Random Finite Sets for Multitarget Tracking with Applications

Trevor M. Wood
St Anne's College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

Michaelmas Term 2011

This thesis is dedicated to my Mum and Dad whose unwavering support over the years has made my work possible.

Acknowledgements

Firstly, I would like to thank my academic supervisors: Philip Bond, Irene Moroz and David Allwright for their guidance and insight. I am very grateful to both Thales Underwater systems and Thales Aerospace for their financial support of this project, but more importantly for the useful discussions that I have had with Stephen Long, Roger Benton and Glen Davidson which have greatly improved this thesis. I am also grateful to Thales Underwater Systems for making real sonar data available which has been invaluable. Thanks must also be given to EPSRC for their role in funding my studentship.

I would like to thank the people I have been fortunate to collaborate with over the course of this project. Many thanks to the Bacteria Boys: Christian Yates, David Wilkinson and Gabriel Rosser. Our collaborative project has been the most fun part of my DPhil. Cheers, guys. Thanks also to Daniel Clark and his research group at Heriot-Watt University. My visit to Edinburgh was of great help and I look forward to seeing the fruits of your work. Finally, thanks to Branko Ristic and Ba-Ngu Vo. Your advice in our email correspondence has been valuable.

I have received good advice regarding my work from many sources over the past three years. I would like to thank Chris Farmer, Charles Offer and David Allwright who grilled me in intermediate examinations and internal seminars and always provided constructive criticism that eventually improved my work. The same may also be said of the anonymous reviewers of the papers relating to this work, so thanks should also go to them.

Finally, and most importantly, I would like to thank those friends and family who I have relied on during my time in Oxford for the past eight years. Thanks to my parents for their inexhaustible supply of love and support. Thanks to my wife(!) Helen for her love, strength and good humour. I couldn't have done this without her. Thanks to Craig and Nan who have always been there for me. Thanks to Chris, for his attentive proof-reading and for his sense of humour in the office.

Abstract

Random Finite Sets for Multitarget Tracking with Applications

Trevor M. Wood, St. Anne's College

Submitted towards *Doctor of Philosophy, Michaelmas Term 2011*

Multitarget tracking is the process of jointly determining the number of targets present and their states from noisy sets of measurements. The difficulty of the multitarget tracking problem is that the number of targets present can change as targets appear and disappear while the sets of measurements may contain false alarms and measurements of true targets may be missed. The theory of random finite sets was proposed as a systematic, Bayesian approach to solving the multitarget tracking problem. The conceptual solution is given by Bayes filtering for the probability distribution of the set of target states, conditioned on the sets of measurements received, known as the multitarget Bayes filter. A first-moment approximation to this filter, the probability hypothesis density (PHD) filter, provides a more computationally practical, but theoretically sound, solution.

The central thesis of this work is that the random finite set framework is theoretically sound, compatible with the Bayesian methodology and amenable to immediate implementation in a wide range of contexts. In advancing this thesis, new links between the PHD filter and existing Bayesian approaches for manoeuvre handling and incorporation of target amplitude information are presented. A new multitarget metric which permits incorporation of target confidence information is derived and new algorithms are developed which facilitate sequential Monte Carlo implementations of the PHD filter.

Several applications of the PHD filter are presented, with a focus on applications for tracking in sonar data. Good results are presented for implementations on real active and passive sonar data. The PHD filter is also deployed in order to extract bacterial trajectories from microscopic visual data in order to aid ongoing work in understanding bacterial chemotaxis. A performance comparison between the PHD filter and conventional multitarget tracking methods using simulated data is also presented, showing favourable results for the PHD filter.

List Of Abbreviations

CRLB - Cramer-Rao Lower Bound
EKF - Extended Kalman Filter
GM - Gaussian Mixture
GM-PHD - Gaussian Mixture Probability Hypothesis Density
HMM - Hidden Markov Model
JMLS - Jump Markov Linear Systems
IMM - Interacting Multiple Models
JPDA - Joint Probabilistic Data Association
MHT - Multiple Hypothesis Tracking
MoE - Measure of Effectiveness
NN - Nearest Neighbour
OSPA - Optimal Sub-pattern Assignment
PDA - Probabilistic Data Association
PHD - Probability Hypothesis Density
SMC - Sequential Monte Carlo
SMC-PHD - Sequential Monte Carlo Probability Hypothesis Density
SNR - Signal-to-Noise Ratio
SPRT - Sequential Probability Ratio Test
TBD - Track Before Detect
UKF - Unscented Kalman Filter

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation and Scope

Among the first practitioners of sonar were marine mammals like dolphins and sperm whales which use the technique of echolocation for navigation and hunting. Outside of the water, the same trick is used by numerous species of microbat which are able to perform the impressive feat of simultaneously navigating, detecting and catching mobile prey while flying. The first recorded human use of sonar was not until 1490 when Leonardo Da Vinci is said to have placed a tube into the water in order to hear ships large distances away. The latter is an example of passive sonar which involves listening without transmitting. Active sonar, which involves transmitting, as in echolocation, was first proposed by humans in the early 20th century as a means of locating the wreck of the Titanic. Shortly after the sinking of that vessel in 1912, the First World War broke out and military applications of sonar began to be investigated. Sonar is used to this day to detect submarines and other vessels. The process of detecting and tracking submarines in sonar data, and targets in more general data, will be the motivation for this thesis.

Let us stay, for the moment, with the example of submarine tracking in sonar data due to its intuitive appeal. In active sonar, a sound pulse, or 'ping', is sent out from an emitter, bounces off anything it encounters and returns to a receiver. For each point in the region of surveillance an amplitude is calculated based on the magnitude of the returning signal using the technique of beamforming [119]. Peaks in this amplitude data are extracted as target detections. It is with the measurements due to these detections that the majority of this work will be concerned.

In single target tracking, for each time step, k, a measurement $\mathbf{z}_k$ is received due to a target with unknown state $\mathbf{x}_k$. The measurements may be corrupted by noise. The goal is to infer the unknown state of the target, $\mathbf{x}_k$, conditioned on the measurements,

$\mathbf{z}_k$, received at each time step k . When the measurements are noisy, no unambiguous 'true' state may be discerned. The Bayesian solution to this problem is to compute, at each time step, the posterior density for the target state, conditioned on all of the measurements received up to that time step: $f(\mathbf{x}_k|\mathbf{z}_1, ..., \mathbf{z}_k)$. The calculation of these posterior densities is known as the Bayes filter. The Bayesian approach is the de facto standard methodology for single target tracking. The Kalman filter, developed in 1960 by Rudolf E. Kalman [58] provides an optimal solution to the Bayes filter in the special case when all noise is Gaussian and all forward and measurement models are linear. Finding tractable, sufficiently accurate approximations to the Bayes filter remains an active area of research which has thus far provided the Extended Kalman filter [2], Unscented Kalman filter [57], Gaussian Sum filter [111], the particle filter and the latter's many variants [5].

Unfortunately the realities of tracking are often less simple. Multitarget tracking is a generalisation of the above problem and is the problem many practical tracking systems must deal with. In multitarget tracking, a set of indistinguishable measurements is extracted at each time step. Some of these may be false alarms, not due to any target. Furthermore, detections of actual targets may be missed. In multitarget tracking, the goal is to infer how many targets are present and their states using the sets of detections at each time step.

Conventional approaches to this problem are based on data association: associating a particular measurement with each target present. However, it will be shown that there are theoretical problems with this approach. In the last decade, a new Bayesian approach, called random finite set theory, has been attracting attention. The random finite set approach, largely due to Mahler [75], involves treating the set of states for targets present at time k as a finite set, denoted X_k, with an unknown number of members. The problem may be viewed as jointly determining the number of elements in the set, and its members. In random finite set theory, however, the goal is to infer a posterior density for the set X_k conditioned on the sets of measurements received, $Z_1, ..., Z_k$, at each time step. Note that the posterior density is over the space of finite sets. This approach permits a rigorous Bayesian treatment of the multitarget tracking problem.

The central thesis of the present work is that the random finite set framework is theoretically sound, compatible with the Bayesian methodology and amenable to immediate implementation in a wide range of contexts, particularly those relating to sonar tracking. In advancing this thesis, the following are presented:

- The background of random finite set theory, including an elaboration of the link with point process theory and the derivation of the multitarget Bayes filter and the probability hypothesis density (PHD) filter along with details of implementation.
- New links between the PHD filter implementations and existing Bayesian approaches to manoeuvre handling and incorporation of target amplitude information.
- Development of new techniques for the sequential Monte Carlo implementation of the PHD filter.
- A new multitarget metric incorporating target confidence information.
- Applications of the PHD filter in real active and passive sonar data showing good results by comparison with existing methods.
- New applications of the PHD filter for tracking cell trajectories in microscopic visual data.

Additionally, single target tracking methods are presented and evaluated in a Bayesian manner, unlike conventional filter evaluation. This methodology suggests an alternative method for target prior selection which is shown to improve results.

