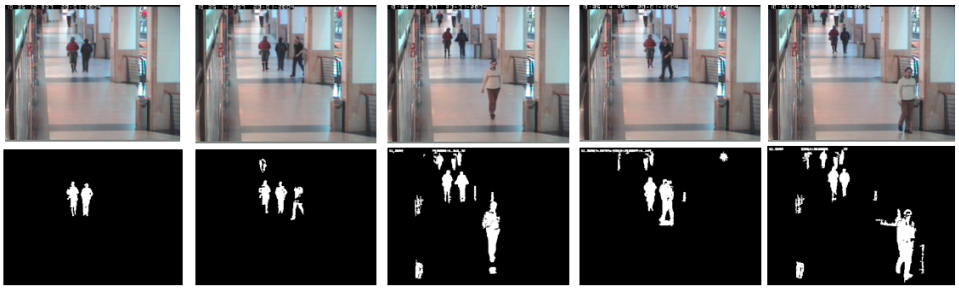
Background subtraction results

Fig. 6.2 shows some selected frames and results from the background subtraction for three datasets, from which, it is clear that many targets appear in the scenario. However, there is still much noise from the environment, which may cause false alarms and hence influence the performance. Moreover, sometimes it may fail to detect the targets because of occlusion and poor lightning conditions. In this case, the OCSVM classifier is employed to aid calculation for each particle. In this way, the noise is mitigated, in the later section, the improvement made by employing OCSVM will be shown.

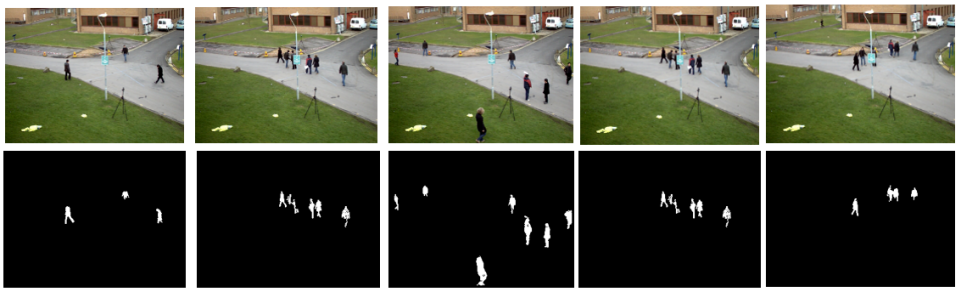
Social force model results

By employing the parameters in Table 6.2, and three example sequences selected from different datasets, the proposed social force model is first compared with the traditional SFM [5] and the S-SFM [6]. Fig. 6.3 shows the comparison of Eulidean tracking error in each frame between the above methods for the three sequences. The mean of the Euclidean error in each frame (MEE) over all the frames and their standard deviation (SD) are also compared in Table 6.4.

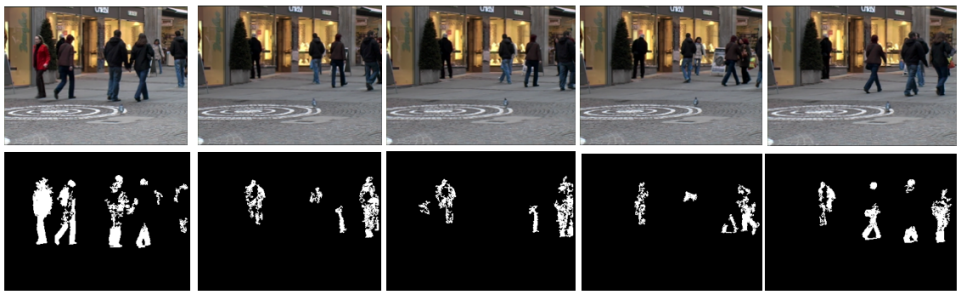
From Table 6.4 and Fig. 6.3 it is evident that the proposed social force model consistently attains better performance for the three sequences in terms of both the MEE and SD, as compared with the two baseline social force models. The improvement of the proposed social force model comes from the exponential-term model employed to describe the parameters such as the distance, angle, change of velocity, and the destination used in the model, with their influence controlled by the variance terms in the exponential-term model. In addition, in the proposed social force model, a threshold has been employed to control the modelling of the social forces between two targets, by excluding those that are far apart from each other in terms of distance (i.e. greater than the pre-defined threshold). This es-



(a) Frames from the CAVIAR dataset and background subtraction results



(b) Frames from the PETS2009 dataset and background subtraction results



(c) Frames from the TUD dataset and background subtraction results

Figure 6.2. Selected frames examples and their related background subtraction results from the three selected sequences from three different datasets, i.e. (a) is from the 'EnterExitCrossingPaths1cor' sequence from the CAVIAR dataset, (b) is from the 'PETS09_View001_S2_L1' sequence from the PETS2009 dataset and (c) is from 'TUD_Stadtmitte' sequence from the TUD dataset, and it can be found that the human target boundaries are extracted successfully, but there is still much environmental noise which may cause missed detections and false alarms. In order to mitigate such noise, an OCSVM classifier will be employed based upon the features from both colour and oriented gradient histograms of human targets.

sentially avoids the influence from unnecessary targets, hence improving the tracking accuracy when more targets are present in the environment.

Table 6.4. Comparison of the mean of the Euclidean tracking errors over the frames and their standard deviation by three social force models for the 'EnterExitCrossingPaths1cor' sequence from the CAVIAR dataset, the 'PETS09_View001_S2_L1' sequence from the PETS2009 dataset and the 'TUD_Stadtmitte' sequence from the TUD dataset.

