

State Estimation and Smoothing for the Probability Hypothesis Density Filter

by

Sergio I. Hernandez

A thesis
submitted to the Victoria University of Wellington
in fulfilment of the
requirements for the degree of
Doctor of Philosophy
in Computer Science.

Victoria University of Wellington

2010

Abstract

Tracking multiple objects is a challenging problem for an automated system, with applications in many domains. Typically the system must be able to represent the posterior distribution of the state of the targets, using a recursive algorithm that takes information from noisy measurements. However, in many important cases the number of targets is also unknown, and has also to be estimated from data.

The Probability Hypothesis Density (PHD) filter is an effective approach for this problem. The method uses a first-order moment approximation to develop a recursive algorithm for the optimal Bayesian filter. The PHD recursion can be implemented in closed form in some restricted cases, and more generally using Sequential Monte Carlo (SMC) methods. The assumptions made in the PHD filter are appealing for computational reasons in real-time tracking implementations. These are only justifiable when the signal to noise ratio (SNR) of a single target is high enough that remedies the loss of information from the approximation.

Although the original derivation of the PHD filter is based on functional expansions of belief-mass functions, it can also be developed by exploiting elementary constructions of Poisson processes. This thesis presents novel strategies for improving the Sequential Monte Carlo implementation of PHD filter using the point process approach. Firstly, we propose a post-processing state estimation step for the PHD filter, using Markov Chain Monte Carlo methods for mixture models. Secondly, we develop recursive Bayesian smoothing algorithms using the approximations of the filter backwards in time. The purpose of both strategies is to overcome the problems arising from the PHD filter assumptions. As a motivating example, we analyze the performance of

the methods for the difficult problem of person tracking in crowded environments.

Acknowledgments

I would first like to thank my primary and secondary supervisors, Dr. Marcus Frean and Dr. Paul Teal for their enthusiasm and guidance. This thesis has been largely motivated by their constant motivation and encouragement for making quality research.

Not less important, the support from the students and staff members from both the School of Mathematics and Operations Research and the School of Engineering and Computer Science of the University of Victoria of Wellington, who not only provided highly valuable academic enlightenment, but also provided personal support for making easier the process of moving to study in a new country. It was this complete environment which enabled the completion of this thesis, but special thanks would have to be made to the Associate Dean of the Science Faculty Office Mrs. Shona de Sain, who was strong enough not to let me fall apart during the hard times.

I am grateful to Chilean Commission in Science and Technology¹ and the University of Victoria of Wellington, for providing the necessary funds for my studies.

To my friends, who provided the much needed personal support during the time spent on this dissertation. Specially, the Jara's and Gallego's families whose warm hospitality made more tolerable being away from my own family. And finally, Elizabeth, Aurora and Ivan, to whom this thesis is dedicated.

¹Comisión Nacional de Ciencia y Tecnología (CONICYT), beca otorgada mediante resolución exenta N 2036 datada el 30 de noviembre de 2005.

List of Publications

The following publications were produced during the course of this dissertation:

1. **S. Hernández**, “Population-based Monte Carlo,” in Proc. New Zealand Computer Science Research Student Conference, 2007.
2. **S. Hernández** and P. Teal, “Multi-target tracking with Poisson processes observations,” in Advances in Image and Video Technology, 2007.
3. **S. Hernández**, “On-line Bayesian tracking and detection of multiple objects,” in Proc. New Zealand Computer Science Research Student Conference, 2008.
4. **S. Hernández**, P. Teal and M. Freaan, “Two filter probability hypothesis density smoothing,” in Proc. Jornadas Chilenas de la Computación, 2009.

Contents

1	Introduction	1
1.1	Thesis scope	1
1.2	Related work	4
1.3	Contributions of the thesis	5
1.3.1	Marked point process formalism for the PHD filter	5
1.3.2	PHD with spatial discretization	6
1.3.3	Bayesian state estimation for the PHD filter	7
1.3.4	Improved PHD filtering with smoothing	7
1.3.5	People tracking and counting with PHD filtering and smoothing	8
2	Probabilistic inference in single target tracking	9
2.1	State space models	10
2.1.1	Filtering	12
2.1.2	Smoothing	13
2.2	Filtering and smoothing recursions for linear state-space models	15
2.2.1	Kalman filter and smoother	15
2.2.2	Numerical example	17
2.3	Filtering for non-linear and non-Gaussian state-space models .	18
2.3.1	Sequential Importance Resampling : Particle Filter	21
2.3.2	Auxiliary particle filter	22
2.3.3	Numerical example	23

2.4	Smoothing for non-linear/non-Gaussian state-space models . . .	24
2.4.1	Forward-Backward smoothing	24
2.4.2	MAP particle smoothing	26
2.4.3	Two-filter and generalized information smoother	26
2.4.4	Numerical example	29
2.5	Summary and Contributions	31
3	Probabilistic inference in multi-target tracking	33
3.1	Finite point processes	35
3.1.1	Moments and transformations of a point process	37
3.1.2	Poisson process	40
3.1.3	Distribution and density of a point process	41
3.1.4	Probability generating functionals and their derivatives	45
3.2	The Probability Hypothesis Density (PHD) filter	47
3.2.1	The Probability Hypothesis Density filter	48
3.2.2	Bayesian point process predictor	49
3.2.3	Bayesian point process update	51
3.3	Spatial discretization for the PHD filter	53
3.4	Sequential Monte Carlo implementation of the PHD recursion	55
3.4.1	Numerical example	56
3.5	Gaussian mixture implementation of the PHD recursion	58
3.6	State estimation for the particle PHD filter	59
3.6.1	Bayesian state estimation	61
3.7	Limitations of the PHD filter	62
3.8	Summary and Contributions	64
4	Smoothing algorithms for the PHD filter	66
4.1	Forward-Backward PHD smoother	67
4.2	Two-Filter PHD smoother	71
4.2.1	SMC implementation for the Two-Filter PHD smoothing	74
4.3	Summary and contributions	75

5	Performance Evaluation	78
5.1	Distance metrics	80
5.1.1	Hausdorff distance	81
5.1.2	Wasserstein distance	82
5.2	Performance evaluation of PHD filters and smoothers	83
5.2.1	Low clutter scenario no births and deaths	84
5.2.2	Low clutter scenario with missed detections	88
5.2.3	High clutter scenario	91
5.2.4	High clutter scenario with target births and deaths	94
5.3	Summary and contributions	98
6	PHD filter and smoother for visual tracking	100
6.1	PHD filter for multiple person tracking	103
6.2	Person tracking model	105
6.3	Visual Tracking Examples	107
6.4	Summary and contributions	117
7	Conclusion	119
7.1	Future Work	122
A	Appendix	125
A.1	Bayesian state estimation	125
A.2	Forward-Backward smoothing source code	127
A.3	Two-Filter smoothing source code	128

List of Tables

- 2.1 Performance comparison for the Kalman filter and smoother. . . 18
- 2.2 Performance comparison of the SIR and ASIR particle filters. . . 23
- 2.3 Performance comparison of particle smoothing strategies. . . . 29

- 5.1 Cardinality and OSPA ($c=30, p=2$) error for the PHD filter and smoothers. 87
- 5.2 Cardinality and OSPA ($c=30, p=2$) error for the PHD filter and smoothers with missed detections 90
- 5.3 Cardinality and OSPA ($c=30, p=2$) error for the PHD filter and smoothers in high clutter 93
- 5.4 Cardinality and OSPA ($c=30, p=2$) error for the PHD filter and smoothers with births and deaths 97

- 6.1 Parameter settings for the background subtraction model . . . 108
- 6.2 Parameter settings for the PHD filter and smoother. 110
- 6.3 Cardinality and OSPA error ($p=2, c=2$) for the PHD filter and fixed-interval smoothing for visual tracking. GP(1) corresponds to the RMS error for the Gaussian process regression method using texture and area features, and GP(2) to the same method using area only features. Crowd counting results for the PHD filter and fixed-interval implementations of the FB-PHD and TF-PHD smoothers are comparable to the GP(1) method which uses area only features. 110

6.4	Speed of the PHD filtering and smoothing algorithms	111
6.5	Cardinality and OSPA ($c=2, p=2$) error for the PHD filter and smoothers for visual tracking	115

List of Figures

2.1	Schematic representation of a state space model. The state of the system x_k at time k is a Markov process observed via the measurement z_k	11
2.2	Kalman filter and smoother. Ground truth (red line) is plotted against the observed values (circles). Kalman filter and smoother estimates are also plotted (blue and black lines respectively).	17
2.3	Schematic diagram for the importance sampling procedure. A proposal distribution $q(x)$ (plotted in red line) is used to generate weighted samples x (circles with radiuses being proportional to the importance weights) from a target distribution $f(x)$ (plotted in blue line).	19
2.4	Schematic diagram for the importance resampling procedure. A target distribution (plotted in blue line) is approximated by a set of weighted samples (circles with radiuses proportional to the weights). When resampling, particles with low weights are discarded and particles with high weights are multiplied. New particles have equal weights.	22
2.5	SIR and ASIR particle filter estimation for a linear Gaussian model.	23

2.6	Particle smoothing for the linear Gaussian model. The Forward-Backward in (a) and the Two-Filter smoother in (b) provide improved results over the filtered estimates. This is not the case for the MAP smoother in Figure (c), which requires multiple iterations in order to converge.	29
3.1	Realizations of a point process.	36
3.2	Spatial point process. The number of events in a region B is a point process in \mathbb{R}^2	36
3.3	Multiple object tracking model. Targets are represented with blue squares and observations with red circles. Several targets appear at random times (horizontal axis) and random locations (vertical axis).	56
3.4	Particle filter approximation to the PHD filter. The posterior multi-target density is approximated using particles (plotted in gray). Particles are clustered around targets locations, so additional techniques have to be used for state estimation. Different strategies for state estimation are discussed in Section 3.6.	58
3.5	Positions and velocities of 3 target states are represented by the particle PHD filter. Monte Carlo samples are tightly clustered around target states, making it difficult to compute the model parameters using the EM algorithm.	60
3.6	Bivariate Gaussian mixture model estimated with the EM algorithm and the Gibbs sampler. After 100 iterations, the Gibbs sampler has explored different configurations of the parameters (Gaussian mean parameter being represented by red crosses), while the EM algorithm estimates are more dependent on the initial setup.	61

5.1	Multi-target estimation ambiguity. 3 frames with different number of targets (ground truth plotted in red) and estimated states (plotted in blue). At frame k the first multi-target estimator in Figure 5.1a has detected 3 of 4 targets, and for the same frame the second estimation in Figure 5.1b detected a different multi-target configuration. The same situation occurs at each frame, so it is not clear which one of the multi-target estimators is closer to the ground truth.	79
5.2	Three Targets following linear trajectories and no births or deaths of targets occurs. (a) Targets (ground truth plotted in black squares) are observed with low signal-to-noise ratio and there is a small number of false alarms per scan. (b) Due to the variability on the PHD filter cardinality estimates, state estimates also presents several discrepancies with the ground truth. Figures (c) and (d) show the improved estimates by using the FB-PHD and TF-PHD smoothers.	84
5.3	Cardinality estimates for the PHD filter (dashed line) and smoothers (blue line) are obtained by particle approximations to the multi-target posterior densities. Both PHD smoothers are able to recover the actual number of targets (ground truth plotted in red) in the backward pass.	85
5.4	Illustration of particle PHD smoothing in frame 37, Ground truth of target locations (black stems) is plotted along with the particle approximation (gray histogram), the estimated state densities (blue and red line plots) and location estimates (blue and read stems). Figures (a) and (b) show the EM (blue stems) and Gibbs sampler (red stems) state estimation for the PHD filter with cardinality error, Figures (c), (d), (e) and (f) show the EM and Gibbs sampler state estimation for the PHD smoothers.	86

5.5	State estimation errors for the PHD filter, FB-PHD smoother and TF-PHD smoother. (a) EM state estimation for the FB-PHD smoother, (b) Gibbs sampler state estimation for the FB-PHD smoother, (c) EM state estimation for the TF-PHD smoother, (d) Gibbs sampler state estimation for the TF-PHD smoother	86
5.6	Multi-target tracking model with missed detections. Positions of 3 targets have to be estimated with low observation noise and moderate clutter. Missed detections are present in several time steps.	88
5.7	Cardinality estimates for the PHD filter and smoother. The PHD filter has unstable cardinality estimates due to the missed detections, but both PHD smoothers are able to recover from the error.	89
5.8	Illustration of particle PHD smoothing with missed detections in frame 25. (a) EM state estimation for the PHD filter, (b) Gibbs sampler state estimation for the PHD filter, (c) EM state estimation for the FB-PHD smoother, (d) Gibbs sampler state estimation for the FB-PHD smoother, (e) EM state estimation for the TF-PHD smoother, (f) Gibbs sampler state estimation for the TF-PHD smoother.	90
5.9	OSPA Error for the PHD filter, FB-PHD smoother and TF-PHD smoother with missed detections. Cardinality errors have an adverse effect on the location estimates, so the OSPA error of the PHD filter present several spikes where missed detections occurs. The EM and Gibbs sampler estimators of the FB-PHD and the TF-PHD smoothers provide improved estimates.	90
5.10	Multi-target tracking model with high clutter volume.	91
5.11	Cardinality estimates for the PHD filter and smoother.	91

5.12	OSPA Error for the PHD filter, FB-PHD smoother and TF-PHD smoother with high clutter.	92
5.13	Multi-target tracking model in high clutter with targets birth and death. As seen in Figure (b), increasing the clutter volume deteriorates the performance of the PHD filter. Figures (c) and (d) shows the estimates from the FB-PHD and TF-PHD smoothers. In this example, the FB-PHD smoother can recover a lost trajectory in the forward pass, but that is not the case of the TF-PHD smoother.	94
5.14	Cardinality estimates for the PHD filter and smoother.	95
5.15	OSPA Error for the PHD filter, FB-PHD smoother and TF-PHD smoother with targets birth and death.	95
6.1	A single person is represented as an ellipsoid consisting of a 2-dimensional centroid and the equidensity contour of the Gaussian observation noise	107
6.2	Crowded scenario with multiple people walking in different directions. A single camera captures images at 10 frames per second and the goal is to track and count individual persons. .	107
6.3	After the background subtraction step, from the image in Figure 6.2 a binary image of the foreground is used to extract observations for the tracking algorithm. A single blob might contain either a group or a single person.	108
6.4	Each pixel is weighted according to a perspective map. In order to compare areas from different areas of the image, objects closer to the camera have lower weights than objects being far away.	108
6.5	Schematic diagram of the tracking procedure.	109

6.6	A rough estimate of the number of persons and their locations is given to the PHD filter as an input. Pixels within a blob are clustered to the expected number of persons according to a reference area. Gray dots denote observations for the PHD filter and smoother. Over-segmentation and under-segmentation of the blobs causes clutter and mis-detections.	109
6.7	Particle PHD filter and TF-PHD smoother estimates for frame 20.	112
6.8	Cardinality estimates for the PHD filter and smoother. The TF-PHD and the FB-PHD smoothers give an improved estimate of the number of targets.	113
6.9	Particle approximations for frame 14 of the pedestrian tracking sequence. Location estimates from the PHD filter suffers from an incorrectly estimated number of persons. Since the Gibbs sampler is less sensitive to the initial conditions, it manages to allocate person locations more accurately and with less variance than the EM algorithm. Monte Carlo approximations by means of the FB-PHD and the TF-PHD smoothers provide improved estimates over the PHD filter alone.	113
6.10	OSPA Error for the PHD filter and fixed-interval implementations for the FB-PHD smoother and TF-PHD smoother for visual tracking. A fixed-interval of 50 frames is used to evaluate the benefit of smoothing over filtering.	114

List of Acronyms

PDA	Probabilistic Data Association
MHT	Multiple Hypothesis Tracking
SMC	Sequential Monte Carlo
MCMC	Markov Chain Monte Carlo
MRF	Markov Random Field
EM	Expectation Maximization
SNR	Signal-to-Noise Ratio
EKF	Extended Kalman Filter
UKF	Unscented Kalman Filter
SIS	Sequential Importance Sampling
SIR	Sequential Importance Resampling
ASIR	Auxiliary Sequential Importance Resampling
MAP	Maximum A Posteriori
RMS	Root Mean Square
MMSE	Minimum Mean Squared Error

OMAT Optimal Mass Transfer

OSPA Optimal Sub-Pattern Assignment

RFS Random Finite Set

FISST Finite-Set Statistics

PHD Probability Hypothesis Density

FB-PHD Forward-Backward Probability Hypothesis Density

TF-PHD Two-Filter Probability Hypothesis Density

GP Gaussian Process

List of Symbols

x_k vector-valued state at time k

z_k vector-valued measurement at time k

$\hat{x}_{k|k}$ mean of state at time k

$\Sigma_{k|k}$ covariance of state at time k

\mathbb{B} state space

\mathcal{B} Borel σ -field generated by \mathbb{B}

B n-fold product space $B_1 \times \cdots \times B_n$

$N(B)$ number of points in B

$\lambda(x) = \lambda^*(t, k) dt dk$ intensity function of a marked point process on $\mathbb{R} \times \mathbb{R}^n$

$\Lambda(B)$ intensity measure

$M_{[n]}(B)$ n-th order factorial moment measure

$m_{[n]}(x_1, \dots, x_n)$ n-th order factorial moment density

$J_n(B)$ Janossy measure

$j_n(x_1, \dots, x_n)$ Janossy density

$G[h]$ probability generating functional (p.g.fl.) of N

X_k set-valued state at time k

Z_k set-valued measurement at time k

$\pi_s(x)$ target probability of survival

$\pi_d(x)$ target probability of detection

$\gamma(x)$ target probability of spawning

$b(x)$ target spontaneous birth density

$c(z)$ clutter spatial density

λ_c clutter intensity

$D_{k|k}$ probability hypothesis density filter at time k

$D_{k|T}$ probability hypothesis density smoother at time $k < T$

$d_p(x, y)$ Euclidean distance between two vectors

$d^H(X, Y)$ Hausdorff distance between two sets

1

Introduction

1.1 Thesis scope

Scientific and engineering communities have long been interested in signal tracking. Signals arise in different forms, such as price commodities in finance, mammal migrations in population ecology or ship trajectories in radar signal processing. Despite being very different in nature, all these applications share the need to reconstruct a signal from noisy observations while incorporating all available information. The theoretical foundation for signal tracking can be regarded as a temporal data fusion problem, where one or multiple sensors collect data that has to be integrated in order to perform inference about the unobserved dynamic system.

While the statistical analysis of univariate and multivariate time series of linear and non-linear dynamical models has a long history, the development of

modern computational methods have broadened the interest and scope of applications of signal tracking. The recursive estimation of a latent Markovian process contaminated with noise can be naturally achieved in the Bayesian framework [10], where conditional probabilities are propagated in time. In many important cases, the expectation over the state of the dynamical system can be written in closed form using the Kalman filter equations. In cases where no closed form is available, Monte Carlo computational methods can be used for the estimation task.

One of the key points in Bayesian inference is the idea of incorporating and integrating information using basic probability rules. In the case of tracking systems, the position and velocity of a single target are represented as a dynamic parameter or random vector, whose conditional expectation is handled using Bayesian reasoning. However, when the number of targets is not known apriori, it is not straightforward to extend the parametric inference approach. In these cases, the model dimensionality is also part of the uncertainty, so the expectation has to be computed over both parameters and models [39]. Standard parametric inference, such as maximum likelihood estimation requires a model selection step which consists of estimating the order of the model, and this has to be performed before making inference about the state of the system. Non-parametric inference on the other hand can handle the model selection and on-line processing duality using the same underlying Bayesian principle. Furthermore, the number of parameters in non-parametric inference does not scale with the model order, so these models are more amenable to use in the non-constant dimensionality problem [28].

Instead of using a parametric family for the prior and posterior distributions, Bayesian non-parametric models use more general families of functions to represent the prior and posterior distributions over infinitely many targets. Suitable choices for non-parametric Bayesian inference are point processes. Point processes are models for random numbers of random events with irregular spatial and temporal distributions. Estimation of the state of

the system is performed using a time-varying function that represents the sufficient statistics of a stochastic point process [99]. With all the flexibility of the non-parametric approach, in order to develop tractable recursive filters, computational models must rely on approximations and assumptions about the underlying dynamic system. In target tracking, workable algorithms can be achieved by assuming a particular family of density functions for the prior and posterior distributions. The Probability Hypothesis Density (PHD) filter developed by Ronald Mahler is a recursive algorithm in which the Poisson process is used as a best-fit approximation to the real posterior distribution [68]. The Poisson approximation allows us to factorize the complete posterior distribution into the product of the marginal single target densities while being characterized by its first-order moment. Nevertheless, a remarkable property of the PHD filter approach is the direct interpretation of multi-target problems in terms of the classic non-linear non-Gaussian filtering theory.

Apart from having an unknown number of targets, multi-target tracking scenarios are usually contaminated with false alarms or clutter. In order to estimate the state of the system, one has also to resolve which measurement corresponds to which target. This process is usually named *data association*, and depending on the application it has to be solved using hard or soft assignments. Probabilistic Data Association (PDA) approaches update the marginal probability of each single target by means of normalized weights that represent the total probability of a measurement coming from a target or being a false alarm [30]. Multiple Hypothesis Tracking (MHT) algorithms on the other hand, enumerate all possible hard assignments, but unfortunately the number of association hypotheses increases exponentially with the number of targets [7].

The point process formalism provides a natural interpretation for the non-constant dimensionality problem, and also allows both data association schemes under the same framework [102]. By using a Poisson process approximation, the PHD filter can be categorized as a soft PDA approach, so it can

be appropriate for cases where one cannot distinguish between a new-born and an existing target. One such application can be found in visual tracking in crowded scenarios, where it is still a challenge to have unambiguous measurements from a single person using standard image processing techniques [29].

1.2 Related work

One of the major challenges in statistical visual tracking consists of associating multiple and unknown objects [130]. The problem becomes even more challenging when targets are dense enough not to be able to detect a single object from image measurements. The problem with closely spaced objects is that data association methods would have to perform reasoning from ambiguous measurements due to partial or complete occlusions. Moreover, objects might not be well defined so it is not easy to differentiate false alarms from object generated measurements.

Early approaches for tracking multiple targets managed data association implicitly, assuming that each object generates a single measurement [53]. A more sophisticated method considers Markov Chain Monte Carlo (MCMC) sampling method with an optimization criterion [129]. A drawback of the method is that MCMC is only guaranteed to converge in the limit of an infinite number of iterations, so reaching an equilibrium distribution is domain specific [92], making MCMC based methods ill suited to on-line analysis.

Contrary, Sequential Monte Carlo (SMC) methods are well suited for on-line processing. One of the first approaches for incorporating data association into the SMC context can be found in [48]. The proposed method combines SMC with MCMC for estimating a distribution for the association hypotheses. In order to achieve on-line performance, data association priors have to be carefully designed from some optimal proposal distribution. A more advanced approach by [59] considers MCMC data association for interacting targets using a Markov Random Field (MRF) prior. The method is well

suitable for tracking interacting targets, but is not easy to extend when the number of targets is also unknown.

1.3 Contributions of the thesis

This thesis is concerned with sequential Bayesian inference for multi-target tracking using a first-order moment approximation. In this context, the state of the system is regarded as being composed of an unknown number of targets characterized by a Poisson point process. PHD filters are emerging as potential replacements to standard multi-target tracking algorithms using data association, however most developments assume a perfect model or knowledge of the true underlying parameters. Parameter estimation in multi-target tracking problems is not a direct extension of the single-target case, therefore the first-order moment approximation might not be optimal in more realistic scenarios.

The main objective of this dissertation is to propose improvement strategies for the PHD filter in order to cope with real world multi-target tracking problems, and more specifically visual tracking of pedestrians in crowded environments. The novel contributions of this thesis can be summarized as:

1.3.1 Marked point process formalism for the PHD filter

The original approach for the PHD filter by Mahler is based on the random finite sets (RFS) formalism, where most of the standard probability rules are replaced by similar concepts in the RFS notation, using an "almost parallel worlds" argument. The method is highly appealing for general data fusion problems, where imprecise information can be represented using set-valued analysis. Instead, this thesis follows a Poisson point process approach and develops a marked point process formalism for the PHD filter. The PHD filter algorithm has no modification under this new approach, but separating the intensity function into different *marks* is appealing to visual tracking,

where dynamic and geometric features can be separated into different metric spaces.

Computer vision and image analysis applications are well characterized by object level models, such as marked point processes [21]. A novel contribution of this thesis is to highlight the marked point process approach as a natural extension to the PHD filter, which can be used in visual tracking applications. In that context, The PHD filter achieves a good compromise between algorithmic complexity and model stability and can be implemented using standard SMC techniques. The theoretical foundations required to specify a density and compute estimates from a marked point process are outlined in Section 3.1.

1.3.2 PHD with spatial discretization

The PHD filter calculates the probability of finding a target in any particular region of the observation space by recursively computing the marginal posterior density of a single target. Sequential Monte Carlo methods can then be used for approximating the sufficient statistics or intensity of the Poisson process that represents the unknown number of targets. The method replaces the parametric densities of standard Bayesian filtering equations by means of set-valued integrals, so the resulting algorithm is well rooted in non-linear and non-Gaussian filtering theory. However, a more intuitive interpretation can be achieved considering a fixed-grid approximation to the spatio-temporal intensity. In cases where the state-space can be separated into spatial and (discrete-time) temporal components, a spatial discretization can be used for estimating the posterior distribution of the locations of the targets. We demonstrate in Section 3.3 that the conditional intensity of a Poisson point process can be estimated by discretizing the state-space into bins.

A similar model named the “bin-occupancy” approach, was recently proposed in [26] for deriving the PHD filter equations. The new technique presented on this thesis, is based on a discretization of the state space and

provides a natural interpretation for a point process with independent scattering. Both approaches are described in Section 3.3, and the application of the spatial discretization technique to acoustic multi-target tracking was proposed in [43].

1.3.3 Bayesian state estimation for the PHD filter

The PHD filter is a computationally attractive alternative to the traditional multi-target tracking algorithms, especially when the Signal-to-Noise Ratio (SNR) is sufficiently high that the posterior distribution can be approximated by its first-order moment, and the number of targets can be estimated directly from data. However, whenever the basic assumptions of the PHD filter are broken, the algorithm loses its well-behaved performance.

In cases where the SNR is not high enough, target state uncertainty increases so we propose Markov chain Monte Carlo methods for Bayesian state estimation. Parametric estimation for the PHD filter using the EM algorithm has been discussed in the literature for the standard SNR case [15]. More recently, Bayesian state estimation using MCMC was proposed in [63]. In Section 3.6, a new MCMC method for state estimation is provided, which takes advantage of the noise conditions commonly found in visual tracking applications.

1.3.4 Improved PHD filtering with smoothing

The sequential Monte Carlo implementation of the PHD filter has been successfully used in highly cluttered tracking scenarios. Despite being a computationally attractive alternative to traditional multi-target tracking algorithms, in the presence of false alarms and missed detections, the filtered estimates are unstable, degrading the overall performance of the algorithm.

While the instability issue can easily be remediated by averaging the cardinality estimates in a time window, a more principled Bayesian alternative would consider smoothing or retrodiction [44]. Bayesian smoothing is a back-

ward recursive alternative to filtering, which considers data frames later than the current time step. An original implementation of the two-filter smoother for the PHD filter is proposed in Section 4.2. This is then compared with standard PHD filtering and an alternative smoothing scheme, using both Bayesian and maximum likelihood state estimation.

1.3.5 People tracking and counting with PHD filtering and smoothing

One application where the independence property of the PHD filter is easily violated is found in pedestrian tracking in crowded scenarios. Persons can walk alone or in groups, so in principle, the PHD filter is well suited for overriding the combinatorial problem of associating detected persons with existing tracks. Moreover, most of the state of the art pre-processing steps required to generate observations from a single person in video are far from being optimal, so existing data association methods have to maintain a large set of association hypotheses in order to keep track of an unknown number of targets.

In the case of pedestrian tracking in crowded scenes, occlusions and unknown depth ordering makes it difficult to create and maintain association hypotheses between observations and persons [42]. Chapter 6 discusses an original application of PHD filtering and smoothing for the estimation and tracking of an unknown number of persons in crowded environments.

2

Probabilistic inference in single target tracking

Tracking a single object using measurements from one or several sensors is an application of data fusion methods for temporal information processing. Usually, the state of the system being tracked consists of kinematic information, like position and velocity of a target, and conveniently modelled as a random vector. The measurements or observations are also assumed to be realizations from a random vector which is related to the state vector.

This chapter introduces probabilistic inference in single target tracking. We will see that in order to achieve a tractable representation of the dynamical state of the system, Markovian dependence between the current and the previous state as well as conditional independence between the current observation and all previous states has to be assumed. Moreover, the current

observation is also assumed to be independent from the previous states and observations. Therefore, the resulting dynamical model is a stochastic process with time-ordered Markovian structure, and this means that the state of the system can be estimated recursively using all measurements available.

Estimation of the state of the system is performed recursively using *filtering* and *smoothing* equations. These techniques are explained in Section 2.1, which also describes the general form for modeling the dynamical system in tracking applications. Section 2.2 describes filtering and smoothing methodologies for the estimation problem in the special case when the model is linear and Gaussian. Filtering for the non-linear and non-Gaussian case is described in section 2.3 and the related smoothing approaches are then described in Section 2.4.

2.1 State space models

A dynamic system can be written as a first-order probabilistic generative model, known as a *hidden Markov model* in the discrete case or more generally as a *state space model*. These models are used to characterize the sequential structure of the dynamical system, and the evolution of a hidden state of the system. The state is considered to be not directly observable, but some of its properties are available at discrete or continuous observation times. The state-space model can be regarded as a dynamic Bayesian network, where the nodes are connected through a sequence of pairs with a state and its observation.

The state-space model contains two equations for describing the dynamical behavior of the system and the observational process. As seen in Figure 2.1, the state-space representation is conceptually a graph for sequential probabilistic inference over a partially observed stochastic process. The state x is an unobserved first-order Markov process and the observations z are conditionally independent given the state process.

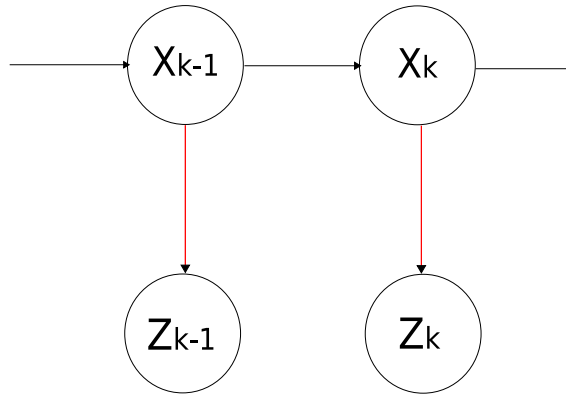


Figure 2.1: Schematic representation of a state space model. The state of the system x_k at time k is a Markov process observed via the measurement z_k .

$$\begin{array}{ll}
 x_k = f(x_{k-1}, v_k) & \text{process equation} \\
 z_k = g(x_k, w_k) & \text{observation equation}
 \end{array}$$

The noise sources $\{v_k\}$ and $\{w_k\}$ are assumed as being mutually independent and identically-distributed (i.i.d.) sequences of random variables, which are also independent of the state and the observations x_k and z_k respectively. The functions f and g represents possibly non-linear mapping from x_{k-1} to x_k and from x_k to z_k respectively.

Bayesian inference provides a sound framework for building a model for a sequence of unobserved states given noisy measurements. The information about the hidden state (or latent variable) and the observations can be integrated recursively, leading to a functional form for the conditional expectation of the posterior distribution of the current state of the system given all previous observations. However, a general solution for recursively estimating the posterior distribution of the state is not always available in analytical form. In the case of linear Gaussian models, the recursive formula admits a closed form and an optimal estimator can be achieved using standard methods like the Kalman filter or the recursive least squares algorithm [1].

For non-linear or non-Gaussian models, the marginal distribution of the current state has no analytical expression in general and global or local approximations must be made. Local approximations can be achieved by means of analytic expansions of the posterior distribution. Methods like the extended Kalman filter (EKF) and the unscented Kalman filter (UKF) are examples of local approximations using Taylor series expansions and sigma-points regression respectively. On the other hand, global approximations to the posterior distribution are possible when the expectation of the latent variable is calculated by means of deterministic or non-deterministic integration [91]. In that sense, we will see that Monte Carlo integration plays a crucial role in sequential methods for the global approximation of non-linear and non-Gaussian dynamic models.

2.1.1 Filtering

Under the Bayesian inference framework we are interested in recursively estimating the posterior distribution of the hidden states $x_{1:k}$ given the past observations $z_{1:k}$, and more specifically the marginal or *filtering density* $p(x_k|z_{1:k})$. This procedure is also called *Bayesian filtering*, and consists of a prediction step and an update step to calculate the following quantities:

$$p(x_k|z_{1:k-1}) \quad \text{(Prediction)} \quad (2.1)$$

$$p(x_k|z_{1:k}) \quad \text{(Update)} \quad (2.2)$$

The prediction step uses all information from the past in order to estimate a value of the hidden state x_k . Under the assumption of a first-order Markovian process dynamic system, the distribution of x_k conditional on the history $x_{1:k-1}$ is only determined by the previous state x_{k-1} . Consequently, the conditional distribution of the current observation y_k given all previous states and observations $y_{1:k-1}$ is also determined by its relationship with x_k . A recursive formula for the state x_k at time k can be achieved by first con-

sidering a prior density $p(x_0)$, so the prediction density $p(x_k|z_{1:k-1})$ can be decomposed by using the Chapman-Kolmogorov equation [54]:

$$p(x_k|z_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|z_{1:k-1}) dx_{k-1} \quad (2.3)$$

As a result of Bayes theorem, the filtering density can be decomposed as:

$$p(x_k|z_{1:k}) = p(x_k|z_k, z_{1:k-1}) \quad (2.4)$$

$$= \frac{p(z_k|x_k, z_{1:k-1})p(x_k|z_{1:k-1})}{p(z_k|z_{1:k-1})} \quad (2.5)$$

where the normalization $p(z_k|z_{1:k-1})$ constant can be written as:

$$p(z_k|z_{1:k-1}) = \int p(z_k|x_k)p(x_k|z_{1:k-1}) dx_k \quad (2.6)$$

Using the filtering density $p(x_k|z_{1:k})$, a *filtered estimate* $\hat{x}_{k|k}$ of the hidden state can be calculated as the expected value or first-order moment of the posterior distribution:

$$\hat{x}_{k|k} \triangleq E[x_k|z_{1:k}] \quad (2.7)$$

$$= \int x_k p(x_k|z_{1:k}) dx_k \quad (2.8)$$

2.1.2 Smoothing

Although the filtering density provides a fairly accurate way to calculate an instant estimate of the state of the system, we might expect an improvement if we incorporate more information in the production of the estimate. Rather than considering only the past and current observations, the accuracy of the filtered estimate can be improved by also taking into account future observations. This procedure, which is also named *smoothing*, can be performed

by either delaying the production of state estimates by some time-lag or by post-processing the entire observations data set [1]. In both cases, the *smoothed estimate* $\hat{x}_{k|k+L}$ considers a lengthier data frame $k + L$ than the filtered estimate, which only examines observations up to the current time step k . In the case of maximum-likelihood estimation of the inferred state, the improvement due to smoothing at any time k is monotonically increasing with the smoothing lag $L > 0$ [87]. Nevertheless, a more general formulation of the smoothing problem can be achieved by considering the *smoothing distribution*:

$$p(x_k|z_{1:k+L}) = \int p(x_k|x_{k+1})p(x_{k+1}|z_{1:k+L}) dx_{k+1} \quad (2.9)$$

The estimator $\hat{x}_{k|k+L}$ takes more information into account, so it is expected to be more accurate than its filtered counterpart $\hat{x}_{k|k}$. Since the smoothed estimator $\hat{x}_{k|k+L}$ depends on future, current and past observations of the process, it is also said to be a non-causal function of the observations. Just as in the filtered case, the smoothed estimate can be obtained from the smoothing distribution:

$$\hat{x}_{k|k+L} \triangleq E[x_k|z_{1:k+L}] \quad (2.10)$$

$$= \int x_k p(x_k|z_{1:k+L}) dx_k \quad (2.11)$$

A remarkable feature of smoothing algorithms is that they are completely determined by their corresponding filters. Once we have computed a filtered estimate, it is possible to derive a backward Markovian recursion (or a suitable re-parameterization) that represents the smoothing distribution [10].

Any method for computing the recursion in Equation 2.9 gives a smoothing algorithm, and depending on how much extra information is incorporated, three different strategies have traditionally been followed for implementing the method [57]:

-
1. Fixed-interval: Estimates the state $\hat{x}_{k|T}$ at time $k < T$, where T represents the final time step for all available observations.
 2. Fixed-lag: Estimates the state $\hat{x}_{k-L|k}$ at time $k - L$ with L being a positive valued number for the time lag.
 3. Fixed-point: Estimates the state $\hat{x}_{k|k+N}$ at time k using an observation from N steps ahead.