1.2 Organisation of the Thesis

The work in *Chapter 2* concerns the single target tracking problem. In Section 2.1, there is a presentation of the Bayes filter and its relationship with the Kalman filter along with a review of the most popular approximate methods for single target filtering. In Section 2.2, the single target approximate filters are evaluated by comparison with the 'correct' answer: the Bayes filter, computed here using an adaptive grid. An alternative method for choosing single target priors is given in Section 2.3, while Section 2.4 summarises the method of using jump Markov linear systems (JMLS) to model target manoeuvres.

Chapters 3 and 4 contain the background material relating to multitarget tracking using random finite sets. A summary of the conventional approaches to multitarget tracking is given in Section 3.1 along with a critique of the methods based on association hypotheses. Sections 3.2 to 3.3 give the background to random finite set theory including detailed derivations of the multitarget Bayes filter. The derivation of the

PHD filter is given in Section 4.1. Example calculations are also given for illustrative purposes. Full details of the sequential Monte Carlo (SMC) and Gaussian mixture (GM) implementations of the PHD filter are given in Section 4.2.

Chapter 5 focuses on analysis of, and extensions to the PHD filter. In Section 5.1, new techniques are presented culminating in the improved SMC-PHD filter with track continuity. In Section 5.2, a general scheme for the PHD filter with JMLS manoeuvre handling is presented which uses the framework of interacting multiple models. It is shown that the scheme reduces to the established JMLS-GM-PHD filter in the linear, Gaussian case and the JMLS Bayes filter in the single target case. In Section 5.3, an analysis of the PHD filter with target amplitude information demonstrates a similarity with the Bayesian track-before-detect methodology in a special case. A new variation of the optimal sub-pattern assignment (OSPA) metric which incorporates target confidence information (which is important to many multitarget filters) is presented in Section 5.4.

A range of applications of the PHD filter is presented in *Chapter 6*. Active sonar data with a synthetic target is studied in Section 6.1, with a focus on detecting and tracking low signal-to-noise ratio (SNR) targets. Passive sonar data is considered in Section 6.2 where the combination of a new denoising scheme with a random finite set formulation of the problem of detection tones in the frequency domain permits the detection of low SNR tones. In Section 6.3, the PHD filter is deployed on microscopic visual data to extract bacterial trajectories for motion analysis. Simulated data is studied in Section 6.4 to allow a performance comparison of GM-PHD with conventional methods evaluated using practical performance indicators and the OSPA metric.

A summary and conclusions, including a discussion of possible avenues of future work is given in *Chapter 7*.

1.3 Original Contributions

- *Single target filter performance comparison (see Sections 2.2 and 2.3):*
  - Performance evaluation of popular approximate single target filters based on a comparison with Bayes filter posteriors.
  - Implementation of an adaptive grid method for computing the Bayes filter posterior.

  - Proposal of a new method for choosing single target priors and preliminary test results showing improvement in performance using this choice of prior.

- *Contributions to the presentation of the random finite set theory background (see Chapters 3 and 4):*
  - Presentation of simple worked example calculations of the multitarget Bayes filter and PHD filter.
  - Original proof of an intermediate result from the derivation of the PHD filter given in full in Appendix B.1.

- *New methods for implementation of the SMC-PHD filter (see Section 5.1 and [136]):*
  - Identification of a new class of problems with the clustering step in the SMC-PHD filter.
  - Proposal of a method for jointly determining the number of clusters and clustering for track extraction in the SMC-PHD filter. Performance of this method is verified by testing on simulated data.
  - Proposal of stratified resampling method and optimisation of maintained track labels in order to permit track continuity in the improved SMC-PHD filter of Ristic et al. [96].
  - Gating method proposed to increase computational efficiency in the improved SMC-PHD filter with track continuity.
  - Test results demonstrating the effectiveness of the improved SMC-PHD filter with track continuity and the gating method.

- *Jump Markov Linear System (JMLS) PHD Filter (see Section 5.2 and [131]):*
  - New method proposed using interacting multiple models to allow implementation of PHD filter for targets with JMLS dynamics. This removes the need for further assumptions on target dynamics.
  - The JMLS-PHD filter reduces to the established JMLS-GM-PHD filter of Pasha et al. [90] when forward and measurement models are linear and Gaussian.
  - The JMLS-PHD filter reduces to the JMLS Bayes filter in the single target case.

- *Comparison of the PHD filter with target amplitude information of Clark et al. [29] (SNR-PHD filter) with track-before-detect methodology (see Section 5.3):*
  - The track-before-detect filter in the case of 0 or 1 targets is equivalent to a multitarget Bayes filter.
  - Existence of a close link between track-before-detect particle filter of Salmond and Birch [105] and a special case of the SNR-PHD filter.
  - Existence of a close link between the HMM implementation of track-before-detect filter and a special case of the SNR-PHD filter.
  - The above suggest that the SNR-PHD filter provides a rigorous method for extending the track-before-detect methodology to the multitarget case.
  - Illustrative results demonstrate the effectiveness of the SNR-PHD filter for a multitarget scenario.
- *New variation of OSPA metric incorporating track confidence information (see Section 5.4):*
  - Standard formulation of the OSPA metric does not take into account track confidence information such as the weights in the PHD filter, which is important to many multitarget trackers.
  - A new distance taking this information into account is proposed and it is shown that this distance is a multitarget metric.
  - Properties of the new metric include preferring higher confidence true tracks, preferring lower confidence false tracks and being equivalent to the standard OSPA metric when all weights are equal to one.
- *Implementation of the PHD filter on active sonar data (see Section 6.1):*
  - Demonstration of the effectiveness of the PHD filter for detecting and tracking a low SNR synthetic target in active sonar data.
- *Implementation of the PHD filter on passive sonar data (see Section 6.2 and [135]):*
  - A new denoising method based on a combination of the Radon and Wavelet transforms.
  - Formulation of the problem of detection and monitoring low SNR tones in audio data as a multitarget tracking problem in the frequency domain.

  - Results on real and simulated passive sonar data demonstrating good performance at detecting and monitoring tones.

- *Tracking bacteria in microscopic visual data using the PHD filter (see Section 6.3 and [137]):*
  - Implementation of the PHD filter on microscopic visual data to extract bacterial trajectories.
  - A significantly larger number of trajectories extracted than was possible using older methods.
  - Trajectories extracted used to verify a prediction from fluid dynamics relating to the behaviour of bacteria near surfaces.

- *Performance comparison of the GM-PHD filter with data association methods (See Section 6.4 and [133]):*
  - Results showing superior performance of the GM-PHD filter by comparison with nearest neighbour and probabilistic data association methods, particularly in high levels of false alarms.

1.4 Publications

- [131] T.M. Wood, "Interacting methods for manoeuvre handling in the GM-PHD filter", *IEEE Trans. on Aerospace and Electronic Systems*, 47(4):3021-3025, 2011.

- [135] T.M. Wood, D. Allwright, P. Bond, S. Long and I. Moroz, "A new method for processing passive sonar data", *Proc. 13th International Conference on Information Fusion*, Edinburgh, 2010.

- [136] T.M. Wood, D. Clark, B. Ristic, "Efficient resampling and basic track continuity for the SMC-PHD filter", *Proc. Cognitive Systems with Interactive Sensors*, 2010.

- [133] T.M. Wood, "Tracking in dense clutter with the PHD filter", *Proc. IMA Mathematics in Defence*, 2009.

- [137] T.M. Wood, C.A. Yates, D. Wilkinson, G. Rosser, "Simplified multitarget tracking using the PHD filter for microscopic visual data", In Press, *IEEE Trans. on Circuits and Systems for Video Technology.*

- [134] T.M. Wood, "Detecting and tracking multiple stealthy targets: comparison of PHD filter and track-before-detect approaches", *Proc. IMA Mathematics in Defence*, 2011.

1.4.1 Author's Contribution to Jointly Authored Papers

"*A new method for processing passive sonar data*" with D. Allwright, P. Bond, S. Long and I. Moroz.

- Implementation of the denoising method, which was conceived by the other authors.
- Formulation of the detection and tracking of tones in an audio signal as a multitarget tracking problem in the frequency domain.
- Results for real and simulated passive sonar data.

"*Efficient resampling and basic track continuity for the SMC-PHD filter*" with D. Clark and B. Ristic.

- The idea of achieving clustering using the resampling step was due to D. Clark but the details of the stratified resampling scheme were due to the author.
- Scheme for track continuity by optimising the number of track labels maintained.
- Idea and implementation of the gating step jointly with D. Clark.
- Implementations and results obtained using numerical simulations.

"*Simplified multitarget tracking using the PHD filter for microscopic visual data*" with C.A. Yates, D. Wilkinson and G. Rosser

- Idea and implementation details for PHD filter applied to microscopic visual data.
- Manual analysis for verifying performance jointly with G. Rosser.
- Simulations for performance analysis.
- Analysis of bacterial trajectories leading to probability densities for stepwise angle changes.

Chapter 2

Single Target Tracking

2.1 Filtering

The term 'filtering' is taken from the idea of removing impurities (as if from water) and is applied, in this instance, to removing noise from information. Anderson and Moore [2], in their textbook on "Optimal Filtering", highlight this comparison, writing that measurements will "contain random inaccuracies or be contaminated by unwanted signals and filtering is necessary to make the control close to that desired".