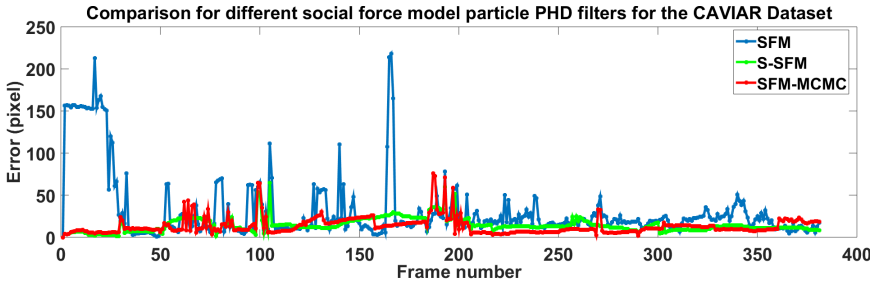
		MEE (pixel)	SD (pixel)
CAVIAR	SFM [5]	31.81	39.55
	S-SFM [6]	14.28	9.99
	SFM-MCMC	13.22	8.26
PETS2009	SFM [5]	68.25	17.20
	S-SFM [6]	40.76	14.95
	SFM-MCMC	39.41	13.29
TUD	SFM [5]	188.70	85.12
	S-SFM [6]	89.60	41.06
	SFM-MCMC	77.0	33.64

After the SFM-MCMC resampling, the predicted weights for particles are updated. Fig. 6.4 shows an example distribution of predicted particle weights, for frame 11 of the 'PETS09_View001_S2_L1′ sequence from the PETS2009 dataset.

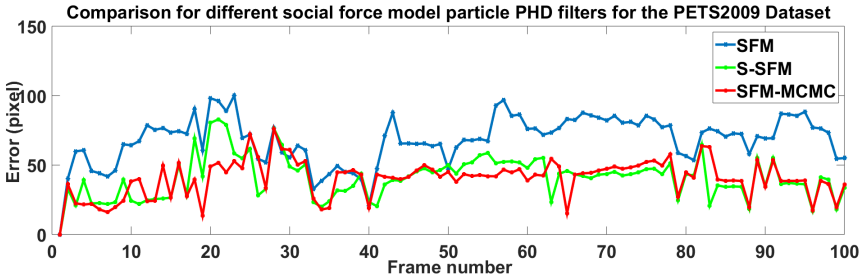
The figure shows that the particles with higher social force are given higher weights than others, the redundant peaks in the figure are because of the noise patches which will be mitigated in the updating step. Compared with the traditional particle PHD filter, where the particles are given the same weights in the prediction stage, the weights in the SFM-MCMC-PHD filter are determined based on the SFM which leads to more accurate prediction.

OSPA evaluation

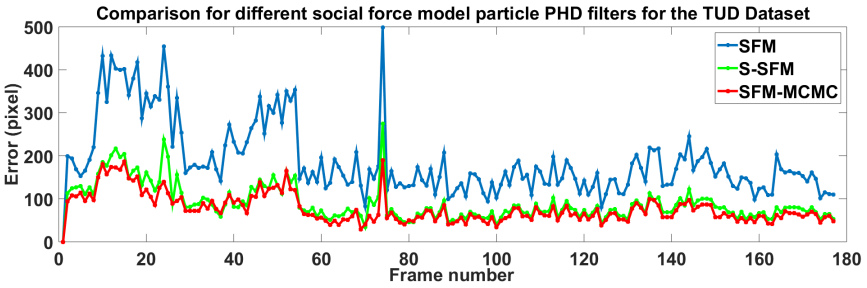
In order to evaluate the proposed system in terms of both localization and cardinality, the OSPA metric has also been employed. In this work, $c = 20$ and $p = 2$ are used. Comparisons for the three example sequences as in the previous experiment are shown in Fig. 6.5, where the black line denotes the



(a) Social force model comparison for the CAVIAR dataset



(b) Social force model comparison for the PETS2009 dataset



(c) Social force model comparison for the TUD dataset

Figure 6.3. Comparison in terms of Euclidean error of the three social force models when employed by the particle PHD filter for multiple human tracking. Subfigure (a) is the comparison for the 'EnterExitCrossingPaths1cor' sequence from the CAVIAR dataset, (b) is for the 'PETS09_View001_S2_L1' sequence from the PETS2009 dataset and (c) is for the 'TUD_Stadtmitte' sequence from the TUD dataset. The blue line denotes the traditional SFM [5], the green line denotes S-SFM [6] and the red line denotes the SFM-MCMC algorithm proposed in this work.

OSPA value from the traditional PHD filter, the blue line corresponds to the proposed SFM-MCMC-PHD particle PHD filter and the red line denotes the proposed SFM-MCMC-OCSVM-PHD algorithm.

To perform more reliable evaluation, the average OSPA values for all the 20 sequences based upon different methods have been obtained and are

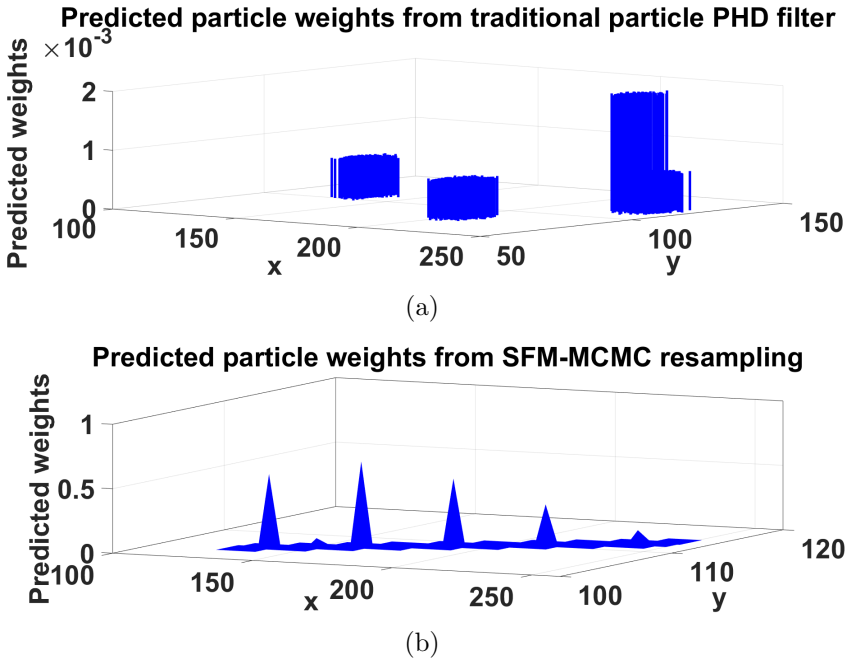


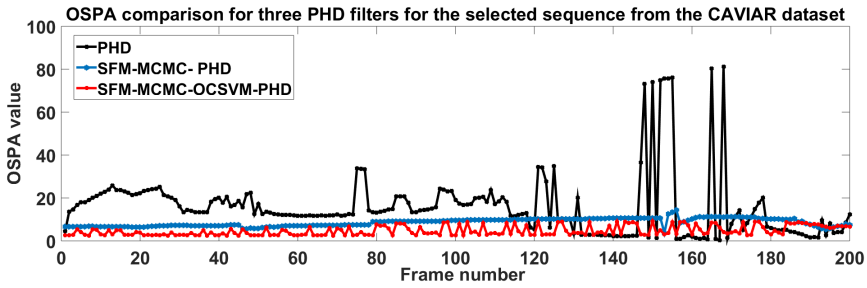
Figure 6.4. Comparison between the two distributions of the predicted particle weights, where (a) is from the traditional particle PHD filter and (b) is from the SFM-MCMC-PHD filter; for frame 11 of the 'PETS09_View001_S2_L1′ sequence from the PETS2009 dataset, where the ground truth position of the targets are (142,102), (233,115) and (200,95).

shown in Table 6.5.