2.2 Filtering and smoothing recursions for linear state-space models

When the state-space is linear with Gaussian additive noise, the well-known Kalman filter [58] achieves the solution for the optimal estimation problem. Next, we formulate the filtering and smoothing equations for the linear Gaussian case.

2.2.1 Kalman filter and smoother

The Kalman filter is the most popular technique for handling linear models with Gaussian distributed noise. Since its formulation in the seminal paper by Kalman [58], it has been widely adopted in several engineering branches such as automatic control, computer vision and signal processing. Nevertheless, its use has also been extended to more general filtering and time series problems like economic forecasting and neural networks modeling [40].

When the state space can be written as a linear dynamic model with zero-mean Gaussian noise sources $v_k \sim \mathcal{N}(0, Q_k)$ and $w_k \sim \mathcal{N}(0, R_k)$, the posterior density is also Gaussian so it can be completely parameterized by its mean and covariance. Let A_k and B_k be two matrices defining a linear transformation for the process and observation equations. Q_k and R_k represents the process and observation noise covariance respectively.

The linear Gaussian state space can be written now as:

$$x_k = A_k x_{k-1} + v_k \quad \text{Linear process equation} \quad (2.12)$$

$$z_k = B_k x_k + w_k \quad \text{Linear observation equation} \quad (2.13)$$

In this case, equations 2.3 and 2.4 have known closed form, so the solutions can be written in terms of Gaussian distributions:

$$p(x_{k-1}|z_{1:k-1}) = \mathcal{N}(\hat{x}_{k-1|k-1}, \Sigma_{k-1|k-1}) \quad (2.14)$$

$$p(x_k|z_{1:k-1}) = \mathcal{N}(\hat{x}_{k|k-1}, \Sigma_{k|k-1}) \quad (2.15)$$

$$p(x_k|z_{1:k}) = \mathcal{N}(\hat{x}_{k|k}, \Sigma_{k|k}) \quad (2.16)$$

The Kalman filter [58] computes the optimal conditional mean and covariance of x_k by recursively predicting and updating a Gaussian distribution. The recursive method is optimal since using the following equations minimizes the mean square error of the observations and the predicted state:

$$\hat{x}_{k|k-1} = A_k \hat{x}_{k-1|k-1} \quad (2.17)$$

$$\Sigma_{k|k-1} = Q_k + A_k \Sigma_{k-1|k-1} A_k^T \quad (2.18)$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (z_k - B_k \hat{x}_{k|k-1}) \quad (2.19)$$

$$\Sigma_{k|k} = \Sigma_{k|k-1} - K_k S_k K_k^T \quad (2.20)$$

The term S_k denotes the covariance of an innovation matrix $\epsilon_k = z_k - B_k \hat{x}_{k|k-1}$ that (disregarding the orthogonality condition of the observations) generates a sequence of uncorrelated terms. The Kalman filter and Kalman smoother algorithms can then be derived from least squares techniques and by using properties of orthogonal projections in the Hilbert space of linear combinations [57]. The superscript T denotes matrix transposition and K_k is the so-called Kalman gain. Both terms S_k and K_k can also be written as:

$$S_k = B_k \Sigma_{k|k-1} B_k^T + R_k \quad (2.21)$$

$$K_k = \Sigma_{k|k-1} B_k^T S_k^{-1} \quad (2.22)$$

The smoothed estimate $\hat{x}_{k|T}$ is also a Gaussian distribution with covariance $\Sigma_{k|T}$. The optimal solution for the linear Gaussian problem was presented in [88], and is widely known as the Rauch-Tung-Striebel (RTS) smoother. The corresponding equations are:

$$\hat{x}_{k|T} = \hat{x}_{k|k} + G_k(\hat{x}_{k+1|T} - A_k\hat{x}_{k|k}) \quad (2.23)$$

$$\Sigma_{k|T} = \Sigma_{k|k} - G_k(\Sigma_{k+1|k} - \Sigma_{k+1|T})G_k^T \quad (2.24)$$

where the smoothing gain G_k is computed as:

$$G_k = \Sigma_{k|k}A_k^T\Sigma_{k+1|k}^{-1} \quad (2.25)$$

Although the optimal linear estimator can be completely derived from a “geometric” formulation [57], the algorithms can also be formulated in terms of probability density functions, providing a more general Bayesian filtering framework for solving the filtering and smoothing equations [45].

2.2.2 Numerical example

In order to exemplify the improvement of the smoothed estimate over the filtered one, a linear Gaussian dynamical model is estimated with the Kalman filter and smoother. A 1-dimensional linear Gaussian tracking model with constant velocity $\dot{x}_k = \dot{x}$ was simulated using $A_k = 1$, $B_k = 1$, $\dot{x}_k = .1$, $v_k = \mathcal{N}(0, 2)$, $w_k = \mathcal{N}(0, 5)$. The priors $x_0 = 0$ and $v_0 = 2$ were used to simulate the data.

The model simulates a constant velocity trajectory of a target observed with Gaussian noise. Table 2.1 and Figure 2.2 shows the estimation results of the Kalman filter and smoother. As expected, the Kalman smoother outperforms the estimates from the Kalman filter.

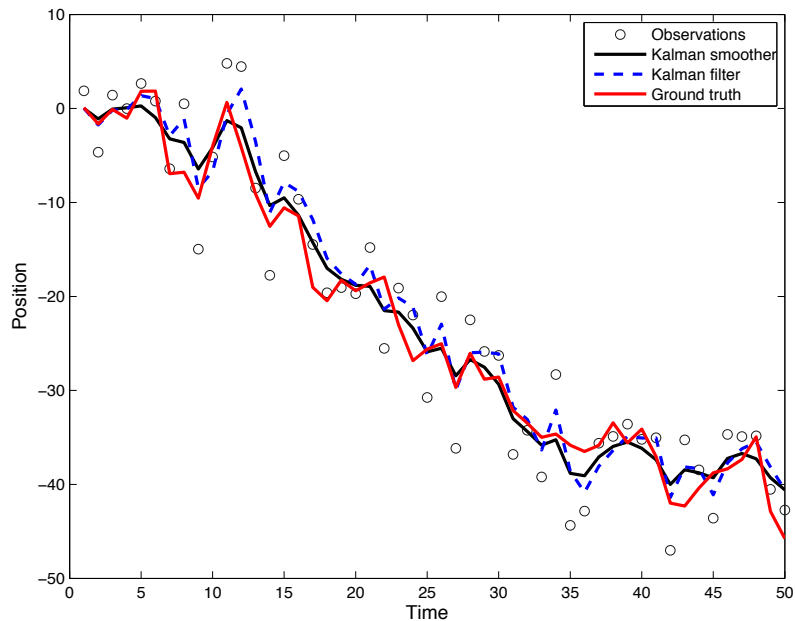


Figure 2.2: Kalman filter and smoother. Ground truth (red line) is plotted against the observed values (circles). Kalman filter and smoother estimates are also plotted (blue and black lines respectively).

	Kalman filter	Kalman smoother
RMS Error	8.54	4.01

Table 2.1: Performance comparison for the Kalman filter and smoother.

2.3 Filtering for non-linear and non-Gaussian state-space models

The Chapman-Kolmogorov equation (see Equation 2.3) forms the basis for the optimal Bayesian filtering solution, but in general it cannot be solved through an analytical formulation. Apart from the linear Gaussian and the discrete state space models, the equations have no closed form and their solution usually requires high-dimensional numerical integration. For that reason, several sub-optimal methods have been proposed in the literature for

solving the problem of estimating the state of a dynamical system under a noisy observation model.

Monte Carlo and Markov chain Monte Carlo (MCMC) methods have a long tradition in statistical physics, since their first appearance in 1949 in the seminal paper by Metropolis and Ulam [73]. Their use has been widespread among several disciplines where the use of a randomized sampling scheme can be helpful. In Bayesian statistics for example, some problems require high dimensional integration to compute expectations, therefore the analytical result might be forbidden in these cases [91].

Importance sampling is a Monte Carlo method for performing stochastic integration. When it is not feasible to draw samples from a posterior distribution, a Monte Carlo method using importance sampling can draw several independent samples from a *proposal distribution*, such that the samples are easy to draw and are in the support region of the posterior distribution.

Let's suppose we are trying to approximate a multidimensional integral $I = \int g(x)dx$, where $g(x) = x f(x)$ represents the expectation of a probability density function (pdf) $f(x)$, such that $f(x) \geq 0$ and $\int f(x)dx = 1$. Since $f(x)$ is a proper density, the integral $\int x f(x)dx$ is also defined and finite for all x . When it is not possible to directly take samples from $f(x)$, we can generate samples from a density $q(x)$, which is also termed an *importance or proposal density* and which is similar to $f(x)$ [92]. In this case, the integral I can be written in terms of the proposal density:

$$I = \int x \frac{f(x)}{q(x)} q(x) dx \quad (2.26)$$

Let $\{x^i, w^i\}_{i=1}^N$ be a set of N i.i.d. samples from $q(x)$ with the weights $w^i = f(x^i)/q(x^i)$. According to the law of large numbers, the stochastic approximation of the self-normalized integral \hat{I} represents an unbiased estimate of x , such that:

$$\begin{aligned}\hat{I} &= \frac{\frac{1}{N} \sum_{i=1}^N x^i w^i}{\sum_{i=1}^N w^i} \\ &= \sum_{i=1}^N x^i \tilde{w}^i\end{aligned}\tag{2.27}$$

Where \tilde{w}^i represents the normalized importance weights. Figure 2.3 shows a schematic diagram outlining the importance sampling principle.

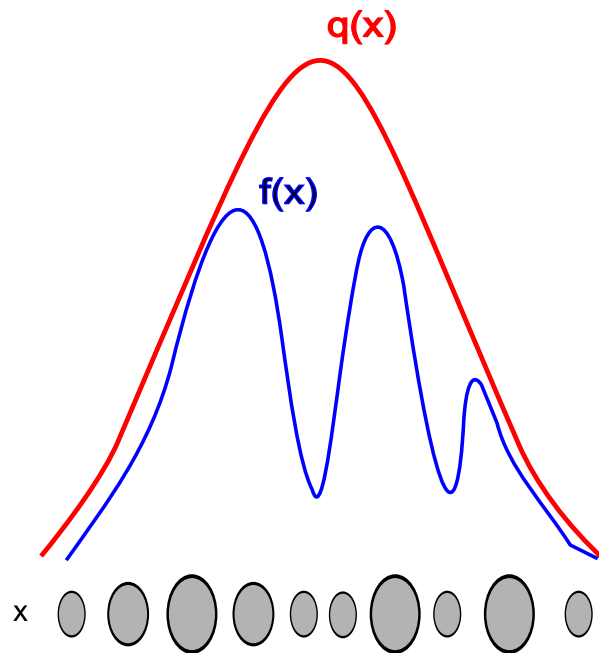


Figure 2.3: Schematic diagram for the importance sampling procedure. A proposal distribution $q(x)$ (plotted in red line) is used to generate weighted samples x (circles with radiuses being proportional to the importance weights) from a target distribution $f(x)$ (plotted in blue line).

In the case of filtering, the target distribution $p(x_{1:k}|z_{1:k})$ is a valid probability density, but due to non-linearity, multi-modality or non-Gaussianity it becomes difficult to take samples from directly. However, it can be evaluated using a related density $q(x_{1:k}|z_{1:k})$ that is designed to be easy to sample, so

it is used for generating weighted samples $\{x_{1:k}^i, w_{1:k}^i\}_{i=1}^N$. In this case, the importance weights are proportional to the posterior distribution:

$$w_{1:k}^i \propto \frac{p(x_{1:k}^i | z_{1:k})}{q(x_{1:k}^i | z_{1:k})} \quad (2.28)$$

Using the samples generated from $q(x_{1:k} | z_{1:k})$, the posterior distribution can be described by a set of discrete stochastic points $x_{1:k}^i$. For the sequential imputation of importance sampling, the proposal distribution can be chosen to have the ability to be factorizable. The importance density can then be written incrementally as:

$$q(x_{1:k}^i | z_{1:k}) \triangleq q(x_k^i | x_{1:k-1}^i, z_{1:k}) q(x_{1:k-1}^i | z_{1:k-1}) \quad (2.29)$$

Replacing the importance density from Equation 2.29 into Equation 2.28 leads to the following recursion on the weights:

$$\begin{aligned} w_k^i &= \frac{p(x_{1:k}^i | z_{1:k})}{q(x_k^i | x_{1:k-1}^i, z_{1:k}) q(x_{1:k-1}^i | z_{1:k-1})} \\ &= \frac{p(z_k | x_{1:k}^i, z_{1:k-1}) p(x_{1:k}^i | z_{1:k-1})}{p(z_k | z_{1:k-1}) q(x_k^i | x_{1:k-1}^i, z_{1:k}) q(x_{1:k-1}^i | z_{1:k-1})} \\ &= \frac{p(z_k | x_k^i) p(x_k^i | x_{k-1}^i) p(x_{1:k-1}^i | z_{1:k-1})}{p(z_k | z_{1:k-1}) q(x_k^i | x_{1:k-1}^i, z_{1:k}) q(x_{1:k-1}^i | z_{1:k-1})} \end{aligned} \quad (2.30)$$

Taking out the normalization constant $p(z_k | z_{1:k-1})$ yields the following approximation for the weights:

$$\begin{aligned} w_k^i &\propto \frac{p(y_k | x_k^i) p(x_k^i | x_{k-1}^i) p(x_{1:k-1}^i | z_{1:k-1})}{q(x_k^i | x_{1:k-1}^i, z_{1:k}) q(x_{1:k-1}^i | z_{1:k-1})} \\ &= w_{k-1}^i \frac{p(z_k | x_k^i) p(x_k^i | x_{k-1}^i)}{q(x_k^i | x_{1:k-1}^i, z_{1:k})} \end{aligned} \quad (2.31)$$

A special case of the algorithm is when the prior distribution of the states is used as an importance density, leaving $q(x_k^i | x_{1:k-1}^i, z_{1:k}) = p(x_{1:k})$. In this

case, the weight recursion can be written just in terms of the likelihood of the samples:

$$w_k^i \propto w_{k-1}^i p(z_k | x_k^i) \quad (2.32)$$

This methodology is known as sequential importance sampling (SIS) [90] and yields the following approximation for the marginal filtering density:

$$p(x_k | z_{1:k}) \approx \sum_{i=1}^N w_k^i \delta(x_k - x_k^i) \quad (2.33)$$

The SIS algorithm is computationally simple and straightforward to implement, and it forms the basis of the sequential Monte Carlo approach for non-linear and non-Gaussian state space models. However, it also has some undesired behavior that leads to a degeneracy of the algorithm. The following subsections show some strategies to overcome those problems.

2.3.1 Sequential Importance Resampling : Particle Filter

The optimal proposal distribution $q(x_{1:k} | z_{1:k})$ should be chosen to be as close as possible to the target distribution $p(x_{1:k} | z_{1:k})$. Because this is not possible in most situations, sub-optimal choices must be taken. However this can also lead to an increase in the estimator variance over time. This issue has been called the *degeneracy problem* and means that most of the samples will have negligible weights after a few steps. The degeneracy problem has a deep impact on the SIS estimate, leaving the filtering distribution $p(x_k | z_{1:k})$ being approximated by a single Dirac measure [10].

The sequential importance resampling (SIR) filter (aka particle filter) [38] solves the problem of the skewed distribution of the weights of the SIS method, by means of eliminating low weighted samples (particles) and multiplying samples in the important regions. This is achieved by resampling with replacement N times from the discrete representation of the marginal

posterior density in Equation 2.33. The rationale behind this is that after having acquired a set of weighted samples with normalized weights, a new set of samples can be generated by sampling with replacement with probability of sampling proportional to the weights. Figure 2.4 shows a schematic diagram for this procedure.

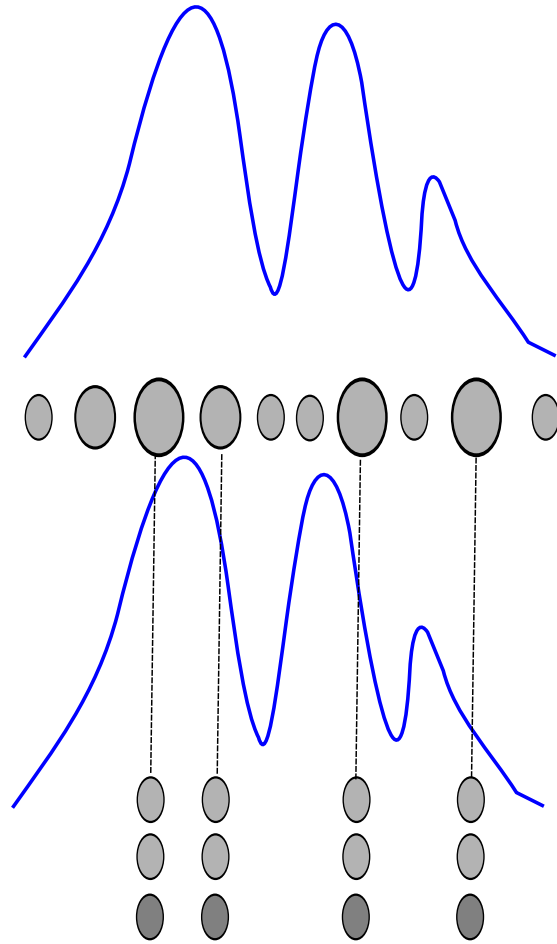


Figure 2.4: Schematic diagram for the importance resampling procedure. A target distribution (plotted in blue line) is approximated by a set of weighted samples (circles with radiuses proportional to the weights). When resampling, particles with low weights are discarded and particles with high weights are multiplied. New particles have equal weights.

The weight w_k^i of each particle is used for eliminating low weighted particles and resampling the surviving samples. This procedure is the same as sampling the number of offspring for each particle according to a multinomial distribution with parameter w_k^i , and returning a set of equally weighted propagated particles. The weighted measure $\{x_k^i, w_k^i\}_{i=1}^N$ is transformed into an un-weighted measure $\{x_k^{i*}, \frac{1}{N}\}_{i=1}^N$, but each x_k^{i*} is not an i.i.d. sample from the posterior density anymore. However, theoretical results shows that the new set of samples is asymptotically independent as the sample size N goes to infinity [10].

A comparison of different strategies for implementing resampling on the SIR algorithm can be found in [22].

2.3.2 Auxiliary particle filter

The SIR algorithm uses resampling schemes to help the degeneracy problem of the SIS filter. Although this is a great improvement on the basic methodology, there is still no clue on how to approximate the optimal proposal distribution. In that sense, the introduction of auxiliary variables into the SIR filter can overcome some of the weaknesses of such loss of information [85].

The key idea in the auxiliary SIR (ASIR) is to perform resampling at time step $k - 1$ using information from time k , before propagating the particles. This is achieved by introducing an auxiliary index j in the importance density $q(x_k, j, z_{1:k})$, such that $\{x_k^j, j\}_{j=1}^N$ is a sample from $p(z_k | \mu_k^j) p(x_k | x_{k-1}^j) w_{k-1}^j$, where μ_k^j is a projection of x_k given x_{k-1}^j .

2.3.3 Numerical example

In order to achieve better understanding of the methodologies, the SIR and ASIR algorithms are implemented for the test problem in the previous example. Both sequential Monte Carlo methods are implemented for estimating the position and velocity of the target. Figure 2.5 shows the particle approx-

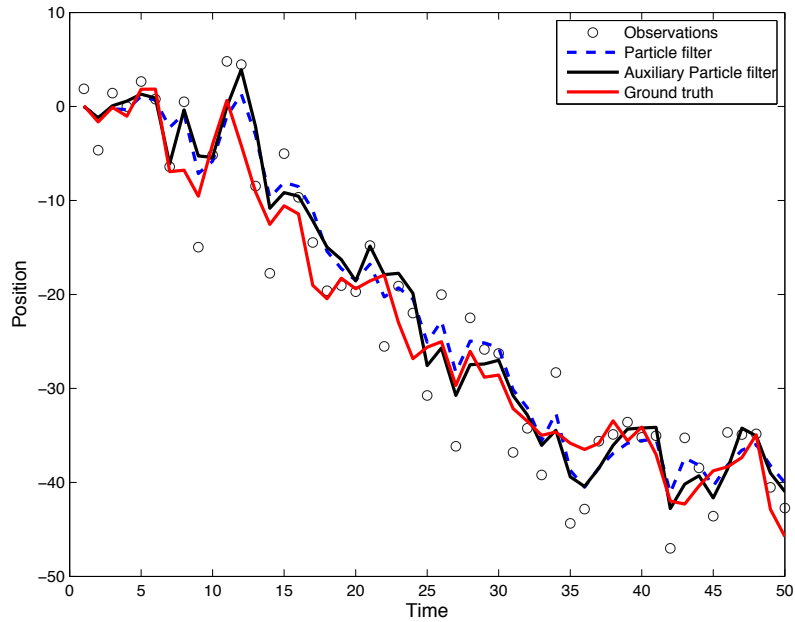


Figure 2.5: SIR and ASIR particle filter estimation for a linear Gaussian model.

imation of the SIR algorithm and the SIR and ASIR estimates along with the ground truth and the observations.

Table 2.2 shows the performance of the SIR and ASIR particle filters with the same dataset using a different number of particles each time. In this example, the ASIR algorithm does not provide an improved estimate over the SIR algorithm. Nevertheless, it is easy to see that both algorithms improve the RMS error of the estimated values as the number of particles increases.

Sample size	SIR	ASIR
100 particles	9.85	9.78
300 particles	9.44	9.45
500 particles	9.31	9.12
1000 particles	9.08	9.11

Table 2.2: Performance comparison of the SIR and ASIR particle filters.

2.4 Smoothing for non-linear/non-Gaussian state-space models

Monte Carlo approximations to the posterior distribution of the filtering density $p(x_k|z_{1:k})$ can be achieved by sequentially propagating a set of weighted samples. The output of the algorithm is the approximate expected value of the state x_k , computed from the weighted samples. However, a by-product of the algorithm is a set of particle trajectories and weights describing different paths of the system state. Interestingly, the weights of the particles can also be seen as an approximation to the smoothing distribution of the past trajectories [61]. However, this approach will only be effective for small fixed-lag smoothing, and becomes less reliable due to the successive resampling schedules.

An alternative approach for approximating the smoothing distribution would consider a backward recursion, which is an extension of the Kalman smoother to the non-linear non-Gaussian case. Once a Monte Carlo particle filtering has been performed in a forward-pass, a backward-pass can be used to modify the existing weights conditioning on future states. Using the weighted samples from the forward-step, it is relatively easy to elaborate a particle approximation to smoothing distribution, by re-weighting the existing particles and leaving the locations of the particles x_k^i without any change.

2.4.1 Forward-Backward smoothing

It is possible to derive the *marginal smoothing distribution* $p(x_k|z_{1:T})$ using the following factorization:

$$p(x_k|z_{1:T}) = \int p(x_k, x_{k+1}|z_{1:T}) dx_{k+1} \quad (2.34)$$

$$= \int p(x_{k+1}|z_{1:T}) p(x_k|x_{k+1}, z_{1:T}) dx_{k+1} \quad (2.35)$$

$$= \int p(x_{k+1}|z_{1:T}) p(x_k|x_{k+1}, z_{1:k}) dx_{k+1} \quad (2.36)$$

$$= p(x_k|z_{1:k}) \int \frac{p(x_{k+1}|z_{1:T}) p(x_{k+1}|x_k)}{p(x_{k+1}|z_{1:k})} dx_{k+1} \quad (2.37)$$

In [23], it has been shown that algorithm (1) can be derived from such factorization:

Algorithm 1 Forward-Backward smoother

- 1: Forward pass.
 - 2: **for** $k = 0, \dots, T$ **do**
 - 3: Perform SMC to get particles and weights $\{x_k^i, w_k^i\}_{1 \leq i \leq N}$.
 - 4: **end for**
 - 5: Choose $w_{T|T}^i = w_T^i$.
 - 6: Backward pass
 - 7: **for** $k = T - 1, \dots, 0$ **do**
 - 8: **for all** $i \in \{1, \dots, N\}$ **do**
 - 9: Calculate $w_{k|T}^i = \sum_{j=1}^N w_{k+1|T}^j \frac{w_k^i p(x_{k+1}^j|x_k^i)}{\sum_{l=1}^N w_k^l p(x_{k+1}^j|x_k^l)}$
 - 10: **end for**
 - 11: Normalize modified weights $w_{k|T}$
 - 12: **end for**
-

The resulting algorithm has $O(N^3)$ complexity for each smoothed time-step, but it can be easily reduced when computing the normalization constant outside of the main loop, so the complexity is reduced to $O(N^2)$.

2.4.2 MAP particle smoothing

Since the *joint smoothing distribution* $p(x_{1:T}|z_{1:T})$ can be factorized as:

$$p(x_{0:T}|z_{1:T}) = p(x_T|z_{1:T}) \prod_{k=1}^{T-1} p(x_k|x_{k+1|T}, z_{1:T}) \quad (2.38)$$

where:

$$p(x_k|x_{k+1|T}, z_{1:T}) \propto p(x_k|z_{1:k})p(x_{k+1}|x_k) \quad (2.39)$$

and the new weights can be constructed as:

$$w_{k|k+1}^i = \frac{w_k^i p(x_{k+1}|x_k^i)}{\sum_{j=1}^N w_k^j p(x_{k+1}|x_k^j)} \quad (2.40)$$

As given in [35], algorithm (2) describes a particle re-weighting method for the smoothing recursion.

The algorithm is again independent of the filtering scheme that has been used, but the smoothing recursion has the nice property of having $O(N)$ complexity with respect to the number of particles. Although backward simulation is implicitly assumed when sampling \tilde{x}_k , this is automatically achieved when simulating an auxiliary backward index, so no backward filtering has to be performed.

Despite the $O(N)$ complexity, the method achieves convergence to the joint smoothing distribution by iterating the algorithm several times.

2.4.3 Two-filter and generalized information smoother

An alternative solution for the marginal smoothing estimation problem can be achieved when considering a *Markovian backward kernel*. In this case, apart from the forward filtering step, a backward prediction step is also conducted. The marginal smoothed distribution can now be written as the product of two filtering schemes:

Algorithm 2 MAP particle smoother

- 1: Forward pass.
 - 2: **for** $k = 0, \dots, T$ **do**
 - 3: Perform SMC to get particles and weights $\{x_k^i, w_k^i\}_{1 \leq i \leq N}$.
 - 4: **end for**
 - 5: Choose $\tilde{x}_T = x_T^i$ with probability w_T^i .
 - 6: Backward pass
 - 7: **for** $k = T - 1, \dots, 0$ **do**
 - 8: **for all** $i \in \{1, \dots, N\}$ **do**
 - 9: Calculate $w_{k|k+1}^i \propto w_k^i p(\tilde{x}_{k+1} | x_k)$
 - 10: **end for**
 - 11: Normalize modified weights $w_{k|k+1}$
 - 12: Choose $\tilde{x}_k = x_k^i$ with probability w_k^i .
 - 13: **end for**
 - 14: $\tilde{x}_{1:T} = \{\tilde{x}_1, \dots, \tilde{x}_T\}$ is an approximate realization from $p(x_{0:T} | z_{1:T})$.
-

$$p(x_k | z_{1:T}) = p(x_k | z_{1:k-1}, z_{k:T}) \quad (2.41)$$

$$= \frac{p(x_k | z_{1:k-1}) p(z_{k:T} | x_k)}{p(z_{k:T} | z_{1:k-1})} \propto p(x_k | z_{1:k-1}) p(z_{k:T} | x_k) \quad (2.42)$$

While the first part of equation (2.41) remains as the usual forward filtering recursion, the second part $p(z_{k:T} | x_k)$ is sometimes named the *backward information filter* [71]. The backward in time equation can be computed recursively using the following equation:

$$p(z_{k+1:T} | x_k) = \int p(z_{k+1:T} | x_{k+1}) p(x_{k+1} | x_k) dx_{k+1} \quad (2.43)$$

$$p(z_{k:T} | x_k) = p(z_k | x_k) p(z_{k+1:T} | x_k) \quad (2.44)$$

In the specific case of linear Gaussian models, this decomposition is achieved by the so-called *two-filter formula* [32]. It has been noted in the

literature that the integral in Equation 2.43 might not be defined as a valid probability density function, so it is not possible (in general) to approximate as a probability measure [9]. However, in certain cases the integral is known to be finite and therefore can be approximated using common stochastic approximation techniques [61],[52]. In these cases, the prior distribution of the states has known analytical form and the following algorithm can be written:

Algorithm 3 Two-Filter smoother

- 1: Forward pass.
 - 2: **for** $k = 0, \dots, T$ **do**
 - 3: Perform SMC to get particles and weights $\{x_k^i, w_k^i\}_{1 \leq i \leq N}$.
 - 4: **end for**
 - 5: Choose $w_{T|T}^i = w_T^i$.
 - 6: Backward pass
 - 7: **for** $k = T - 1, \dots, 0$ **do**
 - 8: **for all** $i \in \{1, \dots, N\}$ **do**
 - 9: Calculate $w_{k|T}^i = \sum_{j=1}^N w_{k|T}^j \frac{w_k^i p(x_{k+1}^j | x_k^i)}{\sum_{l=1}^N p(y_k | x_k^l) p(x_{k+1}^j | x_k^l)}$
 - 10: **end for**
 - 11: Normalize modified weights $w_{k|T}$
 - 12: **end for**
-

A solution for the general backward filtering problem is to define an *artificial backward distribution* $\gamma_k(x_k)$, such that $\tilde{p}(x_k | z_{1:T}) \propto \gamma_k(x_k) p(z_{1:T} | x_k)$. From Equation 2.43:

$$\tilde{p}(x_k | z_{k:T}) \propto \gamma_k(x_k) p(y_k | x_k) \int p(x_{k+1} | x_k) \frac{\tilde{p}(x_{k+1} | z_{k+1:T})}{\gamma_{k+1}(x_{k+1})} dx_{k+1} \quad (2.45)$$

Having achieved a forward-pass, a backward-pass filtering is used to simulate new particles from the marginal smoothing distribution. A re-weighting scheme can be used to combine the estimates from the forward-pass with the ones from the backward-pass.

2.4.4 Numerical example

The forward-backward, the MAP and the two-filter smoothers are implemented for the linear Gaussian problem. Figure 2.6 shows the filtered and smoothed estimates.

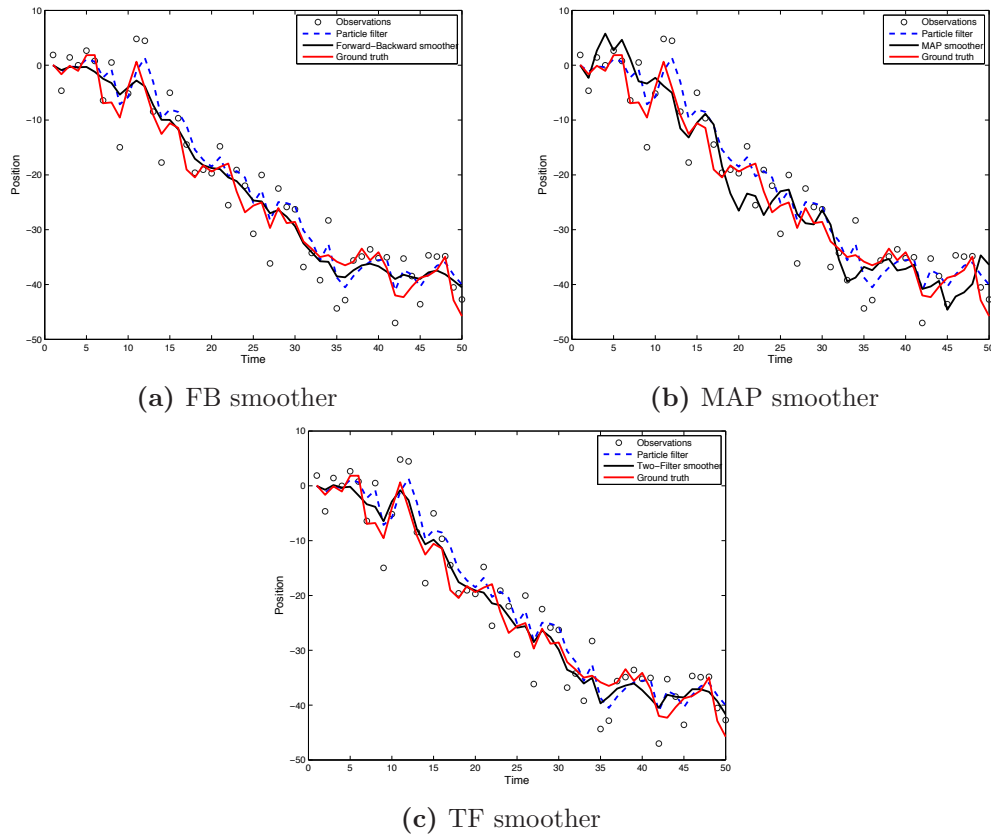


Figure 2.6: Particle smoothing for the linear Gaussian model. The Forward-Backward in (a) and the Two-Filter smoother in (b) provide improved results over the filtered estimates. This is not the case for the MAP smoother in Figure (c), which requires multiple iterations in order to converge.

Table 2.3 compares the forward-backward, the MAP and the two-filter smoothers for the linear Gaussian problem. Both, the forward-backward and the two-filter smoothers provides improved estimates, but this is not the case for the MAP smoother. The reason is that the MAP smoother requires

several iterations to converge.

Sample size	Forward-Backward	MAP	Two-Filter
100 particles	5.41	15.97	4.48
300 particles	5.34	14.95	4.29
500 particles	4.52	11.22	4.04
1000 particles	4.34	10.19	4.02

Table 2.3: Performance comparison of particle smoothing strategies.

Comparing the results of the three smoothers in Table 2.3 with the SIR and the ASIR particle filters in Table 2.2, shows that forward-backward and the two-filter smoothers both improve the RMS error over the filtered estimates. Increasing the sample size also improves the RMS error of all the three smoothers, but more dramatically in the MAP smoother which locates a candidate particle from the forward pass. In this linear Gaussian example, the support of the filtering distribution provides a good approximation to the optimal smoothing distribution, therefore the forward-backward smoother achieves the best performance of the three algorithms. In strongly non-linear cases, we should expect some level of discrepancy between the filtering and smoothing distributions, so a two-filter approach would be a better alternative.

2.5 Summary and Contributions

In this background chapter we introduced the most common approaches for estimation in dynamic models. The Kalman filter was first discussed for the task of recursively obtaining a filtered estimate of a dynamic linear model. Since these models are commonly found in visual tracking problems in surveillance, this case was also illustrated with numerical examples. Once we established the correspondence between state-space models and the sequential estimation procedure, smoothing was introduced as a backward-recursive method for reducing the variance of the estimator. More specifically, filtered estimates use previous and current observations, while smoothed estimators depend on arbitrary larger time frames.

When the state-space is discrete, the expected value of the system state can be calculated as a sum of discrete probabilities. But when a continuous state-space is under consideration, the sum becomes an integral, so the estimation problem consists of computing the integral of the random vector over the sample space with respect to a base reference measure [4]. This interpretation of the state-space model is not needed for describing linear Gaussian dynamic systems, but it becomes the key to the non-linear and non-Gaussian case.

Sequential Monte Carlo approaches, such as the SIR and ASIR algorithms can be regarded as more general simulation-based algorithms for the recursive estimation task. Although the SMC approach can approximate complex non-linear and non-Gaussian posterior distributions, the method can be also used for the dynamic linear case, where all related densities are Gaussian. Moreover, the importance sampling procedure also allows the linear Gaussian state equation to be used as proposal distribution, and this leads to the aforementioned methods having straightforward implementations.

The Forward-Backward and the Two-Filter Monte Carlo smoothers were then introduced. The Forward-Backward particle smoother performs a backward-recursive pass where particles are re-weighted by means of the state transi-

tion model. The Two-Filter smoother on the other hand, can theoretically re-weight a new set of samples in the backward pass, however there is no obvious proposal distribution as in the forward pass. Furthermore, the existence of a backward filter is not guaranteed, so artificial kernels have to be used instead. Again, the choice of the artificial distributions has to be carefully designed for the application, with no general recommendations for approximating the optimal backward kernel. Fortunately, this is not the case for linear Gaussian models, where the backward information filter is also Gaussian and the smoother can be evaluated without changing the support from the filtering distribution.