An important development in the study of filtering occurred in 1960 with a discovery by Rudolf Emil Kalman [58]. Grewal and Mohinder [45] refer to the Kalman filter as "one of the greatest discoveries in the history of statistical estimation theory". A Kalman filter was at the heart of the navigation computer for the Apollo space missions [51] and Kalman filters remain important to control systems, for example, in manufacturing processes, aircraft and ships. Just as impressive is the simplicity of the Kalman filter; an implementation needs no more than six lines of MATLAB code.

In this subsection, single target tracking will be formulated as a filtering problem. A conceptual solution will be given the form of the Bayes filter and a key relationship between the Bayes filter and the Kalman filter will be demonstrated. Finally, some important developments of the theory of filtering in the 50 years since Kalman's seminal paper will be presented.

2.1.1 Problem Formulation and the Bayes filter

Consider an evolving system with unknown state $\mathbf{x}_k$ and a measurement, $\mathbf{z}_k$ made of the system, both at (discrete) time k where $\mathbf{x}_k$ and $\mathbf{z}_k$ belong to $\mathbb{R}^n$ and $\mathbb{R}^m$ respectively for some integers n and m. Let the evolution of the state and the measurement

process be given by

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k) + \mathbf{v}_{k+1}, \tag{2.1}$$

$$\mathbf{z}_{k+1} = h(\mathbf{x}_{k+1}) + \mathbf{w}_{k+1}, \tag{2.2}$$

respectively. Here f, h are the deterministic (generally nonlinear) terms of the forward and measurement processes and $\mathbf{v}_{k+1}$, $\mathbf{w}_{k+1}$ are the random terms, considered 'noise' and assumed here to be additive.

In single target tracking, $\mathbf{x}_k$ is taken to be the state of the target. This might include position, velocity and possibly acceleration of the target in some cartesian coordinate system or any other relevant parameters. Typically, the dimension of the measurement, $\mathbf{z}_k$, will be lower than that of $\mathbf{x}_k$.

Consider, as an example, active sonar in which at discrete time intervals a sound signal (or 'ping') is sent out and is detected as it is reflected from a target. It is not possible to measure the velocity or acceleration of the target directly, but the range of the target may be calculated from the time taken for the sound wave to return and the bearing may be calculated using beamforming [119]. In this case, the measurement is $\mathbf{z}_k = [r, \theta] = [\sqrt{x_k^2 + y_k^2}, \arg(x_k + iy_k)]$.

The problem is: infer the unknown state $\mathbf{x}_k$, at time k, given the measurements up to time k, $Z^k = (\mathbf{z}_1, ..., \mathbf{z}_k)$. Due to the multiple sources of noise in (2.1-2.2), and the lower dimension of the measurement in the example, it will not be possible to infer $\mathbf{x}_k$ exactly. Therefore, the solution to the inference problem is: compute the probability density for $\mathbf{x}_k$, conditioned on Z^k, for all k. This probability density will be written $f_{k|k}(\mathbf{x}_k|Z^k)$.

Note that for the rest of this thesis, the notation f will be used to denote probability density functions. The corresponding random variable for the probability function and any conditioning should be made clear and the subscript indices will denote the time step. A subscript index such as $f_{k+1|k}$ will denote a prediction from time step k to time step $k+1$.

The Bayes Filter

Assume that there is a known prior probability density, $f_{0|0}(\mathbf{x}_0)$ encapsulating the information known about the system state before the first measurement arrives. For the active sonar tracking example, this might be a uniform density over some finite region of surveillance, or it may have a peak at some region where targets are known to hide. The essential feature of the method is induction over the time step. It will be

shown how to compute the probability density at time step $k+1$, $f_{k+1|k+1}(\mathbf{x}_{k+1}|Z^{k+1})$, given the probability density at time step k, $f_{k|k}(\mathbf{x}_k|Z^k)$.

First, the predicted density $f_{k+1|k}(\mathbf{x}_{k+1}|Z^k)$ will be computed for known Z^k using the forward model (2.1). Begin by considering the full joint density, $f(\mathbf{x}_{k+1}, \mathbf{x}_k, \mathbf{v}_{k+1}|Z^k)$ from which the non-relevant variables will be marginalised out using the law of total probability. Standard conditional probability gives

$$f(\mathbf{x}_{k+1}, \mathbf{x}_k, \mathbf{v}_{k+1}|Z^k) = f(\mathbf{x}_{k+1}|Z^k, \mathbf{x}_k, \mathbf{v}_{k+1}) f_V(\mathbf{v}_{k+1}|Z^k) f(\mathbf{x}_k|Z^k). \tag{2.3}$$

Here, $f_V(\mathbf{v}_{k+1}|Z^k)$ is the probability density for $\mathbf{v}_{k+1}$ conditioned on Z^k and the independence of $\mathbf{x}_k$ and $\mathbf{v}_{k+1}$ is assumed.

Using the fact that if $\mathbf{x}_k$ and $\mathbf{v}_{k+1}$ are given then $\mathbf{x}_{k+1}$ is uniquely determined (and so has a multidimensional delta function for a probability density), and also assuming the independence of $\mathbf{v}_{k+1}$ and Z^k,

$$f(\mathbf{x}_{k+1}, \mathbf{x}_k, \mathbf{v}_{k+1}|Z^k) = \delta(\mathbf{x}_{k+1} - f(\mathbf{x}_k) - \mathbf{v}_{k+1}) \cdot f_V(\mathbf{v}_{k+1}) \cdot f_{k|k}(\mathbf{x}_k|Z^k). \tag{2.4}$$

Integrating out $\mathbf{v}_{k+1}$ and using the law of total probability and the sifting property of the delta function gives

$$f(\mathbf{x}_{k+1}, \mathbf{x}_k|Z^k) = f_V(\mathbf{x}_{k+1} - f(\mathbf{x}_k)) \cdot f_{k|k}(\mathbf{x}_k|Z^k). \tag{2.5}$$

Similarly, integrating out $\mathbf{x}_k$ gives

$$f_{k+1|k}(\mathbf{x}_{k+1}|Z^k) = \int f_V(\mathbf{x}_{k+1} - f(\mathbf{x}_k)) \cdot f_{k|k}(\mathbf{x}_k|Z^k) \mathrm{d}\mathbf{x}_k, \tag{2.6}$$

which is the predicted density.

Using a similar method, it is possible to derive the updated, or 'posterior', density $f_{k+1|k+1}(\mathbf{x}_{k+1}|Z^{k+1})$ for known Z^{k+1} using (2.2). Again, begin with the full joint density

$$f(\mathbf{x}_{k+1}, \mathbf{z}_{k+1}, \mathbf{w}_{k+1}|Z^k) = \delta(\mathbf{z}_{k+1} - h(\mathbf{x}_{k+1}) - \mathbf{w}_{k+1}) f_W(\mathbf{w}_{k+1}) f_{k+1|k}(\mathbf{x}_{k+1}|Z^k), \tag{2.7}$$

using the same logic as previously and assuming that the measurement noise is independent of the target state and previous measurements and with $f_W(\mathbf{w}_{k+1})$ being the probability density of $\mathbf{w}_{k+1}$. Integrating out $\mathbf{w}_{k+1}$ gives

$$f(\mathbf{x}_{k+1}, \mathbf{z}_{k+1}|Z^k) = f_W(\mathbf{z}_{k+1} - h(\mathbf{x}_{k+1})) f_{k+1|k}(\mathbf{x}_{k+1}|Z^k). \tag{2.8}$$

A further application of conditional probability gives

$$f_{k+1}(\mathbf{x}_{k+1}|Z^{k+1}) = C^{-1} f(\mathbf{x}_{k+1}, \mathbf{z}_{k+1}|Z^k) \tag{2.9}$$

where C is the normalising constant

$$C = f(\mathbf{z}_{k+1}|Z^k) = \int f(\mathbf{x}_{k+1}, \mathbf{z}_{k+1}|Z^k)\mathrm{d}\mathbf{x}_{k+1}. \tag{2.10}$$

Combining (2.6), (2.8) and (2.9) and using the standard notation

$$f_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{x}_k) = f_V(\mathbf{x}_{k+1} - f(\mathbf{x}_k)), \tag{2.11}$$

$$f_{k+1}(\mathbf{z}_{k+1}|\mathbf{x}_{k+1}) = f_W(\mathbf{z}_{k+1} - h(\mathbf{x}_{k+1})), \tag{2.12}$$

where the former is referred to as the Markov density and the latter is the likelihood function. Using this notation, the Bayes filter may be written as [20] [63]:

$$f_{k+1|k}(\mathbf{x}_{k+1}|Z^k) = \int f_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{x}_k) \cdot f_{k|k}(\mathbf{x}_k|Z^k)\mathrm{d}\mathbf{x}_k, \tag{2.13}$$

$$f_{k+1|k+1}(\mathbf{x}_{k+1}|Z^{k+1}) = \frac{f_{k+1|k}(\mathbf{x}_{k+1}|Z^k)f_{k+1}(\mathbf{z}_{k+1}|\mathbf{x}_{k+1})}{f(\mathbf{z}_{k+1}|Z^k)}. \tag{2.14}$$

This is a conceptual solution for calculating the probability density for $\mathbf{x}_k$ conditioned on Z^k. Only in special cases can this be reduced to known functions and in general an approximation is required. In Section 2.1.2 a special case where the Bayes filter can be easily computed will be presented and in Section 2.1.3 a series of approximation strategies will be considered.