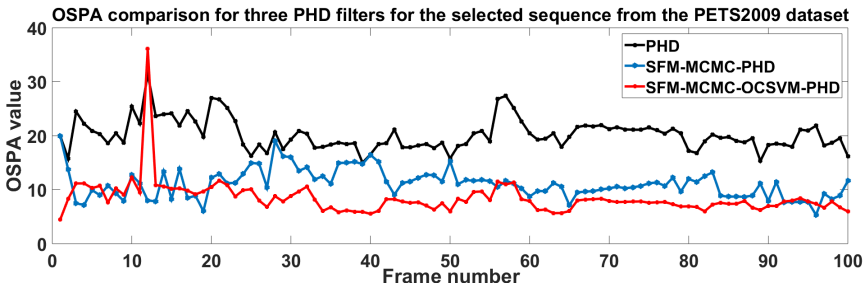
Table 6.5. Comparison of OSPA over 20 sequences for proposed PHD filters

	PHD [4]	SFM-MCMC-PHD	OCSVM-PHD	SFM-MCMC-OCSVM-PHD
OSPA (pixel)	21.93	13.54	12.42	8.83
Improvement	-	38.25%	43.36%	59.73%

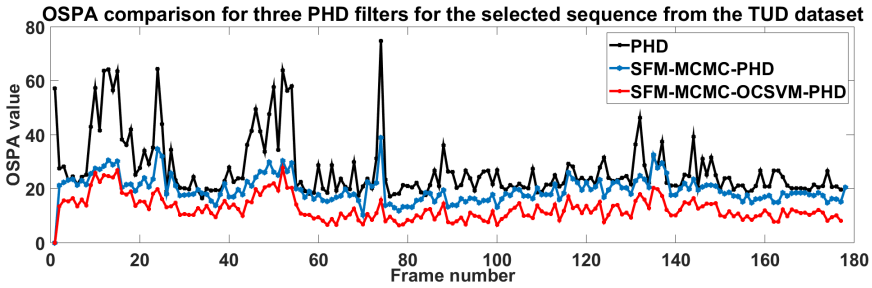
From the above comparison, it can be observed that the improvement of the proposed system comes from both the exponential-term based SFM-MCMC resampling step and the OCSVM likelihood calculation step. The



(a) OSPA comparison for the CAVIAR dataset



(b) OSPA comparison for the PETS2009 dataset



(c) OSPA comparison for the TUD dataset

Figure 6.5. Performance evaluation with OSPA performance measure for the proposed social force model aided MCMC particle PHD filter and the traditional particle PHD filter for multiple human tracking. The performance is examined with the 'EnterExitCrossingPaths1cor' sequence from the CAVIAR dataset, the 'PETS09_View001_S2_L1' sequence from the PETS2009 dataset and the 'TUD_Stadtmitte' sequence from the TUD dataset. The black line in the figure denotes the OSPA value from the traditional PHD particle PHD filter; the blue line shows the result by adding a social force model aided MCMC resampling step in the prediction stage of the particle PHD filter; and the red line denotes the OSPA value from the proposed SFM-MCMC-OCSVM-PHD filter.

background subtraction is an integrated component as in [4] which is used to determine the measurement foreground pixels. Without background subtraction, none of the methods under study would operate, so the individual

improvement from background subtraction is not provided in this work. By using the OCSVM, the average OSPA value for the 20 sequences is further reduced by 4.71 pixels since the OCSVM can distinguish the measurement of the human targets from the noise in the environment. To examine the difference in OSPA results between the traditional particle PHD filter and the proposed SFM-MCMC-OCSVM-PHD filter, the one-way ANOVA based F-test [96] is performed. $F = 8.74$, p-value $= 0.0051$ are obtained and the degree of freedom (1,42), where the F value is the ratio of the between-group variability to the within-group variability and the p-value is the probability of a more extreme result than the value actually achieved when the null hypothesis is true. Using the degree of freedom value and significant value 0.05, the critical value F_{crit} is found to be 4.07 from the F-distribution table given in [96]. According to the test, the results are accepted as statistically significant if $F \geq F_{crit}$ and the p-value is less than the significant value. The test results confirm the difference in OSPA results between the proposed SFM-MCMC-OCSVM-PHD and traditional PHD filter is statistically significant.

MOTP evaluation

Besides the Euclidean error in each frame, standard deviation and OSPA used for evaluation, MOTP is also employed to evaluate the proposed tracking system. The MOTP results for the three selected sequences from different datasets are shown in Table 6.6.