In this Chapter, sequential Monte Carlo filters and smoothers for the linear Gaussian model are compared each other. Kalman and Monte Carlo smoothers outperform the corresponding filters in the dynamic linear model numerical example. The Forward-Backward smoother does so by comparing samples at different time steps, while the Two-Filter smoother also takes into account the observation data. It has been recently noticed that since the standard particle smoothers do not change the sample support, they could also cause degeneracy problems when some level of discrepancy exists between distributions and specially when the smoothing interval increases [27]. Section 2.4 shows that this discrepancy problem does not affect the performance of fixed-interval particle smoothers for the linear Gaussian model. However, in a more general (non-linear or non-Gaussian) setting, fixed-lag particle smoothing would be preferred where limited dissimilarity between the successive distributions may exist.

3

Probabilistic inference in multi-target tracking

Real world tracking applications like urban surveillance, radar and sonar usually involve finding more than one moving object in the scene. In these cases, it remains a great challenge for the standard single-target Bayesian filter to effectively track multiple targets. The dynamic system in consideration must take into account targets appearing and disappearing possibly at random times, while also coping with the possibility of clutter and non-detected targets. Moreover, additional uncertainty in the dynamic and the sensor model can make the single target likelihood function a poor characterization of the real system, so the performance of multiple coupled filters can be far from optimal [108].

A Bayesian method for dealing with multiple and unknown numbers of

targets was developed by Ronald P. Mahler [68] and the resulting algorithm was called the *probability hypothesis density* (PHD) filter. The method uses a first-order moment approximation for the posterior distribution of a random set, or equivalently a multi-dimensional random point process. A previous formulation of the multi-target problem using the point process formalism was proposed in [124], where a time-varying finite set of observations is represented as a *random counting measure* being defined as a point process. Instead of considering the multiple target observation model as a list of measurements, the approach proposed by Washburn used a representation in terms of random scattered points. The method was developed theoretically by assuming a known number of targets, and had a naturally appealing interpretation in terms of radar or sonar returns being monitored on a screen. Further explorations of the point processes approach considered an unknown and time-varying number of targets that has to be estimated alongside the individual target states. In this context, the approach taken by Miller, Srivastava and Grenader considered a continuous-time model for the estimation and recognition of multiple targets [74]. The approach taken by Mahler was theoretically developed using a re-formulation of point process theory named *finite-set statistics* (FISST) [67] and significantly differs from the previous works, since it considered random finite sets (RFS) evolving in discrete time. In FISST, the multi-target and multi-sensor data fusion problem is treated as an estimation problem of a single “global target” observed by a single “global sensor”, both being represented by random sets. Given that the expected value of an RFS cannot be mathematically expressed using the standard Bayesian filtering equations from Chapter 2, filtering and update equations for the PHD filter were developed using set derivatives and set integrals of belief mass functions [37].

The expectation of a random set requires set integration, which can be calculated by considering all singletons (sum of indicator functions) that almost surely belongs to the random set [2]. In FISST, the expectation of a random set is calculated (in a similar way as generating functions of random

variables) by taking functional derivatives of a sum of bounded functions that characterize the distribution of a point process [70]. This derivation leads to analytical forms for the Bayesian prediction and update equations of the recursive multi-target densities. However, an alternative derivation can be also achieved if the state-space is discretized into a finite number of disjoint bins containing a single target. The multi-target density is then recovered by taking the sum of the probabilities of all bins and the volume of each bin taking infinitesimally small values. This technique was called the *physical-space* approach and was explored in [25] and [43]. More recent formulations suggest that filtering and update equations can be also obtained by means of transformations of a point processes (see references [97] and [105],[104]). The posterior multi-target density is not (in general) a Poisson process, but it can be approximated to be Poisson distributed by taking the product of all normalized and unit-less marginal single-target densities.

In this Chapter, we present the point process formalism for the multi-target problem. Section 3.1 describes some aspects from point processes theory that will be useful for modeling multi-target densities, and Section 3.2 presents the PHD filter. The PHD filter develops a recursive approach for the multi-target problem by propagating multi-target densities and interestingly, as shown in Section 3.4, the SIR algorithm from Chapter 2 can be used without any extra modification. However, instead of a single (vector-valued) Monte Carlo estimate, the particle approximation to the PHD filter is used to estimate multiple targets. Section 3.6 presents a novel Bayesian approach to the multi-target state estimation problem.

3.1 Finite point processes

A *point process* is a random pattern of points in a possibly multi-dimensional space. A simple or finite point process can be defined in one dimension, which is usually time, and can be used to describe the random times where some event can occur. Examples of such point processes are the times of

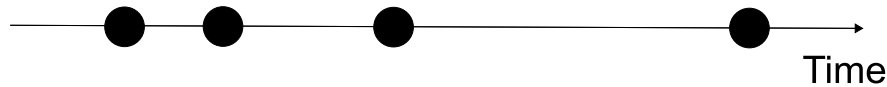


Figure 3.1: *Realizations of a point process.*

vehicle collisions in a particular street, or the times when a hospital receives emergency calls. A stochastic point process can be used to describe the random number of events in a specified time frame and the instant of time of the events.

Point processes in spaces of higher dimensionality are called *spatial point processes* [51]. For example, if we analyze earthquake epicenters or crime locations in a determined time period, the locations of the events are situated in a 2-D or 3-D region. As in the 1-dimensional case, there are a random number of points located at random positions, but the points may have also some extra associated information (“marks”). In the crime analysis case, the times of occurrence are the points so the marks could be the locations of the crimes in a particular neighborhood and time frame. In this case the stochastic process would be named a *marked point process*.

A 1-dimensional point process in time can be studied by observing either the number of events or the inter-arrival times. Sometimes, we can take advantage of the independence of the events in disjoint subsets of the temporal dimension, so the number of events arriving up to a specific time can be obtained by summing all of them, and the counting process is a continuous function of time. In higher dimensions, the events do not follow any natural ordering, so there is no analogue for the inter-arrival times of the counting process. For this reason, it is more useful to generalize the disjoint intervals defined in a state space \mathbb{B} . Figure 3.2 shows a realization of the spatial point process in a subset of \mathbb{R}^2 .

An approach to define a RFS is to count the numbers of events in a specified region. If the region B is also defined as a closed subset, then a locally finite point process can be used to construct the RFS for any bounded subset $B \in \mathbb{B}$ [89].

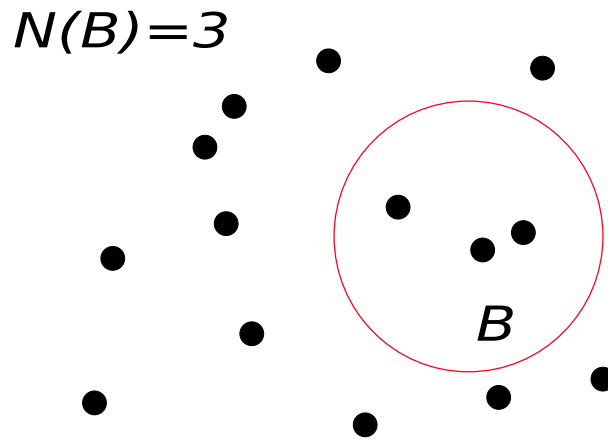


Figure 3.2: Spatial point process. The number of events in a region B is a point process in \mathbb{R}^2 .

Definition 1. A measure is called a counting measure $N(B)$ if it only takes non-negative integer values and is locally finite.

A finite point process can also be defined by a *counting measure* $N(B)$. A consequence of the counting measure formalism is that a random finite set Ξ can be generated by a point process. The number of occurrences of the process in the set B can then be expressed as $N(B) = |\Xi \cap B|$. Furthermore, B can be also expressed as a family of bounded closed sets in a generic topological space \mathbb{B} , so it can also be written as a union of disjoint sets $B = B_1 \cup B_2 \cup \dots \cup B_n$, such that:

$$N(B) = \sum_i^n N(B_i) \quad (3.1)$$

Equation 3.1 gives the intuitive notion of a counting measure as a summation over disjoint sets B_i . Such a formulation is useful for describing a random set Ξ , since any bounded subset B of \mathbb{B} is also defined as being locally finite.

3.1.1 Moments and transformations of a point process

An important feature of point processes is their invariance under several transformations. It is possible to construct new point processes by modifying or changing an existing point process, so the basic properties remains closed under transformations [16]. Although these properties are fairly general for most point process models, it is particularly useful to define *superposition*, *thinning* and *translation* for a Poisson process. The Poisson process is the simplest point process, where events are independent of each other. Furthermore, if there is also no preferred location of the events, the process is said to be homogeneous and the rate or *intensity* λ is a constant number that specifies the average density of the points per unit area. The expected number of points is then proportional to the total area of B :

$$E [N(B)] = \lambda \text{Area}(B) \quad (3.2)$$

A more general formulation for the expectation of a point process given in Equation 3.2 would allow the process to take different values from place to place. In this case, the intensity of the process is a function $\lambda(x)$. The probability of an event has to be estimated in a small area dx around a location x , so the number of points in a unitary area dx is $\lambda(x) dx$. The expected number of points in a region B can then be calculated by integrating the *intensity function* $\lambda(x)$:

$$E [N (B)] = \int_B \lambda(x) dx \quad (3.3)$$

The first-order moment of a point process, given by the *intensity measure* $\Lambda(B)$ can then be defined as:

Definition 2. Given that $\Lambda(B) \leq \infty$ for all B . The intensity measure $\Lambda(B)$ also defines a measure on \mathbb{B} such that:

$$\Lambda(B) = \int_B \lambda(x) dx \quad (3.4)$$

Let's suppose now we are observing multiple queues in the branch of a bank. If each queue has its own Poisson (homogeneous) arrival rate, then we can say that the total number of customers at a particular time is Poisson distributed and the mean corresponds to the sum of the individual arrival rates of each cue. This is called the superposition property of a Poisson process and can be defined as:

Definition 3. If N_1, N_2, \dots, N_n are independent Poisson processes in \mathbb{B} with rates $\lambda_1, \lambda_2, \dots, \lambda_n$, then $N = N_1 + N_2 + \dots + N_n$ is a Poisson process with intensity $\lambda_1 + \lambda_2 + \dots + \lambda_n$.

Another transformation of a Poisson process occurs when splitting the process into several processes. Let's suppose now we want to estimate how many bank customers are male or female. If we know the probability of each independent event (male or female), then the number of arrivals of each gender is still a Poisson process with mean rate proportional to the probability of survival ("retention") p of each event. Splitting a Poisson point process into one or several point processes with probabilities p is called a p -thinning:

Definition 4. Let $N(B) = N(B_1) + N(B_2) + \dots + N(B_n)$ be a Poisson process with intensity λ . $N(B_1), N(B_2), \dots, N(B_n)$ are independent Poisson processes with rates $\lambda_1, \lambda_2, \dots, \lambda_n$, then $N(B_j)$ is a π_j -thinned Poisson process of N with $\sum_j^n \pi_j = 1$ and rate $\lambda_j = \pi_j \lambda$.

A useful transformation for studying the evolution of a spatial point process is *translation*. Mapping each individual point of a point process to another point process by means of fixed transformation can be useful for describing the dynamics of individual targets, while the resulting process is still a Poisson process. A more formal definition is:

Definition 5. A point process N_k is *stationary* if for each $x_k \in B$ there is a transition kernel $p(x_{k+1}|x_k)$ such that the translated point process $N_{k+1|k}$ is a point process with intensity function:

$$\lambda_{k+1|k}(x) = \int_B p(x_{k+1}|x_k)\lambda(x_k)dx_k \quad (3.5)$$

Example : Binomial process Suppose we take a fixed number of n targets in a surveillance region $W \subset \mathbb{R}^2$. Let x_1, x_2, \dots, x_n be i.i.d. random samples uniformly distributed in B . Since we have only conditioned the number of points in a subset B of W , the intensity function $\lambda(x)$ of the point process is constant, so the probability density of each point can be written as:

$$f(x) = \begin{cases} 1/\text{Area}(W) & \text{if } x \in B \\ 0 & \text{otherwise} \end{cases} \quad (3.6)$$

Since each target x_i is uniformly distributed in a 2-dimensional space, for any bounded set B we have:

$$P(x \in B) = \int_B f(x)dx \quad (3.7)$$

$$= \frac{\text{Area}(B \cap W)}{\text{Area}(W)} \quad (3.8)$$

The resulting process $N(B)$ has a binomial distribution with parameters n and $p = \text{Area}(B \cap W)/\text{Area}(W)$, and is called a *binomial process*.

3.1.2 Poisson process

Another characterization for the number of counts in a point process is given by the Poisson process. For this case, the number of elements $N(B)$ are Poisson distributed. The model has been used extensively for studying queues and other discrete processes. Let's take for example an unreliable bus driver, for which we might want to know the time of arrival given that we have waited a certain amount of time. If the arrival time of a new bus does not depend on the time we have already waited, then all subsequent events will

be independent and identically distributed with the exponential distribution. However, the total number of arrivals follows a Poisson distribution with the intensity rate being proportional to the time spent [16]. A general definition of the Poisson process would be:

Definition 6. A Poisson process in \mathbb{B} with intensity function $\lambda(x)$ satisfies:

1. For every bounded closed set B , the number of points $N(B)$ has a Poisson distribution with mean measure $M(B) = \int_B \lambda(x)dx$.
2. If B_1, \dots, B_n are disjoint regions, the random measures $M(B_1), \dots, M(B_n)$ are independent.

Following the same derivation of the binomial process, the number of counts for the Poisson process with the intensity measure $M(B)$ can be written as:

$$P\{N(B) = k\} = [M(B)^k/k!] e^{-M(B)} \quad (3.9)$$

Example : Homogeneous Poisson process A homogeneous Poisson process with intensity $\lambda > 0$ has $N(B) \sim Poisson(\lambda Area(B))$, so the intensity measure is proportional to the area of B .

3.1.3 Distribution and density of a point process

A point process can be described in statistical terms by defining the space of all possible events and then assigning probabilities to the different outcomes. The event space of a point process in \mathbb{B} is the set of all counting measures on \mathbb{B} . So far, we have considered finite point processes defined bounded sets $B \subset \mathbb{B}$. Therefore, a point process in \mathbb{B} is a mapping from the probability space generated by the Borel σ -algebra of \mathbb{B} to the set of all counting measures.

By considering the space of all counting measures on \mathbb{B} , the distribution of a point process is then specified by its finite-dimensional distributions

$N(B)$. Using Definition 1, we can calculate the probability of finding exactly k elements in B :

$$E[N(B)] = P\{N(B) = k\} \quad (3.10)$$

$$= M(B) \quad (3.11)$$

Moreover, the distribution of the point process also specifies joint probabilities such as $P\{N(B_i) = n_i \wedge N(B_j) = n_j\}$, and the probability of finding no points $P\{N(B_i \times B_j) = 0\}$ in any disjoint sets B_i and B_j . Higher-order moments of a general finite point process (where points may have multiplicity greater than 1) can be defined then by considering a moment measure M_n on the joint distribution of $(N(B_1), \dots, N(B_n))$. Higher-order moments, such as the covariance may now be defined as:

$$Cov(N(B_i), N(B_j)) = M_2(B_i \times B_j) - M(B_i)M(B_j) \quad (3.12)$$

where $M_2(B_i \times B_j) = E[N(B_i)N(B_j)]$.

The general n -th order moment measure M_n describes the properties of a set of events distributed over disjoint regions B_i . However, if the regions are not independent each other, a single event x_i can occur in more than one region of the state space. This multiplicity leads to a diagonal concentration or singularity on the density when $x_i = x_j$ with $i \neq j$. For point processes with non-repeated events, the n th-order factorial moment measure is defined as follows:

Definition 7. Given a product space $B = B_1 \times \dots \times B_n$, the n th-order factorial moment measure $M_{[n]}(B_1 \times \dots \times B_n)$ can be written as:

$$\begin{aligned} M_{[n]}(B) &= E \left[\prod_{k=1}^n N(B_k) - N(B_1 \cap \dots \cap B_n) \right] \\ &= E \left[\sum_{\substack{\neq \\ x_1, \dots, x_n}} f(x_1, \dots, x_n) \right] \end{aligned}$$

where f is any non-negative bounded function.

The factorial moment measure $M_{[n]}$ is useful for describing the expectation of a random number of elements with no multiplicity. The n -th order product density $m_{[n]}(x_1, \dots, x_n)$ is related to the factorial moment measure by:

$$M_{[n]}(B) = \int m_{[n]}(x_1, \dots, x_n) dx_1 \dots dx_n \quad (3.13)$$

For the homogeneous Poisson process, the factorial moment measure can be written in terms of the intensity function $\lambda(x) = \lambda$ and is proportional to the area of B .

$$M_{[n]}(B) = \int_{B_1} \lambda(x_1) dx_1 \dots \int_{B_n} \lambda(x_n) dx_n \quad (3.14)$$

$$= \lambda^n \text{Area}(B_1) \times \dots \times \text{Area}(B_n) \quad (3.15)$$

When considering the more general finite point process, a further distributional simplification can be achieved by restricting ourselves to a family of symmetric distributions over all permutations on the numbers $1, \dots, n$ and a discrete probability distribution p_n for the number of points.

$$\Pi_n^\sigma(B_1 \times \dots \times B_n) = \frac{1}{n!} \sum_{perm} \Pi_n(B_{\sigma_1} \times \dots \times B_{\sigma_n}) \quad (3.16)$$

The factorial moment measure can then be represented as a superposition of a finite or countably infinite family of permutation invariant measures called Janossy measures $J_n(\cdot)$.

$$J_n(B_1 \times \dots \times B_n) = n! p_n \Pi_n^\sigma(B_1 \times \dots \times B_n) \quad (3.17)$$

$$= p_n \sum_{perm} \Pi_n(B_{\sigma_1} \times \dots \times B_{\sigma_n}) \quad (3.18)$$

In this case, all events $\{x_1, \dots, x_n\}$ are permutation invariant so no preference is made over any particular order. The probability of finding exactly n points, one at each of infinitesimal locations dx_1, \dots, dx_n can then be specified by Janossy densities j_n [18].

Lemma 8 (Daley and Vere-Jones (2003)). If the k -th factorial moment measure $M_{[k]}$ exists, then it is absolutely continuous if and only if the superposition of measures are absolutely continuous for all $n \geq k$, in which case the k -th order product density $m_{[k]}(\cdot)$ and the Janossy densities $j_n(\cdot)$ are related by:

$$m_{[k]}(x_1, \dots, x_k) = \sum_{n=0}^{\infty} \frac{1}{n!} \int_B j_{k+n}(x_1, \dots, x_k, x_{k+1}, \dots, x_{k+n}) dx_{k+1} \dots dx_{k+n} \quad (3.19)$$

Janossy densities also have the property to be jointly normalized by the factor $1/n!$, so the total probability sums to one:

$$\sum_{n=0}^{\infty} \frac{1}{n!} \int j_n(x_1, \dots, x_n) dx_1 \dots dx_n = 1 \quad (3.20)$$

For the homogeneous Poisson process, we can demonstrate that the intensity function $\lambda(\cdot)$ is a valid probability density since:

$$\begin{aligned} \sum_{n=0}^{\infty} \frac{1}{n!} \int j_n(x_1, \dots, x_n) dx_1 \dots dx_n &= e^{-\lambda} \sum_{n=0}^{\infty} \frac{1}{n!} \int \lambda(x_1) dx_1 \dots \lambda(x_n) dx_n \\ &= e^{-\lambda} \sum_{n=0}^{\infty} \frac{\lambda^n}{n!} \\ &= e^{-\lambda} e^{\lambda} \\ &= 1 \end{aligned} \quad (3.21)$$

Now we can use factorial moment measures and symmetric densities from Equation 3.21 to calculate the expectation of the sum over all permutations on the events x_i . Let $h : \mathbb{B} \mapsto \mathbb{R}$ be a measurable function, such that the summation of the test function $h(x)$ is a random variable with expected value:

$$\begin{aligned} M(B) &= E\left[\sum_{x \in B} h(x)\right] \\ &= \int_B h(x) \lambda(x) dx \end{aligned} \tag{3.22}$$

This is also true for indicator functions $h(x) = \mathbf{1}_B(x)$, and the general result is known as Campbell's formula [51]. In the case of the Poisson process, the factorial moment measure can be written as:

$$M_{[n]}(B_1 \times \cdots \times B_n) = E\left[\sum_{x_1, \dots, x_n}^{\neq} \mathbf{1}_{B_1}(x_1) \cdots \mathbf{1}_{B_n}(x_n)\right] \tag{3.23}$$

And the first-order moment measure can now be written as:

$$M(B) = E\left[\sum_{x \in B} \mathbf{1}_B(x)\right] \tag{3.24}$$

3.1.4 Probability generating functionals and their derivatives

Moment generating functions and more specifically, *factorial moment generating functions* can be used to evaluate the mean, variance and higher-order moments of discrete random variables. The n th-order derivative of the moment generating function evaluated at some point, gives the n th-order moment of the random variable. Construction of generating functions by means of sums of convergent series has been demonstrated to be a valuable tool in statistics and engineering for a long time [110]. In the same way, factorial moment measures and Janossy measures can also be approximated by

means of sums of convergent series, where the power series used in generating functions for random variables are replaced by the power expression of a real-valued test function $h(x)$, such that $0 \leq h(x) \leq 1$.

Definition 9. Let $j(x_1, \dots, x_n)$ be a density of X and let h^X be a power series expansion of a test function $h(x)$, then the probability generating functional (p.g.fl) $G[h]$ is denoted by:

$$\begin{aligned} G[h] &\triangleq \int h^X p(X) dX \\ &= p(\emptyset) + \int h(x_1) j(x_1) dx_1 + \frac{1}{2} \int h(x_1) h(x_2) j(x_1, x_2) dx_1 dx_2 + \dots \end{aligned} \quad (3.25)$$

where

$$h^X \triangleq \begin{cases} 1 & \text{if } X=\emptyset \\ \prod_{x \in X} h(x) & \text{otherwise} \end{cases}$$

The sum in Equation 3.25 can be identified with an integral with respect to the factorial moment $M_{[n]}(\cdot)$. As a result of the *fundamental theorem of calculus*, the moments of the point process $N(\cdot)$ can be computed from functional derivatives of $G[h]$:

$$\frac{\delta G}{\delta X} [h] \triangleq \lim_{\varepsilon \searrow 0} \frac{G[h + \varepsilon \delta_x] - G[h]}{\varepsilon} \quad (3.26)$$

The Janossy density $j(\cdot)$ can now be written using p.g.fl. notation by means of iterated functional derivatives:

$$j_n(x_1, \dots, x_n) = \frac{\delta^n G}{\delta x_1 \dots \delta x_n} [0] \quad (3.27)$$

And the first-order factorial moment measure:

$$M_{[1]}(B) = \frac{\delta G}{\delta x_1 \dots \delta x_n} [1] \quad (3.28)$$

The extension to conditional Janossy densities and conditional expectation is done by simply introducing the conditioning factor on each equation [18]. Using this notation, Bayesian filtering equations are developed in the following section.

3.2 The Probability Hypothesis Density (PHD) filter

The problem of performing joint detection and estimation of multiple objects has a natural interpretation as a *dynamic point process*, namely a random counting measure $N(\dots)$ where the stochastic intensity $\lambda(\dots)$ of the model is a space-time function. In this case, a model for detection and estimation of multiple objects can be performed using the conditional expectation of the point process under transformations.

Using the point processes approach, we can predict the state of multiple objects, given the history of observations. The resulting model is powerful enough to allow a time-varying number of objects to appear and disappear. Furthermore, if we approximate the prior density by a Poisson process, we can also use Poisson processes for target births and clutter, therefore the posterior multi-target density is also a Poisson process.

Let $X_k = \{x_1, x_2, \dots, x_n\}$ and $Z_k = \{z_1, z_2, \dots, z_m\}$ be two RFS. Each one of the random finite sets has time-varying dimensions, so two random counting measures $N(X)$ and $N(Z)$ can be also defined. Bayesian filtering equations can be constructed in a similar fashion to the single target filtering equations 2.3 and 2.4 from Chapter 2:

$$p(X_k | Z_{1:k-1}) = \int p(X_k | X_{k-1}) p(X_{k-1} | Z_{k-1}) \delta X_{k-1} \quad (3.29)$$

$$p(X_k|Z_{1:k}) = \frac{p(Z_k|X_k)p(X_k|Z_{1:k-1})}{p(Z_k|Z_{k-1})} \quad (3.30)$$

Since X_k is also represented by a point process $N(X)$, the expectations for the RFS prediction and update equations (Equations 3.29 and 3.30) can be calculated as sums of counts in disjoint intervals. As we saw in Section 3.1, these expectations are also random measures having associated symmetric densities. Using this framework, the rest of the section develops a recursive algorithm for computing multi-target densities.

3.2.1 The Probability Hypothesis Density filter

The Kalman filter propagates the sufficient statistics for the single target case, namely the mean and covariance of the target state. If the SNR ratio is high enough, the second-order moment can be neglected and the prediction and update operators can just use the first-order moment (i.e., constant gain Kalman filter) [5]. While this assumption is too restrictive for most tracking applications, it might be justifiable when extending the case to a multiple object tracking problem. An estimate of the first-order moment of a point process can be calculated in closed form in the case of a Poisson process prior. In this case, the probability hypothesis density (PHD) $D(\cdot)$ is defined as the first-order moment measure or intensity measure of the posterior distribution of a point process with density $p(\{x_1, \dots, x_n\}|Z_{1:k}) = j_n(x_1, \dots, x_n)$. More formally, the PHD can be defined as:

Definition 10 (Probability Hypothesis Density). The PHD represents the family of Janossy densities of a point process $N(\cdot)$ using a single function $D(x)$ that takes the place of the first-order factorial moment density $m_{[1]}(x)$, such that:

$$D(x) \triangleq \int p(\{x\} \cup X) \delta X \quad (3.31)$$

$$= \sum_{n=0}^{\infty} \frac{1}{n!} \int p(\{x, x_1, \dots, x_n\}) dx_1, \dots, dx_n \quad (3.32)$$

The recursive PHD corresponds to the first-order moment of the multi-target densities in Equations 3.29 and 3.30.

$$D_{k|k-1}(x) \triangleq \int p(\{x\} \cup X_k | Z_{1:k-1}) \delta X_k \quad (3.33)$$

$$D_{k|k}(x) \triangleq \int p(\{x\} \cup X_k | Z_{1:k}) \delta X_k \quad (3.34)$$

3.2.2 Bayesian point process predictor

The PHD $D_{k|k-1}(x)$ of the predicted point process can be written as the linear superposition of a thinned point process with Markov translations and a Poisson birth process. If we let $\pi_s(x)$ and $\pi_d(x)$ be the probabilities of survival and detection of each target, then the point process representing the predicted number of targets can be written as the sum of the number of targets that have survived from time $k-1$ to k , the number of targets being spawned from other targets with probability $\gamma(x)$, and the number of targets appearing from spontaneous birth with density $b(x)$. The predicted point process $N_{k|k-1}$ for survival and spawning targets with spontaneous birth can be written as:

$$N_{k|k-1} = N_{k|k-1}^s + N_{k|k-1}^b + N_{k|k-1}^g \quad (3.35)$$

where:

$$N_{k|k-1}^s = \int D_{k-1|k-1}^s(x) dx \quad (3.36)$$

$$N_{k|k-1}^g = \int D_{k-1|k-1}^g(x) dx \quad (3.37)$$

$$N_{k|k-1}^b = \int D_{k-1|k-1}^b(x) dx \quad (3.38)$$

Here $\Phi(x) = \pi_s(x)p(x'|x) + \gamma(x)$ is an operator that describes the evolution of survived targets from time $k-1$ to k . The number of surviving targets $N_{k|k-1}^s$ is then a π_s -thinned Poisson process, $p(\cdot)$ is a Markov transition kernel, $\gamma(x)$ is a probability law for spawning targets and $b_{k|k-1}$ represents the distribution of spontaneous births. It follows from definition 3 that the result from the superposition, translation and thinning transformations is a point process with intensity $D_{k|k-1}$.

The probability of surviving targets is a π_s -thinned point process corresponds to a an i.i.d. sequence of Bernoulli probabilities with probability π_s of success. Using the fact that each target is subject to independent displacements and independent deletions, also known as *independent scattering* which by Definition 6 gives a form for a Poisson process. Therefore, the number of survival targets follow a Binomial distribution with mean or expected value being dependent on the number of targets $n' = |X_{k-1}|$. However, it can be shown that the limiting posterior Poisson process has intensity function $\int (1 - \pi_s(x) + \pi_s(x)p(x)) dx$ [105]. Leaving the density function

$$p(X_k|X_{k-1}) = \begin{cases} \prod_{i=1}^{n'} (1 - \pi_s(x_i)) & \text{if } X_k = \emptyset, \\ & \text{if } X_k = \{x_1, \dots, x_{n_k}\} \end{cases}$$

Using Equation 3.33, the PHD equation for surviving targets $D_{k|k-1}^s$ is then calculated by induction:

$$D_{k|k-1}^s(x) = \int \pi_s(x) p(x|x') D_{k-1|k-1}(x') dx' \quad (3.39)$$

Using the same induction principle, $D_{k|k-1}^g$ can be written as:

$$D_{k|k-1}^g(x) = \int \gamma_{k|k-1}(x|x') D_{k-1|k-1}(x') dx' \quad (3.40)$$

Since target births are modeled via an independent Poisson process, the density for N^b new born targets can be written as:

$$p(X_k|X_{k-1}) = e^{b(x)} \prod_{i=1}^n b_{k|k-1}(x_i) \quad (3.41)$$

with the PHD $D_{k|k-1}^b$ for the new born targets being:

$$D_{k|k-1}^b(x) = b_{k|k-1}(x) \quad (3.42)$$

Therefore, the total predicted PHD yields:

$$D_{k|k-1}(x) = \int (\pi_s(x_k)p(x|x') + \gamma_{k|k-1}(x)) D_{k-1|k-1}(x') dx' + b_{k|k-1}(x) \quad (3.43)$$

3.2.3 Bayesian point process update

Using the Poisson process prior, the observed number of measurements $|Z_k| = m$ can be approximated as a Poisson process by taking into account a probability of detection π_d of the targets and spatial Poisson clutter with intensity $\kappa_k(z) = \lambda_c c_k(z)$. Similarly to the case of the prediction step, the posterior point process consists of the superposition of a π_d -thinned point process of the detected targets $N_{k|k-1}^d$ and the non-detected targets $N_{k|k-1}^{\setminus d}$.

The density for the detected targets corresponds to the summation on the power set of equally likely hypotheses of targets and observation assignments. This density cannot be factorized (in a general setting) as a Poisson process, therefore the updated point process $N_{k|k}$ is not itself a Poisson process. Even

so, Mahler showed that a best-fit approximation can then be estimated from the current observations [68].

Firstly, the non-detected targets point process $N_{k|k}^{\setminus d}$ is then a $1-\pi_d$ -thinned Poisson process for the surviving targets in the prediction step. The likelihood for the non-detected targets can be written as a symmetric density over all permutations σ_{k-1} of X_{k-1} :

$$p(X_k|Z_{1:k}) \propto p(\emptyset|X_k) p(X_k|X_{k-1}) \quad (3.44)$$

$$= \sum_{\sigma_{k-1}} \prod_{i=1}^{n_k} (1 - \pi_d(x_i) p(x_i|x_{\sigma_{k-1}})) \quad (3.45)$$

The PHD $D_{k|k}(x)^{\setminus d}$ for the non-detected targets can then be written as:

$$D_{k|k}(x_k)^{\setminus d} = 1 - \pi_d(x_k) D_{k|k-1}(x_k) \quad (3.46)$$

Secondly, the detected point process $N_{k|k}^d$ can be estimated in the update step by taking into account the observations Z_k . $N_{k|k}^d$ is not a Poisson process but it can be approximated by taking the product of the single target densities [105]. The Poisson approximation was discussed by Mahler in terms of the Kullback-Leibler distance [68], and a quantitative evaluation was performed in [77].

If we don't take into account any false alarm, the observations are almost surely being originated by a target. Therefore, we can write a the likelihood function of the observed data as a symmetric density over all permutations σ_k of X_k :

$$p(Z_k|X_k) = e^{\Lambda_z(Z)} \sum_{\sigma_k} \prod_{i=1}^{m_k} p(z_i|x_{\sigma_k}) \quad (3.47)$$

In order to estimate the locations of the targets from data, Streit developed a framework using likelihood inference for point processes [105]. This

derivation was later extended, by comparing the data update step of the PHD filter with a previously proposed algorithm used for Positron Emission Tomography [104]. The later derivation uses the EM algorithm to estimate the locations of the targets as being missing information, so the conditional density of the locations becomes the ratio of two densities. The resulting PHD for detected targets $D_{k|k}^d(x)$ can be written as:

$$D_{k|k}(x) = \sum_{z \in Z_k} \frac{\pi_d(x) p(z|x) D_{k|k-1}(x)}{\int \pi_d(x) p(z|x) D_{k|k-1}(x) dx} \quad (3.48)$$

Introducing clutter into Equation 3.47 yields an augmented state space for permutations between target originated observations and clutter originated observations.

$$D_{k|k}^d(x) = \int \frac{\pi_d(x) p(z|x)}{\lambda_c c(z) + \int \pi_d(x) p(z|x) D_{k|k-1}(x) dx} D_{k|k-1}(x) dx \quad (3.49)$$

And the total number of targets corresponds to the targets being detected in clutter and the targets non-detected $N_{k|k} = N_{k|k}^d + N_{k|k}^{\setminus d}$.

Consequently:

$$N_{k|k}(E) \triangleq \int_E D_{k|k}(x) dx \quad (3.50)$$

The PHD update equation can be finally written as:

$$D_{k|k}(x) = \left[1 - \pi_d(x) + \sum_{z \in Z_k} \frac{\pi_d(x) p(z|x)}{\lambda_c c(z) + \int \pi_d(x) p(z|x) D_{k|k-1}(x) dx} \right] D_{k|k-1}(x) \quad (3.51)$$

The updated posterior multi-target density consists of the summation of the undetected and the detected targets. This is not an independent scattering process, so it is not a Poisson process. Using a first-order moment

approximation to the posterior density leaves an averaged intensity from all the conditional likelihoods of a measurement being originated by any particular target. Since the underlying Janossy density is symmetric under permutations, all association hypothesis are equally weighted and no preferred association is considered. Therefore, the PHD filter keeps no track of the individual target identities or “tracks”, so additional steps have to be taken for performing track-to-estimate associations [15].

3.3 Spatial discretization for the PHD filter

A physical interpretation for the PHD filter was given in [26]. In this approach, the state space is discretized into a large number of small cells c_i , so the first-order moment represents the probability surface of a single target existing on each one of the cells.

Using the physical-space approach, the predicted and updated PHD equations can be written as:

$$D_{k|k-1}(c) \triangleq \int_c D_{k|k-1}(x) dx = Pr(c \in X'_k) \quad (3.52)$$

$$D_{k|k}(c) \triangleq \int_c D_{k|k}(x) dx = Pr(c \in X_k) \quad (3.53)$$

As the volume of the cells $|c|$ tends to zero and the number of cells increases to infinity, the equations for the spatial discretization model are equivalent to the continuous-time prediction and update equations obtained from the corresponding p.g.fl..

$$\frac{D_{k|k-1}(c)}{|c_x|} \mapsto D_{k|k-1}(x) \quad (3.54)$$

$$\frac{D_{k|k}(c)}{|c_x|} \mapsto D_{k|k}(x) \quad (3.55)$$

Due to the infinitesimally small volume of the cells c_i , all transformations are mutually exclusive, so the total probability that the particular bin c_i contains a target can be written as:

$$D_{k|k-1}(c_i) = p_b(c_i) + \sum_e p_s(c_e) p_{k|k-1}(c_i|c_e) D_{k-1|k-1}(c_e) + \sum_e p_\gamma(c_i|c_e) D_{k-1|k-1}(c_e) \quad (3.56)$$

And so:

$$\frac{D_{k|k-1}(c_i)}{|c_i|} = \frac{p_b(c_i)}{|c_x|} + \sum_e \left[p_s(c_e) \frac{p_{k|k-1}(c_i|c_e)}{|c_i|} + \frac{p_\gamma(c_i|c_e)}{|c_i|} \right] \frac{D_{k-1|k-1}(c_e)}{|c_i|} |c_e| \quad (3.57)$$

Taking the limit $|c_i| \mapsto 0$ and $i \mapsto \infty$, the equation for the spatial discretization PHD predictor converges to the standard PHD equation. In the case of spatial Poisson measurement process, the number of observations $|Z_k| = m_k$ on each disjoint bin of the state space follows a Poisson distribution with non-homogeneous intensity measure $\Lambda_z(Z) = \int_Z \lambda(z) dz$, such that:

$$p(m|\Lambda_z(Z)) = \Lambda_z(Z)^{m_k} \frac{e^{-\Lambda_z(Z)}}{m_k!} \quad (3.58)$$

This gives an expression for the likelihood $p(Z_k|X_k)$ in terms of the conditional intensity $\lambda(z|x)$:

$$p(Z_k|X_k) = e^{\lambda_z(Z)} \prod_{i=1}^{m_k} \lambda(z_i|X) \quad (3.59)$$

$$= e^{\lambda_z(Z)} \prod_{i=1}^{m_k} \left[\lambda_c + \sum_{j=1}^n \lambda_j(z_i|x_j) \right] \quad (3.60)$$

The spatial discretization approach described here provides a less formal derivation for the multiple target tracking equations, without requiring measure-theoretic concepts, while at the same time, being specifically well suited for signal processing applications such as joint detection and estimation of multiple acoustic signals [43].