2.1.2 Bayes Formulation of the Kalman Filter

The Kalman filter is typically derived by minimising expected squared estimation errors, as in Kalman's original paper [58]. However, Kalman himself points out that minimising expected estimation errors is only one possible way of choosing an estimator and furthermore it might be noted that focusing on square errors is not the only possible option. By contrast, in this section the Kalman filter is derived as a special case of the Bayes filter (2.13-2.14), following the approach in [48].

The special case considered is where all of the deterministic terms of the forward and measurement models (2.1-2.2) are linear and are represented by the matrices F_k and H_k respectively. Furthermore, all noise is assumed to be zero-mean, white and Gaussian with covariance Q_k for process noise (i.e. in the forward process) and R_k for measurement noise. Hence, (2.1-2.2) may be written as

$$\mathbf{x}_{k+1} = F_k\mathbf{x}_k + \mathbf{v}_{k+1} \qquad \mathbf{v}_{k+1} \sim \mathcal{N}(\mathbf{0}, Q_{k+1}), \tag{2.15}$$

$$\mathbf{z}_{k+1} = H_{k+1}\mathbf{x}_{k+1} + \mathbf{w}_{k+1} \qquad \mathbf{w}_{k+1} \sim \mathcal{N}(\mathbf{0}, R_{k+1}). \tag{2.16}$$

The notation $\mathcal{N}(\mathbf{p}, P)$ denotes the normal distribution with mean $\mathbf{p}$ and covariance matrix P where $P \in \mathbb{R}^{n\times n}$ when $\mathbf{p} \in \mathbb{R}^n$. Similarly, the notation $\mathcal{N}(\mathbf{x}; \mathbf{p}, P)$ denotes the probability density function (p.d.f.) of the same normal distribution evaluated at $\mathbf{x}$. Using these conventions, the Markov density and likelihood function from (2.13-2.14) may be expressed as

$$f_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{x}_k) = \mathcal{N}(\mathbf{x}_{k+1}; F_k\mathbf{x}_k, Q_{k+1}), \tag{2.17}$$

$$f_{k+1}(\mathbf{z}_{k+1}|\mathbf{x}_{k+1}) = \mathcal{N}(\mathbf{z}_{k+1}; H_{k+1}\mathbf{x}_{k+1}, R_{k+1}). \tag{2.18}$$

For the derivation, the following algebraic identity will be required, the proof of which may be found in [75, App. D]:

$$\mathcal{N}(\mathbf{r}; H\mathbf{x}, R)\mathcal{N}(\mathbf{x}; \mathbf{p}, P) = \mathcal{N}(\mathbf{r}; H\mathbf{p}, R + HPH^T)\mathcal{N}(\mathbf{x}; \mathbf{e}, E), \tag{2.19}$$

where

$$E = (P^{-1} + H^T R^{-1} H)^{-1}, \tag{2.20}$$

$$\mathbf{e} = E(P^{-1}\mathbf{p} + H^T R^{-1}\mathbf{r}). \tag{2.21}$$

Prediction

Assume that $f_{k|k}(\mathbf{x}_k|Z^k) = \mathcal{N}(\mathbf{x}_k; \mathbf{x}_{k|k}, P_{k|k})$. Then, from (2.13),

$$f_{k+1|k}(\mathbf{x}_{k+1}|Z^k) = \int \mathcal{N}(\mathbf{x}_{k+1}; F_k\mathbf{x}_k, Q_{k+1})\mathcal{N}(\mathbf{x}_k; \mathbf{x}_{k|k}, P_{k|k})\mathrm{d}\mathbf{x}_k \tag{2.22}$$

$$= \int \mathcal{N}(\mathbf{x}_{k+1}; F_k\mathbf{x}_{k|k}, Q_{k+1} + F_k P_{k|k} F_k^T)\mathcal{N}(\mathbf{x}_k; \mathbf{e}, E)\mathrm{d}\mathbf{x}_k, \tag{2.23}$$

using (2.19). The term on the right is a probability density and hence integrates to 1. Therefore,

$$f_{k+1|k}(\mathbf{x}_{k+1}|Z^k) = \mathcal{N}(\mathbf{x}_{k+1}; F_k\mathbf{x}_{k|k}, Q_{k+1} + F_k P_{k|k} F_k^T). \tag{2.24}$$

Update

Re-writing $f_{k+1|k}(\mathbf{x}_{k+1}|Z^k) = \mathcal{N}(\mathbf{x}_{k+1}; \mathbf{x}_{k+1|k}, P_{k+1|k})$, (2.14) gives

$$f_{k+1|k+1}(\mathbf{x}_{k+1}|Z^{k+1}) = \frac{\mathcal{N}(\mathbf{z}_{k+1}; H_{k+1}\mathbf{x}_{k+1}, R_{k+1}) \cdot \mathcal{N}(\mathbf{x}_{k+1}; \mathbf{x}_{k+1|k}, P_{k+1|k})}{\int \mathcal{N}(\mathbf{z}_{k+1}; H_{k+1}\mathbf{x}_{k+1}, R_{k+1}) \cdot \mathcal{N}(\mathbf{x}_{k+1}; \mathbf{x}_{k+1|k}, P_{k+1|k})\mathrm{d}\mathbf{x}_{k+1}}. \tag{2.25}$$

Using (2.19) again:

$$f_{k+1|k+1}(\mathbf{x}_{k+1}|Z^{k+1}) = \frac{\mathcal{N}(\mathbf{z}_{k+1}; H_{k+1}\mathbf{x}_{k+1|k}, R_k + H_{k+1}P_{k+1|k}H_{k+1}^T)\mathcal{N}(\mathbf{x}_{k+1}; \mathbf{e}, E)}{\int \mathcal{N}(\mathbf{z}_{k+1}; H_{k+1}\mathbf{x}_{k+1|k}, R_k + H_{k+1}P_{k+1|k}H_{k+1}^T)\mathcal{N}(\mathbf{x}_{k+1}; \mathbf{e}, E)\mathrm{d}\mathbf{x}_{k+1}} \tag{2.26}$$

$$= \mathcal{N}(\mathbf{x}_{k+1}; \mathbf{e}, E), \tag{2.27}$$

where

$$E = (P_{k+1|k}^{-1} + H_{k+1}^T R_{k+1}^{-1} H_{k+1})^{-1}, \tag{2.28}$$

$$\mathbf{e} = E(P_{k+1|k}^{-1}\mathbf{x}_{k+1|k} + H_{k+1}^T R_{k+1}^{-1}\mathbf{z}_{k+1}). \tag{2.29}$$

Using [48], these may be re-written as

$$E = P_{k+1|k} - P_{k+1|k}H_{k+1}^T(H_{k+1}P_{k+1|k}H_{k+1}^T + R_{k+1})^{-1}H_{k+1}P_{k+1|k}, \tag{2.30}$$

$$\mathbf{e} = \mathbf{x}_{k+1|k} + P_{k+1|k}H_{k+1}^T(H_k P_{k+1|k}H_{k+1}^T + R_{k+1})^{-1}\cdot (\mathbf{z}_{k+1} - H_{k+1}\mathbf{x}_{k+1|k}). \tag{2.31}$$

Therefore it can be seen that in the linear, Gaussian case, as long as the prior probability density is Gaussian, all subsequent predicted and updated probability densities will also be Gaussian. Furthermore, the means and covariances of these densities can be calculated using only matrix algebra. Doing so permits the calculation of means and covariances for the state $\mathbf{x}_k$ using the scheme below, which is the well known Kalman filter.

The Kalman Filter - Writing $\mathbf{x}_{k|k}$ and $P_{k|k}$ for the mean and covariance of $f_{k|k}(\mathbf{x}_k, Z^k)$ and $\mathbf{x}_{k+1|k}$ and $P_{k+1|k}$ for the mean and covariance of $f_{k+1|k}(\mathbf{x}_{k+1}, Z^k)$, these may be computed as:
Prediction:

$$\mathbf{x}_{k+1|k} = F_k\mathbf{x}_{k|k} \tag{2.32}$$

$$P_{k+1|k} = Q_{k+1} + F_k P_{k|k} F_k^T \tag{2.33}$$

Update:

$$K_{k+1} = P_{k+1|k}H_{k+1}^T(H_{k+1}P_{k+1|k}H_{k+1}^T + R_{k+1})^{-1} \tag{2.34}$$

$$\mathbf{x}_{k+1|k+1} = \mathbf{x}_{k+1|k} + K_{k+1}(\mathbf{z}_{k+1} - H_{k+1}\mathbf{x}_{k+1|k}) \tag{2.35}$$

$$P_{k+1|k+1} = (\mathbb{I} - K_{k+1}H_{k+1})P_{k+1|k} \tag{2.36}$$

2.1.3 Approximation of the Bayes filter

It was shown above that in the case where (2.1-2.2) were linear and Gaussian, the calculation of the Bayes filter reduces to a matrix calculation. However, if these equations are either nonlinear or have correlated or non-Gaussian noise then other strategies will be required to calculate the Bayes filter. There are a plethora of methods available and it is clearly not possible to provide anything approaching a detailed

survey. Instead, a very brief introduction to five important classes of methods will be presented. These methods of approximation are: Taylor series, sigma point, Gaussian sum, sequential Monte Carlo and grid-based methods.