From the MOTP comparison, it can be observed clearly that the proposed method can greatly improve the tracking accuracy over the traditional PHD filter. For the CAVIAR dataset, the MOTP value is reduced by 6.14 pixels by employing the SFM-MCMC-PHD filter and then further reduced by 2.71 by employing the SFM-MCMC-OCSVM-PHD filter. For the PETS2009 dataset, the MOTP value is reduced by 1.67 and 2.92 pixels respectively. For

Table 6.6. MOTP comparison for three sequences using the traditional and proposed particle PHD filters

	CAVIAR	PETS2009	TUD
PHD [4]	12.49	8.23	16.42
SFM-MCMC-PHD	6.35	6.56	13.86
OCSVM-PHD	5.73	6.15	14.21
SFM-MCMC-OCSVM-PHD	3.64	4.84	12.07

the TUD dataset, the MOTP value is reduced by 2.56 and a further 1.79 pixels respectively. The average MOTP value over all the 20 sequences is reduced from 10.70 to 8.63 pixels by employing the SFM-MCMC-PHD filter and a further 6.31 pixels by the SFM-MCMC-OCSVM-PHD filter. The reduction of MOTP is mainly due to the utilization of the social force model aided MCMC step for resampling in the prediction stage, so that a more accurate posterior distribution is achieved. Moreover, the OCSVM classifier in the updating step helps to mitigate the measurement noise from the environment so that the problems of missed detections and false alarms are mitigated. By performing a one way ANOVA based F test for the proposed SFM-MCMC-OCSVM-PHD filter and the traditional particle PHD filter, F = 6.86, p-value = 0.0131 are obtained and the degree of freedom (1,34). By setting the significant value to be 0.05, the critical value F_{crit} is found to be 4.13 from the F-distribution table given in [96]. From the test results, it is obvious that the difference in MOTP results between the proposed SFM-MCMC-OCSVM-PHD and the traditional PHD filter is statistically significant.

Computational complexity

The computational complexity is also examined through the run-time. Since the particle PHD filter is used in this system, the number of particles plays an important role in affecting the computational complexity. In order to select

the most suitable number of particles, as an example, the OSPA measure and run-time as a function of the number of particles are calculated by employing a sequence from the PETS2009 dataset, the results of which are shown in Table 6.7. It can be observed that the increase in particle number

Table 6.7. Comparison of OSPA related to the number of particles

Number of particles	50	100	500	1000
Run-time/frame	1.44s	1.84s	4.85s	5.10s
OSPA (pixel)	22.52	21.32	21.06	21.03

has a bigger impact on the computational cost as compared with that on the OSPA results. Similar results have been observed for other sequences. In the simulations, the number of particles is selected empirically based on these experiments. The comparison shows that the number of particles chosen as 100 tends to provide a good compromise between run time and tracking performance.

The computational complexity of the proposed tracking system has also been considered. Compared with RFS, the particle PHD filter has a smaller computational cost since only the first moment of the posterior is employed instead of the posterior itself. However, the main growth of the time complexity is from the background subtraction and the OCSVM part of the proposed tracking system. If the times needed for determining the brightness and colour conditions are denoted as T_B and T_C respectively, and the update time is T_U, the total processing time for a single image pixel can be expressed as

$$T = N_B T_B + N_C(T_B + T_C) + N_U(T_B + T_C + T_U) \tag{6.7.4}$$

where N_B is the number of codewords rejected after testing the brightness condition, N_C is the number of codewords rejected after testing both the

brightness and colour conditions and N_U is 1 if a matching codeword is found and 0 otherwise. Furthermore, the computational complexity of the OCSVM classifier is $O(m^3)$ where m is the number of the training patterns. As compared with the traditional PHD filter [4], the time complexity of the proposed algorithm becomes higher due to the introduction of the SFM and OCSVM steps. The average run-time (calculated using 20 video sequences from the three different datasets) is shown in Table 6.8. This run-time comparison is made by implementing the algorithms with MATLAB (version R2015a) with a 3.4GHz I5 processor.

Table 6.8. Run-time comparisons for proposed PHD filters

	PHD [4]	SFM-MCMC-PHD	SFM-MCMC-OCSVM-PHD
Run-time/frame	1.39s	1.43s	1.95s

From this table it can be found that the run time for the traditional particle PHD filter is 1.39s/frame, and the overhead for the social force model aided MCMC resampling step is 0.04s/frame and for the one class SVM is 0.52s/frame, the run-time increases 2.9% by employing the social force based MCMC resampling step, and grows 36% by employing the OCSVM. However, the tracking accuracy has been improved by 38.25% and 34.9% by employing them respectively. The comparison show that the increase of time complexity is mainly due to the use of the OCSVM classifier when calculating the features from the colour and oriented gradient histograms for each particle.

Comparison with state-of-the-art methods

The proposed method is also compared with two recent multiple human tracking methods proposed in [97] and [98]. In [97], online learning of non-

linear motion patterns and robust appearance models are used for multiple target tracking and in [98] a background subtraction based multi-Bernoulli filtering method is proposed for visual tracking. The mean of the OSPA measure is employed for evaluation, which is shown in Table 6.9. All the three methods are evaluated on the 20 sequences from the CAVIAR, PETS2009 and TUD datasets.

Table 6.9. OSPA comparison of three recent methods and the proposed method over 20 sequences

	PHD [4]	Method in [98]	Method in [97]	Proposed method
OSPA (pixels)	21.93	15.39	12.95	8.83

From the comparison in the table it can be observed that the background subtraction based multi-Bernoulli filter proposed in [98] performs better than the particle PHD filter proposed in [4]. However, in [98], a kernel based background subtraction method was employed instead of the codebook method which is employed in this system. As such, the quality of the measurement is generally worse than that in the proposed system. Moreover, the social force model based MCMC resampling step provides more accurate predictions of targets states. The appearance modelling based method in [97], however, does not address the challenge of varying number of targets hence it generates an OSPA value that is higher than the proposed method.

Examples of tracking failures

Nevertheless, the tracking results of the proposed system can be degraded by the following factors: increase in the number of targets and the variations in lighting and colour of the targets which will influence the results from background subtraction. For example, for the sequence Browse1 from the CAVIAR dataset, the lighting in the environment is varying, and there is a large amount of noise in the results obtained from the background sub-

traction, hence it fails to provide accurate foreground measurement, which also leads to possible failure of the tracking system. What is more, when human targets are very crowded in the visual scene, the social force model can be influenced by false alarms. These tracking scenarios pose common challenges to many existing methods such as [99–101] as well as the baseline methods [4] [98]. The proposed method gives advantages over the baselines in the scenarios especially with group target movements, varying number of targets, and large amount of environmental noise. However, to address the aforementioned common challenges, more powerful techniques in appearance modeling and occlusion handling are required which will be the focus of the suggestions for future research.

6.8 Summary

In this chapter, contributions for improving tracking performance were proposed by enhancing both the prediction and updating steps. In the prediction stage, an exponential term based social force model was proposed and was exploited in the likelihood of an MCMC resampling step, from which, a more robust predicted particle set was obtained. In the updating stage, an OCSVM classifier which was trained on human features including histograms of colour and oriented gradient was employed in order to mitigate the measurement noise from background subtraction. From the above contributions, the accuracy of the tracking system was improved. In Section 6.7, comparisons were made between the proposed novel social force model and traditional ones. Moreover, the evaluation confirms the improvement from each of the proposed steps.