3.4 Sequential Monte Carlo implementation of the PHD recursion

A particle implementation for the PHD recursion was proposed by [127, 116, 96] as a method for simulating the conditional expectation of the unknown number of targets given the current observations. The SMC implementation of the PHD filter approximates the intensity measure of a dynamic point process with a set of weighted particles. Convergence results and a central limit theorem for the particle approximation are given in [14, 55].

The importance sampling estimator for random variables given in Chapter 2 is used now to approximate the first-order moment $D_{k|k}(x)$. Using the discrete estimation formula from Equation 3.24, Monte Carlo integration is used here instead for approximating the first-order moment measure of the multi-target posterior density. A set of particles $\{x^i, w^i\}_{i=1}^{L_k}$ is sampled from a proposal density $q(\cdot)$ leaving $D_{k|k}$ as being:

$$\int_B D_{k|k}(x) dx = E\left[\sum_{x_k \in B} \mathbf{1}_B(x_k)\right] \quad (3.61)$$

$$\approx \sum_{i=1}^{L_k} \mathbf{1}_B(x_k^i) w_k^i \quad (3.62)$$

Algorithm 4 describes the particle approximation to the PHD recursion as given in [116].

Algorithm 4 Particle PHD filter

Require: $k \geq 1 \wedge \{w_{k-1}^i, x_{k-1}^i\}_{i=1}^{L_{k-1}}$

- 1: Step 1: Prediction step
 - 2: **for** $i = 1, \dots, L_{k-1}$ **do**
 - 3: Sample $\tilde{x}_k^i \sim q_k(\cdot | x_{k-1}^i, Z_k)$
 - 4: Compute predicted weights $\tilde{w}_{k|k-1}^i = \frac{\phi(\tilde{x}_k^i, Z_k)}{q_k(\tilde{x}_k^i | x_{k-1}^i, Z_k)} w_{k-1}^i$
 - 5: **end for**
 - 6: **for** $i = L_{k-1} + 1, \dots, J_k$ **do**
 - 7: Sample $\tilde{x}_k^i \sim p_k(\cdot | Z_k)$
 - 8: Compute predicted weights for the new born particles

$$\tilde{w}_{k|k-1}^i = \frac{1}{J_k} \frac{\gamma(\tilde{x}_k^i)}{p_k(\tilde{x}_k^i | Z_k)}$$
 - 9: **end for**
 - 10: Step 2: Update step
 - 11: **for all** $z \in Z_k$ **do**
 - 12: Compute $C_k(z) = \sum_{j=1}^{L_{k-1}+J_k} \psi_{z,k}(\hat{x}_k^j) w_{k|k-1}^j$
 - 13: **end for**
 - 14: **for** $i = 1, \dots, L_{k-1} + J_k$, update weights **do**
 - 15: $\tilde{w}_k^i = \left[\nu(x) + \sum_{z \in Z_k} \frac{\psi_{z,k}(\tilde{x}_k^i)}{\kappa_k(z) + C_k(z)} \right] \tilde{w}_{k|k-1}^i$
 - 16: **end for**
 - 17: Step 3: Resampling step
 - 18: Compute the total mass $\hat{N}_{k|k-1} = \sum_{j=1}^{L_{k-1}+J_k} \tilde{w}_k^j$
 - 19: Resample $\{\tilde{w}_k^i / \hat{N}_{k|k-1}, \tilde{x}_k^i\}_{i=1}^{L_{k-1}+J_k}$
 - 20: **return** $\{w_k^i, x_k^i\}_{i=1}^{L_k}$
-

3.4.1 Numerical example

As an example of a multiple target tracking model, consider a 1-dimensional scenario with targets appearing at random with uniform distribution over the surveillance region $[-80, 80]$. As shown in Figure 3.3, the state of each target consists of position and velocity $x_k = [p_x, \dot{p}_x]$ and the observations z_k are linear measurements of the position embedded in Gaussian noise $w_k \sim$

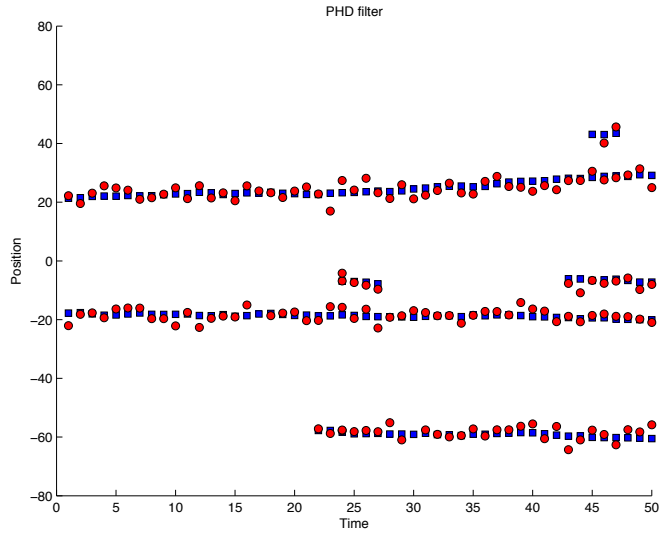


Figure 3.3: Multiple object tracking model. Targets are represented with blue squares and observations with red circles. Several targets appear at random times (horizontal axis) and random locations (vertical axis).

$\mathcal{N}(0, 5)$. Target dynamics are also linear with Gaussian noise $v_k \sim \mathcal{N}(0, 1)$.

Probabilities of survival and detection are state independent, $p_s = 0.98$ and $p_d = 0.99$ respectively. Birth rate is set at $\lambda_b = 1e - 3$, giving an average number of 0.16 targets appearing on each scan. Clutter is also simulated as a spatial Poisson process with intensity rate $\lambda_c = 4e - 2$ with uniform distribution over the region $[-100, 100]$, giving an average number of 8 clutter returns. For simplicity, target spawning is not considered here.

Figure 3.4 shows the SMC approximation to the PHD filter. The posterior multi-target density is approximated using the sampling and re-sampling scheme from the Algorithm 4.

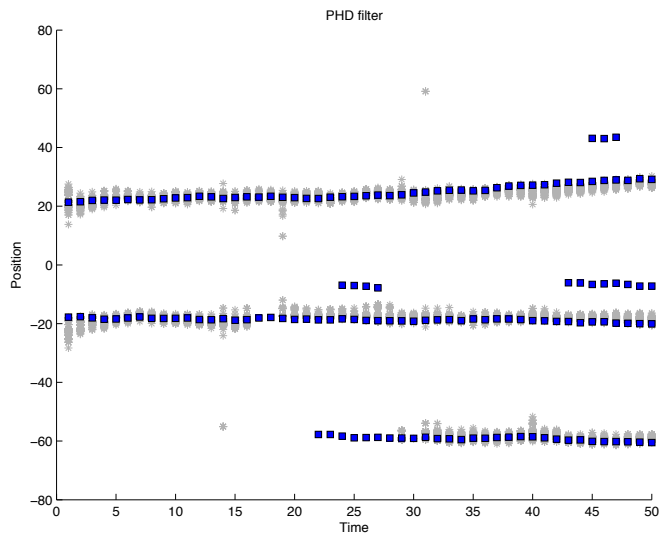


Figure 3.4: Particle filter approximation to the PHD filter. The posterior multi-target density is approximated using particles (plotted in gray). Particles are clustered around targets locations, so additional techniques have to be used for state estimation. Different strategies for state estimation are discussed in Section 3.6.

3.5 Gaussian mixture implementation of the PHD recursion

An alternative implementation can be used for the PHD recursion in the case of *linear Gaussian multi-target models*. In this case, each single-target Markov transition kernel is associated with a linear Gaussian model, so the resulting intensity measure can be represented using a mixture of Gaussian distributions also known as Gaussian mixture model [72]. Using a similar derivation for the Gaussian-sum filter for the single target case [61], a closed form solution for the PHD filter can be found by approximating the intensity function by a piecewise deterministic function such as a Gaussian mixture model [114].

In this approach, each target follows a linear Gaussian kinematic model and the observation model is also linear and Gaussian, and the equations for

each target can be written using the Kalman filter. Probabilities of detection and survival are state independent, such that $p_{d,k}(x) = p_{d,k}$ and $p_{s,k}(x) = p_{s,k}$. Moreover, intensities for target birth $b_{k|k-1}(x)$ and spawning $\gamma_{k|k-1}(x)$, as well as the clutter probability $c(z)$ are also approximated with Gaussian mixtures:

$$b_k(x) = \sum_{i=1}^{J_{b,k}} w_{b,k}^i \mathcal{N}(\hat{x}_{b,k}^{(i)}, \Sigma_{b,k}^{(i)}) \quad (3.63)$$

$$\gamma_{k|k-1}(x_k|x_{k-1}) = \sum_{i=1}^{J_{\gamma,k}} w_{\gamma,k}^i \mathcal{N}(A_{\gamma,k}^{(i)} \hat{x}_{\gamma,k}^{(i)} + d_{\gamma,k}, \Sigma_{\gamma,k}^{(i)}) \quad (3.64)$$

$$c(z) = \sum_{i=1}^{J_{c,k}} w_{c,k}^i \mathcal{N}(\hat{x}_{c,k}^{(i)}, \Sigma_{c,k}^{(i)}) \quad (3.65)$$

The linear Gaussian multi-target model has a straightforward interpretation in the physical-space approach. A Gaussian mixture model for the spontaneous births locates each mean $\hat{x}_{b,k}^{(i)}$ as the peak of the location in a discrete state-space, where a target is likely to appear scattered with covariance $\Sigma_{b,k}^{(i)}$. The weights $w_{b,k}^i$ are chosen to sum the birth rate on each bin. Similarly, target spawning is modeled with an affine transformation of the position occupied by the target x_{k-1} .

Under these conditions, Vo et.al. [114] derived an analytical expression for the posterior intensity of the resulting point process. More precisely, the predicted and updated intensities are also Gaussian mixtures whose parameters are computed from Kalman prediction and update equations for each component of the mixture.

3.6 State estimation for the particle PHD filter

Being the first-order moment of a multi-target posterior distribution, the PHD represents the intensity function of an information-theoretic best fit Poisson process approximation. The integral of that function over the field

gives the estimated number of targets, and the positions of targets can be estimated from the peaks of the same function.

In the SMC implementation of the PHD filter, Monte Carlo samples are used to represent the intensity function, so a larger number of particles are used in areas where targets are more likely to exist. Assuming that we have sample from the posterior PHD distribution, clustering methods can be used for estimating the targets states. K-means and the Expectation-Maximization (EM) algorithms are the main approaches for state estimation for the PHD filter [15]. The total number of targets corresponds to the total particle mass, so target states are computed by clustering particles and using the centroids of each cluster. Furthermore, the authors in [15] also incorporated track continuity in the particle PHD filter by using validation techniques in the state estimation.

K-means is an iterative algorithm which separates the data points into K partitions and estimates the centres of each partition [6]. Each data point is associated to only one of the clusters, and the algorithm iterates until some utility function is minimized. On the other hand, the EM algorithm performs probabilistic classification using a finite mixture model. In this case, the probability of each data point belonging to a particular class is evaluated using a normalized allocation vector to all latent classes.

Since the PHD filter assumes low observation noise, parametric estimation using EM can be difficult. All data points would potentially be tightly clustered around their centres, introducing numerical instability in the calculation of the variances [107]. Furthermore, having only access to a resampled particle approximation, could also produce a mismatch between model complexity and the amount of available data.

Maximum likelihood approaches for parametric estimation suffer from local minima and over-fitting, as well dependency on the starting point. Bayesian approaches can be used instead in order to overcome the problems of deterministic estimation using limited data [34]. Figure 3.6 shows the EM algorithm and Bayesian estimation of a bivariate Gaussian mixture

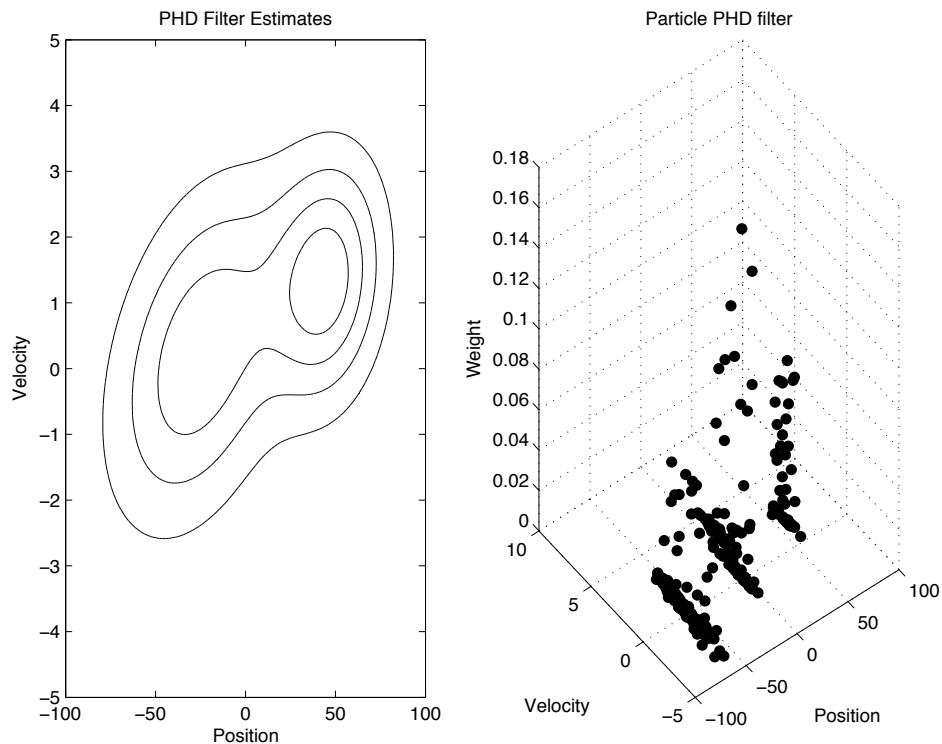


Figure 3.5: Positions and velocities of 3 target states are represented by the particle PHD filter. Monte Carlo samples are tightly clustered around target states, making it difficult to compute the model parameters using the EM algorithm.

model using the Gibbs sampler. Both methods are implemented with the same number of iterations and the estimated parameter is plotted at each step. It is easy to see the dependence of the EM algorithm on the starting point, while the Gibbs sampler is able to explore the parameter space.

3.6.1 Bayesian state estimation

In parametric state estimation for the particle PHD filter, we represent the target states as mixture model with parameters $\Theta_{\mathbf{k}} = (\theta_k^1, \dots, \theta_k^{N_k})$ and mixing proportions $\pi_{\mathbf{k}} = (\pi_k^1, \dots, \pi_k^{N_k})$, and each particle is sampled from the mixture distribution. The mixture density is a weighted linear combination

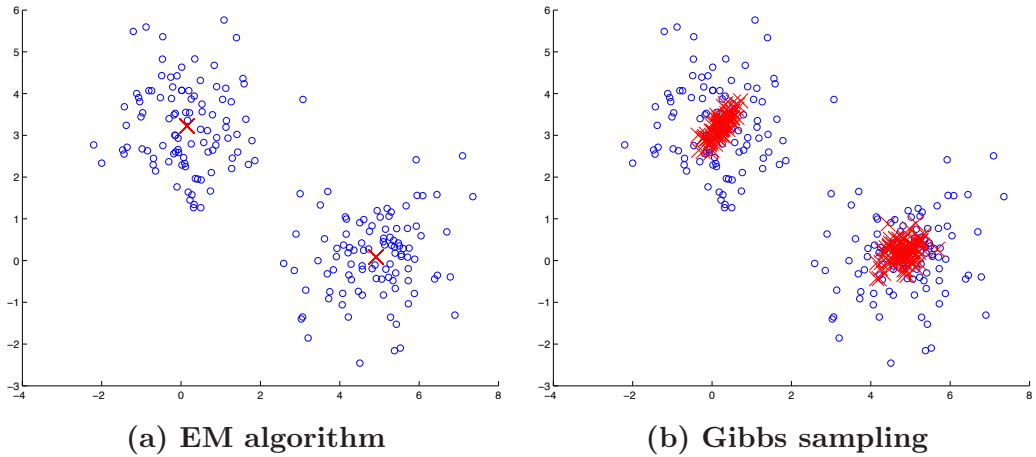


Figure 3.6: Bivariate Gaussian mixture model estimated with the EM algorithm and the Gibbs sampler. After 100 iterations, the Gibbs sampler has explored different configurations of the parameters (Gaussian mean parameter being represented by red crosses), while the EM algorithm estimates are more dependent on the initial setup.

of all the components of the mixture, with known parametric form. Estimation consists of estimating the values of the weights for all components, as well as the specific parameters for each one of the components. The number of components is usually estimated from data, using information theoretic methods such as the Akaike Information Criteria or the Bayesian Information Criteria [92]. When performing state estimation for the PHD filter we already have an estimate of the number of targets, which can instead be used as the expected number of clusters.

$$p(x_i|\Theta_k) = \sum_{j=1}^{N_k} \pi_j p(x_i|\theta_k^j) \quad (3.66)$$

The Bayesian approach for finite mixture modeling uses priors on the parameters, usually called hyper-parameters and the likelihood of the available data with respect to the mixture density. Similarly to the EM algorithm, allocation of the data instances to the mixture components is treated as a

missing variable, but in this case a density over the unknown allocations is used to perform inference. Estimation can be performed using Markov Chain Monte Carlo methods such as the Gibbs sampler [92], where the available data corresponds to the particle approximation to the PHD. After the update step of the PHD filter at time k , the available data is a sample from the real PHD distribution which is taken as the input for the parametric Bayesian state estimator.

In the PHD filter case, each component is usually modelled as a multivariate Gaussian distribution $\theta_k^j = \mathcal{N}(\hat{x}_k^j, \Sigma_k^j)$. State estimation using the Gibbs sampler was also considered in [63], where the authors compared estimation using the EM algorithm and Bayesian estimation with the Gibbs sampler. However, the authors proposed a complete mixture modeling estimation procedure and results were validated in a simple toy example.

We describe an algorithm for Bayesian state estimation using the Gibbs sampler. State estimation from mixtures of multivariate normals with known variances. This assumption holds when the target observation noise is used as a prior for the Gibbs sampler, so estimation is constrained to the means and mixing proportions. Instead of using a convergence criterion, both algorithms are described using a fixed number of iterations. Algorithm 5 describes the Gibbs sampler for PHD state estimation with known variances. In this case, we use independent hyper-parameters for the means $\hat{x}^j \sim \mathcal{N}(\mu^o, \Sigma^o)$ and the mixing proportions from the Dirichlet distribution $(\pi^1 \dots \pi^{N_k}) \sim \mathcal{D}(\gamma^1, \dots, \gamma^{N_k})$.

3.7 Limitations of the PHD filter

As noted by Mahler, the motivation for using a first-order moment approximation is the same as that for using a constant gain Kalman filter. Because of the computational cost of the Kalman gain matrix, which involves inversions of potentially singular covariance matrices, the full posterior distribution can be replaced with a scalar value for all time steps. In this case, the filter only

Algorithm 5 Gibbs sampler mean state estimation for PHD filter

Require: Given a set of L_k particles $\{x_k^1, \dots, x_k^{L_k}\}$, target independent observation noise Σ_k , and the estimated number of targets N_k .

- 1: Initialize Θ_k^0
 - 2: Perform N_i Gibbs sampler iterations
 - 3: **for** $i = 1, \dots, N_i$ **do**
 - 4: Generate component allocations $S = (s^1, \dots, s^{L_k})$
 - 5: **for** $\theta^n = \{x_k^n, \pi^n\} \in \Theta_k$ **do**
 - 6: $s^j = n \propto \pi^n \mathcal{N}(x_k^j - x_k^n)$
 - 7: **end for**
 - 8: Compute sufficient statistics
 - 9: **for** $j = 1, \dots, N_k$ **do**
 - 10: $\nu^j = \sum_{l=1}^{L_k} \mathbf{1}_{s^j=n}$
 - 11: $\eta^j = \sum_{l=1}^{L_k} \mathbf{1}_{s^j=n} x^l$
 - 12: **end for**
 - 13: Sample mixture proportions $\pi_k \sim \mathcal{D}(\gamma^1 + \nu^1, \dots, \gamma^{N_k} + \nu^{N_k})$
 - 14: Update means
 - 15: $\Sigma^j = ((\Sigma^0)^{-1} + \nu^j (\Sigma_k)^{-1})^{-1}$
 - 16: $\mu^j = \Sigma_k ((\Sigma^0)^{-1} \mu_o + \sigma_k^{-1} \nu(j) \eta^j)$
 - 17: $\hat{x}^j \sim \mathcal{N}(\mu^j, \Sigma^j)$
 - 18: **end for**
-

propagates the mean, which corresponds to the first-order moment of the whole distribution [5].

This assumption is only valid in cases where the sensor noise is small compared with the signal power. In the case of the PHD filter, apart from the sensor noise uncertainty, there is also uncertainty on the number of targets as well as the corresponding interactions. Furthermore, in order to make the process recursive, the functional form of the posterior distribution has to be preserved. The Poisson approximation to the complete distribution corresponds to a deterministic approximation, whose error is also propagated

in time.

In order to improve these limitations, a generalization of the PHD recursion has been proposed which also propagates the posterior cardinality distribution [69]. The Cardinalized PHD (CPHD) is still first-order in the states of individual targets, but is higher-order on the number of targets. Analytic implementations have been proposed in [120], but their scope is much more limited when compared with the PHD filter. Furthermore, it has also been noted that the complexity is cubic with respect to the observations and the posterior cardinality distribution is also sensitive to missed detections.

3.8 Summary and Contributions

The PHD filter can be discussed in terms of random sets and point processes theories. In both cases, a recursive solution for the multi-target tracking problem can be achieved. There are subtle distinctions between the two approaches, which might not be evident to practitioners, but which can make a difference when modeling particular tracking scenarios. In visual tracking problems, a richer representation of the targets might be required, so the point process formalism can be used to model variations in the geometry of an object [3]. Since the complete evaluation of all possible models with unknown number of components is not tractable, it becomes reasonable to approximate the complete posterior distribution by the first-order moment of a Poisson process.

The PHD filter achieves a parsimonious representation of the multi-target problem when data association is too expensive to evaluate. When considering a point process formalism, a richer object representation is also achieved with no extra theoretical requirements. At the same time, the PHD filter can be derived from the spatial discretization of a Poisson point process, which provides an intuitive derivation of the intensity recursion. Birth and deaths of objects can be modeled as transformations of a stochastic process that remains closed under Poisson approximations to these events. Moreover, the

number and locations of the birth and death events can be represented as non-homogeneous Poisson processes, whose spatial distribution can be fitted to the particular scenario.

A contribution of this Chapter is a modeling framework for object processes. In order to reconstruct an unknown number of signals observed in clutter, the PHD recursion assumes that every signal has a SNR that compensates the loss of information incurred in the first-order moment posterior density approximation. Nevertheless, the distributional approximation also implies the density to be a completely random process, which neglects any particular orientation or minimum distance between objects. While the usual setup for stationary marked point processes considers a density with respect to the Poisson measure [103], where mark-sum intensities and second-order moments on the locations and marks describes correlations on the posterior point process. Section 3.1 presents the theoretical foundations for the distribution and density of such process, being the first-order moment posterior density an approximation to a non-independent scattering process.

As a way to overcome the sensor noise assumption without leaving the first-order formulation, a novel Gibbs sampler algorithm for state estimation is proposed in Section 3.6. Since MCMC methods usually converge in the limit of high number of iterations, the new algorithm consists of a constrained version of the Gibbs sampler intended for real time estimation. The special case of Bayesian estimation of mixtures of multivariate Gaussian distributions was illustrated. The proposed algorithm uses the observation noise model for the covariance structure, so only the means and mixing coefficients are left to the stochastic algorithm. Furthermore, the use of a proper prior is also recommended in order to find estimates in computational times closer to real time.

4

Smoothing algorithms for the PHD filter

The PHD filter algorithm provides an approximation to the expectation or first-order moment of the intensity measure of a Poisson point process. The method has the property of being able to explicitly model the birth and deaths of targets, as well as clutter and mis-detections, which can also be subject to spawning or merging. This model-based approach can be appealing in multiple tracking systems where the data association step is non-trivial or cannot be optimally solved.

An alternative solution for improving the PHD filter's instantaneous estimates is to perform smoothing or retrodiction. Filtered estimates of the individual target states and the posterior cardinality distribution can be considerably improved by considering a higher data frame than the history of

observations. More specifically, PHD filtering can be extended to smoothing and is expected to correct the abrupt changes on the estimated number of targets and their states, that originate from errors propagated by the filtered distributions.

Let $X_k = \{x_1, \dots, x_{n_k}\}$ be a set target states and $Z_{1:T}$ a collection of set-valued measurements collected up to time $T \geq k$. The smoothed PHD can be written as follows:

$$D_{k|T}(x) \triangleq \int p(\{x\} \cup X_k | Z_{1:T}) \delta X_k \quad (4.1)$$

Accordingly, the smoothed number of targets can then be written as:

$$N_{k|T} \triangleq \int D_{k|T}(x) dx \quad (4.2)$$

As with the standard linear and non-linear smoothing equations in Chapter (2), the PHD smoothing problem might be approached as fixed-interval smoothing, fixed-lag smoothing or fixed-point smoothing. The algorithms presented in this chapter are not dependent on the data interval size, so they can be implemented under each one of these schemes. Notice that, since the PHD is only available for non-ordered sets, full PHD smoothing distributions $p(X_{1:k} | Z_{1:T})$ are not available, so only the *marginal PHD smoothing* $D_{k|T}(x)$ in Equation 4.1 can be approximated. Sections 4.1 and 4.2 describes two possible approximations.

4.1 Forward-Backward PHD smoother

Following a similar approach to the particle forward-backward smoother in Chapter 2, Nandakumaran et al. developed a Forward-Backward PHD (FB-PHD) smoother [79] based on the physical-space approach presented previously in Section 3.3. Here, we present an alternative formulation based on factorial moment measures. Using a factorization similar to the single target

marginal smoothing case, Bayesian filtering equations can be extended to compute the marginal smoothing distribution:

$$p(X_k|Z_{1:T}) = \int p(X_k, X_{k+1}|Z_{1:T})\delta X_{k+1} \quad (4.3)$$

$$= \int p(X_{k+1}|Z_{1:T})p(X_k|X_{k+1}, Z_{1:T})\delta X_{k+1} \quad (4.4)$$

$$= p(X_k|Z_{1:k}) \int \frac{p(X_{k+1}|Z_{1:T})p(X_{k+1}|X_k)}{p(X_{k+1}|Z_{1:k})}\delta X_{k+1} \quad (4.5)$$

The formulation seems to be an extension to the vector-valued smoothing case, but in this case all related quantities are representing symmetric densities of non-probability measures. In FISST notation, equations (4.5) are used to represent the backward recursion required to obtain the marginal PHD smoothing. Once the filtered and smoothed intensities are obtained in the forward pass, the smoothed intensities are computed in the backward pass.

Equation 4.5 is used to represent the backward recursion required to obtain the marginal PHD smoothing. Once obtained the filtered and smoothed intensities in the forward pass, the smoothed intensities are computed in the backward pass. Replacing the first-order moment expression in Equation 3.34 into 4.5 leaves the first-order moment of the smoothing PHD $D_{k|T}$ as being:

$$\begin{aligned}
D_{k|T}(x) &= \int p(\{x\} \cup X_k | Z_{1:T}) \delta X_k \\
&= \int p(\{x\} \cup X_k | Z_{1:k}) \\
&\quad \left[\int \frac{p(X_{k+1} | Z_{1:T}) p(X_{k+1} | \{x\} \cup X_k)}{p(X_{k+1} | Z_{1:k})} \delta X_{k+1} \right] \delta X_k \\
&= \int p(\{x\} \cup X_k | Z_{1:k}) \delta X_k \left[\int \frac{p(X_{k+1} | Z_{1:T}) p(X_{k+1} | \{x\} \cup X_k)}{p(X_{k+1} | Z_{1:k})} \delta X_{k+1} \right] \\
&= \int \Phi(x) D_{k|x}(x) dx
\end{aligned} \tag{4.6}$$

We can evaluate the backward kernel $\Phi(x)$ as:

$$\Phi(x) = \begin{cases} 1 - \pi_s(x) & \text{if } X_{k+1} = \emptyset, \\ \int \frac{p(\{x\} \cup X_{k+1} | Z_{1:T}) p(\{x\} \cup X_{k+1} | X_k)}{p(\{x\} \cup X_{k+1} | Z_{1:k})} \delta X_{k+1} & \text{if } X_{k+1} = \{x_1, \dots, x_{n_{k+1}}\} \end{cases} \tag{4.7}$$

Consequently, taking the first-order moment of $p(X_{k+1} | Z_{1:T})$ can be also achieved by the PHD approximation $D_{k+1|T}(x) = \int p(\{x\} \cup X_{k+1} | Z_{1:T}) \delta X_{k+1}$.

The denominator of Equation 4.6 can be evaluated in a similar fashion to the prediction step of the PHD filter in Equation 3.43, so that the first-order moment $D_{k+1|k}(x)$ becomes:

$$D_{k+1|k}(x) = \int p(\{x\} \cup X_{k+1} | Z_{1:k}) \delta X_{k+1} \tag{4.8}$$

$$= \int (\pi_s(x) p(x|x') + \gamma_{k+1|k}(x|x')) D_{k|k}(x') dx' + b_{k+1|k}(x) \tag{4.9}$$

The term $p(\{x\} \cup X_{k+1} | X_k)$ in the numerator can be evaluated as a density for persisting targets at time step $k + 1$ from step k :

$$p(\{x\} \cup X_{k+1} | X_k) = \sum_{\sigma_k} \prod_{i=1}^{n_{k+1}} \pi_s(x) p(x'_i | x_{\sigma_k}) \tag{4.10}$$

where σ_k represents all permutations on X_k .

The smoothed PHD can now be written as:

$$D_{k|T}(x) = D_{k|k}(x) (1 - \pi_s(x)) + D_{k|k}(x) \int \frac{D_{k+1|T}(x') \pi_s(x) p(x'|x)}{D_{k+1|k}(x')} dx' \quad (4.11)$$

$$= D_{k|k}(x) \left[(1 - \pi_s(x)) + \int \frac{D_{k+1|T}(x') \pi_s(x) p(x'|x)}{D_{k+1|k}(x')} dx' \right] \quad (4.12)$$

A particle approximation to the smoothing multi-target density can be written as:

$$\int_B D_{k+1|T}(x) dx = E \left[\sum_{x_{k+1} \in B} \mathbf{1}_B(x_{k+1}) \right] \quad (4.13)$$

$$\approx \sum_{i=1}^{L_{k+1}} \mathbf{1}_B(x_{k+1}^i) w_{k+1|T}^i \quad (4.14)$$

By replacing $D_{k|k}(x)$ by its particle approximation in Equation 3.62, we can now write the backward recursion for the smoothing particle approximation as shown in Algorithm 6. For the SMC implementation of the PHD filter, it has been shown that the SMC approximation of the PHD under any bounded test function (i.e. indicator functions), the empirical measure converges almost surely when both the true and the particle approximation have finite masses [55]. In the case of the FB-PHD smoothing, there seems no reason why the same results should not hold, however the theoretical analysis becomes more complicated than the filter. Recently, Duouc et al. proposed a framework for studying SMC approximations of smoothing distributions, but it is not clear how it can be applied to the PHD smoothing case.

Algorithm 6 Forward-Backward PHD smoother

Require: $k < T \wedge \{w_k^i, x_k^i\}_{i=1}^{L_k} \wedge \{w_{k+1}^i, x_{k+1}^i\}_{i=1}^{L_{k+1}}$

- 1: **for all** $i \in \{1, \dots, L_k\}$ **do**
 - 2: $\mu_{k+1|k}^j = b_{k+1|k}(x_{k+1}^j) + \sum_{l=1}^{J_k} w_k^l [\pi_s(x_k^l) p(x_{k+1}^j | x_k^l) + \gamma_{k+1|k}(x_{k+1}^j | x_k^l)]$
 - 3: $w_{k|T}^i = w_{k|k}^i \left[\frac{\sum_{j=1}^{L_{k+1}} w_{k|T}^j \pi_s(x_k^i) p(x_{k+1}^j | x_k^i)}{\mu_{k+1|k}^j} \right]$
 - 4: **end for**
 - 5: Compute smoothed estimated number of targets $\hat{N}_{k|T} = \sum_{i=1}^{L_k} w_{k|T}^i$
 - 6: Normalize $\{w_{k|T}^i\}_{1 \leq i \leq L_k}$ to get $\{\frac{\hat{N}_{k|T}}{L_k}\}_{1 \leq i \leq L_k}$.
-

4.2 Two-Filter PHD smoother

Another approach for PHD smoothing can be achieved by means of the two-filter formula shown in Section 2.4.3. In this case, the PHD filter has to be combined with the output of a backward information filter, which propagates the posterior distribution of the random counting measure $N_{K|T}$ from Equation 4.2 to be represented by the following factorization:

$$p(X_k | Z_{1:T}) = p(X_k | Z_{1:k-1}, Z_{k:T}) \quad (4.15)$$

$$= \frac{p(X_k | Z_{1:k-1}) p(Z_{k:T} | X_k)}{p(Z_{k:T} | Z_{1:k-1})} \quad (4.16)$$

$$\propto p(X_k | Z_{1:k-1}) p(Z_{k:T} | X_k) \quad (4.17)$$

where the *backward information* $p(Z_{k:T} | X_k)$ filter can be written as:

$$p(Z_{k:T} | X_k) = \int p(Z_k, Z_{k+1:T}, X_{k+1} | X_k) \delta X_{k+1} \quad (4.18)$$

$$= \int p(Z_k x | X_k) p(X_{k+1} | X_k) p(Z_{k+1:T} | X_{k+1}) \delta X_{k+1} \quad (4.19)$$

Similar to the vector-valued smoothing case, the backward information PHD filter can be expressed as a conditional intensity of future observed point

processes given the current state. It is worth noticing that this entity is not defined as a proper density in X_k , so the integral might not (in general) be finite, leaving SMC methods being not applicable. Nevertheless, if we consider a sequence of finite Poisson processes $\{X_1, \dots, X_k\}$, the backward PHD filter can be also constructed using an information-theoretic best-fit Poisson approximation for linearly superimposed Poisson processes. In this case, the requirement for priors $p(X_k)$ and $p(X_{k+1})$ is solved by considering the filtering multi-target densities from the forward pass:

$$P(X_k) = \int \dots \int p(X_1) \prod_{i=1}^k p(X_i | X_{i-1}) \delta X_{1:k} \quad (4.20)$$

$$\propto \int p(X_k | X_{k-1}) p(X_{k-1} | Z_{1:k-1}) \delta X_{k-1} \quad (4.21)$$

Using the principles for Bayes equations of FISST [115], the backward information PHD filter can be expressed as:

$$p(Z_{k+1:T} | X_k) = \frac{p(X_k | Z_{k+1:T}) p(Z_{k+1:T})}{p(X_k)} \quad (4.22)$$

$$\propto \frac{p(X_k | Z_{k+1:T})}{p(X_k)} \quad (4.23)$$

which when replaced in Equation 4.18, gives an expression for the predicted smoothing density:

$$p(X_k | Z_{k+1:T}) = \int \frac{p(X_k | X_{k+1}) p(X_k) p(X_{k+1} | Z_{k+1:T})}{p(X_{k+1})} \delta X_{k+1} \quad (4.24)$$

$$= \int p(X_{k+1} | X_k) p(X_{k+1} | Z_{k+1:T}) \delta X_{k+1} \quad (4.25)$$

and the backward updated PHD:

$$\begin{aligned} p(X_k | Z_{k:T}) &= \frac{p(Z_k | X_k) p(X_k | Z_{k+1:T})}{p(Z_k | Z_{k+1:T})} \\ &= \frac{p(Z_k | X_k) p(X_k | Z_{k+1:T})}{\int p(Z_k | X_k) p(X_k | Z_{k+1:T}) \delta X_k} \end{aligned} \quad (4.26)$$

We now need to evaluate the multi-target densities in 4.24 and 4.26, so a backward recursion is constructed as follows.