Taylor Series Approximation

A simple and popular [95] method of approximation, based upon a Taylor series truncation, is the Extended Kalman Filter (EKF). In order to derive this, start with (2.1-2.2), assuming that f_k and h_{k+1} are sufficiently differentiable, and Taylor expand about the expected trajectory as follows (using 'h.o.t.' to denote 'higher order terms'):

$$f_k(\mathbf{x}_k) = f_k(\mathbf{x}_{k|k}) + \frac{\partial f_k(\mathbf{x})}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\mathbf{x}_{k|k}}(\mathbf{x}_k - \mathbf{x}_{k|k}) + h.o.t. \tag{2.37}$$

$$h_{k+1}(\mathbf{x}_{k+1}) = h_{k+1}(\mathbf{x}_{k+1|k}) + \frac{\partial h_{k+1}(\mathbf{x})}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\mathbf{x}_{k+1|k}}(\mathbf{x}_k - \mathbf{x}_{k|k}) + h.o.t. \tag{2.38}$$

Here, the derivatives are the Jacobian matrices of f_k and h_{k+1} respectively and will be denoted $F_k^{[1]}$ and $H_{k+1}^{[1]}$ hereafter. Neglecting the higher order terms leads to the linearised form of (2.1-2.2)

$$\mathbf{x}_{k+1} = F_k^{[1]}\mathbf{x}_k + \mathbf{v}_k + \mathbf{u}_k, \tag{2.39}$$

$$\mathbf{z}_{k+1} = H_{k+1}^{[1]}\mathbf{x}_{k+1|k} + \mathbf{w}_{k+1} + \mathbf{y}_{k+1}, \tag{2.40}$$

where

$$\mathbf{u}_k = f_k(\mathbf{x}_{k|k}) - F_k^{[1]}\mathbf{x}_{k|k} \quad , \quad \mathbf{y}_{k+1} = h_{k+1}(\mathbf{x}_{k+1|k}) - H_{k+1}^{[1]}\mathbf{x}_{k+1|k}. \tag{2.41}$$

The Extended Kalman filter is then derived via the Kalman filter for the above linearised equations and is given by [2, ch. 8]

Prediction:

$$\mathbf{x}_{k+1|k} = f_k(\mathbf{x}_{k|k}), \tag{2.42}$$

$$P_{k+1|k} = F_k^{[1]}P_{k|k}(F_k^{[1]})^T + Q_{k+1}. \tag{2.43}$$

Update:

$$K_{k+1} = P_{k+1|k}(H_{k+1}^{[1]})^T(H_{k+1}^{[1]}P_{k+1|k}(H_{k+1}^{[1]})^T + R_{k+1})^{-1}, \tag{2.44}$$

$$\mathbf{x}_{k+1|k+1} = \mathbf{x}_{k+1|k} + K_{k+1}(\mathbf{z}_{k+1} - H_{k+1}^{[1]}\mathbf{x}_{k+1|k}), \tag{2.45}$$

$$P_{k+1|k+1} = (\mathbb{I} - K_{k+1}H_{k+1}^{[1]})P_{k+1|k}. \tag{2.46}$$

There are some obvious problems with the Extended Kalman filter as an approximate Bayes filter calculation. The most obvious is that the linearisation in (2.39-2.40) may not be a good approximation if the higher order terms in (2.37-2.38) make a significant contribution to the sum and this will introduce error. Just as importantly, by linearising and then using the Kalman filter framework, there is an inherent restriction to Gaussian probability densities, but of course not all probability densities are Gaussian. In Section 2.2 an example based on a simple tracking example is presented which highlights this shortcoming. Simulations conducted in [132] demonstrate that the EKF can return state estimates which are incorrect by several orders of magnitude more than the uncertainty in the measurements, and so the EKF should be implemented with caution.

Regardless, the EKF is considered the most widely used estimation algorithm for filtering nonlinear systems [57], and especially for nonlinear tracking [15], due to its simplicity and ease of calculation.

Sigma Point Approximation

The Unscented Kalman Filter (UKF) was developed by Julier and Uhlmann [55] who state that it is "founded on the intuition that it is easier to approximate a probability distribution than it is to approximate an arbitrary nonlinear function or transformation". To this end, the UKF does not linearise the non-linearities in (2.1-2.2) but instead propagates statistical information through the nonlinear transform using so-called sigma points.

These sigma points are chosen so that using $2N_x + 1$ sigma points (where N_x is the dimension of the state space) it is possible to capture the mean and covariance of $\mathbf{x}_k$. Julier and Uhlmann show in [56] that consequently the mean and covariance of $f_k(\mathbf{x}_k)$ are correct to second order, an order of accuracy higher than the EKF. The full details of the UKF are presented in Appendix B.4.

It has been demonstrated that the UKF can provide a better approximation to the Bayes filter in situations where the EKF approximations break down. In particular, this is shown in the case of a tracking scenario where bearing uncertainty is large in comparison with range uncertainty in [57]. As this is particularly relevant to sonar tracking scenarios, this was investigated further in [132], which includes simulated results showing that the UKF was more able to return reliable estimates for target state than the EKF. A further investigation will be presented in Section 2.2.

As with the EKF, the filter calculation is reduced to a matrix computation, which is fast and simple in low dimensions. However, there is again a restriction to Gaussian

probability densities. The next three methods will not require this restriction.

Gaussian Sum Approximation

In [111], it is shown that any probability density can be approximated with arbitrary accuracy with a Gaussian sum, which is a weighted sum of Gaussian densities and is also known as a Gaussian mixture. The Gaussian sum filter utilises this by approximating the Markov density and likelihood function in (2.13-2.14) by Gaussian sums, avoiding the reliance on any particular form of these equations. Approximating the prior density by a Gaussian sum, it will be possible to calculate the Bayes filter exactly using (2.19). It should be noted that in general there is an exponential increase in the number of components in the Gaussian sum as time increases. Some methods for managing this growth in the number of Gaussian components will be discussed in Section 4.2.2. Further details are given in Appendix B.5.

Sequential Monte Carlo

In Sequential Monte Carlo, first proposed in [44], a set of states known as 'particles' $\{\mathbf{x}^i_{k|k}\}_{i=1}^N$ with weights $\{w^i_{k|k}\}_{i=1}^N$ (whose sum is equal to one) are used to approximate the probability density as

$$f_{k|k}(\mathbf{x}_k|Z^k) \approx \sum_{i=1}^{N} w^i_{k|k}\delta_{\mathbf{x}^i_{k|k}}(\mathbf{x}_k), \tag{2.47}$$

where $\delta_{\mathbf{x}}$ is the Dirac delta centred on $\mathbf{x}$. The particles may be interpreted as weighted samples from the posterior density $f_{k|k}(\mathbf{x}_k|Z^k)$.

Assuming that the posterior density at time k can be approximated as in (2.47) and using this in the Bayes filter (2.1-2.2) gives

$$f_{k+1|k+1}(\mathbf{x}_{k+1}|Z^{k+1}) \propto \sum_{i=1}^{N} w^i_{k|k} f_{k+1|k}(\mathbf{x}_{k+1}, \mathbf{x}^i_{k|k}) f_{k+1}(\mathbf{z}_{k+1}|\mathbf{x}_{k+1}). \tag{2.48}$$

Therefore, if it was possible to sample $\mathbf{x}^i_{k+1|k+1} \sim f_{k+1|k}(\cdot, \mathbf{x}^i_{k|k}) f_{k+1}(\mathbf{z}_{k+1}|\cdot)$ then the posterior at time $k+1$ could be approximated as

$$f_{k+1|k+1}(\mathbf{x}_{k+1}|Z^{k+1}) \approx \sum_{i=1}^{N} w^i_{k|k}\delta_{\mathbf{x}^i_{k+1|k+1}}(\mathbf{x}_{k+1}). \tag{2.49}$$

However, it is not generally possible to sample from this distribution. Instead the technique of importance sampling is used. Importance sampling (see, for example, [36]) is a technique whereby samples are drawn from a proposal density, instead of

the target density, and are subsequently weighted. Let π be the target density which cannot be sampled but can be evaluated and let q be the proposal density. The idea then is to sample $\mathbf{x}^i \sim q(\cdot)$ and assign importance weights for $\mathbf{x}^i$ as

$$w^i \propto \frac{\pi(\mathbf{x}^i)}{q(\mathbf{x}^i)}. \tag{2.50}$$

A common choice for the proposal density is $q = f_{k+1|k}(\cdot|\mathbf{x}_k)$, if it is possible to sample from this function. Note that it can be seen from (2.48) that with this choice of proposal density, the corresponding importance weights in (2.49) are $w^i_{k+1|k+1} = w^i_{k|k} f_{k+1}(\mathbf{z}_{k+1}|\mathbf{x}^i_{k+1})$.

Degeneracy and Resampling

It has been shown in [36] that the variance of the importance weights increases with time. This leads to a case where some particles have negligible weight and hence are not contributing to the approximation of the posterior but still require computation. In the extreme case, this results in particle degeneracy where all of the weight for the set of samples is on one particle.