However, as mentioned in Section 6.7.2, failure cases are given, moreover, by evaluating the run-time of the proposed method, the proposed SFM-MCMC-OCSVM particle PHD filter had a more complex computa-

tional complexity caused by the MCMC resampling step and the OCSVM calculation. In order to address these challenges, approaches such as sparse dictionary learning method will be employed, which could be described in the suggestions for future work introduced in the next chapter.

Chapter 7

CONCLUSIONS AND FUTURE WORK

In this chapter, the contributions of this thesis are summarized in Section 7.1, and the suggestions for future work are given in Section 7.2.

7.1 Conclusions

This thesis gives solutions to video-based multiple human tracking, in particular to handle the challenges of varying number of target, interaction between targets and occlusions. In order to achieve these targets, in particular to handle a varying number of targets, the particle PHD filter was employed as the fundamental framework of the tracking system. The contributions to improve the particle PHD filter satisfy the four objectives mentioned in the introduction chapter. The first contribution is to replace the Gaussian distribution by the Student's-t distribution in the state and measurement models of the particle PHD filter; the second contribution is building a joint distribution between the measurement noise covariance matrix and the state model, then to use the variational Bayesian approach to find the best solution for the joint distribution, which is distributed by the multi-variate Student's-t distribution; the third contribution is improving the tracking performance by applying the delayed measurement adaptively, namely the adaptive Retro-PHD filter; the fourth contribution is to use the human be-

haviour and an MCMC resampling step in the prediction model to improve the accuracy; and the last contribution is to improve the robustness of the measurement model by employing an OCSVM classifier to mitigate the environment noise. The details of the contributions are as follows:

In Chapter 4, the Student's-t distribution was employed to improve the tracking performance. Firstly, the Student's-t distribution was used to replace the Gaussian distribution in the state and measurement models in the traditional particle PHD filter due its heavier tails than the Gaussian distribution, which can achieve predicted particles with larger amplitude. The simulation results confirm the outcome from the proposed method, where the OSPA for the sequences from CAVIAR and PETS2009 datasets were improved by 19.92% and 11.70% respectively. Then the Student's-t distribution was employed as a joint distribution between the measurement noise covariance matrix and the state model, and the variational Bayesian approach was employed to estimate the most suitable parameters for the joint distribution, the tracking performance was improved by 12.45% and 27.16% for the sequences from CAVIAR and PETS2009 dataset respectively when evaluated by the OSPA measure.

In Chapter 5, following the concept of the fixed-lag smoothing algorithm, the backward measurements were employed in order to achieve more observable measurements; firstly, the non-linear and non-Gaussian properties of the particle PHD filter were considered, so the term 'smoother' was replaced by the term 'Retrodiction', which employed the backward retrodiction step after the PHD filtering step. The performance of the tracking system was improved because more observable measurements were obtained, by evaluating by the OSPA measure with sequences from the CAVIAR and PETS2009 datasets, the performances of the tracking results were improved by 18.42% and 15.30%. In order to obtain further improvement, an adaptive approach was developed to combine the forward and retrodiction step adap-

tively, namely the adaptive Retro-PHD filtering, where the convex adaptive parameter was calculated by the similarity between the measurement set over discrete samples; the outcome of this approach can be shown by evaluating the OSPA measure, which was improved by 13.12% and 17.94% over the Retro-PHD filtering for the sequences from CAVIAR and PETS2009 dataset respectively.

In Chapter 6, the social force model aided MCMC-OCSVM particle PHD filter was proposed, which can be separated into three main contributions to improve the tracking performance from both the prediction model and the updating models to improve the tracking performance. Firstly, a novel exponential term based social force model was employed to represent human behaviour in order to predict more robust particles; secondly, the particles were resampled by an MCMC step, where the acceptance ratio was obtained with the aid of the social force model; thirdly, in the measurement model, an OCSVM classifier was employed to mitigate the environmental noise. The proposed tracking system was evaluated with 20 sequences from the CAVIAR, PETS2009 and TUD datasets, which showed the improved accuracy when compared with the traditional particle PHD filter. By employing the novel social force model aided MCMC resampling step and the OCSVM, the OSPA values of the tracking system were reduced by 38.25% and 43.36% respectively, and the overall reduction of OSPA from the proposed tracking sytem was 59.73%, which confirmed the significant improvement of employing the proposed social force model aided MCMC-OCSVM particle PHD filter.

7.2 Suggestions for future work

In order to further improve this study, there are several topics which could be further researched.

Firstly, in order to obtain further improvement in respect of a more robust measurement model, besides the traditional classifier such as the OCSVM, the dictionary learning method can be considered, which can help to build the template measurement set with the human features. Moreover, an online classification method can be exploited, which can update the measurement from human targets overtime, which can also improve the ability to handle the partial occlusion in the multiple tracking system.

Secondly, apart from the histograms of colour and oriented gradient, other human features could be employed in the likelihood function, for example, the SIFT feature and features obtained from other sensors could be employed to aid the information from the camera, for example, the audio signal and features extracted from the Depth sensor. By achieving a measurement set with richer human features, the tracking accuracy can potentially be improved.

Thirdly, data association methods can be considered to be exploited in the tracking framework and several such approaches have been proposed such as [20, 92]. This data association scheme can help to identify the new born targets and the survived targets, which can help particle sampling in the prediction stage of the particle PHD filter.

Fourthly, although the improvement from employing the social force model can be seen in Chapter 6, the tracker may fail when more than one targets move together; in order to give a solution to this challenge, the group social force model can be employed, which uses a hierarchy framework to separate the multiple human targets into groups, then the social forces are firstly calculated between the groups, and the inter-forces are calculated within each group. This development can also handle the occlusion challenge and hence achieve more accurate tracking performance.