Backward PHD prediction

The backward predicted number of targets can be estimated as targets surviving $N_{k|k+1}^s$ and not surviving $N_{k|k+1}^{\setminus s}$ from $k+1$ to k :

$$N_{k|k+1} = N_{k|k+1}^{\setminus s} + N_{k|k+1}^s \quad (4.27)$$

Assume that we already have a first-order moment:

$$D_{k+1|T}(x) \triangleq \int P(\{x\} \cup X_{k+1} | Z_{k+1:T}) \delta X_{k+1} \quad (4.28)$$

Consequently, the PHD $D_{k|k+1}(x)$ can be written as:

$$D_{k|k+1}(x) = \int p(\{x\} \cup X_k | Z_{k+1:T}) \delta X_k \quad (4.29)$$

$$= \int \left[\int p(X_{k+1} | \{x\} \cup X_k) p(X_{k+1} | Z_{k+1:T}) \delta X_{k+1} \right] \delta X_k \quad (4.30)$$

$$= \int \Psi(x) D_{k+1|T}(x) dx \quad (4.31)$$

Now we can evaluate $\Psi(x)$ as follows:

$$\Psi(x) = \begin{cases} (1 - \pi_s(x)) & \text{if } X_k = \emptyset, \\ p(X_{k+1} | \{x\} \cup X_k) & \text{if } X_k = \{x_1, \dots, x_{n_{k+1}}\} \end{cases}$$

The conditional density $p(X_{k+1} | X_k)$ is taken along all permutations σ_{k+1} of X_{k+1} , and $\gamma_{k+1|k}(x_{\sigma_{k+1}} | x)$ represents targets being spawned from times $k+1$ to k .

$$p(X_{k+1} | X_k) = \sum_{\sigma_{k+1}} \prod_{i=1}^{n_{k+1}} (\pi_s(x_i) p(x_{\sigma_k} | x_i) + \gamma_{k+1|k}(x_{\sigma_k} | x_i)) \quad (4.32)$$

The backward predicted number of targets $N_{k|k+1}$ can now be written as a superposition of the above two point processes, so the backward PHD prediction can be written as:

$$D_{k|k+1}(x) = 1 - \pi_s(x) + \int \left[\pi_s(x)p(x'|x) + \gamma_{k+1|k}(x') \right] D_{k+1|T}(x') dx' \quad (4.33)$$

We can use now this result for the backward updated PHD.

Backward PHD update

Similar to the filtering case, we approximate the posterior multi-target density $p(X_k|Z_{k:T})$ by a Poisson process with mean measure $N_{k|T}$. The first-order moment of the backward multi-target density in 4.26 can be written then as:

$$D_{k|T}(x) = \int \frac{p(Z_k|\{x\} \cup X_k)p(\{x\} \cup X_k|Z_{k+1:T})}{p(Z_k|Z_{k+1:T})} \delta X_k \quad (4.34)$$

$$\propto \int p(Z_k|\{x\} \cup X_k)p(\{x\} \cup X_k|Z_{k+1:T}) \delta X_k \quad (4.35)$$

$$= \int p(Z_k|\{x\} \cup X_k) D_{k|k+1}(x) \delta X_k \quad (4.36)$$

The pseudo-likelihood $p(Z_k|X_k)$ can be evaluated in a similar fashion to the filter, so replacing Equation 3.47 into 4.36 leaves:

$$D_{k|T}(x) = \left[1 - \pi_d(x) + \sum_{z \in Z_k} \frac{\pi_d(x)p(z|x)}{\lambda_c c(z) + \int \pi_d(x)p(z|x)dx} \right] D_{k|k+1}(x) \quad (4.37)$$

4.2.1 SMC implementation for the Two-Filter PHD smoothing

The backward predicted density can be approximated as:

$$\int_B D_{k|k+1}(x)dx = E\left[\sum_{x_k \in B} \mathbf{1}_B(x_k)\right] \quad (4.38)$$

$$\approx \sum_{i=1}^{L_k} \mathbf{1}_B(x_k^i) w_{k|k+1}^i \quad (4.39)$$

And the particle approximation to the multi-target updated density at time $k = T$ can be written as:

$$\int_B D_{k|T}(x)dx = E\left[\sum_{x_k \in B} \mathbf{1}_B(x_k)\right] \quad (4.40)$$

$$\approx \sum_{i=1}^{L_k} \mathbf{1}_B(x_k^i) w_{k|T}^i \quad (4.41)$$

with the smoothing weights $w_{k|T}^i = w_{k|k}^i$.

The recursion on the backward predicted PHD in Equation 4.33 can then be approached as:

$$D_{k|k+1}(x) = 1 - \pi_s(x) + \sum_{j=1}^{L_{k+1}} [\pi_s(x)p(x|x_{k+1}^j) + \gamma_{k|k+1}(x|x_{k+1}^j)] w_{k+1|k+1}^j \quad (4.42)$$

and the backward predicted PHD smoothing PHD in Equation 4.37:

$$D_{k|k+1}(x) = 1 - \pi_s(x) + \int [\pi_s(x)p(x'|x) + \gamma_{k+1|k}(x')] w_{k|k+1}^i \quad (4.43)$$

The SMC approximation for the backward predicted smoother is plugged into 4.37 and the recursion can then be written as Algorithm 7.

4.3 Summary and contributions

In this chapter, a novel derivation for the FB-PHD smoother using the Poisson point process approach is presented in Section 4.1. Also, an original

Algorithm 7 Two-Filter PHD smoother

Require: $k < T \wedge \{w_k^i, x_k^i\}_{i=1}^{L_k} \wedge \{w_{k+1}^i, x_{k+1}^i\}_{i=1}^{L_{k+1}}$

1: **for all** $i \in \{1, \dots, L_k\}$ **do**

2: $\psi_k^l = \sum_{h=1}^{J_k} \pi_d(x_k^l) p(z|x_k^l)$

3: $L_z(x_k^l) = 1 - \pi_d(x_k^l) + \sum_{z \in Z_k} \frac{\pi_d(x_k) p(z|x_k^l)}{\lambda_c C_k(z) + \psi_k^l}$

4: $\alpha_{k+1|k}^j = \sum_{l=1}^{J_k} w_{k|k}^l p(x_{k+1}^j | x_k^l)$

5: $w_{k|T}^i = L_z(x_k^i) \left[\sum_{j=1}^{L_{k+1}} \frac{w_{k|T}^j \pi_s(x_k^i) p(x_{k+1}^j | x_k^i)}{\alpha_{k+1|k}^j} + 1 - \pi_s(x_k^i) \right]$

6: **end for**

7: Compute smoothed estimated number of targets $\hat{N}_{k|T} = \sum_{i=1}^{L_k} w_{k|T}^i$

8: Normalize $\{w_{k|T}^i\}_{1 \leq i \leq L_k}$ to get $\{\frac{\hat{N}_{k|T}}{L_k}\}_{1 \leq i \leq L_k}$.

PHD smoothing implementation based on the two-filter formula, the TF-PHD smoother is presented in Section 4.2. Both particle smoothing techniques re-weight samples obtained in the forward pass, so no backward filtering step is required. The computational complexity of the backward step is quadratic in the number of particles for both smoothers, but the TF-PHD smoother requires a likelihood evaluation which also depends on the number of observations.

Particle methods have an inherent ‘‘curse of dimensionality’’ when representing high dimensional state spaces, therefore an optimal instrumental distribution is essential when a finite number of samples is used to approximate a probability density function [19]. In order to avoid a poor representation of the target pdf by means of a single point mass, re-sampling is also performed in the sequential Monte Carlo setup. In the case of the PHD filter, particles are used to approximate a non-probability density which is usually multi-modal. Consequently, the dimensionality problem becomes even more severe in the PHD filter than it is in traditional particle approximations.

The sequential Monte Carlo approximation of a multi-modal and possibly high dimensional function imposes a fundamental limit in the achievable performance of the particle PHD smoothers. The FB-PHD smoother relies

heavily on the choice of the importance density, which apart from modeling the individual target translations, also takes into account independent birth of new targets and spontaneous deaths. Whenever there is a discrepancy between the importance density and the posterior distribution, the particle representation would fail to approximate the intensity function. Furthermore, when support is lacking in areas where a genuine target is located, smoothing can not alleviate this problem by merely re-weighting the existing particles. Furthermore, resampling could also have an unpleasant effect in the FB-PHD smoother. On the other hand, the TF-PHD smoother makes use of the likelihood function, so it could potentially ameliorate the dependency of the backward pass on the individual target dynamic model. Moreover, the TF-PHD could potentially implement a carefully designed artificial backward importance density, which could sample in areas not being explored in the forward pass.

Particle PHD smoothers have larger complexity than the PHD filter and at the same time inherit the same limitations from the forward pass. However, PHD smoothers can be implemented with a fixed time-lag, so a delayed estimator can be obtained with only modest extra computational effort. The smoothed estimators are expected to reduce the variance of the number of targets in the filtering step. Accordingly, a more stable estimate of cardinality should also lead to improved estimates of the individual target locations.

5

Performance Evaluation

Distance metrics play a crucial role in any performance assessment for state estimation in discrete dynamical systems. The most widely used criterion for performance evaluation in single target tracking is the Euclidean distance between the estimated state and the ground truth. It is important to note that the notion of convergence which is widely used for parameter estimation, has no direct meaning in state estimation for tracking systems. Because of the dynamic nature of the system, the consistency and accuracy of the estimator are used instead [5]. A common practice is to study the consistency of the estimator by means of the first and second moment characterization of the innovations:

$$\mathbb{E}[x_k - \hat{x}_{k|k}] \triangleq 0 \tag{5.1}$$

$$\mathbb{E} [(x_k - \hat{x}_{k|k})(x_k - \hat{x}_{k|k})'] \triangleq \Sigma_{k|k} \tag{5.2}$$

The best estimator in terms of the Minimum Mean Square Error (MMSE) criteria requires the estimation error in Equation 5.2 to have zero-mean and to be uncorrelated to the observations. As seen in Chapter 2 for the case of the Kalman filter, the optimal estimator of a random variable relies in the orthogonality condition between the estimation error and the observed data (see Section 2.2), and the first and second order moments being the sufficient statistics for the optimal estimator of linear Gaussian models [1].

In the case of multi-target estimation, performance evaluation is not as straightforward as in the single target case, where the consistency is analyzed by means of the residuals of the estimator (see Equation 5.2). In this case, the number of targets has to be estimated alongside with the individual states, and erroneous associations between targets and observations or targets being closely spaced might lead to different estimation results [24]. Evaluating the performance under different configurations between estimated states and ground truth is challenging because of the ambiguity of evaluating all possible associations. The ambiguity of the estimation performance is shown in Figure 5.1 for a hypothetical multi-target scenario with two different estimators.

Because of the combinatorial nature of the data association problem, the consistency of the estimator cannot be correctly captured by the MMSE criteria. For that reason, different distance metrics have to be used in order to evaluate the performance of multi-target filters and smoothers. Section 5.1 introduces two different options proposed in the literature.

5.1 Distance metrics

In order to measure the performance of multi-target tracking algorithms, we first have to establish a suitable distance metric. A distance metric is a mathematical concept for defining dissimilarity between two different objects. Let $d: X \times Y \mapsto \mathcal{R}$ be a positive valued function of two objects x, y defined in two Euclidean n -dimensional spaces $X \in \mathcal{R}^n$ and $Y \in \mathcal{R}^m$ respectively. The space $(X \times Y)$ is called a *metric space* if the following conditions are

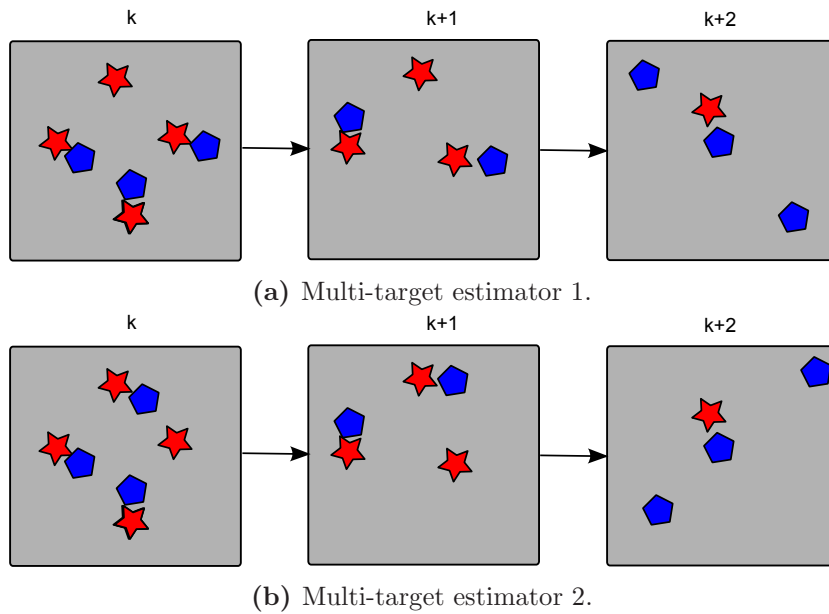


Figure 5.1: *Multi-target estimation ambiguity. 3 frames with different number of targets (ground truth plotted in red) and estimated states (plotted in blue). At frame k the first multi-target estimator in Figure 5.1a has detected 3 of 4 targets, and for the same frame the second estimation in Figure 5.1b detected a different multi-target configuration. The same situation occurs at each frame, so it is not clear which one of the multi-target estimators is closer to the ground truth.*

satisfied:

1. Non-negativity, $d(x, y) \geq 0$
2. Identity, $d(x, y) = 0 \implies x = y$
3. Symmetry, $d(x, y) = d(y, x)$
4. Triangle inequality, $d(x, y) \leq d(x, z) + d(z, y)$

The most commonly used metric for measuring distance between two points is the Euclidean distance, denoted as:

$$d_p(x, y) = \sqrt[p]{\sum_{i=1}^n d_i(x_i, y_i)^p} \quad (5.3)$$

A more general formulation is the L_p distance, which accounts for function spaces and is defined as:

$$d_p(f, g) = \sqrt[p]{\int |f(x) - g(x)|^p dx} \quad (5.4)$$

$$d_\infty(f, g) = \sup_x |f(x) - g(x)| \quad (5.5)$$

Now we are in position to define different multi-object distance metrics suitable for performance evaluation.

5.1.1 Hausdorff distance

Instead of considering vectors, we want to consider now differences between sets. A widely used measure for the difference between two sets is the Hausdorff distance. Let X and Y be two finite non-empty subsets of \mathbb{R}^n , where the n -dimensional vectors x and y are defined. If $d(x, y)$ is some underlying norm on X and Y (e.g. the Euclidean norm $d(x, y) = d_s(x, y)$), the Hausdorff distance is defined as:

$$d^H(X, Y) = \max \left(\max_{x \in X} \min_{y \in Y} d(x, y), \max_{y \in Y} \min_{x \in X} d(x, y) \right) \quad (5.6)$$

The quantity $\max_{x \in X} \min_{y \in Y} d(x, y)$ is computed by identifying the farthest point $x \in X$ to any point in Y , and calculating the distance from x to its nearest neighbor in Y .

The Hausdorff distance has been widely accepted in the image processing and stochastic geometry community for comparing binary images. This is

mainly because it quantifies the extent to which a pixel in a reference image lies besides a pixel in another image [49]. Furthermore, the Hausdorff distance is a valid metric over all closed and bounded sets, so it is used to generate the so called Matheron or hit-and-miss topology which is used in stochastic geometry for defining the Borel σ -algebra for random closed sets in locally compact spaces [75].

Despite being well suited for multi-object error estimation, it has been noted that the Hausdorff distance has an adverse behavior when comparing sets with different cardinalities [46]. Furthermore, due to being defined over non-empty sets, it cannot correctly handle the case of no targets in the field of view. Moreover, since the Hausdorff metric compares maximum distances between the two different sets, it is also highly sensitive to outliers.

5.1.2 Wasserstein distance

In order to override the problems of the Hausdorff metric for the multi-object evaluation purpose, Hoffman and Mahler [46] proposed a construction based on the Wasserstein distance. The Wasserstein distance has a long tradition in theoretical statistics for comparing probability distributions on metric spaces. Intuitively speaking, the Wasserstein distance estimates the cost of turning one probability distribution into another distribution with probably different total mass, by means of the minimal average distance between the two sets. The associated cost is calculated as an optimal-assignment problem for all $m \times n$ transportation matrices C , that satisfies $C_{i,j} \neq 0 \forall i, j$ and:

$$\sum_{j=1}^n C_{i,j} = \frac{1}{m} \text{ for } 1 \leq i \leq m \quad (5.7)$$

$$\sum_{i=1}^m C_{i,j} = \frac{1}{n} \text{ for } 1 \leq j \leq n \quad (5.8)$$

The Optimal Mass Transfer (OMAT) metric of order p for non-empty sets X and Y is defined as:

$$d_p(X, Y) = \min_C \left(\sum_{i=1}^m \sum_{j=1}^n C_{i,j} d(x_i, y_j)^p \right)^{\frac{1}{p}} \quad (5.9)$$

The OMAT metric mitigates the problem of comparing sets with different cardinalities, but still suffers from the non-empty sets comparison problem, as well as some geometry dependent behavior [119].

More recently, a new distance for multi-object error estimation was proposed in [95]. Intuitively speaking, the distance chooses the assignment between the points of two sets, that minimizes the truncated sum of the distances with a cut-off value c . The minimum sum of distances on the assigned points is regarded as a total localization error and the remaining points are interpreted as cardinality errors. The total error corresponds to the sum of the localization error and the cardinality error. The Optimal Sub-Pattern Assignment (OSPA) metric is then defined as follows:

$$\bar{d}_p^c(X, Y) = \left(\frac{1}{n} \left(\min_{\pi \in \Pi_n} \sum_{i=1}^m d_c(x_i, y_{\pi(i)})^p + c^p(n - m) \right) \right)^{\frac{1}{n}} \quad (5.10)$$

The magnitude order parameter p makes the OSPA metric become more sensitive to outliers as p increases, making the metric more focused on location errors. On the other hand, the cut-off parameter c penalizes cardinality errors instead of wrongly placed locations. Because it is a consistent metric for multi-target error estimation, we use the OSPA¹ metric defined in Equation 5.10 for performance evaluation.

¹The author would like to acknowledge Dr. Ba-Ngu Vo from the Department of Electrical & Electronic Engineering of the University of Melbourne for the code implementing the OSPA metric.

5.2 Performance evaluation of PHD filters and smoothers

In this section we study the performance of the PHD filter and smoothers under different conditions. Four different scenarios are simulated and the performance of the particle PHD filter and smoothers is compared using the root mean square (RMS) error for the cardinality estimates and the OSPA metric (see Equation 5.10) for the locations.

5.2.1 Low clutter scenario no births and deaths

The following example shows three targets on the real line in a surveillance region bounded by $[-100, 100]$. The targets follow linear trajectories with unknown and nearly constant velocity embedded in Gaussian noise with diagonal covariance matrix $\Sigma_x = \text{diag}([1, .1])$. The observation model for each target is also linear with Gaussian noise with unit variance. Birth rate in a unit volume is $\lambda_b = 1e^{-5}$, and each target has equal probability of detection $\pi_d = 1$ and survival $\pi_s = 1$. Clutter is homogeneous (uniform spatial distribution) with rate $\lambda_c = 1e^{-3}$ per unit volume, giving an average of 0.2 false alarms per scan. Figure 5.2 shows data generated from this multi-target model.

The particle PHD filter is implemented with 1000 particles per target. After each prediction, update and resampling step, state estimation is carried out with the EM algorithm and the Gibbs sampler, with the number of Gaussian components estimated by rounding the sum of the particle weights to the nearest integer.

In the presence of low clutter, cardinality estimates from the PHD filter are unstable and biased to the number of observations. The reason for this behavior is the denominator in the PHD filter update equation, which uniformly integrates clutter in the surveillance area (see Equation 3.49). This situation can be illustrated with $\lambda_c \approx 0$:

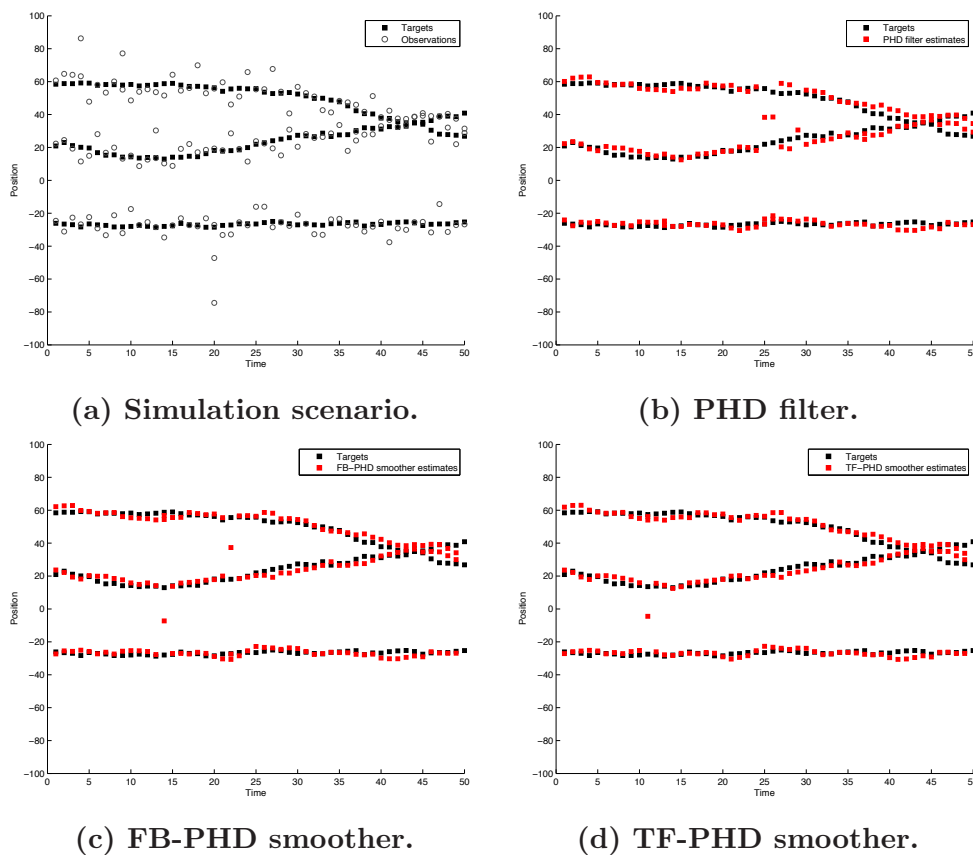


Figure 5.2: Three Targets following linear trajectories and no births or deaths of targets occurs. (a) Targets (ground truth plotted in black squares) are observed with low signal-to-noise ratio and there is a small number of false alarms per scan. (b) Due to the variability on the PHD filter cardinality estimates, state estimates also presents several discrepancies with the ground truth. Figures (c) and (d) show the improved estimates by using the FB-PHD and TF-PHD smoothers.

$$D_{k|k}^d(x) = \int \frac{\pi_d(x) p(z|x)}{\lambda_c c(z) + \int \pi_d(x) p(z|x) D_{k|k-1}(x) dx} D_{k|k-1}(x) dx \quad (5.11)$$

$$\approx \int \frac{\pi_d(x) p(z|x)}{\int \pi_d(x) p(z|x) D_{k|k-1}(x) dx} D_{k|k-1}(x) dx \quad (5.12)$$

which is exactly the situation when no clutter is present, shown in Equa-

tion 3.47.

However, as seen in Figures 5.3(a) and 5.3(b), both PHD smoothers can recover the actual number of targets in a more stable way.

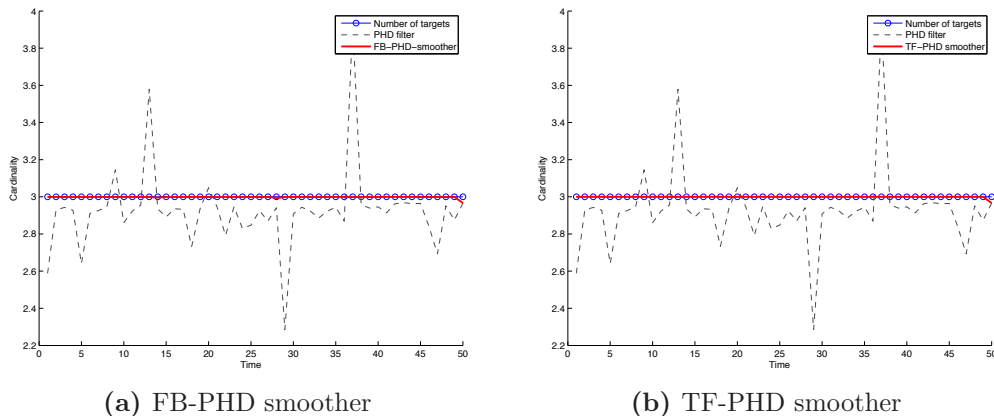


Figure 5.3: Cardinality estimates for the PHD filter (dashed line) and smoothers (blue line) are obtained by particle approximations to the multi-target posterior densities. Both PHD smoothers are able to recover the actual number of targets (ground truth plotted in red) in the backward pass.

A comparative illustration of the two particle smoothing schemes is shown in Figure 5.4 for frame 37 of the time series. The PHD filter incorrectly estimates that there are 5 targets, so state estimates using EM and the Gibbs sampler shown in Figures 5.4(a) and 5.4(b) are also incorrect.

Using the cardinality estimate from the particle implementations of the FB-PHD and TF-PHD smoothers, the EM algorithm and the Gibbs sampler are used to estimate target locations. However, since the TF-PHD uses the current observations in order to re-weight the particles and re-compute the number of targets, the smoother estimates can be more sensitive to outliers. The OSPA metric is used to compare the performance of the PHD filter and the two smoothing schemes. Parameters for the OSPA metric are chosen to be sensitive to location errors. A comparison is shown in Figure 5.5.

Table 5.1 summarizes the performance metrics for the PHD filter and smoothers for the low clutter scenario.

Error	PHD	FB-PHD	TF-PHD
RMS	0.21	0.01	0.00
OSPA (EM)	4.78	1.69	2.00
OSPA (Gibbs)	4.41	2.01	2.33

Table 5.1: Cardinality and OSPA ($c=30, p=2$) error for the PHD filter and smoothers.

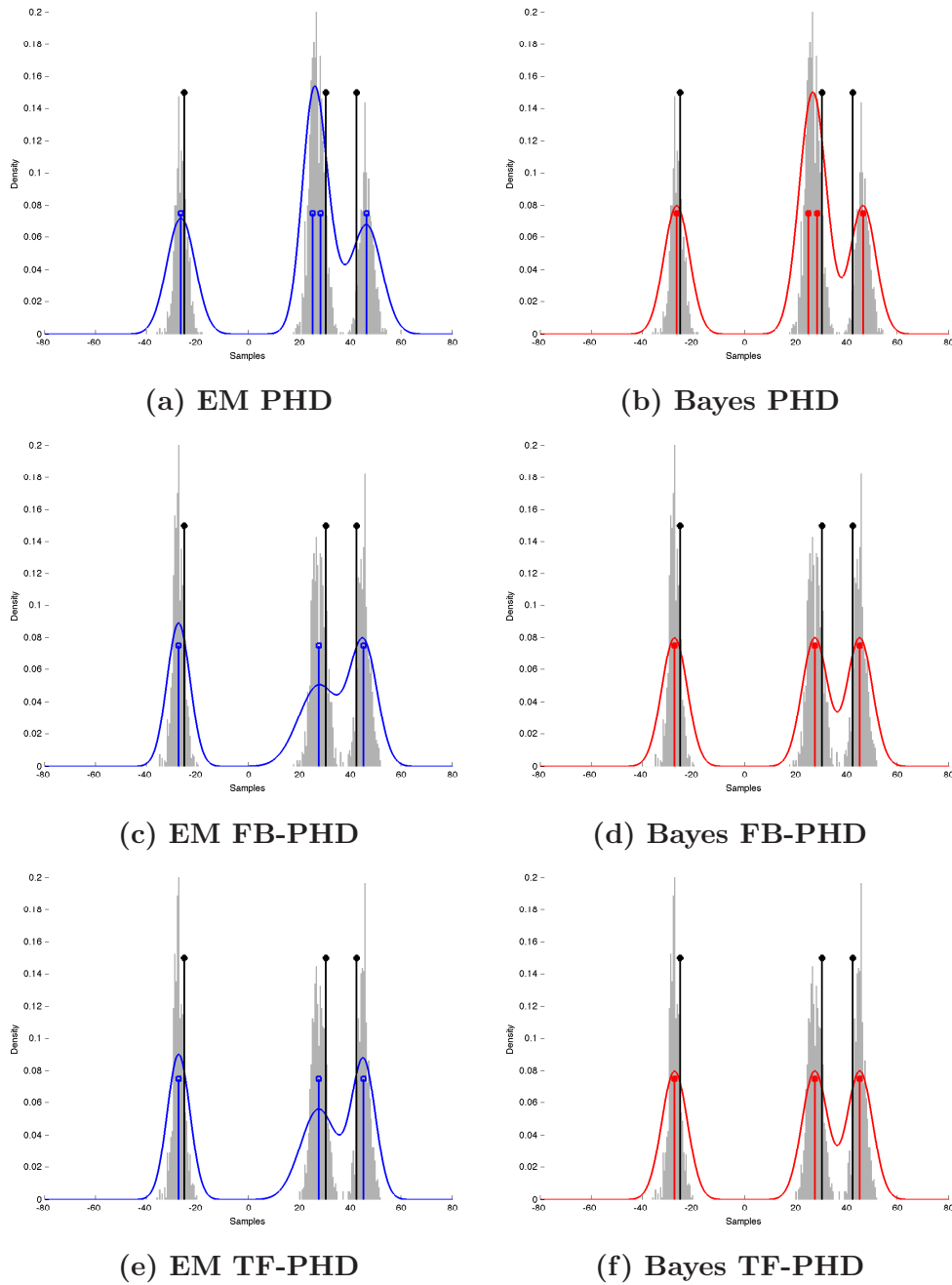


Figure 5.4: Illustration of particle PHD smoothing in frame 37, Ground truth of target locations (black stems) is plotted along with the particle approximation (gray histogram), the estimated state densities (blue and red line plots) and location estimates (blue and read stems). Figures (a) and (b) show the EM (blue stems) and Gibbs sampler (red stems) state estimation for the PHD filter with cardinality error, Figures (c), (d), (e) and (f) show the EM and Gibbs sampler state estimation for the PHD smoothers.

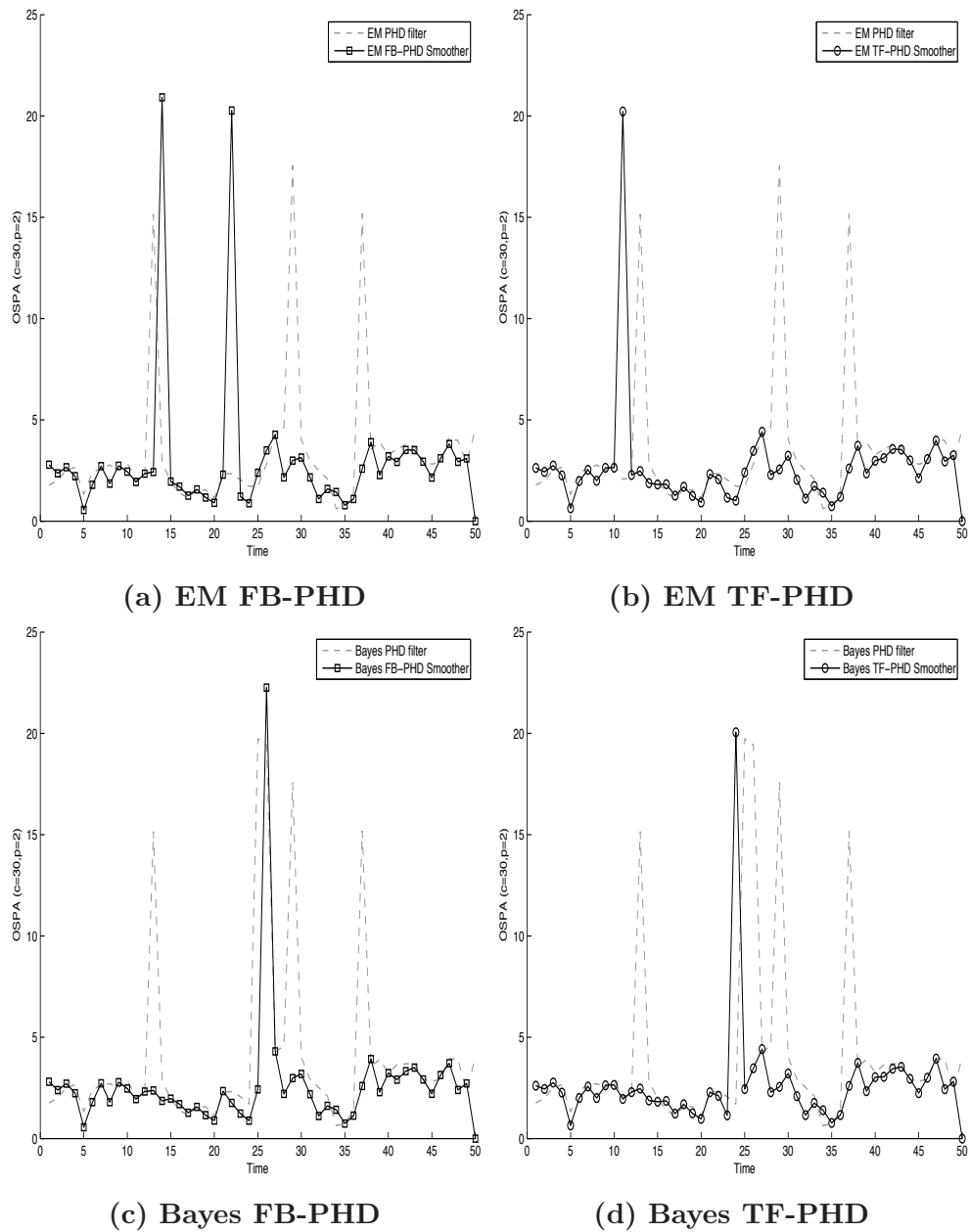


Figure 5.5: State estimation errors for the PHD filter, FB-PHD smoother and TF-PHD smoother. (a) EM state estimation for the FB-PHD smoother, (b) Gibbs sampler state estimation for the FB-PHD smoother, (c) EM state estimation for the TF-PHD smoother, (d) Gibbs sampler state estimation for the TF-PHD smoother

5.2.2 Low clutter scenario with missed detections

Although the PHD is theoretically capable of handling this situation, it has been reported in the literature that failed detections have a counterproductive effect on both the cardinality and state estimates in the PHD filter [25]. In this example, we will focus on the so-called *missed detections problem* and how it can be solved using PHD smoothing strategies.

The same target trajectories from the previous example are used to simulate a different observation model. In this case, each target has a probability of detection $\pi_d \leq 1$, which means that they might not generate an observation at a given time. More precisely, a probability of detection $\pi_d = 0.9$ is used to produce the multi-target scenario shown in Figure 5.6.

Ulmke *et al.* [109] showed that missed detections have a particularly adverse effect on the PHD filter that might result in lost tracks as well as a reduction in the overall sensor performance. In the example, the particle PHD filter is implemented with 1000 particles per target, and we can see that when there are no observations for a given target, the updated particle approximation PHD becomes negligible in that area. When performing Monte Carlo smoothing, particles are re-weighted in a backward pass. The FB-PHD smoother performs re-weighting based on comparing samples from two consecutive time frames, so it is able to recover the actual number of targets, even when no detections are present at a given time step. Cardinality estimates from the PHD filter and FB-PHD smoother are compared in Figure 5.7, where at time step 5 we can see one such missed detection. The cardinality estimate of the PHD filter becomes biased to the missed detection, but both the FB-PHD and the TF-PHD are able to amend that problem.