A solution to this problem is resampling. Resampling is the process of sampling (with replacement) from the set of particles using their importance weights as a discrete probability density (this is sampling from a set of samples, hence 'resampling'). All resampled particles are given equal weight. Intuitively, this results in higher weight particles being chosen with higher probability and negligible weight particles being discarded.

The combination of sequential importance sampling with resampling used at each time step is referred to as Sequential Importance Resampling or, more commonly, the particle filter. The steps in implementing a particle filter may be summarised as:

- Sample from initial prior density to give $\{\mathbf{x}^i_{0|0}, w^i_{0|0}\}_{i=1}^N$.

  For each time step k:

- Sample $\{\hat{\mathbf{x}}^i_{k|k}\}_i$ from the proposal density $q(\cdot)$.
- Calculate the sample weights $\{\hat{w}^i_{k|k}\}_i$ using (2.50).
- Resample N times from $\{\hat{\mathbf{x}}^i_{k|k}, \hat{w}^i_{k|k}\}_{i=1}^N$ to give $\{\mathbf{x}^i_{k|k}, w^i_{k|k}\}_{i=1}^N$.

Further details about sequential Monte Carlo methods may be found in the excellent tutorial articles by Arulampalam et al. [5] and Doucet & Johansen [37]. These

include discussions of important extensions to the standard sequential Monte Carlo filter presented here, such as the auxiliary SMC filter and the resample-move particle filter. There is also a summary of known convergence results for the sequential importance resampling filter in [31].

Grid Approximation

For the grid-based method, the state space is approximated by a regular grid consisting of N cells. These cells are centred on the points $\hat{\mathbf{x}}^i$ for $i = 1, ..., N$. The density is approximated by a constant value $w^i_{k|k}$ on the grid cell centred on $\hat{\mathbf{x}}^i$ at time step k.

Let the pdf at time k be given by

$$f_{k|k}(\mathbf{x}_k|Z^k) = \Sigma_{i=1}^N w_k^i \mathbf{1}_{\tilde{\mathbf{x}}^i}(\mathbf{x}_k) \tag{2.51}$$

where $\mathbf{1}_A$ is the set indicator function on the set A and $\tilde{\mathbf{x}}^i$ is the set of points in the cell centred on $\hat{\mathbf{x}}^i$.

Thus, the Bayes filter can be written as

$$\begin{aligned} f_{k+1|k}(\mathbf{x}_{k+1|k}|Z^k) &= \Sigma_{i=1}^N w^i_{k+1|k} \mathbf{1}_{\tilde{\mathbf{x}}^i}(\mathbf{x}_k), & (2.52)\\ f_{k+1|k+1}(\mathbf{x}_{k+1}|Z^{k+1}) &= \Sigma_{i=1}^N w^i_{k+1} \mathbf{1}_{\tilde{\mathbf{x}}^i}(\mathbf{x}_k) & (2.53) \end{aligned}$$

where

$$\begin{aligned} w^i_{k+1|k} &= \Sigma_{j=1}^N w^j_{k|k} \int_{\mathbf{y}\in\tilde{\mathbf{x}}^j} \int_{\mathbf{x}\in\tilde{\mathbf{x}}^i} f_{k+1|k}(\mathbf{x}|\mathbf{y}) d\mathbf{x} d\mathbf{y}, & (2.54)\\ w^i_{k|k} &= \frac{w^i_{k|k-1} \int_{\mathbf{x}\in\tilde{\mathbf{x}}^i} f_k(\mathbf{z}_k|\mathbf{x}) d\mathbf{x}}{\Sigma_{j=1}^N w^j_{k|k-1} \int_{\mathbf{x}\in\tilde{\mathbf{x}}^j} p f_k(\mathbf{z}_k|\mathbf{x}) d\mathbf{x}}. & (2.55) \end{aligned}$$

A further simplification can be made by approximating $f_{k+1|k}(\mathbf{x}|\mathbf{y})$ and $f_k(\mathbf{z}|\mathbf{x})$ by their values at the centre of the cell. If the cells have equal size, this gives

$$\begin{aligned} w^i_{k+1|k} &= \Sigma_{j=1}^N w^j_{k|k} f_{k+1|k}(\hat{\mathbf{x}}^i|\hat{\mathbf{x}}^j), & (2.56)\\ w^i_{k+1|k+1} &= \frac{w^i_{k+1|k} f_k(\mathbf{z}_k|\mathbf{x}^i)}{\Sigma_{j=1}^N w^j_{k+1|k} f_k(\mathbf{z}_k|\mathbf{x}^j)}. & (2.57) \end{aligned}$$

It is conjectured that this method will converge to the true Bayes filter as the number of grid cells increases. The potential barrier to using this method is that in order to achieve sufficiently fine resolution on a high dimensional state space, the number of grid cells, N, may need to be very large. In particular, (2.56) requires N^2 evaluations. This means that, in general and especially in higher dimensional spaces, a grid-based computation will be infeasible.

2.2 Performance of Single Target Filters

Several of the approximate methods for computing the Bayes filter described in Section 2.1.3 will be tested on a filtering problem relevant to single target tracking in sonar and radar with a linear forward model and a nonlinear measurement model. The aim of nonlinear filtering in this case is to infer the position and velocity of a target moving in a straight line with velocity which drifts with 2-D Brownian motion. The target state and the (linear) deterministic part of the forward model are

$$\mathbf{x}_k = (x_k, y_k, \dot{x}_k, \dot{y}_k), \tag{2.58}$$

$$f(\mathbf{x}_k) = (x_k + dt\dot{x}_k, \dot{x}_k, y_k + dt\dot{y}_k, \dot{y}_k), \tag{2.59}$$

where dt is the length of the time step between measurements. If it is assumed that noise in the forward model is zero mean, white and Gaussian, the appropriate covariance matrix Q_k for the noise is derived in [6] by discretising white noise in the velocity components from the continuous case giving

$$Q = q \begin{pmatrix} \frac{1}{3}dt^3 & \frac{1}{2}dt^2 & 0 & 0 \\ \frac{1}{2}dt^2 & dt & 0 & 0 \\ 0 & 0 & \frac{1}{3}dt^3 & \frac{1}{2}dt^2 \\ 0 & 0 & \frac{1}{2}dt^2 & dt \end{pmatrix}, \tag{2.60}$$

where q is the rate of variance, or *volatility*. For the tests below, noise in the forward model is assumed to be negligible for reasons that will be explained in Section 2.2.1.

The measurements are taken in range and bearing (or r and θ) so that the deterministic part of the measurement model is

$$h(\mathbf{x}_k) = (\sqrt{x_k^2 + y_k^2}, \arg(x_k + iy_k)). \tag{2.61}$$

Measurement noise is assumed to be zero-mean, white and Gaussian with covariance matrix

$$R = \begin{pmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\theta^2 \end{pmatrix}, \tag{2.62}$$

where σ_r and σ_θ are such that uncertainty in position due to bearing error is significantly larger than uncertainty due to range error (i.e. $r_0\sigma_\theta \gg \sigma_r$ where r_0 is a typical target range). This case is particularly relevant to active sonar tracking.

Traditional Filter Performance Evaluation

The standard practice when evaluating nonlinear filters for tracking is to compute

the root mean squared error between the true target state (which is known because it has been generated) and its estimated value, as computed by the nonlinear filter, over a number of Monte Carlo trials (see, for example, [15, ch. 3] [55] [69] [95] [142]). This was the approach taken in the author's transfer thesis [132]. In the performance comparison presented therein, it was demonstrated that as the tracking scenario parameters changed to make the posteriors more non-Gaussian, the performance of the UKF improved relative to the EKF. Similarly, the performance of particle filter methods improved relative to the UKF. These results are supported by the extensive range of examples presented in [95] for nonlinear, non-Gaussian tracking scenarios where the particle filter outperforms more traditional methods.

Whereas the performance comparison in [132] considered various parameters and their effect on performance, in this section this will be distilled down to one key parameter, σ_θ, for clarity of presentation. Figs. 2.1 and 2.2 show an example posterior density (computed with a uniform prior using a method to be described shortly) for $\sigma_\theta = 0.5^o$ and $\sigma_\theta = 5^o$. It can easily be seen that as σ_θ increases, the posterior becomes increasingly poorly approximated by a Gaussian density. The variation of this parameter is sufficient to have an impact on the performance of filters such as the EKF and UKF.

2.2.1 Test Details

Instead of comparing the filter's state estimate with the known truth, as in the traditional performance comparison methods, in this section the full posterior, computed by a selection of the nonlinear filters, will be compared with the 'correct posterior'. The computation of a correct posterior was previously considered to be computationally intractable in all but a few special cases. However, due to some constraints on the forward model, and access to modern hardware, it will be possible to compute a grid-based solution to the Bayes filter equations on an adaptive grid sufficiently dense to capture the posterior.