Finally, one of the most important problems within the video-based multiple human tracking work is its computational complexity, although by em-

ploying the PHD filter, the tracker avoids the exponential growth of the complexity, the run-time of the tracking system still increases because of the large dimension of the measurement. In order to reduce the computational complexity caused by the measurement model, a sparse representation technique has been employed for human tracking recently [102], where the human features are represented by a sparse vector and the likelihood is calculated by a sparse coding method. Moreover, the dictionary learning method can be combined with a sparse representation method to build the measurement set, namely a sparse dictionary, which can be used in the particle PHD updating step. As for the combination, a richer measurement set is achieved and the computational complexity to use this approach is reduced by ignoring the unnecessary elements in the feature vector, and the performance of the tracker can be potentially improved. This is an interesting avenue for future work.

References

[1] I. Goldberg and M. J. Atallah, "Privacy enhancing technologies, 9th international symposium, PETS2009." http://dx.doi.org/10.1007/978-3-642-03168-7. accessed Feburary 2015.

[2] R. Fisher, "Caviar Case Scenarios." http://groups.inf.ed.ac.uk/vision/CAVIAR. accessed Feburary 2013.

[3] E. Maggio and A. Cavallaro, "Video tracking," *John Wiley and Sons, Ltd*, 2011.

[4] Y. D. Wang, J.-K. Wu, A. A. Kassim, and W.-M. Huang, "Tracking a variable number of human groups in video using probability hypothesis density," in *Proc. of IEEE International Conference on Pattern Recognition (ICPR'06)*, vol. 3, pp. 1127–1130, 2006.

[5] M. Luber, J. A. Stork, G. D. Tipaldi, and K. O. Arras, "People tracking with human motion predictions from social forces," in *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, pp. 464–469, 2010.

[6] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *Proc. of IEEE International Conference on Computer Vision*, pp. 261–268, 2009.

[7] M. S. Nixon and J. N. Carter, "Advances in automatic gait recognition,"

in *Proc. of IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 139–144, 2004.

[8] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *Acm computing surveys (CSUR)*, vol. 38, no. 4, p. 13, 2006.

[9] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time human pose recognition in parts from single depth images," *Communications of the ACM*, vol. 56, no. 1, pp. 116–124, 2013.

[10] Q. Hao, F. Hu, and Y. Xiao, "Multiple human tracking and identification with wireless distributed pyroelectric sensor systems," *IEEE Systems Journal*, vol. 3, no. 4, pp. 428–439, 2009.

[11] A. Feldman, M. Hybinette, and T. Balch, "The multi-iterative closest point tracker: An online algorithm for tracking multiple interacting targets," *Journal of Field Robotics*, vol. 29, no. 2, pp. 258–276, 2012.

[12] A. S. Sekmen, M. Wilkes, and K. Kawamura, "An application of passive human-robot interaction: human tracking based on attention distraction," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 32, no. 2, pp. 248–259, 2002.

[13] E. Polat and M. Ozden, "A nonparametric adaptive tracking algorithm based on multiple feature distributions," *IEEE Transactions on Multimedia*, vol. 8, no. 6, pp. 1156–1163, 2006.

[14] X. Li, A. Dick, C. Shen, A. Van Den Hengel, and H. Wang, "Incremental learning of 3d-dct compact representations for robust visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 4, pp. 863–881, 2013.

[15] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, 2012.

[16] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.

[17] M. Barnard, P. Koniusz, W. Wang, J. Kittler, S. M. Naqvi, and J. Chambers, "Robust Multi-Speaker Tracking via Dictionary Learning and Identity Modeling," *IEEE Transactions on Multimedia*, vol. 16, no. 3, pp. 864–880, 2014.

[18] E. Maggio, M. Taj, and A. Cavallaro, "Efficient multitarget visual tracking using random finite sets," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 8, pp. 1016–1027, 2008.

[19] B.-N. Vo and W.-K. Ma, "The Gaussian mixture probability hypothesis density filter," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4091–4104, 2006.

[20] D. E. Clark and J. Bell, "Data association for the PHD filter," in *Proc. of International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pp. 217–222, 2005.

[21] K. Panta, B.-N. Vo, and S. Singh, "Novel data association schemes for the probability hypothesis density filter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 2, pp. 556–570, 2007.

[22] R. P. Mahler, "A theoretical foundation for the Stein-Winter" probability hypothesis density (PHD)" Multitarget Tracking Approach," tech. rep., DTIC Document, 2000.

[23] 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*, vol. 27, no. 11, pp. 1805–1819, 2005.

[24] M. Isard and J. MacCormick, "BraMBLe: A Bayesian multiple-blob tracker," in *Proc. of International Conference on Computer Vision*, vol. 2, pp. 34–41, 2001.

[25] T. De Laet, H. Bruyninckx, and J. De Schutter, "Shape-based online multitarget tracking and detection for targets causing multiple measurements: Variational Bayesian clustering and lossless data association," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 12, pp. 2477–2491, 2011.

[26] D. Reid, "An algorithm for tracking multiple targets," *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979.

[27] B.-T. Vo, B.-N. Vo, and A. Cantoni, "The cardinality balanced multi-target multi-Bernoulli filter and its implementations," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 409–423, 2009.

[28] R. P. Mahler, *Statistical multisource-multitarget information fusion*. Artech House, Inc., 2007.

[29] R. P. Mahler, "Multitarget Bayes filtering via first-order multitarget moments," *IEEE Transactions on Aerospace and Electronic systems*, vol. 39, no. 4, pp. 1152–1178, 2003.

[30] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer, "Multi-camera multi-person tracking for easyliving," in *Proc. of IEEE International Workshop on Visual Surveillance*, pp. 3–10, 2000.

[31] W. Qu, D. Schonfeld, and M. Mohamed, "Distributed bayesian multiple-target tracking in crowded environments using multiple collaborative cameras," *EURASIP Journal on Applied Signal Processing*, vol. 2007, no. 1, pp. 21–21, 2007.

[32] S. L. Dockstader and A. M. Tekalp, "Multiple camera tracking of interacting and occluded human motion," *Proceedings of the IEEE*, vol. 89, no. 10, pp. 1441–1455, 2001.

[33] H. Sidenbladh, M. J. Black, and D. J. Fleet, "Stochastic tracking of 3d human figures using 2d image motion," in *Proc. of European Conference on Computer Vision*, pp. 702–718, 2000.

[34] D. Gatica-Perez, G. Lathoud, J.-M. Odobez, and I. McCowan, "Audiovisual probabilistic tracking of multiple speakers in meetings," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 2, pp. 601–616, 2007.