Figures 5.8(a) and 5.8(b) shows the PHD filter location estimates at time step 25 of the time series. Cardinality errors like this are very important since they then affect location estimates. If we have a certain number targets in a particular frame, and the estimated number of targets differs from the ground truth, then the estimated locations might also suffer a bias from the cardinality error. In the case of the SMC implementation of the PHD

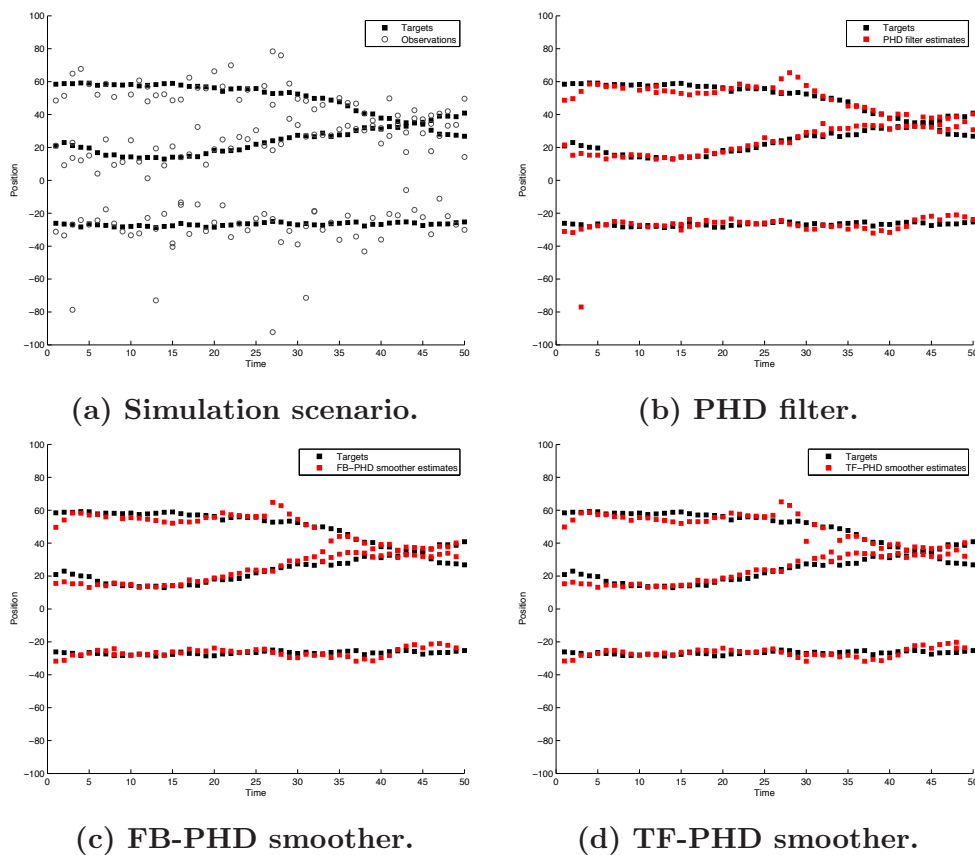


Figure 5.6: Multi-target tracking model with missed detections. Positions of 3 targets have to be estimated with low observation noise and moderate clutter. Missed detections are present in several time steps.

filter, particle weights become too small in areas where missed detections occur so the location estimates are also affected from the cardinality bias. Nevertheless, since smoothing also use future information, it can easily remedy the problem. Figures 5.8(c) and 5.8(d) shows location estimates of the FB-PHD smoother, and Figures 5.8(d) and 5.8(e) shows location estimates of the TF-PHD smoother.

The OSPA metric is then used to compare the performance of the PHD filter and the two smoothing schemes. A comparison is shown in Figure 5.9.

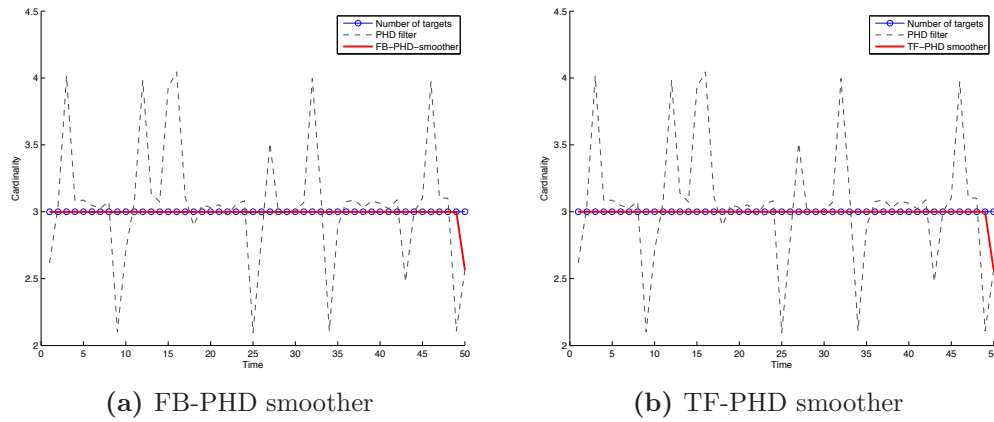


Figure 5.7: Cardinality estimates for the PHD filter and smoother. The PHD filter has unstable cardinality estimates due to the missed detections, but both PHD smoothers are able to recover from the error.

Table 5.2 summarizes the performance metrics for the PHD filter and smoothers with mis-detections.

Error	PHD filter	FB-PHD smoother	TF-PHD smoother
RMS	0.28	0.01	0.00
OSPA (EM)	6.36	2.65	3.12
OSPA (Gibbs)	6.71	3.04	3.20

Table 5.2: Cardinality and OSPA ($c=30, p=2$) error for the PHD filter and smoothers with missed detections

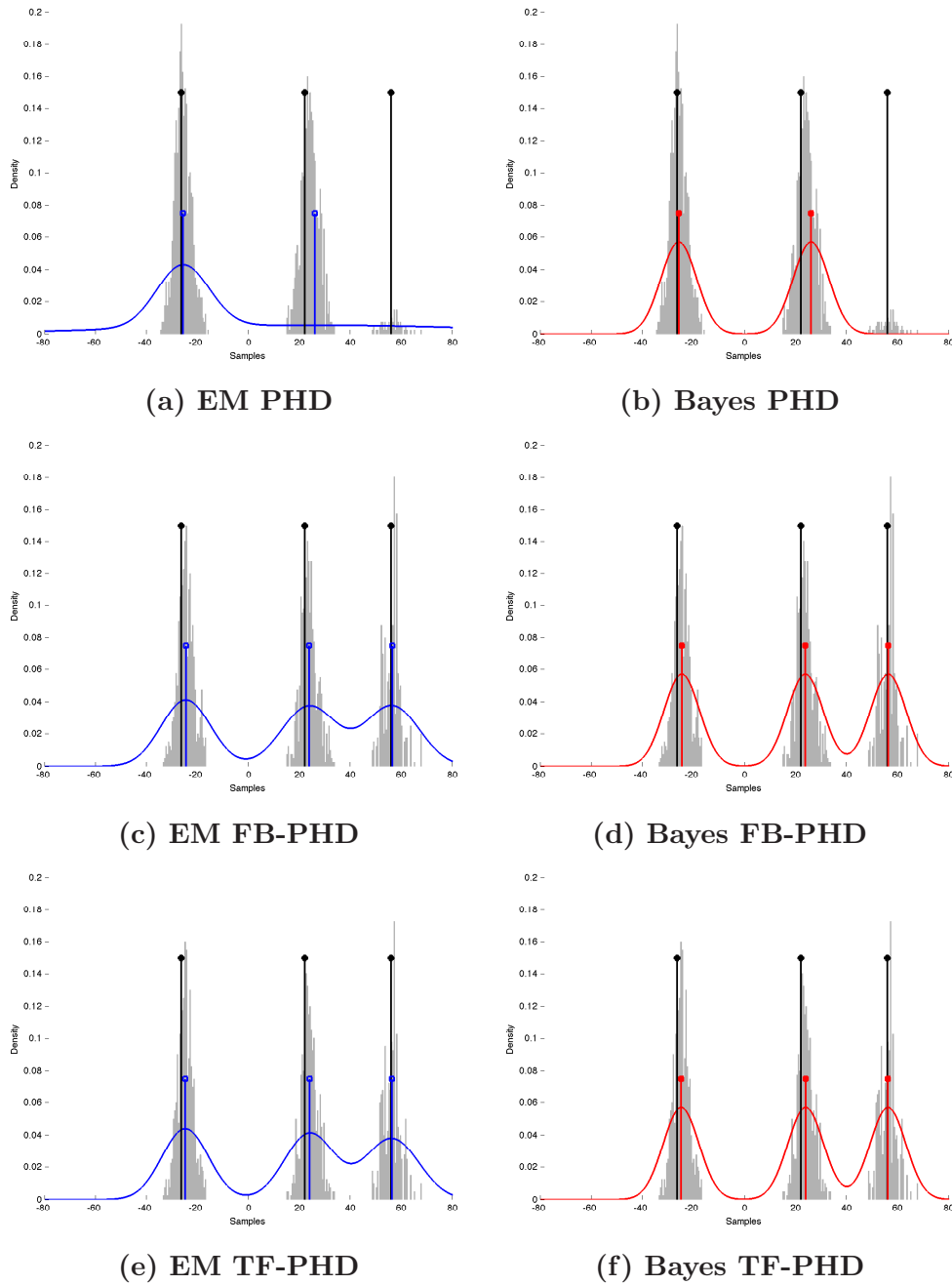


Figure 5.8: Illustration of particle PHD smoothing with missed detections in frame 25. (a) EM state estimation for the PHD filter, (b) Gibbs sampler state estimation for the PHD filter, (c) EM state estimation for the FB-PHD smoother, (d) Gibbs sampler state estimation for the FB-PHD smoother, (e) EM state estimation for the TF-PHD smoother, (f) Gibbs sampler state estimation for the TF-PHD smoother.

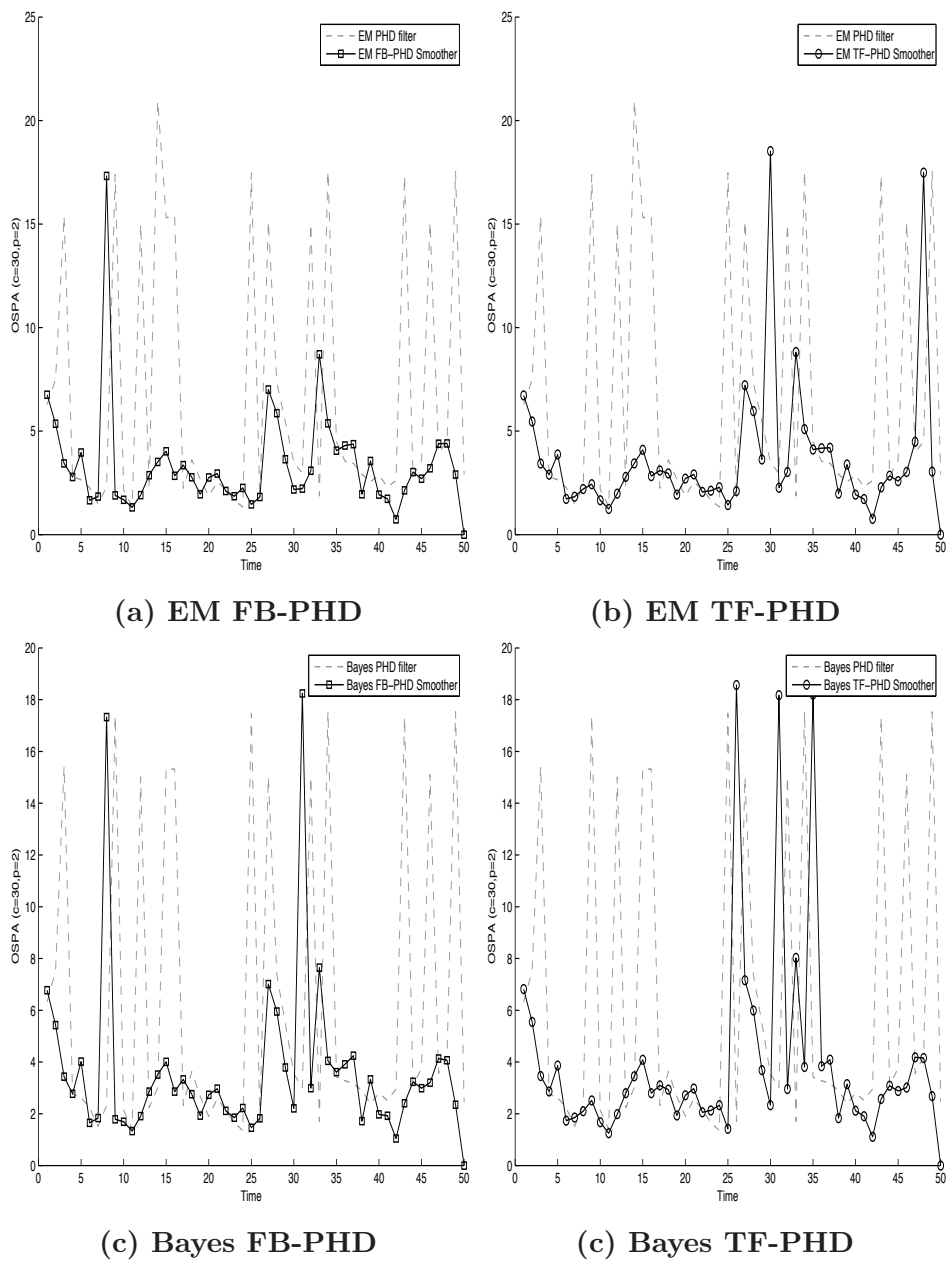


Figure 5.9: OSPA Error for the PHD filter, FB-PHD smoother and TF-PHD smoother with missed detections. Cardinality errors have an adverse effect on the location estimates, so the OSPA error of the PHD filter present several spikes where missed detections occurs. The EM and Gibbs sampler estimators of the FB-PHD and the TF-PHD smoothers provide improved estimates.

5.2.3 High clutter scenario

The PHD filter was originally conceived as a method for the detection and tracking of an unknown number of targets in highly cluttered environments [116]. In this scenario, multi-target tracking algorithms performing data association would not be able to correctly estimate the number of targets so the PHD filter becomes a viable alternative to the dynamic estimation problem. In order to demonstrate the utility of performing smoothing over the standard filtering, a Poisson process with unitary mean rate $10e^{-1}$ is used to simulate false alarms. A sample from the resulting dynamical system is shown in Figure 5.10.

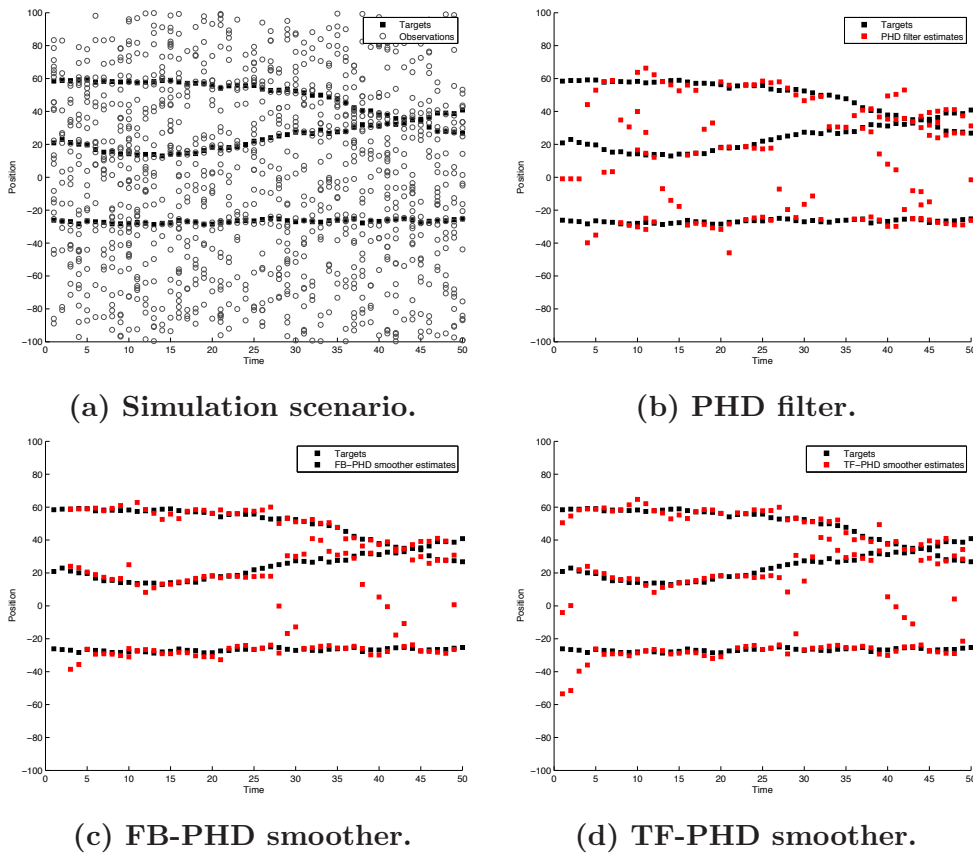


Figure 5.10: Multi-target tracking model with high clutter volume.

Since the PHD filter avoids any data association, the algorithm can cope with a time-varying number of targets, without explicitly modeling target birth or deaths. Particle filtering is used to simulate samples from the Poisson birth density, and a Poisson approximated number of detected targets in clutter. When performing Monte Carlo smoothing, particles are re-weighted in a backward pass. PHD smoothers perform sample re-weighting by comparing particles clouds from one step ahead using the single target dynamic model.

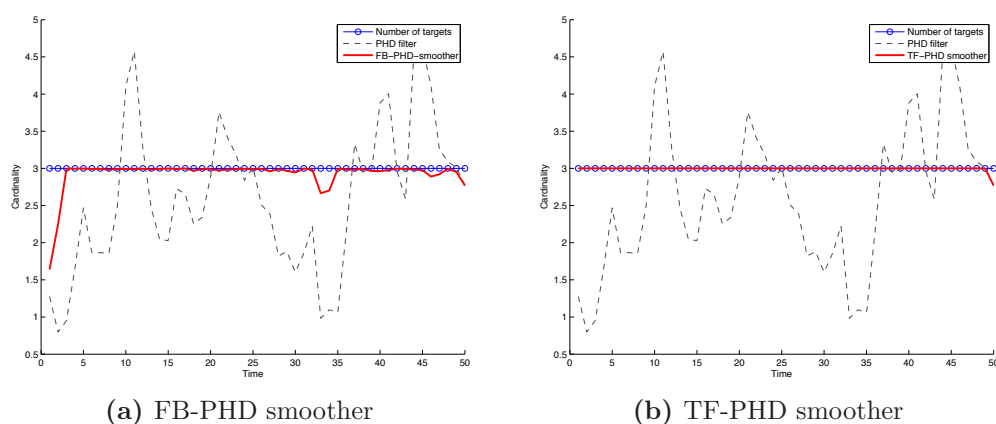


Figure 5.11: Cardinality estimates for the PHD filter and smoother.

As before, the OSPA metric is also used to compare the performance of the PHD filter and the two smoothing schemes. As recently noticed in a research paper submitted as a personal communication to the author [78], performance of the particle PHD filter degrades as clutter volume increases, and so does the performance of the FB-PHD smoother. We see the same pattern for the TF-PHD smoother in this example. Figure 5.12 shows the error for the state estimators using the FB-PHD and the TF-PHD smoothers.

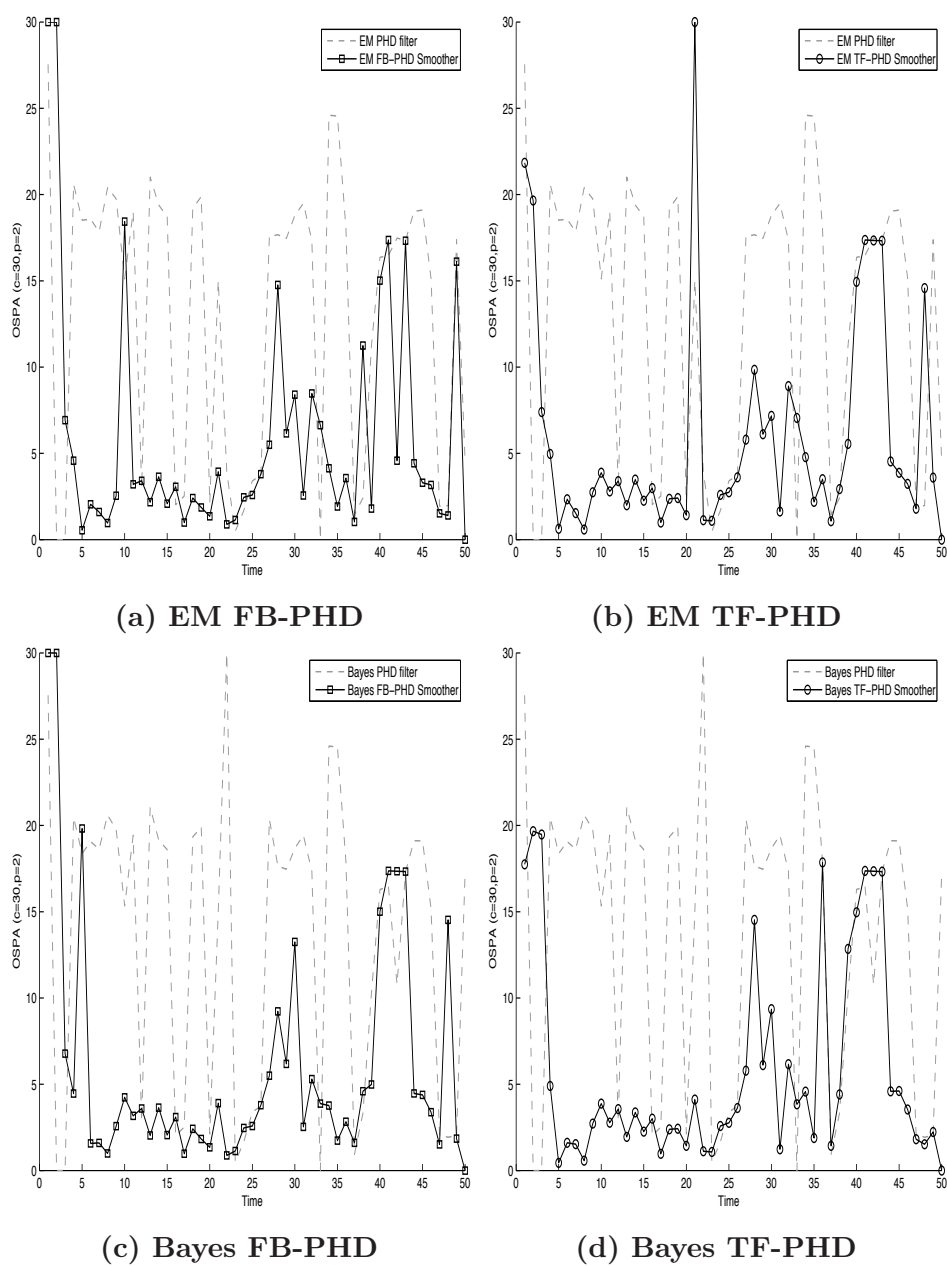


Figure 5.12: OSPA Error for the PHD filter, FB-PHD smoother and TF-PHD smoother with high clutter.

Table 5.3 summarizes the performance metrics for the PHD filter and smoothers in the case of high clutter scenario.

Error	PHD	FB-PHD	TF-PHD
RMS	0.70	0.07	0.01
OSPA (EM)	13.58	5.07	5.04
OSPA (Gibbs)	13.18	5.66	3.71

Table 5.3: Cardinality and OSPA ($c=30, p=2$) error for the PHD filter and smoothers in high clutter

5.2.4 High clutter scenario with target births and deaths

A more challenging example can be considered by taking into account birth and death of targets. Even for a fixed number of targets, standard multi-target algorithms would have to maintain a large number of association hypotheses in order to take into account all possible paths. Births and deaths of targets pose a difficulty, with the change detection for multiple models, so data association can be even less reliable unless clear heuristics can be performed for model evaluation [7].

The example in Figure 5.10 shows targets appearing at random times. As in previous examples, all targets follow linear trajectories with constant velocity. Birth rate of new targets is $\lambda_b = 5e^{-4}$ giving a Poisson rate of 0.1 new targets for each time step. All targets have equal probability of detection $\pi_d = 1$, so all targets are detected. Probability of survival is set to $\pi_s = 0.99$, and clutter rate is $\lambda_c = 10e^{-1}$ in a unit volume, which gives an average of 20 Poisson false alarms per scan. Figure 5.13 shows the multi-target model and the particle estimation.

As we noticed in Section 5.2.3, in the case of high clutter volume, the dynamic model becomes less reliable so cardinality estimates from the PHD filter are also affected. A similar behavior occurs for the smoothers, and more severely in the case of the TF-PHD smoother, which uses the observations for updating the particle weights. As a consequence, as shown in Figure 5.14, cardinality estimates from the TF-PHD and the FB-PHD smoother are also prone to error. Moreover, even for vector-valued smoothers, two different sets of samples with different support leaves almost all particles having small weights [27]. The same problem is inherited in particle PHD smoothers, therefore they cannot be appropriately used to evaluate dynamic scenarios that have some level of discrepancy between the forward and backward densities. Figures 5.14a and 5.14b show the cardinality estimates of the PHD filter and the two smoothers.

The OSPA metric is used to compare the performance of the PHD filter and the two smoothing schemes for the birth and death problem. A compar-

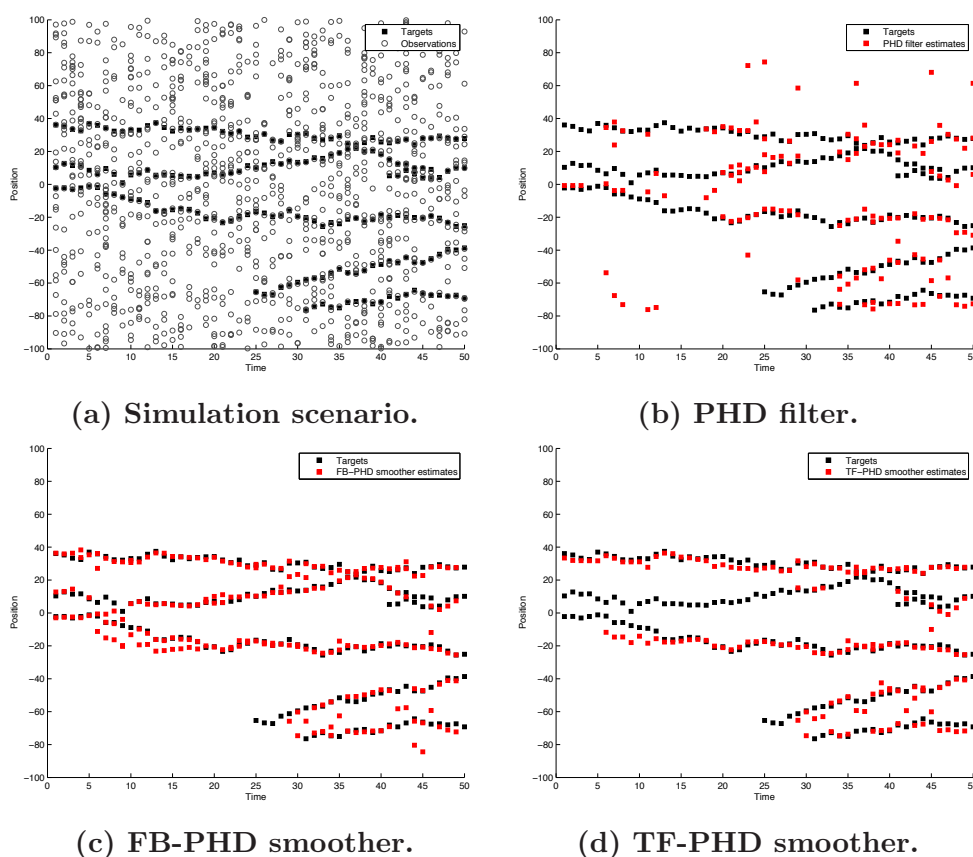


Figure 5.13: Multi-target tracking model in high clutter with targets birth and death. As seen in Figure (b), increasing the clutter volume deteriorates the performance of the PHD filter. Figures (c) and (d) shows the estimates from the FB-PHD and TF-PHD smoothers. In this example, the FB-PHD smoother can recover a lost trajectory in the forward pass, but that is not the case of the TF-PHD smoother.

ison is shown in figure 5.15.

Instead of a fixed-interval smoothing, we now consider a fixed-lag implementation for the FB-PHD and TF-PHD smoothers. Fixed-lag smoothing only requires a certain amount of data in the backward step, so it can be used in real time. At the same time, we can expect to have lower amount of samples

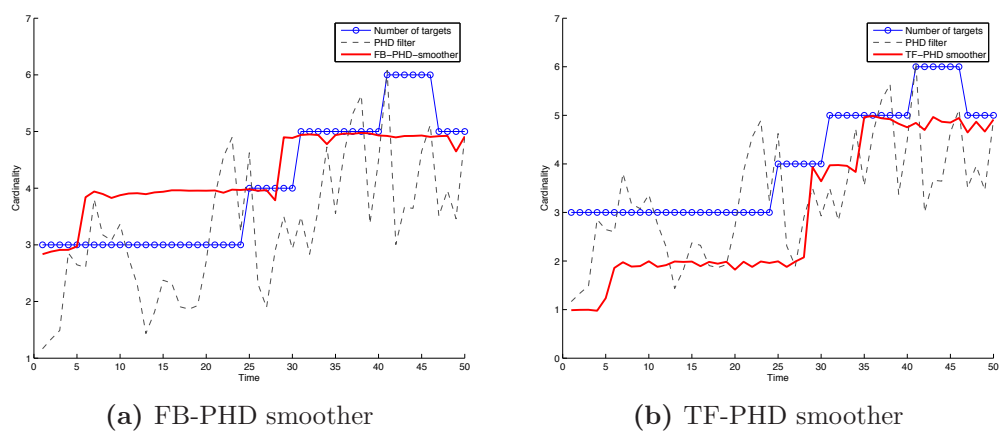


Figure 5.14: Cardinality estimates for the PHD filter and smoother.

discrepancy in small time lags. This is clearly a limitation for particle PHD smoothers, nevertheless improvements on the filter can be achieved when we encounter support in areas not well represented by the approximated filtering distribution.

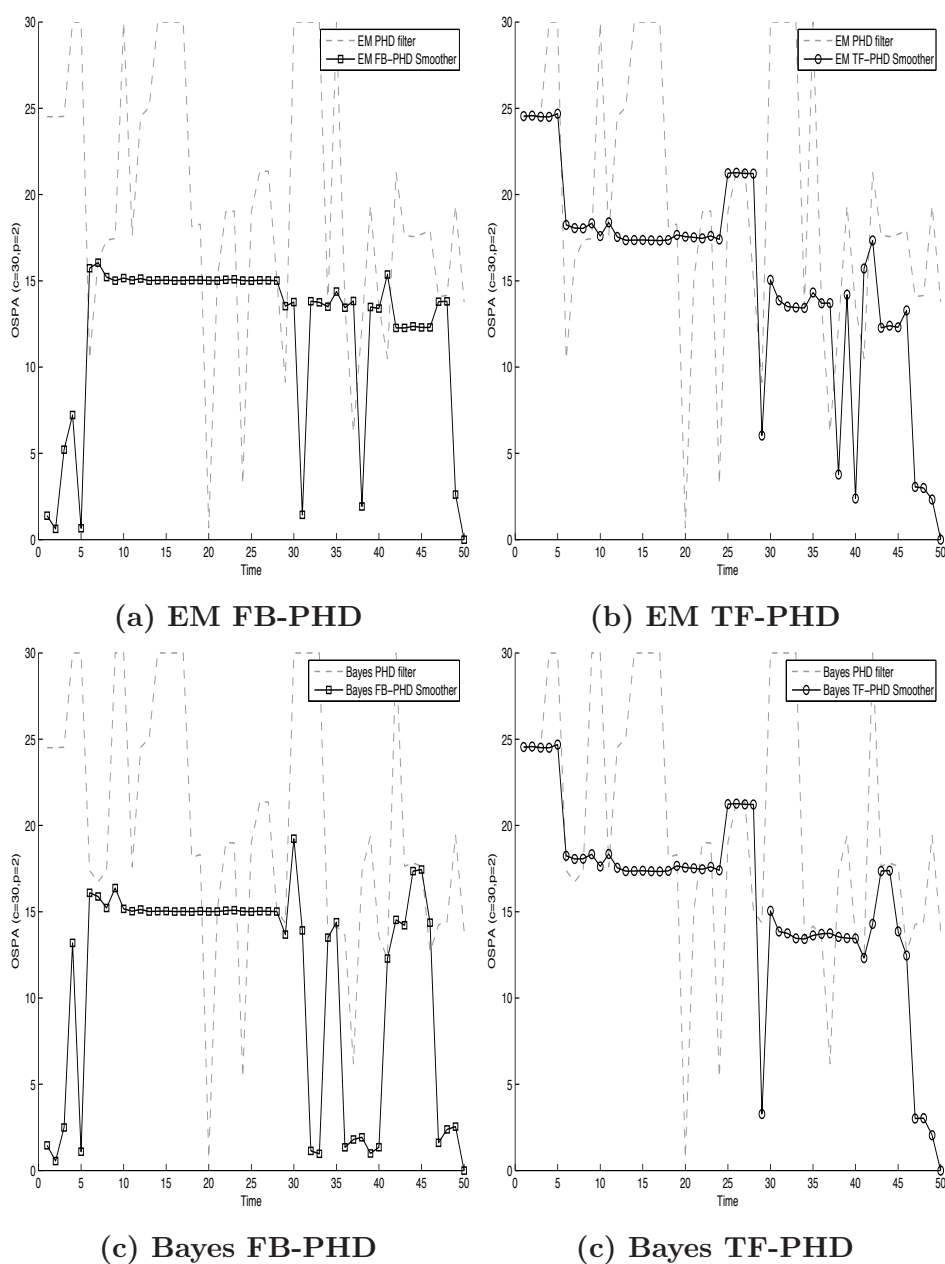


Figure 5.15: OSPA Error for the PHD filter, FB-PHD smoother and TF-PHD smoother with targets birth and death.

Table 5.4 summarizes the performance metrics for the PHD filter and smoothers with three different smoothing lags.

Table 5.4: *Cardinality and OSPA ($c=30, p=2$) error for the PHD filter and smoothers with births and deaths*

Error	PHD	FB-PHD	TF-PHD
Fixed-lag (1 time step) implementation			
RMS	0.68	0.71	0.73
OSPA (EM)	17.61	15.60	15.99
OSPA (Gibbs)	18.05	17.11	16.24
Fixed-lag (2 time steps) implementation			
RMS	0.68	0.65	0.69
OSPA (EM)	17.61	15.22	17.11
OSPA (Gibbs)	18.05	16.13	17.09
Fixed-lag (3 time steps) implementation			
RMS	0.68	0.59	0.68
OSPA (EM)	17.61	14.02	17.11
OSPA (Gibbs)	18.05	14.54	15.97
Fixed-lag (5 time steps) implementation			
RMS	0.68	0.63	0.82
OSPA (EM)	17.61	16.70	17.61
OSPA (Gibbs)	18.05	16.24	20.36

5.3 Summary and contributions

In this chapter, performance of the particle PHD smoothers has been discussed. A suitable metric for comparing multi-target estimation is used and the tracking scenarios considered are relevant to the visual problem. Firstly, a low clutter volume scenario was considered in Section 5.2.1, where the PHD filter is not able to discriminate between a new signal or false alarms. Because of the instantaneous nature of the PHD filter, the estimates are biased to the clutter rate, which is uniformly spread around the field of view. However, the particle support from the forward pass was good enough so both smoothers are able to recover the original signal.

Secondly, in Section 5.2.2 the case of missed detections is considered. When the probability of detection of each target is less than one, the PHD filter update step is not able to recover the signal by integrating out the null probability. Particle weights become negligible in areas where there are no detections, but smoothing can correct the error when the particles in the smoothing distributions are not dramatically distant. Since the dynamic model is informative in the linear Gaussian case with missed detections, the FB-PHD smoother performs seamlessly and recovers the actual number of targets even when no observations are available at a given time. Since the TF-PHD smoother uses the likelihood, it becomes more challenging to recover from a detection failure.

A third tracking scenario in Section 5.2.3 considers targets birth and death with moderate clutter. This is clearly a more challenging environment for particle PHD smoothers, since the number of targets is not constant and the Poisson process approximation will be determined by the reversibility of the Markov birth-death process [86]. In order to have a coupling construction which is integrable with respect to a reference Poisson process, target-independent survival rate and spontaneous (but possibly non-homogeneous) Poisson births are used [76]. Using this construction, the backward birth-death process is also guaranteed to be ergodic leaving PHD smoothers having

analytical priors. However, since the optimal backward density is evaluated point wise by a finite number of samples, no indicator of the jump times is actually estimated and the births and death probabilities are averaged between the existing samples. For that reason, the PHD smoother implementation presented in Section 5.2.4 which uses a small lag is better suited for this scenario.

6

PHD filter and smoother for visual tracking

This chapter we focusses on the visual tracking problem, where an object is defined as anything in a sequence of images that is of interest for analysis. As seen in Chapter 2, tracking can be defined as an estimation problem for the position and trajectory of one or multiple objects, using image measurements as observations. Applications of visual tracking are widely spread across different domains. Some examples are surveillance, multimedia information retrieval, human-computer interaction and cell biology. All these applications share the need to represent a complex object, which might be translated, rotated or occluded, by using noisy information from pixel measurements. Apart from the object reconstruction problem, objects may also present non-linear motion or might be represented by several non-rigid moving parts

[126].