The constraint on the forward model is the absence of process noise. This simplifies (2.56) to

$$w^i_{k+1|k} \quad = \quad w^J_{k|k} f_{k+1|k}(\hat{\mathbf{x}^i}|\hat{\mathbf{x}^J}), \tag{2.63}$$

where J is such that $f^{-1}(\hat{\mathbf{x}}^i) \in \tilde{\mathbf{x}}^J$ where f is the deterministic term in the forward model as in equation (2.59). This eliminates the N^2 factor in the computation, but does restrict the direct applicability of the results here to cases where there is no

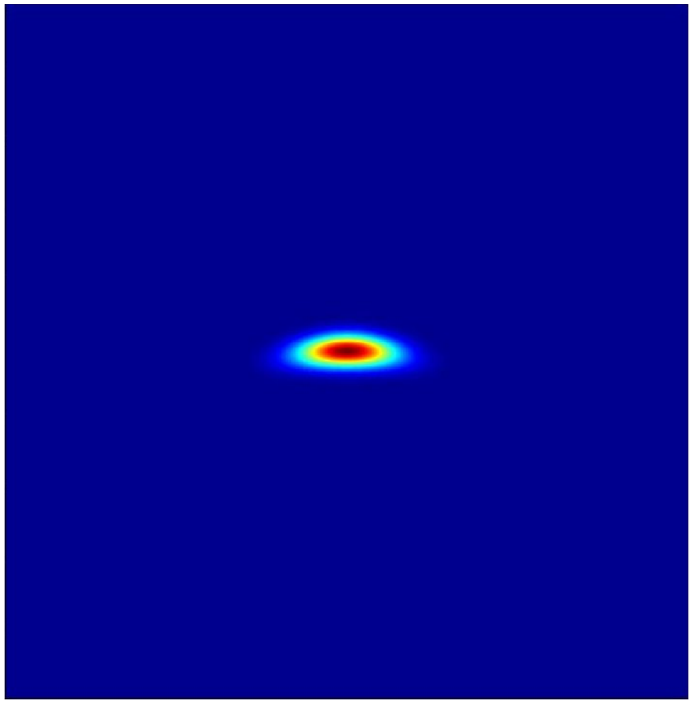


Figure 2.1: Example posterior computed with $\sigma_\theta = 0.5^o$

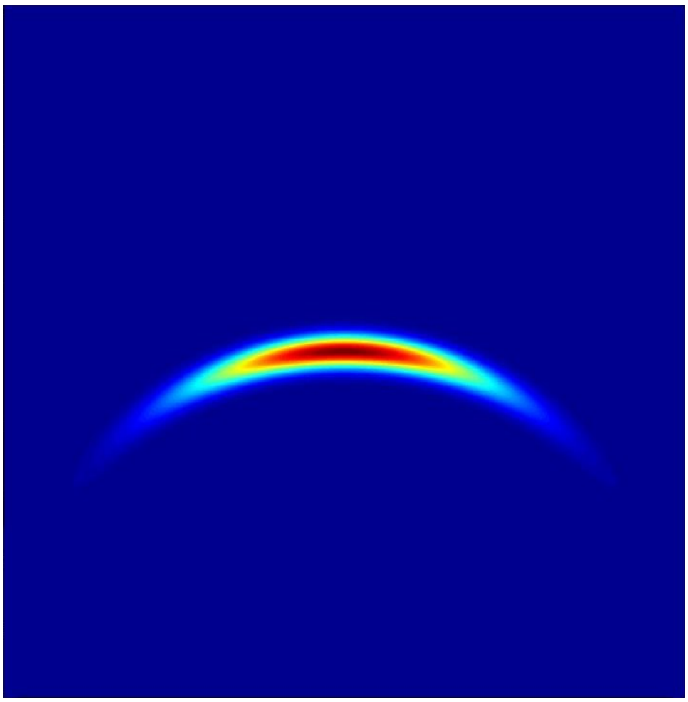


Figure 2.2: Example posterior computed with $\sigma_\theta = 5^o$

process noise. To further facilitate computation of correct posteriors, velocity is restricted to the x-direction.

Adaptive Grid Technique

In order to permit more efficient computation of the posterior, an adaptive grid method is proposed for a regular grid. The method consists of the following for each time step:

1. Compute the predicted density using (2.63).
2. Compute marginal densities in each dimension and determine the subset of each dimension where the probability density is non-negligible.
3. Compute the likelihood and its own marginal densities and determine the subset of each dimension where this is non-negligible.
4. Choose a new grid in the state space restricted to the region where both the predicted density and the likelihood are non-negligible.
5. Interpolate the predicted density and likelihood to the new grid
6. Compute the posterior using (2.57).

Thus, at each time step, the adaptive grid 'zooms' to the relevant subset of the state space. This method has proved effective at enabling a computation at sufficient resolution while using fewer grid points.

Distance Between Probability Densities

The measure used to quantify the distance between the probability densities will be an L_1 norm. Let p and q be two probability density functions, then the L_1-distance, sometimes called the total variation distance, between their respective probability densities is given by

$$L_1(p, q) = \int |p(\mathbf{x}) - q(\mathbf{x})| d\mathbf{x}. \tag{2.64}$$

There are other distances which could have been used, such as the Hellinger distance, but the L_1-norm is easy to compute and has the following equivalent definition:

$$L_1(p, q) = 2 \max_A |p(A) - q(A)|. \tag{2.65}$$

Here, the max is taken over all subsets, A, of the state space and $p(A)$ is the integral of the probability density function over the subset A. This result can easily be verified by choosing $A = \{\mathbf{x} : p(\mathbf{x}) > q(\mathbf{x})\}$ which clearly maximises $|p(A) - q(A)|$.

The L_1 norm lies between zero and two and is twice the maximum difference between the probabilities assigned over the subsets of the state space. It achieves the minimum value only when the probability densities are equal and maximum value when p assigns zero probability to every set to which q assigns a positive probability, and vice versa.

Computing Times

Computing times for methods discussed above were evaluated for a 10 time step run using a 3.99GHz Intel Core 2 Quad CPU. All times are measured in seconds. The results are shown in Table 2.1 in which M is the number of particles used for the particle filter and N is the number of cells used in the grid method.

Method	Time (s)
EKF	0.0004
UKF	0.002
PF	0.3 × (M/1000)
Grid	0.3 × (N/1000000)
Adapt. Grid	9.5 × (N/1000000)

Table 2.1: Computing times in seconds for various filters

There are a few points worth noting about the times for the grid methods.

- If the resolution is increased by a factor of A then the increase to the number of cells will be A^k where k is the dimension of the state space. This means that the computing times can go up rapidly as increased resolution is needed.

- The adaptive grid's larger computing time is mainly due to the interpolation (step 5).

- It can be observed from Table 2.1 that the adaptive grid is only useful if the number of cells can be reduced by a factor of 30. In a simple but representative example with active sonar with range 40km, the adaptive grid was able to narrow down the plausible locations from an 80km×80km grid to a 2km×4km grid within a few time steps, giving an 800-fold reduction in the number of

cells for equivalent resolution. This only takes position states into account, and further improvements could potentially be made with velocity. This suggests that the adaptive grid will help to reduce computation.

- An adaptive grid will tend to use fewer cells, and this will prevent RAM being a limiting factor as might be possible for large grids.

2.2.2 Results

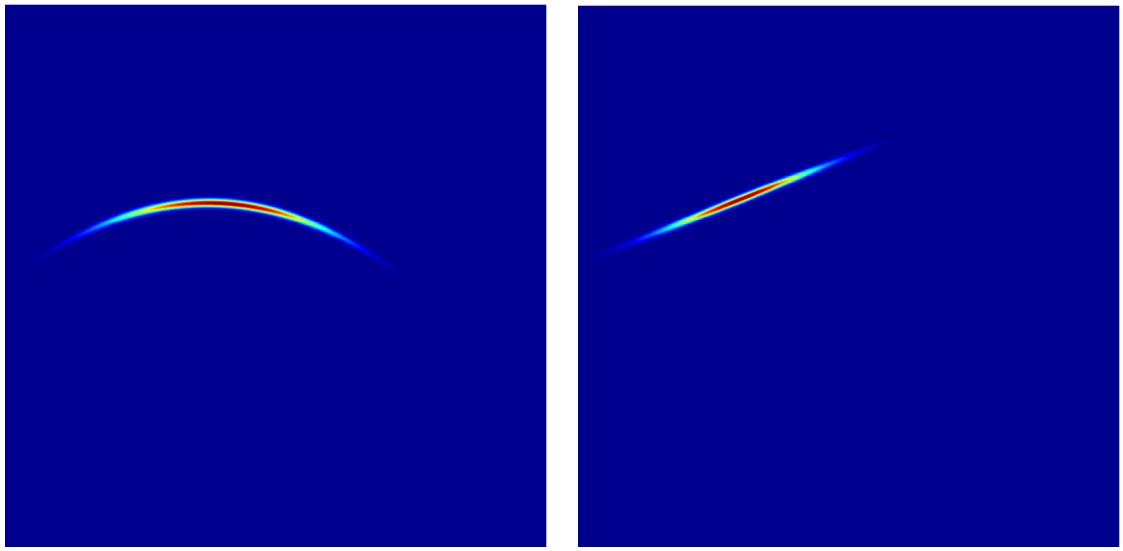
Fig. 2.3 shows some example densities for target location calculated for the same data using the adaptive grid, EKF, UKF and SMC in a case with $\sigma_\theta = 5^o$ and $\sigma_r = 20$m for a target with range 30000m. The posteriors computed using EKF and UKF illustrate the difficulty of capturing the posterior while restricted to the assumption of a Gaussian densities. The posterior for the EKF here assigns almost zero probability density to the maximum point of the true posterior. While UKF alleviates this problem it is clear that its density is too diffuse. Assuming the grid computation to be the true posterior, the L_1 errors of the EKF and UKF filters are 1.627 and 1.641 respectively, worryingly close to the maximum possible error of 2. The posterior density for the SMC filter was calculated using two-dimensional kernel density estimation [18] courtesy of code available at `http://www.maths.uq.edu.au/~botev/`. Visually, the SMC filter is clearly a better representation of the true posterior, but despite using 10^6 particles, the density calculated is still grainy and the L_1 error is 0.726. This is a significant improvement on EKF and UKF but demonstrates the difficulty of obtaining a very accurate posterior estimation in the L_1 sense using particles.