[35] D. N. Zotkin, R. Duraiswami, and L. S. Davis, "Joint audio-visual tracking using particle filters," *EURASIP Journal on Applied Signal Processing*, vol. 2002, no. 1, pp. 1154–1164, 2002.

[36] V. Kılıç, M. Barnard, W. Wang, and J. Kittler, "Audio assisted robust visual tracking with adaptive particle filtering," *IEEE Transactions on Multimedia*, vol. 17, no. 2, pp. 186–200, 2015.

[37] N. Wang, J. Wang, and D.-Y. Yeung, "Online robust non-negative dictionary learning for visual tracking," in *Proc. of IEEE International Conference on Computer Vision*, pp. 657–664, 2013.

[38] T. B. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Computer Vision and Image Understanding*, vol. 81, no. 3, pp. 231–268, 2001.

[39] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 983–990, 2009.

[40] M. D. Fairchild, *Color appearance models*. John Wiley & Sons, 2013.

[41] D. G. Lowe, "Object recognition from local scale-invariant features," in *IEEE International Conference on Computer Vision*, vol. 2, pp. 1150–1157, 1999.

[42] M. Andriluka, S. Roth, and B. Schiele, "Monocular 3d pose estimation and tracking by detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 623–630, 2010.

[43] N. Wang and D.-Y. Yeung, "Learning a deep compact image representation for visual tracking," in *Advances in neural information processing systems*, pp. 809–817, 2013.

[44] P. Feng, M. Yu, S. M. Naqvi, and J. A. Chambers, "Deep learning for posture analysis in fall detection," in *Proc. of IEEE International Conference on Digital Signal Processing*, pp. 12–17, 2014.

[45] Y. Bar-Shalom, *Tracking and data association*. Academic Press Professional, Inc., 1987.

[46] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine IJntelligence*, vol. 25, no. 5, pp. 564–577, 2003.

[47] D. Koller, K. Daniilidis, and H.-H. Nagel, "Model-based object tracking in monocular image sequences of road traffic scenes," *International Journal of Computer*, vol. 10, no. 3, pp. 257–281, 1993.

[48] M. Andriluka, S. Roth, and B. Schiele, "People-tracking-by-detection and

people-detection-by-tracking," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[49] A. Andriyenko, K. Schindler, and S. Roth, "Discrete-continuous optimization for multi-target tracking," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1926–1933, 2012.

[50] P. Brasnett, L. Mihaylova, D. Bull, and N. Canagarajah, "Sequential monte carlo tracking by fusing multiple cues in video sequences," *Image and Vision Computing*, vol. 25, no. 8, pp. 1217–1227, 2007.

[51] S. M. Kay, *Fundamentals of statistical signal processing, vol I.* Prentice Hall PTR, 1993.

[52] N. Gordon, B. Ristic, and S. Arulampalam, "Beyond the Kalman filter: Particle filters for tracking applications," *Artech House, London*, 2004.

[53] S. Sarkka and A. Nummenmaa, "Recursive noise adaptive Kalman filtering by variational Bayesian approximations," *IEEE Transactions on Automatic Control*, vol. 54, no. 3, pp. 596–600, 2009.

[54] X. Yun and E. R. Bachmann, "Design, implementation, and experimental results of a quaternion-based Kalman filter for human body motion tracking," *IEEE Transactions on Robotics*, vol. 22, no. 6, pp. 1216–1227, 2006.

[55] L. Jing and P. Vadakkepat, "Interacting MCMC particle filter for tracking maneuvering target," *Digital Signal Processing*, vol. 20, no. 2, pp. 561–574, 2010.

[56] W. R. Gilks and C. Berzuini, "Following a moving target Monte Carlo inference for dynamic Bayesian models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 1, pp. 127–146, 2001.

[57] R. P. Mahler, "Multitarget Bayes filtering via first-order multitarget mo-

ments," *IEEE Transactions on Aerospace and Electronic systems*, vol. 39, no. 4, pp. 1152–1178, 2003.

[58] B.-N. Vo, S. Singh, and A. Doucet, "Sequential Monte Carlo implementation of the PHD filter for multi-target tracking," in *Proc. of IEEE International Conference on Information Fusion*, pp. 792–799, 2003.

[59] A. Doucet, B.-N. Vo, C. Andrieu, and M. Davy, "Particle filtering for multi-target tracking and sensor management," *Citeseer*, 2002.

[60] D. E. Clark and J. Bell, "Convergence results for the particle PHD filter," *IEEE Transactions on Signal Processing*, vol. 54, no. 7, pp. 2652–2661, 2006.

[61] B. Ristic, B.-N. Vo, and D. Clark, "Performance evaluation of multi-target tracking using the OSPA metric," in *Proc. of IEEE International Conference on Information Fusion (FUSION)*, pp. 1–7, 2010.

[62] T. Vu and R. Evans, "A new performance metric for multiple target tracking based on optimal subpattern assignment," in *Proc. of IEEE International Conference on Information Fusion (FUSION)*, pp. 1–8, 2014.

[63] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, no. 1, pp. 1–10, 2008.

[64] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "A consistent metric for performance evaluation of multi-object filters," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3447–3457, 2008.

[65] H. B. Shitrit, J. Berclaz, F. Fleuret, and P. Fua, "Multi-commodity network flow for tracking multiple people," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 8, pp. 1614–1627, 2014.

[66] Y. Liang, G. Chen, S. Naqvi, and J. Chambers, "Independent vector

analysis with multivariate Student's t-distribution source prior for speech separation," *Electronics Letters*, vol. 49, no. 16, pp. 1035–1036, 2013.

[67] R. Piché, S. Särkkä, and J. Hartikainen, "Recursive outlier-robust filtering and smoothing for nonlinear systems using the multivariate Student-t distribution," in *Proc. of IEEE International Workshop on Machine Learning for Signal Processing*, pp. 1–6, 2012.

[68] S. Li, H. Wang, and T. Chai, "A t-distribution based particle filter for target tracking," in *2006 American Control Conference*, pp. 1–6, 2006.

[69] P. Feng, M. Yu, S. M. Naqvi, W. Wang, and J. A. Chambers, "A Robust student's-t distribution PHD filter with OCSVM updating for multiple human tracking," in *Proc. of European Signal Processing Conference (EUSIPCO)*, pp. 2396–2400, 2015.