Detection and tracking of a moving object in a video can be achieved by comparing the current image to a reference frame [47]. This technique is widely known as background subtraction, where the reference frame is usually termed the *background model* [125]. The background model is a representation of the scene without moving parts, and the complexity level of the model depends on the specific scenario. The output of the background subtraction step is a set of regions from the image that are assumed to belong to the foreground. These regions are widely known in the computer vision literature as “blobs” [81]. Features from the blob (such as bounding box, contour, etc) can then be used as measurements of the position of an object to be tracked. For example, one of the simplest forms of tracking consists of following the trajectory of the center of mass of a blob.

A basic background subtraction technique can use a single image as the background model, however this technique easily fails with small changes of luminance or geometry settings [84]. More advanced techniques can use a temporal sequence building a more robust background model. These techniques typically use a fixed number of images for calculating the variation of the pixel values across past scenes. The model gains benefits from the data acquired so far, but with an increased cost in computation and memory requirements. Moreover, the background model needs to be made in real-time with each new image frame, so there is a trade-off between foreground detection accuracy and the amount of data used for estimating the model [100].

One possible background model is to use the median value of the last k frames. The “temporal median” filter provides an accurate model even if data used is sub-sampled with respect to the original frame rate [64, 33]. However a limitation is the lack of a deviation measure for comparing the current image frame from the background model. A Gaussian probability density function can be also used for representing the background [125]. In this case, pixel values from past frames are assumed to be independent random

variables from a Gaussian density, so the sufficient statistics can be computed independently at each pixel location. At each frame time, an image pixel can be compared with the mean from its background model. The variance of the distribution can be used then for choosing a threshold for deciding whether a pixel belongs to the background or not. Online updating consists of replacing the model parameters with *cumulative averages* and using a learning parameter α for updating [8]. The cumulative average $\mu_k(x, y)$ can be written as a function of the image pixels values $I(x, y)$ and updated as:

$$\mu_k(x, y) = \alpha I(x, y) + (1 - \alpha)\mu_{k-1}(x, y) \quad (6.1)$$

Further improvement from the Gaussian model can be achieved by considering a mixture model of the background [101]. Complex scenes may not be represented by single mean values, so a mixture model can be used to represent a multi-modal density by mixtures of Gaussian distributions. This model is especially well suited in environments where background objects may appear at a faster rate compared to the background model update rate. In this case, estimation is performed iteratively with the expectation maximization (EM) algorithm, so dynamic updating requires fixing the algorithm for the online purpose. Nevertheless, the computation time for the iterative estimation may be improved by updating only those regions of the background where changes are occurring by creating a codebook representation [100].

Depending on each application domain, there might be several features that can be used to recognize the presence of an object within an image. Texture, color histograms, edges and optical flow are commonly used regional descriptors for object recognition [36]. Because of the dynamic nature of tracking, objects have to be recognized across frames in order to provide measurements for the tracking algorithm. This have proven to be challenging in most scenarios, and especially hard in the person tracking problem where it becomes hard to initiate tracks [29]. The tracking system must be able to find

all possible people in the scene, and then track them across the forthcoming scenes, but there is no standard method to find clothed people in arbitrary configurations.

When performing visual tracking for multiple targets, the observation model uses a “multi-blob” likelihood function that represent the likelihood that a particular configuration of targets produces an observed image mask [53]. At the same time, the multi-blob likelihood is used to describe several and unknown number of targets observed in clutter. The joint multi-target state space with unknown dimensionality can be addressed by using trans-dimensional Bayesian inference with reversible jump MCMC [59], while occlusion reasoning and object interactions can be carried out by Markov random fields [98]. However, these approaches have proven to be effective only when the number of targets is relatively small and targets are well separated [130].

Because the estimate is a first-order moment approximation to the joint multi-target density (i.e. a single target state space), the PHD filter differs from the previous approach for visual tracking. Vermaak et al. [112] showed that in the visual tracking case, the posterior distribution is characterized by a multi-modal density which is hard to propagate using the Bayesian filtering equations. However, since the PHD filter avoids explicit data association, it maintains the recursive Bayesian formulation in terms of prediction and update equations. Furthermore, the method also provides a tractable approximation to the joint posterior distribution, since the computational complexity is linear with the number of targets (see Section 3.2). Section 6.1 discusses previous applications of the PHD filter in video, while Sections 6.2 and 6.3 provides an extension to the PHD filter and smoother for the multi-person counting and tracking problem.

6.1 PHD filter for multiple person tracking

In this section, person tracking is described and evaluated. Tracking multiple humans is a challenging application because of the difficulty of generating the

likelihood of a person given pixels observations. Quantifying the information in a group of pixels can be potentially intractable, if we consider all possible orientations and occlusions. Early works for person counting considered pixel histograms where the head of the people can be distinguished [20]. More recent works have considered supervised learning techniques using region descriptors [17], Bayesian unsupervised learning using MCMC methods [129], and methods based on camera calibration [60].

The PHD filter has been used for multiple target tracking in video sequences in [96],[50], [121], [42] and [66]. Observations are taken from moments of the blobs generated from the background subtraction step, and extensions to color tracking and multiple cameras have been also proposed in [83] and [82]. Further developments in the application of the PHD filter for visual tracking considered data-driven approaches for designing birth and death proposals using scene information [123],[65].

Due to the complexity of generating a single observation for each person, the PHD filter has been used for tracking groups of people, where the likelihood of a group can be calculated directly from each generated blob [121]. Clutter can best be described as objects that are not of interest and in this case appears due to the moving parts of the scene which do not correspond to humans. Image processing techniques such as erosion and dilation are commonly used for post-processing the pixels from the background subtraction step and producing a foreground mask [111]. Since a closed area is expected to contain a single person or a group of persons, areas below a certain threshold can be discarded as clutter, but this operation depends on the ability to compare areas from different depth maps. In cases where perspective information is not available, the PHD filter is able to remove clutter from unthresholded measurements in the background subtraction step, while avoiding expensive data association from multiple hypotheses [56]. Furthermore, multiple persons merging into a single group, and group splitting, can also be explicitly modeled when using the Gaussian mixture PHD filter [122].

An important remark on the PHD filter in visual tracking is that each

observation is independent of all others. In order to achieve a tractable approximation of the posterior distribution, all regions in the observations space are assumed to factor as independent partitions (i.e. the expectation of the posterior density being a factorial moment measure as being defined in 7), so the likelihood becomes available only for non-superpositional measurements where the region of influence of an object does not overlap with the others [117]. This is an important limitation for the application of human tracking, where each observation in a crowded region might contain information from several targets, and the likelihood cannot be efficiently separated into marginal single target evaluations [62]. Nevertheless, if we compare the PHD filter approach with other approaches like multi-hypothesis tracking whose computational complexity is exponential on the number of targets, the linear complexity of the PHD filter has an attractive appealing for real time applications, while at the same time achieving good performance [80].

6.2 Person tracking model

For the purpose of evaluating the PHD filter and PHD smoothers, multiple person tracking without person recognition is performed. The reason behind not having an explicit person recognition step is the complexity of detecting a single person in a crowded scene, so a simple marked point process can be formulated as a point process in \mathbb{R}^2 for the locations with marks on \mathbb{R}^2 for the velocities. On the other hand, since no explicit model is used, we expect to find a reasonable number of false positives and missed detections, which complicates the data association step of a multi-hypothesis tracking algorithm.

Instead of using an explicit person detection device, a PHD filtering approach is used to estimate the state of an unknown number of persons. A constant velocity model is used as a generative model for the movement of a single person. The forward model calculates the new position of a person using a velocity vector that remains nearly constant in magnitude and direc-

tion. Let $x_k = [x_x, x_y]^T$ be the transpose of a 2-dimensional position of a person in the image plane and $\dot{x}_k = [\dot{x}_x, \dot{x}_y]$ its velocity.

In a state-space representation the state vector of a person is written as an augmented vector $\mathbf{x}_k = [x_k, \dot{x}_k]$. A linear mapping is used to model the dynamic behavior of a person, and Gaussian noise w_k . The position at the discrete time k can be written as:

$$\begin{aligned}\mathbf{x}_k &= F \mathbf{x}_{k-1} + w_k \\ w_k &\sim \mathcal{N}(0, \Sigma_{\mathbf{x}_k})\end{aligned}$$

where F is a linear transformation matrix in which dt represents the sampling time:

$$F = \begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The observations $\mathbf{y}_k = [y_x, y_y]$ only contain information about the position of a person, so velocity has to be estimated from previous measurements [5]. The velocity is related to the object position as $\dot{x}_k = (x_k - x_{k-1})/dt$ for each sampling interval dt . However, since the PHD filter does not perform inter-frame person association, velocity is sampled from a zero-mean Gaussian prior distribution $\mathcal{N}(0, \Sigma_{\dot{x}})$ with diagonal covariance.

The observations are related to the state of a person by means of a linear transportation matrix G plus Gaussian observation noise v_k :

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\begin{aligned}y_k &= G x_k + v_k \\ v_k &\sim \mathcal{N}(0, \Sigma_{y_k})\end{aligned}$$



Figure 6.1: A single person is represented as an ellipsoid consisting of a 2-dimensional centroid and the equidensity contour of the Gaussian observation noise

Figure 6.1 shows different examples for an elliptical person model.

6.3 Visual Tracking Examples

In this section, the practical implications of using the PHD filter and smoother in human tracking in real world surveillance scenarios are studied. For that purpose, a benchmark pedestrian database is used which is publicly available for testing new algorithms in crowd analysis [13]. The UCSDPEDS dataset contains several videos of pedestrians taken from a stationary surveillance camera. The videos are 8-bit grayscale, with dimensions $[238 \times 158]$ at 10 frames per second. We focus on the persons counting and tracking task, and the worked examples will show the PHD performance for this case. Figure 6.2 shows an example of a particular scene from the dataset.

A temporal Gaussian background model using the parameters specified in Table 6.1 was used for generating the foreground blobs, representing the moving parts of the scene. In order to estimate the number of persons within a blob, we have to consider the effect of perspective in the scene. Persons closer to the camera appear larger than the ones further away, so any feature based on area or size would have to be normalized. When the camera is static, the geometry of the scene can be used to create a perspective map, for weighting each pixel in the image [60]. This feature has been exploited



Figure 6.2: Crowded scenario with multiple people walking in different directions. A single camera captures images at 10 frames per second and the goal is to track and count individual persons.

in crowd counting methods using pattern recognition and image processing methods [12]. After the perspective normalization step, each blob has to be segmented according the estimated number of persons inside the containing region. Since we don't take into account important factors such as the average distance between occluded persons (which also has sociological and cultural implications) [128], this operation is inherently inaccurate and subject to approximation errors. Figure 6.3 shows an example of the foreground regions produced by the background subtraction step with parameters specified in Table 6.1.

Small to medium crowds are present and several occlusions are also occurring. Since person density on each blob is not accurately measured, there are a large number of false alarms. At the same time, an incorrectly estimated number of clusters also lead to wrongly observed positions, increasing the complexity of the estimation procedure. From each foreground blob, we want to have a rough estimate of the number of persons inside, so in order

Parameter	value
Frame buffer (frames)	30
Learning rate	0.75
Normalized person area (square pixels)	250

Table 6.1: *Parameter settings for the background subtraction model*



Figure 6.3: *After the background subtraction step, from the image in Figure 6.2 a binary image of the foreground is used to extract observations for the tracking algorithm. A single blob might contain either a group or a single person.*

to compare areas in different locations on the image we use a weight for each pixel taken from the perspective map. For that purpose, we estimate the projection of the area occupied by each blob in the ground plane, to the area that corresponds to a single person. Figure 6.4 shows the perspective map used for this dataset.

A schematic diagram of the procedure for generating observations from image measurements is shown in Figure 6.5.

The SMC implementations of the PHD filter and smoothers are used for

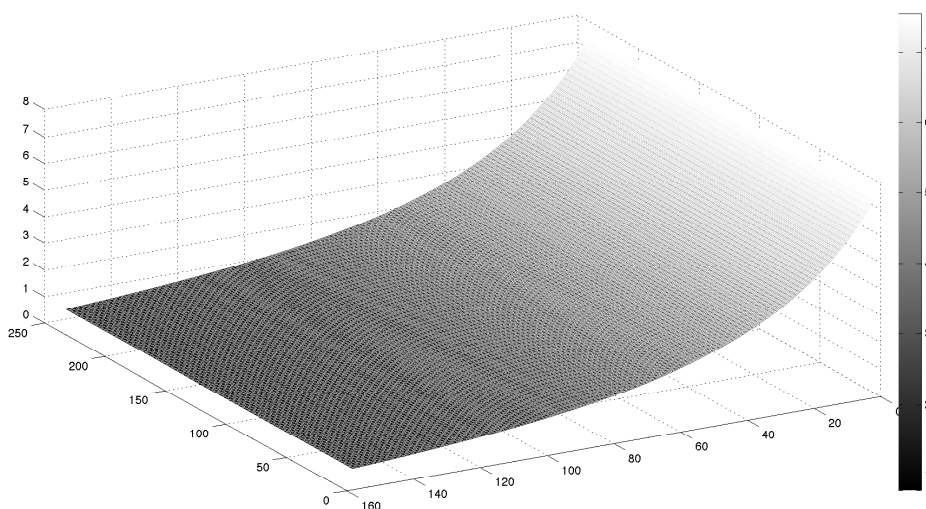


Figure 6.4: Each pixel is weighted according to a perspective map. In order to compare areas from different areas of the image, objects closer to the camera have lower weights than objects being far away.

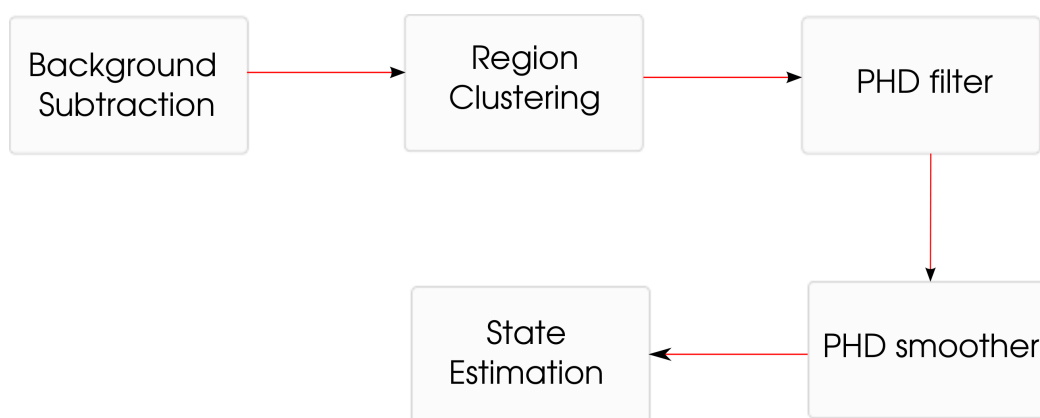


Figure 6.5: Schematic diagram of the tracking procedure.

the task of estimating the position and velocities of each person. Apart from the clutter due to the background subtraction step (e.g. when moving objects appearing at faster rates than update rate of the background subtraction step), clutter also appears due to an *over-segmentation* on each area. Param-

eters for the PHD filter and smoothers are shown in Table 6.2. Whenever the estimated number of persons within a blob is greater than the ground truth, there will be observations not coming from any target. Conversely, an under-estimated number of persons would also produce *under-segmentation*. Moreover, since clustering is applied to pixel positions regardless of the center of mass of each person, observations are also contaminated by substantial noise. Figure 6.6 shows an example of the segmented foreground blobs with the observations.

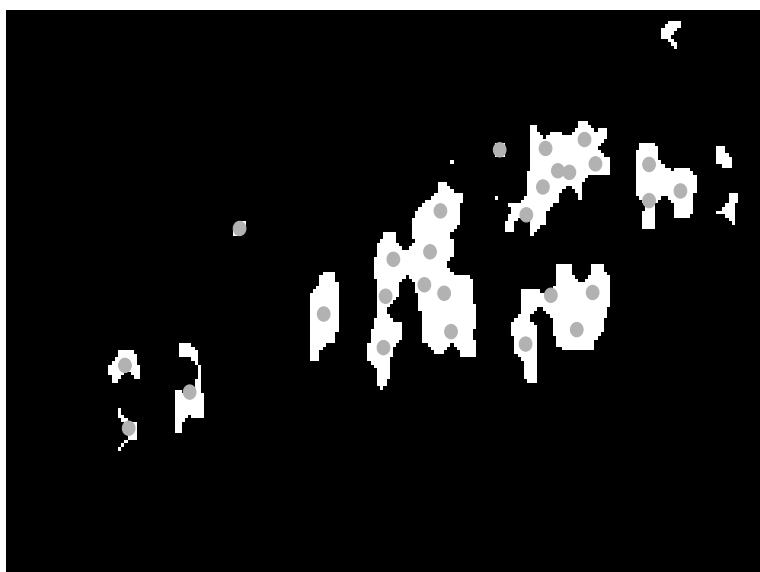


Figure 6.6: A rough estimate of the number of persons and their locations is given to the PHD filter as an input. Pixels within a blob are clustered to the expected number of persons according to a reference area. Gray dots denote observations for the PHD filter and smoother. Over-segmentation and under-segmentation of the blobs causes clutter and mis-detections.

The limitations of the PHD filter have been already highlighted in Section 3.7, so the estimates of the filter are expected to be unstable and have large variance. Even so, and despite the fact that no data association or person recognition has been performed, as shown in Table 6.3¹, the filter

¹Results of the RMS error in red are taken from [12] for the same dataset used here.

Parameter	value
Number of particles per target	150
Poisson clutter rate (per unit value)	$1e - 4$
Poisson birth rate (per unit value)	$1e - 5$
uniform spatial clutter density	$U([1, 238] \times [1, 152])$
uniform spatial birth density	$U([1, 238] \times [1, 152])$
initial Poisson birth rate	10
target process noise (Σ_{x_k})	$diag([5, 5, .1, .1])$
target observation noise (Σ_{y_k})	$diag([8, 4])$
target survival rate	0.95
target detection rate	0.95

Table 6.2: Parameter settings for the PHD filter and smoother.

manages to achieve relatively good performance. Location errors are illustrated with the OSPA error. A large number of outliers are expected, so parameters of the OSPA metric are chosen to be sensitive to them.

Error	PHD filter	FB-PHD	TF-PHD	GP(1)	GP(2)
RMS	2.23	1.62	1.53	0.87	1.30
OSPA (EM)	1.61	1.61	1.60	-	-
OSPA (Gibbs)	1.61	1.62	1.61	-	-

Table 6.3: Cardinality and OSPA error ($p=2, c=2$) for the PHD filter and fixed-interval smoothing for visual tracking. GP(1) corresponds to the RMS error for the Gaussian process regression method using texture and area features, and GP(2) to the same method using area only features. Crowd counting results for the PHD filter and fixed-interval implementations of the FB-PHD and TF-PHD smoothers are comparable to the GP(1) method which uses area only features.

So far, the tractability of the PHD filter approximation has been highlighted, whose computational complexity is linear with respect to the number of targets. However, one of the shortcomings of the particle implementation

of the PHD filter is its dependency on the observed measurements, and more specifically in the case of missed detections and false alarms. In that regard, we have also pointed out the benefit of using smoothing by making the implicit assumption that the computational complexity is the same as in the forward step. Table 6.4 shows the algorithm performance in terms of processing time (in seconds) per video frame. It is worth noting that the implementation used here is not intended for measuring real time performance, but the extension to an operational visual tracking system using PHD filtering and smoothing could be achieved by using state of the art image processing libraries such as OpenCV ².

Algorithm	Time [s]
PHD	21.8378
TF-PHD	19.014424
FB-PHD	19.113789

Table 6.4: *Speed of the PHD filtering and smoothing algorithms*

The results for the crowd counting error are consistent with previous developments using the same dataset, where a more complex background subtraction step was used and a supervised Gaussian Process (GP) regression was performed [12]. The authors reported problems in their approach when not enough training examples were given for a particular person count. Since no person detection step was performed, errors were also produced due to fast moving objects, also requiring more training data to accomplish those cases. In the PHD filter approach, single persons are tracked seamlessly without any effort, but larger areas corresponding to bicycles or golf carts are tracked as multiple persons. This error is also due to the lack of a person detection step, and any larger area continuously moving would be over-segmented and tracked as multiple persons.

Multiple observations from a single person caused by over-segmentation

²<http://opencv.willowgarage.com/>

would cause problems in multi-target tracking methods. Moreover, incorrect person detections would worsen the SNR ratio, deteriorating the performance of the filter. In Figure 6.7, the ellipses are used to enclose detected persons and due to the under-segmentation problem, a group of pedestrians is represented by a single target. Furthermore, because no person recognition has been performed, the estimates are not sensitive to the area occupied by a single person. Therefore, as a consequence of a poor SNR ratio, cardinality and state estimates becomes susceptible to under-segmentation and over-segmentation issues. Also, since the PHD filter does not perform any data association, the assesment of the error on individual person locations and velocities is not straightforward, requiring an additional step.

Figure 6.7 shows the SMC PHD and TF-PHD estimates for frame 20 of the vidf1_33_001.y³ sequence of the dataset. A person with a bicycle has a bigger area than the expected average, and as a result over-segmentation causes the PHD filter in Figure 6.7a to incorrectly estimate the number of targets in that area. Nevertheless, the TF-PHD smoother in figure 6.7b is able to give an improved estimate in the region containing a single person.

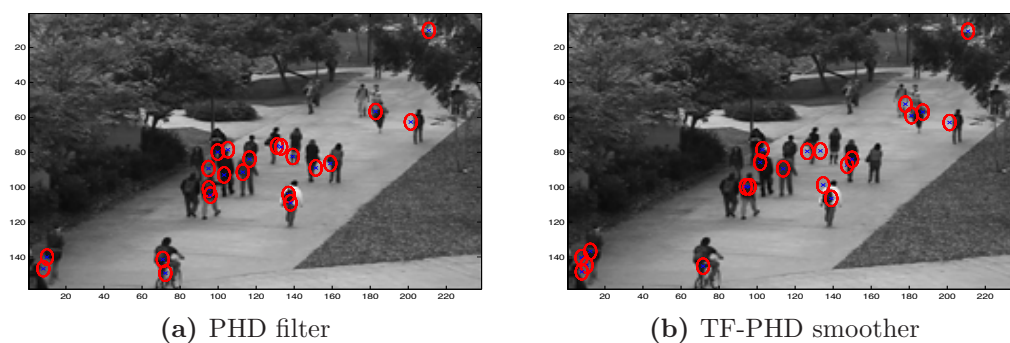


Figure 6.7: Particle PHD filter and TF-PHD smoother estimates for frame 20.

Fixed-interval implementations of TF-PHD and the FB-PHD smoothers achieve improved estimation of the number of persons when the time interval

³Each sequence contains 200 frames and sequences are labeled vidf1_XX_ZZZ.y, where XX is a viewpoint identifier and ZZZ is the sequence identifier.

is relatively small. Figure 6.8 shows the estimated number of targets for the PHD filter and both smoothers for the first 50 frames of the sequence. Consistently with the benchmark results in Chapter 5, the smoothed estimate is more stable than the filtered one.

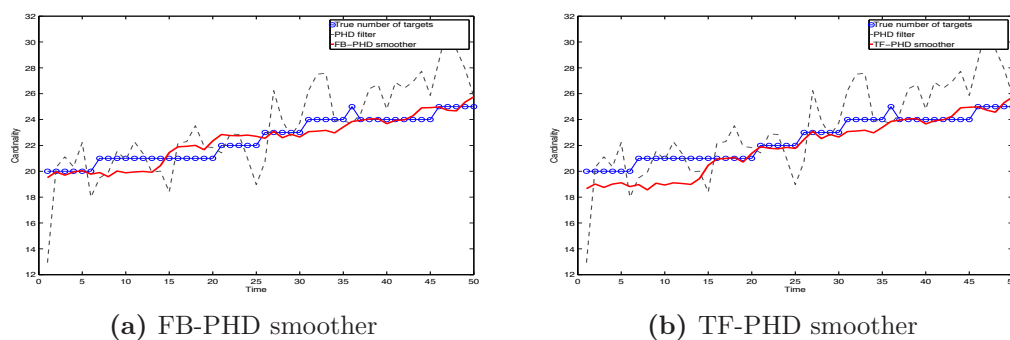


Figure 6.8: Cardinality estimates for the PHD filter and smoother. The TF-PHD and the FB-PHD smoothers give an improved estimate of the number of targets.

Person locations that are incorrectly addressed due to cardinality errors in the forward pass can be re-estimated using the improved estimate. However, since resampling was performed in both steps, it is more challenging for the PHD smoothers to provide improved location estimates. Furthermore, since the PHD filter proposes individual samples for each person, location estimates are not sensitive to inter person distances. This issue is also inherited by particle PHD smoothers, so location estimates suffer from the same problem. Figure 6.9 shows estimates from the PHD filter, the FB-PHD and the TF-PHD smoothers using the EM algorithm and the Gibbs sampler for state estimation (see Section 3.6).

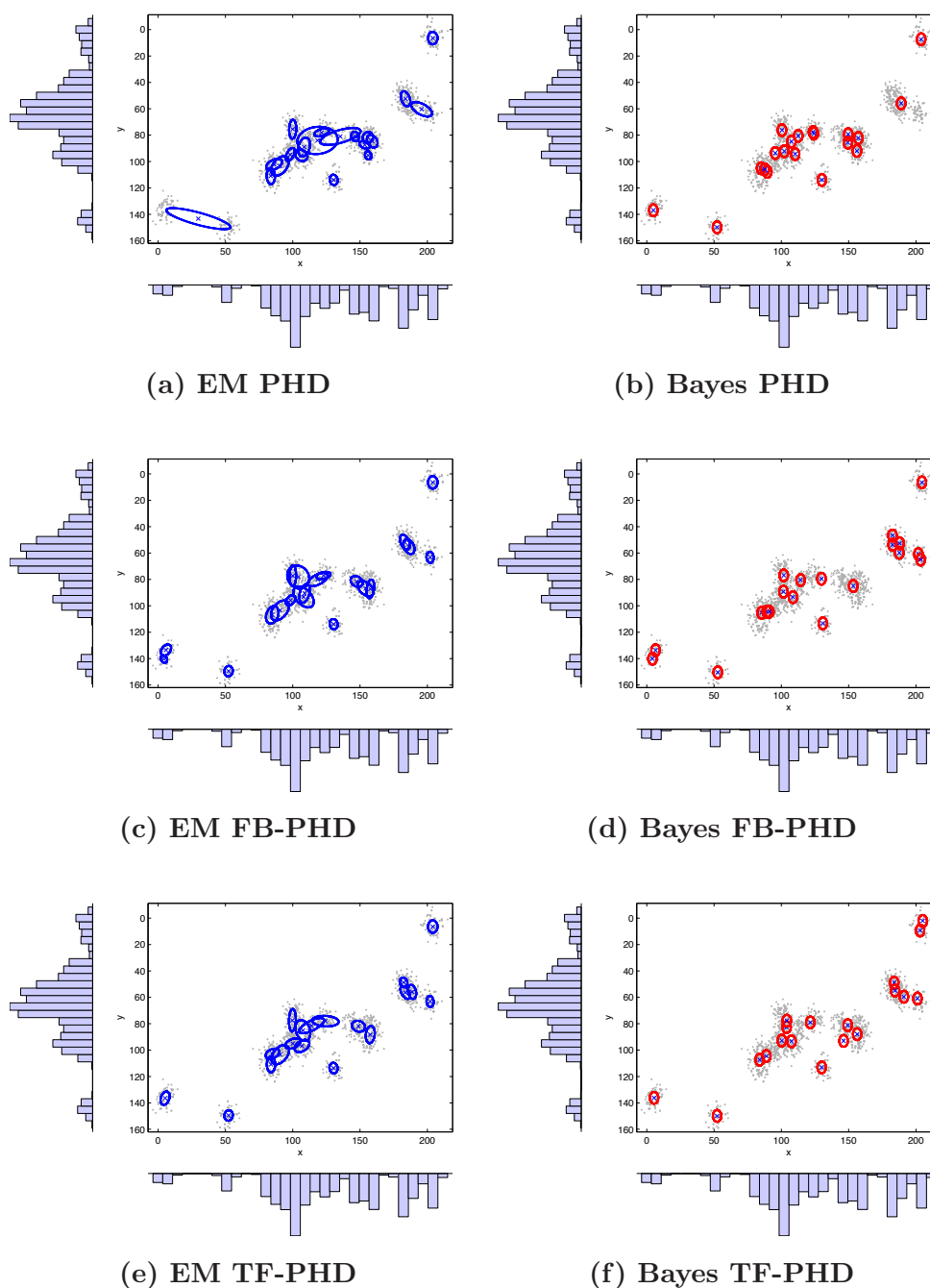


Figure 6.9: Particle approximations for frame 14 of the pedestrian tracking sequence. Location estimates from the PHD filter suffers from an incorrectly estimated number of persons. Since the Gibbs sampler is less sensitive to the initial conditions, it manages to allocate person locations more accurately and with less variance than the EM algorithm. Monte Carlo approximations by means of the FB-PHD and the TF-PHD smoothers provide improved estimates over the PHD filter alone.

As seen in Table 6.3, the estimated number of targets in the backward step is less sensitive to fluctuations in the number of observations. Since estimates and ground truth might have different cardinalities, the OSPA error is used for comparison purposes. Figure 6.10 shows the filter and smoother OSPA errors:

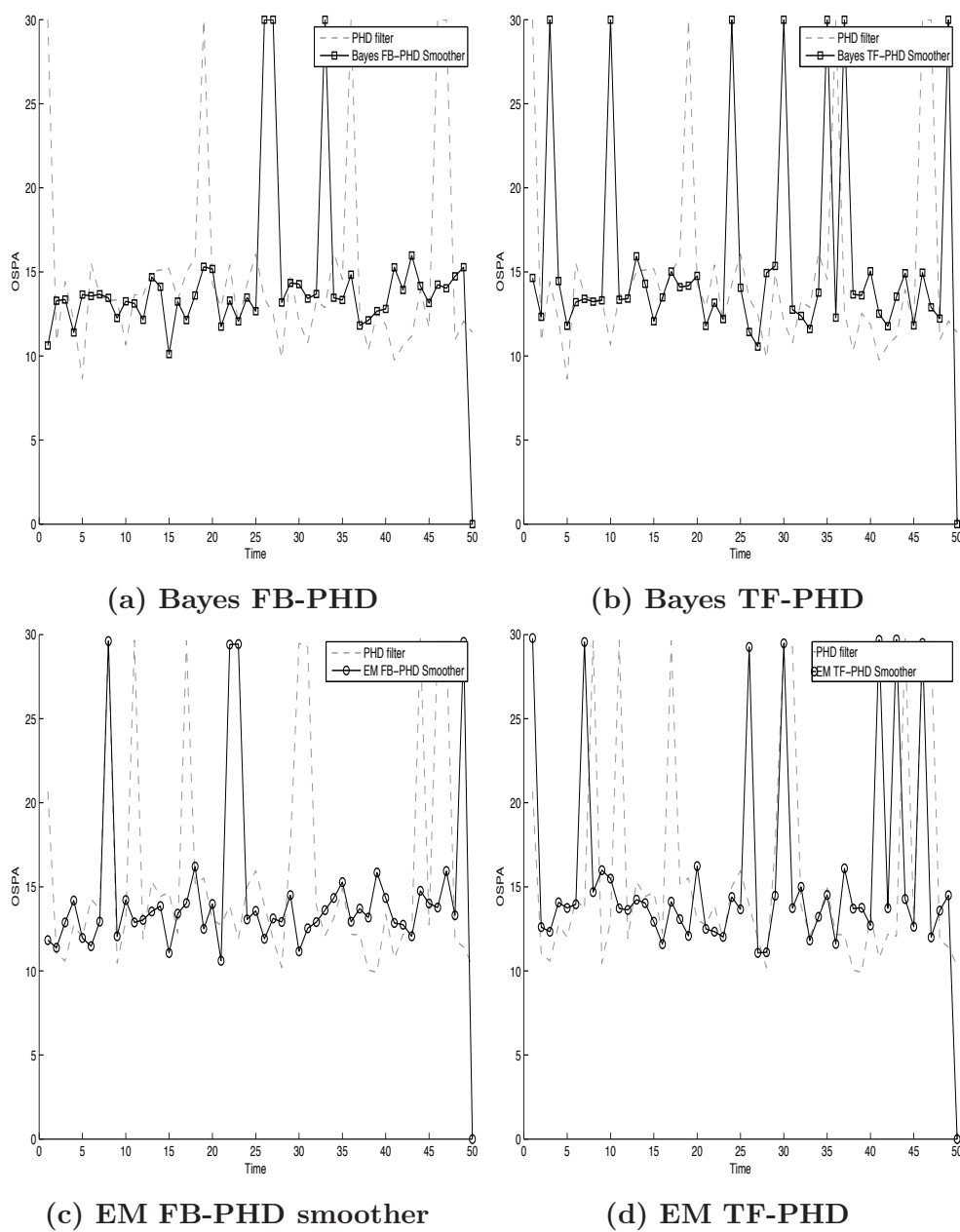


Figure 6.10: OSPA Error for the PHD filter and fixed-interval implementations for the FB-PHD smoother and TF-PHD smoother for visual tracking. A fixed-interval of 50 frames is used to evaluate the benefit of smoothing over filtering.

We will concentrate now on the performance on the sequence vidf1_33_001.y using fixed-lag implementations. As opposed to fixed-interval, fixed-lag implementations can be implemented in real time using a small time lag. Parameters for the PHD filter and smoothers are shown in table 6.2. Four different time lags are considered and Table 6.5 shows the performance of the TF-PHD and the FB-PHD smoothers when the EM algorithm and the Gibbs sampler are used for state estimation. In this case we expected to have a large number of outliers in the estimated locations. Therefore, in order to measure the performance of smoothing over filtering, we choose the OSPA metric to be less sensitive to outliers.

Error	PHD	FB-PHD	TF-PHD
Fixed-lag (1 time step) implementation			
RMS	2.26	2.11	2.11
OSPA (EM)	1.60	1.61	1.60
OSPA (Bayes)	1.62	1.60	1.60
Fixed-lag (2 time steps) implementation			
RMS	2.26	2.04	2.02
OSPA (EM)	1.60	1.59	1.60
OSPA (Bayes)	1.62	1.59	1.60
Fixed-lag (3 time steps) implementation			
RMS	2.26	1.88	1.86
OSPA (EM)	1.60	1.57	1.57
OSPA (Bayes)	1.62	1.58	1.59
Fixed-lag (5 time steps) implementation			
RMS	2.26	1.83	1.81
OSPA (EM)	1.60	1.57	1.57
OSPA (Bayes)	1.62	1.57	1.57

Table 6.5: Cardinality and OSPA ($c=2, p=2$) error for the PHD filter and smoothers for visual tracking

Increasing the time-lag improves performance, but it can be seen that the OSPA error for both EM and Gibbs sampler estimation converges at time lag 5. The numbers in Table 6.5 are also consistent with other reported results for the FB-PHD smoother for a 2-dimensional bearings-only tracking scenario [78], where a fixed-lag of 3 is reported as being sufficiently high that longer time lags will not improve estimates.

6.4 Summary and contributions

This chapter presents an original application of the PHD filter and smoother for tracking multiple persons in crowded scenarios. The model-based approach uses the point process formalism to represent a person as a marked point process for locations and velocities of an unknown and time-varying number of persons. Given the difficulty of obtaining measurements of the position of a single person in a crowded environment, it becomes crucial to adopt a filtering scheme that avoids data association. For that reason, the resulting approach is well suited for on-line surveillance applications. Furthermore, a suitable metric has also been proposed for evaluating the performance of pedestrian tracking algorithms. The OSPA metric is appropriate for comparing estimates of possibly different cardinalities, so it is better suited for evaluating performance of visual tracking algorithms than previously proposed benchmark suites [106].

It is worth noting that the proposed method can be extended to other visual surveillance scenarios and to more detailed pre-processing schemes for person counting. For the dataset considered here, we found that the normalized area occupied by a single person provided a starting indicator for the number of persons within a blob. Interestingly, the RMS error obtained for the number of persons is completely concordant with the results for crowd counting using supervised regression when only the area is used as a feature [12]. Moreover, the results for the supervised approach indicate that the method tended to fail when there are few people in the scene, so we can expect the PHD filter tracker to improve if more features are considered for segmentation.

State estimation is performed by means of the EM algorithm and the Gibbs sampler. Instead of human groups, we are interested in tracking persons. Therefore, each blob has to be segmented in accordance with a rough estimate of the number of persons. This process is conducted disregarding the persons center of mass, so the observation noise also increases. We ob-

served (see Figure 6.9) that state estimation with the Gibbs sampler is less sensitive to the observation noise and provides estimates with less variance than the EM algorithm.

7

Conclusion

This thesis established a marked point process formalism for visual tracking applications. The PHD filter with the underlying Poisson process approximation is used as base methodology for deriving recursive formulas for predicting and updating the positions of an unknown number of objects. Marks can be attached to the point process without making any extra theoretical effort, leaving intact the original PHD filter recursion. By taking the Poisson process approach, tractable Bayesian filtering equations are used to track and count objects with marks.