[70] P. Feng, W. Wang, S. M. Naqvi, and J. Chambers, "Variational Bayesian PHD filter with Deep Learning Network Updating for Multiple Human Tracking," in *Proc. of Sensor Signal Processing for Defence (SSPD), 2015*, pp. 1–5, 2015.

[71] S. Senn and W. Richardson, "The first t-test," *Statistics in medicine*, vol. 13, no. 8, pp. 785–803, 1994.

[72] L. Schmetterer, *Introduction to mathematical statistics*, vol. 202. Springer Science & Business Media, 2012.

[73] S. Li, H. Wang, and T. Chai, "A t-distribution based particle filter for target tracking," in *Proc. of American Control Conference*, pp. 1–6, 2006.

[74] S. Sarkka and A. Nummenmaa, "Recursive noise adaptive Kalman filtering by variational Bayesian approximations," *IEEE Transactions on Automatic Control*, vol. 54, no. 3, pp. 596–600, 2009.

[75] S. Sarkka and A. Nummenmaa, "Recursive noise adaptive Kalman filtering by variational Bayesian approximations," *IEEE Transactions on Automatic Control*, vol. 54, no. 3, pp. 596–600, 2009.

[76] T. S. Jaakkola, "Tutorial on Variational Approximation Methods," *Advanced mean field methods: theory and practice*, p. 129, 2001.

[77] J. Christmas and R. Everson, "Robust autoregression: Student-t innovations using variational Bayes," *IEEE Transactions on Signal Processing*, vol. 59, no. 1, pp. 48–57, 2011.

[78] G. Zhang, F. Lian, C. Han, and S. Han, "An improved PHD filter based on variational Bayesian method for multi-target tracking," in *Proc. of IEEE International Conference on Information Fusion (FUSION)*, pp. 1–6, 2014.

[79] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground–background segmentation using codebook model," *Real-time imaging*, vol. 11, no. 3, pp. 172–185, 2005.

[80] O. Erdinc, P. Willett, and Y. Bar-Shalom, "Probability hypothesis density filter for multitarget multisensor tracking," in *Proc. of International Conference on Information Fusion*, vol. 1, pp. 8–pp, 2005.

[81] N. Nadarajah, T. Kirubarajan, T. Lang, M. Mcdonald, and K. Punithakumar, "Multitarget tracking using probability hypothesis density smoothing," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 4, pp. 2344–2360, 2011.

[82] M. L. Psiaki, "Backward-smoothing extended Kalman filter," *Journal of guidance, control, and dynamics*, vol. 28, no. 5, pp. 885–894, 2005.

[83] M. Klaas, M. Briers, N. De Freitas, A. Doucet, S. Maskell, and D. Lang, "Fast particle smoothing: If I had a million particles," in *Proceedings of*

the 23rd International Conference on Machine learning, pp. 481–488, ACM, 2006.

[84] R. P. Mahler, B.-T. Vo, and B.-N. Vo, "Forward-backward probability hypothesis density smoothing," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 1, pp. 707–728, 2012.

[85] J. Arenas-Garca, L. A. Azpicueta-Ruiz, M. T. M. Sliva, V. H. Nascimento, and A. H. Sayed, "Combinations of Adaptive Filters: Performance and convergence properties," *IEEE Signal Processing Magazine*, vol. 33, pp. 120–140, 2016.

[86] Y. Zhang and J. A. Chambers, "Convex combination of adaptive filters for a variable tap-length LMS algorithm," *IEEE Signal Processing Letters*, vol. 13, no. 10, pp. 628–631, 2006.

[87] S. J. Godsill, A. Doucet, and M. West, "Monte Carlo smoothing for nonlinear time series," *Journal of the American Statistical Association*, 2012.

[88] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 886–893, 2005.

[89] D. A. Forsyth and J. Ponce, *Computer vision: a modern approach*. Prentice Hall Professional Technical Reference, 2002.

[90] P. Feng, W. Wang, S. M. Naqvi, S. Dlay, and J. A. Chambers, "Social force model aided robust particle PHD filter for multiple human tracking," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4398–4402, 2016.

[91] P. Feng, W. Wang, S. M. Naqvi, S. Dlay, and J. A. Chambers, "Social force model based MCMC-OVSVM particle PHD filter for multiple human tracking," *submitted to IEEE Transactions on Multimedia*.

[92] A. Ur-Rehman, S. M. Naqvi, L. Mihaylova, and J. A. Chambers, "Multi-target tracking and occlusion handling with learned variational bayesian clusters and a social force model," *IEEE Transactions on Signal Processing*, vol. 64, no. 5, pp. 1320–1335, 2016.

[93] Z. Zhao and M. Kumar, "An MCMC-based particle filter for multiple target tracking," in *Proc. of International Conference on Information Fusion (FUSION)*, pp. 1676–1682, 2012.

[94] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[95] S. Zhang, X. Yu, Y. Sui, S. Zhao, and L. Zhang, "Object tracking with multi-view support vector machines," *IEEE Transactions on Multimedia*, vol. 17, no. 3, pp. 265–278, 2015.

[96] M. F. Triola, *Elementary statistics*. Pearson/Addison-Wesley Reading, MA, 2006.

[97] B. Yang and R. Nevatia, "Multi-target tracking by online learning of non-linear motion patterns and robust appearance models," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1918–1925, 2012.

[98] R. Hoseinnezhad, B.-N. Vo, and B.-T. Vo, "Visual tracking in background subtracted image sequences via multi-Bernoulli filtering," *IEEE Transactions on Signal Processing*, vol. 61, no. 2, pp. 392–397, 2013.

[99] E. Maggio, E. Piccardo, C. Regazzoni, and A. Cavallaro, "Particle PHD filtering for multi-target visual tracking," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1, pp. 1101–1104, 2007.

[100] L. Wang, N. H. C. Yung, and L. Xu, "Multiple-human tracking by iterative data association and detection update," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 1886–1899, 2014.

[101] C. Huang, Y. Li, and R. Nevatia, "Multiple target tracking by learning-based hierarchical association of detection responses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 4, pp. 898–910, 2013.

[102] X. Mei and H. Ling, "Robust visual tracking and vehicle classification via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2259–2272, 2011.