The PHD filter propagates the first-order moment or intensity function of a Poisson process, so state estimation techniques have to be used in order to calculate the individual target locations. Since a different observation noise condition is found in visual tracking than the one considered in the original formulation of the PHD filter, Chapter 3 introduces Bayesian estimation

using MCMC as a method for improving state estimation by exploiting the inherent uncertainty found in image analysis. Furthermore, the marked point process formalism offers a principled approach for object level models, without compromising either the foundations or the performance of the filter. MCMC methods are well known for escaping from local maxima, so state estimation can be more robust in cases where ambiguity makes maximum-likelihood methods unreliable. However, since Bayesian approaches using MCMC have to run several thousands iterations in order to reach an equilibrium distribution, a constrained Gibbs sampler with an informative prior was proposed. The method has the advantage of combining maximum likelihood for fast inference, but at the same time being able to perform model averaging. Because of the well known label switching problem in MCMC methods, special care must be taken when taking averages of the model components. The reparametrized version of the Gibbs sampler utilizes conjugate priors with posterior feedback, which is particularly helpful for the identifiability problem.

The prior and posterior distributions of the PHD filter are approximated as a Poisson process with independent increments. The predicted and updated number of targets are Poisson distributed having a conditional intensity function where the history is not taken into account. Even in low cluttered environments, estimates are unstable and biased to the number of observations. A principled approach for remedying that condition can be achieved by considering PHD smoothing. PHD smoothers can easily recover the number of targets when the clutter rate is not high enough that the number of observations consisting of clutter and target originated measurements gives information about the actual number of targets. In this case, low false alarms rates cause the filter to bias the estimates to the number of observations, but the FB-PHD and TF-PHD smoothers are able to provide more stable estimates. Another case was studied, in which not all targets were observed due to missed detections. In this case, the PHD filter estimates are also biased to the number of observations. However, when performing

smoothing, cardinality estimates can be recovered from the loss of information. The FB-PHD smoother compares particle clouds from different time frames and the TF-PHD smoother considers both backward dynamics and observations, both providing a robust estimation framework. The case of birth and death dynamics was also considered and proved to be a challenging case for both PHD smoothers. Not surprisingly, the performance of PHD smoothers is strongly tied to the PHD filter behavior in the forward pass. If the filter completely misses a target and there is no support in a particular region of the state space, particle smoothers cannot alleviate the errors from the forward pass. Moreover, additional backward sampling steps cannot be optimally treated in most cases, so artificial or auxiliary distributions have to be carefully designed in order to achieve consistent results. We analyzed the case of linear Gaussian dynamics where the optimal backward distribution can be evaluated using the existing particle support and no backward sampling is required, and this proved to be useful when the filter already performed a reasonably good job.

Motivated by the challenging problem of person tracking in crowded environments, Chapter 6 developed a framework for pedestrian tracking in densely populated scenes from a single camera. Traditional approaches using data association face a combinatorial explosion in the number of association hypotheses that grows with the number of persons, so the PHD filter is an appealing alternative to multi-hypothesis tracking schemes. Furthermore, the PHD filter also offers a top-down approach that can also cope with clutter from the background subtraction step. Being a deterministic approach, the EM algorithm converges to the maximum-likelihood estimate (MLE) of the observed data. However, if no person detection is performed and only area based observations of the persons locations is given, the MLE is not a good indicator of the actual locations when there is uncertainty on the number of persons inside of a blob. Bayesian state estimation using MCMC proved to be a good alternative to the EM algorithm. Instead of point estimates MCMC achieves a more robust estimation procedure by means of model averaging.

As we noted before, the performance of particle PHD smoothers is dependent on the filter behavior in the forward pass. In the case of person tracking, targets can merge or split at several times making it difficult for the filter to establish trajectories, so the filtered estimates are biased to the observed data. Nevertheless, in fixed-interval implementations where no substantial birth or death of targets occur, smoothing showed itself to be beneficial for obtaining more stable estimates.

7.1 Future Work

Bayesian non-parametric state estimation and track identity. State estimation for the particle PHD filter has traditionally been implemented using clustering techniques like K-means or the EM algorithm. K-means seems to work well enough when the SNR is high enough that particles are highly concentrated around the modes, but fails when there is more uncertainty in the estimation procedure. In those cases, the EM algorithm and the proposed Gibbs sampler provide more accurate estimates. Furthermore, Bayesian estimation using the Gibbs sampler avoid problems due to local maxima.

Another advance for state estimation could also be to consider non-parametric mixture models like the Dirichlet process (DP). Apart from a few cases like [31] and [11], the use of DP mixtures has been mostly restricted to non-dynamical systems. Applications to integrated filtering and state estimation for the particle PHD filter have not yet been reported. The application of DP techniques would potentially lead to more stable filtered estimates, since both the number of targets and state estimation are integrated into a single step. Furthermore, the same technique can also be used with no problem for obtaining smoothed estimates from the particle implementations of the FB-PHD and TF-PHD smoothers.

Marginalized particle PHD smoothing. An alternative implementation for the particle PHD filter considers marginalization via Rao-Blackwellization

[113, 94]. In this case, explicit association is performed and the resulting model is a conditionally linear-Gaussian model, so Monte Carlo sampling is only performed on the number of targets and their association variables, while estimation is performed using the standard Kalman filter equations. In the case where the linear-Gaussian assumption holds, particle PHD smoothing can be used for re-computing the number of targets and their association variables, and the RTS Kalman smoother can then be used for estimating target locations. Again, this alternative implementation holds for both the FB-PHD and the TF-PHD smoothers.

Unknown noise and parameter estimation. Since the PHD filter performance seems to be highly sensitive to the specific choice of model parameters, it would be sensible to think that PHD smoothing could be used for parameter estimation. In the case of linear dynamical models, it has been shown that Kalman filtering and smoothing are amenable to the expectation step of the EM algorithm when used for system identification [93]. Although some early papers noted the importance of the parameter estimation problem for random sets models [118], it is not clear that consistent strategies can be readily implemented by practitioners.

However, by mimicking the strategy used for linear dynamical models, we could also use smoothing techniques for estimating parameters of the particle PHD filter. In this case, the expectation of fixed parameters could also be treated as missing data alongside with the unobserved target states.

Fixed-lag estimation using divergence metrics. The choice of the optimal lag for a fixed-lag implementation of PHD smoothers has been done in an empirical way, by running the algorithm several times using different lags and monitoring convergence of the RMS and the OSPA errors. In a more general setup, the optimal lag might depend on the choice of specific parameters of filter. In the case of nonlinear state space models, the Kullback-Leiber divergence has been proposed as a method for estimating the goodness of fit

for each time lag [61]. In the case of the PHD filter and smoother, no clear metric has been devised for the purpose of model comparison, but as in the parameter estimation problem, information-theoretic criteria for random sets could be used for this purpose.

A

Appendix

In this appendix the source code of the main routines used throughout the thesis are given. Programs are implemented using the MATLAB©7.7.0.471 (R2008b) language.

A.1 Bayesian state estimation

```
function [estimates,sigmas]=bayes_estimate(n_particles,onoise,rnum_t,numit)
    [idx,mu]=kmeans(n_particles,rnum_t,'emptyaction','drop','replicates',3);
    [N,dim]=size(n_particles);
    w=ones(1,rnum_t)/rnum_t;
    sigmas=zeros(2,2,rnum_t);
    for j=1:rnum_t
        sigmas(:,:,j)=onoise*eye(dim);
    end
```

```

samp_mean=mean(n_particles);
samp_cov=cov(n_particles);
%eta=mnrnd(1,ones(N,rnum_t)/rnum_t);
b0 = mvnrnd(samp_mean, samp_cov);
B0 = drawWishart(inv(samp_cov)/dim,dim);
c0 = 1/gamrnd(1,1/dim) + dim-1;
C0 = drawWishart(samp_cov/dim,dim);
alpha0=gamrnd(1,1);
prob=zeros(N,rnum_t);
eta=zeros(N,rnum_t);
model_average=zeros(rnum_t,dim,numit);
for i=1:numit
    % E-step
    for j=1:rnum_t
        prob(:,j)=
            w(j)*normalLike(n_particles,mu(j,:),sigmas(:,:,j));
    end
    prob=prob./repmat(sum(prob,2),1,rnum_t);
    eta=mnrnd(1,prob);
    % M-Step
    nu=sum(eta,1);
    w=1./gamrnd(alpha0+nu,1);
    w=w./sum(w);
    bar=(eta'*n_particles)./(repmat(nu',1,dim));
    for j=1:rnum_t
        A = inv(nu(j)*inv(sigmas(:,:,j)) + B0);
        [L p]=chol(A);
        if (p==0 & ~any(isnan(A)))
            Bk=A;
        else
            Bk=onoise*eye(2);
        end
        bk = (nu(j)*bar(j,:)*inv(sigmas(:,:,j)) + b0*B0)*Bk;
    end
end

```



```

        mu(j,:) = mvnrnd(bk, Bk);
    end
    model_average(:, :, i) = mu;
end
estimates = mean(model_average, 3);
sigmas = sigmas;
end

```

A.2 Forward-Backward smoothing source code

```

function [estimates, num_t, weights_s, particles] =
FBSmoothen(weights, s_weights, particles, p_particles, F,
pnoise, onoise, survival_rate, birth_rate)
    % Fixed-interval PHD smoother
    state_dim = 2;
    Lk = size(particles, 1);
    Lk1 = size(p_particles, 1);
    weights_s = repmat(NaN, 1, Lk);
    const = repmat(NaN, 1, Lk1);
    survival = repmat(survival_rate, 1, Lk);
    birth = repmat(birth_rate, 1, Lk1);
    weights_s = phd_forwardlikelihood(p_particles, particles,
weights, s_weights, pnoise, survival, birth);
    num_t = sum(weights_s);
    rnum_t = round(num_t);
    weights_s = weights_s ./ num_t;
    index = randsample(1:Lk, Lk, true, weights_s);
    particles = particles(index, :);
    weights_s = ones(1, Lk) * rnum_t ./ Lk;
    if num_t > 1
        estimates = state_estimate(particles, onoise, rnum_t, 'bayes');
    else
        if rnum_t == 1

```

```

        estimates(1).k=1;
            estimates(1).means=sum(particles(:,1))/Lk;
        estimates(1).sigmas=onoise*diag(state_dim);
    else
        estimates=[];
    end
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function weights_s=
phd_forwardlikelihood(p_particles,particles,weights,
s_weights,pnoise,survival,birth)
    Lk=size(particles,1);
    Lk1=size(p_particles,1);
    Y= repmat(p_particles(:,1)',Lk,1);
    X= repmat(particles(:,1),1,Lk1);
    lik_mat=normpdf(X,Y,pnoise(1));
    survival_mat= repmat(survival',1,Lk1);
    weights_mat= repmat(weights',1,Lk1);
    sweights_mat= repmat(s_weights,Lk,1);
        wlik_mat=sweights_mat.*(survival_mat.*lik_mat);
    wlik2_mat=weights_mat.*(survival_mat.*lik_mat);
    norm=sum(wlik2_mat,1);
    const_mat= repmat((birth+norm),Lk,1);
    marg_mat=wlik_mat./const_mat;
    weights_s=weights.*(sum(marg_mat,2)');
end

```

A.3 Two-Filter smoothing source code

```

function [estimates,num_t,weights_s,particles]=
TFSmoothing(observations,weights,s_weights,particles,p_particles,F,H,pnoise,onoise,

```

```

survival_rate,detection_rate,birth_rate,rate_c,mcmc_steps)
    Lk=size(particles,1);
    Lk1=size(p_particles,1);
    numobs=size(observations,1);
    weights_s= repmat(0,1,Lk);
    %clutter=repmat(rate_c,1,numobs);
    clutter=repmat(rate_c,1,numobs);
    survival=repmat(survival_rate,1,Lk);
    birth=repmat(birth_rate,1,Lk1);
    detection=repmat(detection_rate,1,Lk);

    % compute likelihood
    psi=phd_likelihood(observations,H,particles,onoise,clutter,detection);
    % backward MCMC step
    mcmc_particles=repmat(NaN,Lk,2);
    for s=1:mcmc_steps
        mcmc_particles=particles+repmat([1,.1],Lk,1).*rand(Lk,2);
        mcmc_psi=phd_likelihood(observations,H,
            mcmc_particles,onoise,clutter,detection);
        alpha=min(1,mcmc_psi./psi);
        index1=binornd(1,alpha);
        %size(index1)
        index2=mcmc_psi>psi;
        particles(index2,:)=mcmc_particles(index2,:);
        psi(index2)=mcmc_psi(index2);
    end
    weights_s=phd_forwardlikelihood(p_particles,particles,psi,
        birth,pnoise,weights_s_weights,survival);
    num_t=sum(weights_s)
    rnum_t=round(num_t)
    weights_s=weights_s./num_t;
    if isnan(num_t)
        num_t=0;
    end

```

```
phd_forwardlikelihood(p_particles,particles,psi_mat,
birth,pnoise,weights,s_weights,survival,detection)
    Lk=size(particles,1);
    Lk1=size(p_particles,1);
    Y= repmat(p_particles(:,1)',Lk,1);
    X= repmat(particles(:,1),1,Lk1);
    lik_mat=normpdf(Y,X,pnoise(1));
    weights_mat= repmat(s_weights,Lk,1);
    weights_mat2= repmat(weights',1,Lk1);
    wlik_mat=psi_mat*lik_mat;
    norm=sum(wlik_mat,1);
    const_mat= repmat(norm,Lk,1);
    marg_mat=weights_mat.*lik_mat./const_mat;
    weights_s=(sum(psi_mat',2).*sum(marg_mat,2))';
end
```

Bibliography

- [1] ANDERSON, B. D. O., AND MOORE, J. B. *Optimal filtering*. Prentice-Hall, New Jersey, 1979.
- [2] AUMANN, R. J. Integrals of set-valued functions. *Journal of Mathematical Analysis and Applications* 12, 1 (1965), 1–12.
- [3] BADDELEY, A. J. Stochastic geometry models in high-level vision. *Journal of Applied Statistics* 20, 5&6 (1993), 231–256.
- [4] BAIN, A., AND CRISAN, D. *Fundamentals of Stochastic Filtering*. Springer-Verlag New York, Inc., 2008.
- [5] BAR-SHALOM, Y., LI, X. R., AND KIRUBARAJAN, T. *Estimation with applications to tracking and navigation*. Wiley Interscience, New York, 2001.
- [6] BISHOP, C. M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
- [7] BLACKMAN, S., AND POPOLI, R. *Design and Analysis of Modern Tracking Systems*. Artech House, Norwood, 1999.
- [8] BRADSKI, G., AND KAEHLER, A. *Learning OpenCV*. O’Reilly Media Inc., 2008.

-
- [9] BRIERS, M., DOUCET, A., AND MASKELL, S. Smoothing algorithms for state–space models. *Annals of the Institute of Statistical Mathematics* 62, 1 (Feb. 2010), 61–89.
- [10] CAPPE, O., MOULINES, E., AND RYDEN, T. *Inference in Hidden Markov Models*. Springer series in statistics. Springer, 2005.
- [11] CARON, F., DAVY, M., DOUCET, A., DUFLOS, E., AND VANHEEGHE, P. Bayesian inference for linear dynamic models with Dirichlet process mixtures. *Signal Processing, IEEE Transactions on* 56, 1 (Jan. 2008), 71–84.
- [12] CHAN, A., LIANG, Z.-S., AND VASCONCELOS, N. Privacy preserving crowd monitoring: Counting people without people models or tracking. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* (June 2008), pp. 1–7.
- [13] CHAN, A., AND VASCONCELOS, N. UCSD pedestrian dataset, 2010. <http://www.svcl.ucsd.edu/projects/peoplecnt/>.
- [14] CLARK, D., AND BELL, J. Convergence results for the particle PHD filter. *IEEE Transactions on Signal Processing* 54, 7 (July 2006), 2652–2661.
- [15] CLARK, D., AND BELL, J. Multi-target state estimation and track continuity for the particle phd filter. *Aerospace and Electronic Systems, IEEE Transactions on* 43, 4 (October 2007), 1441–1453.
- [16] COX, D., AND ISHAM, V. *Point processes*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Chapman & Hall, 1980.
- [17] DALAL, N., AND TRIGGS, B. Histograms of oriented gradients for human detection. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*

- (*CVPR'05*) - *Volume 1* (Washington, DC, USA, 2005), IEEE Computer Society, pp. 886–893.
- [18] DALEY, D., AND VERE-JONES, D. *An Introduction to the Theory of Point Processes. Volume I: Elementary Theory and Methods*. Springer, New York, 2003.
- [19] DAUM, F., AND HUANG, J. Curse of dimensionality and particle filters. In *Aerospace Conference, 2003. Proceedings. 2003 IEEE*, vol. 4.
- [20] DAVIES, A., YIN, J. H., AND VELASTIN, S. Crowd monitoring using image processing. *Electronics & Communication Engineering Journal* 7, 1 (Feb 1995), 37–47.
- [21] DESCOMBES, X., AND ZERUBIA, J. Marked point process in image analysis. *Signal Processing Magazine, IEEE* 19, 5 (Sep 2002), 77–84.
- [22] DOUC, R., AND CAPPE, O. Comparison of resampling schemes for particle filtering. In *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on* (Sept. 2005), pp. 64–69.
- [23] DOUCET, A., GODSILL, S., AND ANDRIEU, C. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing* 10, 3 (July 2000), 197–208.
- [24] DRUMMOND, O. E., AND FRIDLING, B. E. Ambiguities in evaluating performance of multiple target tracking algorithms. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series* (Aug. 1992), O. E. Drummond, Ed., vol. 1698 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, pp. 326–337.
- [25] ERDINC, O., WILLETT, P., AND BAR-SHALOM, Y. A physical-space approach for the probability hypothesis density and cardinalized prob-

- ability hypothesis density filters. O. E. Drummond, Ed., vol. 6236, SPIE, p. 623619.
- [26] ERDINC, O., WILLETT, P., AND BAR-SHALOM, Y. The bin-occupancy filter and its connection to the PHD filters. *Signal Processing, IEEE Transactions on* (2009).
- [27] FEARNHEAD, P., WYNOLL, D., AND TAWN, J. A sequential smoothing algorithm with linear computational cost. University of Lancaster Eprints, May 2008.
- [28] FERGUSON, T. S. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics* 1, 2 (1973), 209–230.
- [29] FORSYTH, D. A., AND PONCE, J. *Computer Vision: A Modern Approach*. Prentice Hall Professional Technical Reference, 2002.
- [30] FORTMANN, T., BAR-SHALOM, Y., AND SCHEFFE, M. Sonar tracking of multiple targets using joint probabilistic data association. *Oceanic Engineering, IEEE Journal of* 8, 3 (1983), 173–184.
- [31] FOX, E. B., CHOI, D. S., AND WILLSKY, A. S. Nonparametric Bayesian methods for large scale multi-target tracking. In *Proc. Asilomar Conf. Signals, Systems, and Computers* (November 2006), IEEE.
- [32] FRASER, D., AND POTTER, J. The optimum linear smoother as a combination of two optimum linear filters. *IEEE Transactions on Automatic Control* 14, 4 (August 1969), 387–390.
- [33] FUENTES, L. M., AND VELASTIN, S. Tracking people for automatic surveillance applications. In *Pattern Recognition and Image Analysis* (2003), pp. 238–245.
- [34] GILKS, W. R., AND RICHARDSON, S. *Markov Chain Monte Carlo in Practice*. Chapman Hall, New York, 1996.

-
- [35] GODSILL, S., DOUCET, A., AND WEST, M. Monte Carlo smoothing for nonlinear time series. *Journal of the American Statistical Association* 99, 465 (March 2004), 156–168.
- [36] GONZALEZ, R. C., AND WOODS, R. E. *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.
- [37] GOODMAN, I., MAHLER, R., AND NGUYEN, H. T. *Mathematics of Data Fusion*. Kluwer Academic Publishers, Norwell, MA, USA, 1997.
- [38] GORDON, N., SALMOND, D. J., AND SMITH, A. F. M. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEEE Proceedings* 2, 140 (January 1993), 107–113.
- [39] GREEN, P. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika* 82, 4 (1995), 711–732.
- [40] HAYKIN, S., Ed. *Kalman Filtering and Neural Networks*. Wiley Interscience, 2003.
- [41] HERNANDEZ, S. Population-based Monte Carlo. In *Proc. New Zealand Computer Science Research Student Conference* (2007).
- [42] HERNANDEZ, S. On line Bayesian tracking and detection of multiple objects. In *Proc. New Zealand Computer Science Research Student Conference* (2008).
- [43] HERNANDEZ, S., AND TEAL, P. Multi-target tracking with Poisson processes observations. In *Advances in Image and Video Technology* (2007).
- [44] HERNANDEZ, S., TEAL, P., AND FREAN, M. Two filter probability hypothesis density smoothing. In *Proc. Jornadas Chilenas de la Computacion* (2009).

- [45] HO, Y., AND LEE, R. A Bayesian approach to problems in stochastic estimation and control. *IEEE Transactions on Automatic Control* 9, 4 (October 1964), 333–339.
- [46] HOFFMAN, J., AND MAHLER, R. Multitarget miss distance via optimal assignment. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* 34, 3 (2004), 327–336.
- [47] HORN, B. K. P. *Robot vision*. MIT Press, Cambridge, MA, USA, 1986.
- [48] HUE, C., LE CADRE, J., AND PEREZ, P. Sequential Monte Carlo methods for multiple target tracking and data fusion. *IEEE Trans. on Signal Processing* 2, 50 (February 2002), 309–325.
- [49] HUTTENLOCHER, D., KLANDERMAN, G., AND RUCKLIDGE, W. Comparing images using the Hausdorff distance. *Transactions on Pattern Analysis and Machine Intelligence* 15, 9 (September 1993), 850–863.
- [50] IKOMA, N., UCHINO, T., AND MAEDA, H. Tracking of feature points in image sequence by SMC implementation of PHD filter. In *SICE 2004 Annual Conference* (2004).
- [51] ILLIAN, J., PENTTINEN, A., STOYAN, H., AND STOYAN, D. *Statistical Analysis and Modelling of Spatial Point Patterns*. Wiley, 2008.
- [52] ISARD, M., AND BLAKE, A. A smoothing filter for CONDENSATION. In *Proceedings of the 5th European Conference on Computer Vision* (1998), vol. 1406, Springer-Verlag, pp. 767–781.
- [53] ISARD, M., AND MACCORMICK, J. BraMBLe: a Bayesian multiple-blob tracker. *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on 2* (2001), 34–41.

-
- [54] JAZWINSKI, A. *Stochastic Processes and Filtering Theory*. Academic Press, New York, 1970.
- [55] JOHANSEN, A., SINGH, S., DOUCET, A., AND VO, B.-N. Convergence of the smc implementation of the PHD filter. *Methodology and Computing in Applied Probability* 8, 2 (06 2008), 265–291.
- [56] JOO, S.-W., AND CHELLAPPA, R. A multiple-hypothesis approach for multiobject visual tracking. *Image Processing, IEEE Transactions on* 16, 11 (Nov. 2007), 2849–2854.
- [57] KAILATH, T., SAYED, A. H., AND HASSIBI, B. *Linear estimation*. Prentice-Hall, 2000.
- [58] KALMAN, R. A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering* 82, Series D (1960), 35–45.
- [59] KHAN, Z., BALCH, T., AND DELLAERT, F. MCMC-based particle filtering for tracking a variable number of interacting targets. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27, 11 (2005), 1805–1918.
- [60] KILAMBI, P., RIBNICK, E., JOSHI, A. J., MASOUD, O., AND PAPANIKOLOPOULOS, N. Estimating pedestrian counts in groups. *Computer Vision and Image Understanding* 110, 1 (2008), 43–59.
- [61] KITAGAWA, G. Monte Carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics* 1, 5 (January 1996), 1–25.
- [62] LANZ, O. Approximate bayesian multibody tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 28, 9 (Sept. 2006), 1436–1449.

- [63] LIU, W., HAN, C., LIAN, F., XU, X., AND WEN, C. Multitarget state and track estimation for the probability hypotheses density filter. *Journal of Electronics (China)* 26, 1 (01 2009), 2–12.
- [64] LO, B., AND VELASTIN, S. Automatic congestion detection system for underground platforms. In *Intelligent Multimedia, Video and Speech Processing, 2001. Proceedings of 2001 International Symposium on* (2001), pp. 158–161.
- [65] MAGGIO, E., AND CAVALLARO, A. Learning scene context for multiple object tracking. *Image Processing, IEEE Transactions on* 18, 8 (Aug. 2009), 1873–1884.
- [66] MAGGIO, E., PICCARDO, E., REGAZZONI, C., AND CAVALLARO, A. Particle PHD filtering for multi-target visual tracking. In *IEEE International Conference on Acoustics, Speech and Signal Processing, 2007. ICASSP 2007.* (2007).
- [67] MAHLER, R. *Handbook of multi-sensor data fusion*. CRC Press, London, 2001, ch. Random Set Theory for Target Tracking and Identification.
- [68] MAHLER, R. Multitarget Bayes filtering via first-order multitarget moments. *IEEE Transactions on Aerospace and Electronic Systems* 39, 4 (October 2003), 1152–1178.
- [69] MAHLER, R. PHD filters of higher order in target number. *Aerospace and Electronic Systems, IEEE Transactions on* 43, 4 (October 2007), 1523–1543.
- [70] MAHLER, R. *Statistical Multisource-Multitarget Information Fusion*. Artech House, Norwood, 2007.
- [71] MAYNE, D. Q. A solution of the smoothing problem for linear dynamic systems. *Automatica* 4 (1966), 73–92.

- [72] MCLACHLAN, G., AND PEEL, D. *Finite Mixture Models*. Wiley Series in Probability and Statistics. Wiley-Interscience, October 2000.
- [73] METROPOLIS, N., AND ULAM, S. The Monte Carlo method. *Journal of the American Statistical Association* 44, 247 (1949), 335–341.
- [74] MILLER, M., SRIVASTAVA, A., AND GRENANDER, U. Conditional-mean estimation via jump-diffusion processes in multiple target tracking/recognition. *Signal Processing, IEEE Transactions on* 43, 11 (Nov 1995), 2678–2690.
- [75] MOLCHANOV, I. *Theory of Random Sets*. Springer-Verlag, 2005.
- [76] MØLLER, J., AND WAAGEPETERSEN, R. P. *Statistical inference and simulation for spatial point processes*, vol. 100 of *Monographs on Statistics and Applied Probability*. Chapman & Hall/CRC, Boca Raton, FL, 2004.
- [77] MORI, S., AND CHONG, C.-Y. Quantitative evaluation of Poisson point process approximation. In *Proc. 4th ISIF International Conference on Information Fusion* (2001).
- [78] NANDAKUMARAN, N., KIRUBARAJAN, T., LANG, T., MACDONALD, M., AND PUNITHAKUMAR, K. Multitarget tracking using probability hypothesis density smoothing. submitted as personal communication, December 2009.
- [79] NANDAKUMARAN, N., PUNITHAKUMAR, K., AND KIRUBARAJAN, T. Improved multi-target tracking using probability hypothesis density smoothing. In *Signal and Data Processing of Small Targets 2007* (2007).
- [80] PANTA, K., VO, B.-N., SINGH, S., AND DOUCET, A. Probability hypothesis density filter versus multiple hypothesis tracking. I. Kadar, Ed., vol. 5429, SPIE, pp. 284–295.

-
- [81] PARAGIOS, N., CHEN, Y., AND FAUGERAS, O. *Handbook of Mathematical Models in Computer Vision*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.
- [82] PHAM, N., HUANG, W., AND ONG, S. Probability hypothesis density approach for multi-camera multi-object tracking. *Computer Vision – ACCV 2007* (2007), 875–884.
- [83] PHAM, N. T., HUANG, W., AND ONG, S. Tracking multiple objects using probability hypothesis density filter and color measurements. In *Multimedia and Expo, 2007 IEEE International Conference on* (July 2007), pp. 1511–1514.
- [84] PICCARDI, M. Background subtraction techniques: a review. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on* (Oct. 2004), vol. 4, pp. 3099–3104 vol.4.
- [85] PITT, M., AND SHEPHARD, N. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association* 94, 446 (June 1999), 590–599.
- [86] PRESTON, C. Spatial birth-and-death processes. *Bull. Internat. Statist. Inst.* 46, 2 (1977), 371—391.
- [87] RAUCH, H. Solutions to the linear smoothing problem. *Automatic Control, IEEE Transactions on* 8, 4 (Oct 1963), 371–372.
- [88] RAUCH, H. E., STRIEBEL, C. T., AND TUNG, F. Maximum likelihood estimates of linear dynamic systems. *AIAA JOURNAL* 3, 8 (1965), 1445–1450.
- [89] RIPLEY, B. D. Locally finite random sets: Foundations for point process theory. *The Annals of Probability* 4, 6 (December 1976), 983–994.

- [90] RISTIC, B., ARULAMPALAM, S., AND GORDON, N. *Beyond the Kalman filter. Particle filters for tracking applications*. Artech House, London, 2004.
- [91] ROBERT, C. *The Bayesian Choice*. Springer-Verlag, 2007.
- [92] ROBERT, C. P., AND CASELLA, G. *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.
- [93] ROWEIS, S., AND GHAHRAMANI, Z. A unifying review of linear gaussian models. *Neural Computation* 11, 2 (1999), 305–345.
- [94] SÄRKKÄ, S., VEHTARI, A., AND LAMPINEN, J. Rao-blackwellized particle filter for multiple target tracking. *Information Fusion* 8, 1 (2007), 2–15. Special Issue on the Seventh International Conference on Information Fusion-Part II, Seventh International Conference on Information Fusion.
- [95] SCHUHMACHER, D., VO, B.-T., AND VO, B.-N. A consistent metric for performance evaluation of multi-object filters. *Signal Processing, IEEE Transactions on* 56, 8 (Aug. 2008), 3447–3457.
- [96] SIDENBLADH, H. Multi-target particle filtering for the probability hypothesis density. In *Proceedings of the Sixth International Conference of Information Fusion* (2003).
- [97] SINGH, S. S., VO, B.-N., BADDELEY, A., AND ZUYEV, S. Filters for spatial point processes. *SIAM Journal on Control and Optimization* 48, 4 (2009), 2275–2295.
- [98] SMITH, K., GATICA-PEREZ, D., AND ODOBEZ, J.-M. Using particles to track varying numbers of interacting people. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on* (June 2005), vol. 1, pp. 962–969 vol. 1.

-
- [99] SNYDER, D. L., AND MILLER, M. I. *Random point processes in time and space*. Springer-Verlag, New York, 2001.
- [100] SONKA, M., HLAVAC, V., AND BOYLE, R. *Image processing, analysis and machine vision*. Thomson-Engineering, 2007.
- [101] STAUFFER, C., AND GRIMSON, W. Learning patterns of activity using real-time tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 22, 8 (Aug 2000), 747–757.
- [102] STONE, L. D., CORWIN, T. L., AND BARLOW, C. A. *Bayesian Multiple Target Tracking*. Artech House, Inc., Norwood, MA, USA, 1999.
- [103] STOYAN, D., KENDALL, W. S., AND MECKE, J. *Stochastic Geometry And Its Applications*. John Wiley & Sons Inc, 1995.
- [104] STREIT, R. PHD intensity filtering is one step of a map estimation algorithm for positron emission tomography. In *Information Fusion, 2009. FUSION '09. 12th International Conference on* (July 2009), pp. 308–315.
- [105] STREIT, R., AND STONE, L. Bayes derivation of multitarget intensity filters. In *Information Fusion, 2008 11th International Conference on* (30 2008-July 3 2008), pp. 1–8.
- [106] TAJ, M., MAGGIO, E., AND CAVALLARO, A. Objective evaluation of pedestrian and vehicle tracking on the clear surveillance dataset. *Multimodal Technologies for Perception of Humans* (2008), 160–173.
- [107] TOBIAS, M., AND LANTERMAN, A. Techniques for birth-particle placement in the probability hypothesis density particle filter applied to passive radar. *Radar, Sonar & Navigation, IET* 2, 5 (October 2008), 351–365.

-
- [108] UHLMANN, J. K. *Multisensor Data Fusion*. CRC Press, 2001, ch. Introduction to the Algorithmics of Data Association in Multiple-Target Tracking.
- [109] ULMKE, M., FRANKEN, D., AND SCHMIDT, M. Missed detection problems in the cardinalized probability hypothesis density filter. In *Information Fusion, 2008 11th International Conference on* (2008).
- [110] VAN TREES, H. L. *Detection, Estimation, and Modulation Theory (Part I)*. Wiley-Interscience, New York, 2001.
- [111] VELASTIN, S., AND REMAGNINO, P. *Intelligent Distributed Video Surveillance Systems (Professional Applications of Computing) (Professional Applications of Computing)*. Institution of Engineering and Technology, 2006.
- [112] VERMAAK, J., DOUCET, A., AND PEREZ, P. Maintaining multimodality through mixture tracking. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on* (Oct. 2003), pp. 1110–1116 vol.2.
- [113] VIHOLA, M. Rao-blackwellised particle filtering in random set multitarget tracking. *Aerospace and Electronic Systems, IEEE Transactions on* 43, 2 (April 2007), 689–705.
- [114] VO, B., AND MA, W. The Gaussian mixture probability hypothesis density filter. *IEEE Transactions on Signal Processing* 54, 11 (2006).
- [115] VO, B., AND SINGH, S. On the Bayes filtering equations of finite set statistics. In *5th Asian Control Conference* (2004), vol. 2.
- [116] VO, B., SINGH, S., AND DOUCET, A. Sequential Monte Carlo methods for multitarget filtering with random finite sets. *IEEE Transactions on Aerospace and Electronic Systems* 41, 4 (October 2005), 1224–1245.

- [117] VO, B.-N., VO, B.-T., PHAM, N. T., AND SUTER, D. Bayesian multi-object estimation from image observations. In *Information Fusion, 2009. FUSION '09. 12th International Conference on* (July 2009), pp. 890–898.
- [118] VO, B.-N., VO, B.-T., AND SINGH, S. Sequential monte carlo methods for static parameter estimation in random set models. In *Intelligent Sensors, Sensor Networks and Information Processing Conference, 2004. Proceedings of the 2004* (Dec. 2004), pp. 313–318.
- [119] VO, B. T. *Random Finite Sets in Multi-Object Filtering*. PhD thesis, School of Electrical, Electronic and Computer Engineering. The University of Western Australia, 2009.
- [120] VO, B.-T., VO, B.-N., AND CANTONI, A. Analytic implementations of the cardinalized probability hypothesis density filter. *Signal Processing, IEEE Transactions on* 55, 7 (July 2007), 3553–3567.
- [121] WANG, Y., WU, J., KASSIM, A., AND HUANG, W. Tracking a variable number of human groups in video using probability hypothesis density. In *ICPR 2006. 18th International Conference on Pattern Recognition, 2006.* (2006).
- [122] WANG, Y.-D., WU, J.-K., HUANG, W., AND KASSIM, A. Gaussian mixture probability hypothesis density for visual people racking. In *Information Fusion, 2007 10th International Conference on* (July 2007), pp. 1–6.
- [123] WANG, Y.-D., WU, J.-K., KASSIM, A., AND HUANG, W. Data-driven probability hypothesis density filter for visual tracking. *Circuits and Systems for Video Technology, IEEE Transactions on* 18, 8 (Aug. 2008), 1085–1095.
- [124] WASHBURN, R. A random point process approach to multiobject tracking. In *Proceedings of the American Control Conference* (1987).

-
- [125] WREN, C. R., AZARBAYEJANI, A., DARRELL, T., AND PENTLAND, A. P. Pfunder: Real-time tracking of the human body. *IEEE Trans. Pattern Anal. Mach. Intell.* 19 (July 1997), 780–785.
- [126] YILMAZ, A., JAVED, O., AND SHAH, M. Object tracking: A survey. *ACM Comput. Surv.* 38, 4 (2006), 13.
- [127] ZAJIC, T., AND MAHLER, R. P. S. Particle-systems implementation of the PHD multitarget-tracking filter. *Signal Processing, Sensor Fusion, and Target Recognition XII* 5096, 1 (2003), 291–299.
- [128] ZHAN, B., MONEKOSSO, D. N., REMAGNINO, P., VELASTIN, S. A., AND XU, L.-Q. Crowd analysis: a survey. *Mach. Vision Appl.* 19, 5-6 (2008), 345–357.
- [129] ZHAO, T., AND NEVATIA, R. Bayesian human segmentation in crowded situations. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on* (June 2003), vol. 2, pp. II–459–66 vol.2.
- [130] ZHENG, N., AND XUE, J. *Statistical Learning and Pattern Analysis for Image and Video Processing*. Springer-Verlag, 2009, ch. Multi-Target Tracking in Video –Part II, pp. 319–341.