Verification of Adaptive Grid

In order to verify the ability of the adaptive grid filter to capture the true posterior density, it was tested in a scenario where the true posterior is known: the linear Gaussian case. A 10 time step run of the adaptive grid was run and the results were compared with the Kalman filter which is the true posterior in this case. The L_1 errors for the adaptive grid for this test are shown in Table 2.2. The results show that the adaptive grid method is able to accurately capture the posterior density.

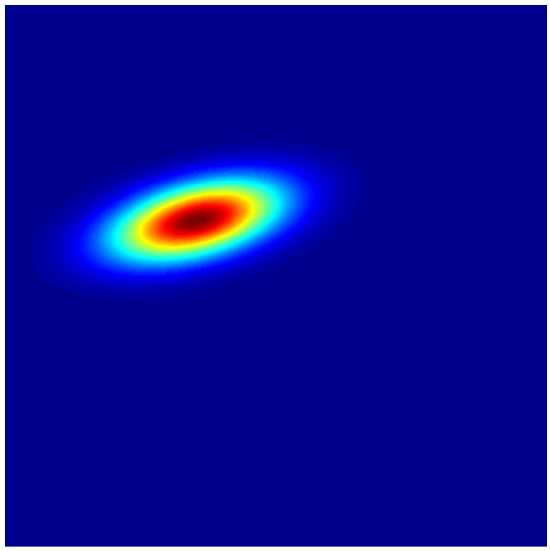
Time	1	2	3	4	5	6	7	8	9	10
L_1 error	0.003	0.003	0.001	0.002	0.002	0.002	0.003	0.003	0.0005	0.001

Table 2.2: The L_1 error in the adaptive grid approximation at each time step in a ten time step run.

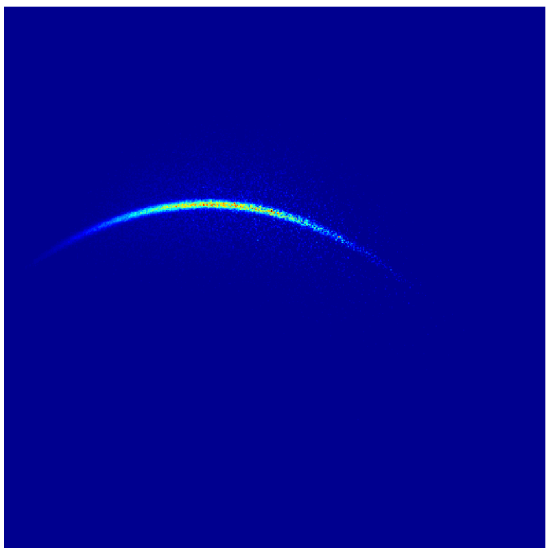


(a)

(b)



(c)



(d)

Figure 2.3: Example densities calculated using:
(a) Adaptive grid (b) EKF (c) UKF (d) SMC

L_1 Error in Posterior Densities

The L_1 errors in probability density were computed for 100 Monte Carlo runs of a 30 time step trajectory for EKF and UKF over a range of values of σ_θ. Difficulties in performing two-dimensional kernel density estimation in increasingly fine grids prevented the inclusion of SMC in the comparison.

A comparison of the average L_1 error for both the EKF and UKF is given in Table 2.3. It can be seen from these results that as σ_θ increases, the error in all estimated densities becomes worse but that UKF becomes the better approximation. Perhaps the most striking aspect of these results is that even for low values of σ_θ the probability densities computed are not a particularly good match to the true densities. For higher values of σ_θ the density errors are close to 2.

σ_θ	1	3	5	10
Error for EKF	0.9620	1.4223	1.7125	1.8971
Error for UKF	0.9406	1.2558	1.4177	1.7232

Table 2.3: Average L_1 errors for EKF and UKF over 100 Monte Carlo runs of 30 time steps.

Comparison with Cramer-Rao Lower Bound

The work so far in this section has focused on the accuracy of the approximate posteriors. A different method for evaluating filter performance is to consider the accuracy of the state estimates derived from a given filter as compared with a known ground truth. For practical purposes, this second method may be more useful.

In order to explore this in a rigorous manner, the Cramer-Rao Lower Bound (CRLB) is introduced. An excellent introduction to the CRLB for nonlinear filtering is given in [95, ch. 4]. Only a brief summary will be attempted here.

The CRLB provides a lower bound for the variance of an estimator. Let $\hat{\mathbf{x}}_k$ be an unbiased estimator for $\mathbf{x}_k$ calculated based on the measurements $\{\mathbf{z}_1, ..., \mathbf{z}_k\}$. The covariance of $\hat{\mathbf{x}}_k$, denoted $\hat{P}_k$ has a lower bound (the CRLB) expressed as

$$\hat{P}_k = \mathbb{E}\big[(\hat{\mathbf{x}}_k - \mathbf{x}_k)(\hat{\mathbf{x}}_k - \mathbf{x}_k)^T\big] \geq J_k^{-1}. \tag{2.66}$$

The inequality here is defined as meaning that $\hat{P}_k - J_k^{-1}$ is a positive semi-definite matrix. J_k here is referred to as the filtering information matrix. Details of how to compute the filtering information matrix are given in [95, ch. 4].

The CRLB cannot be easily calculated in the general nonlinear case but in the special case with no process noise, as in the test scenario, a formula for its computation

is given in [95, p. 76]. A test was conducted using all of the filters described above. There were 100 Monte Carlo runs of a 20 time step trajectory with $\sigma_\theta = 5^o$ and $\sigma_r = 20$m for a target with range 30000m. The grid used was $200 \times 200 \times 200$ and the number of particles used was 10^5. The standard deviation of the position components of the target state estimates were computed. These are shown alongside the corresponding CRLB in fig. 2.4. The results show that the grid filter attains the CRLB, as expected, whereas the EKF, UKF and SMC filter are above the CRLB with the EKF's performance significantly worse than the other two. Note that the errors at time step 1 are entirely dependent on the filter initialisation, which will be discussed shortly.

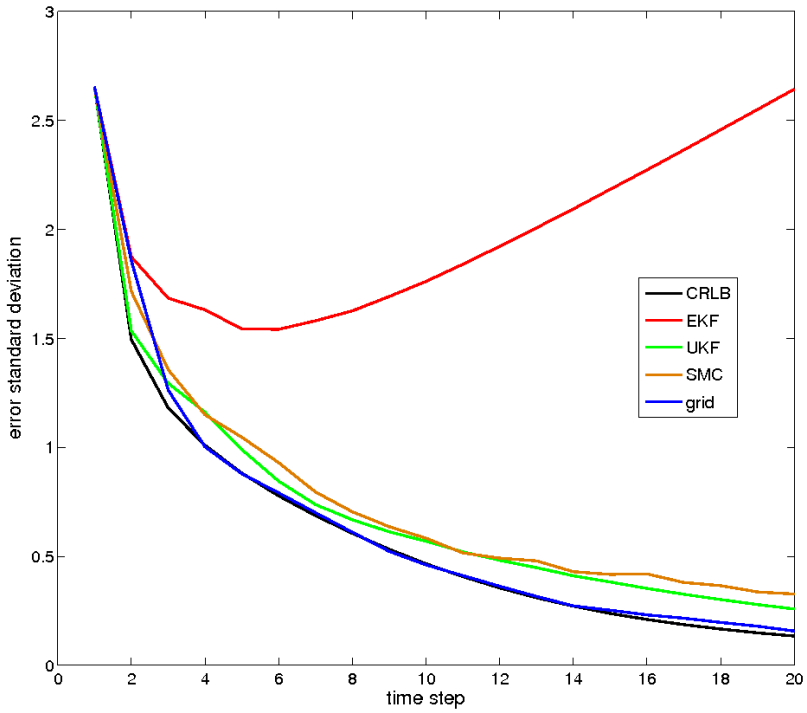


Figure 2.4: Comparison of standard deviation for position estimates for adaptive grid, EKF and UKF filters alongside corresponding CRLB.

Fig. 2.5 shows the ratio of the observed standard deviation in position of the target state estimates with the corresponding CRLB for a range of values of σ_θ averaged over 100 Monte Carlo tests for values of σ_θ between 1^o and 10^o. For low values of σ_θ the difference in performance is small. As σ_θ increases, it is immediately noticeable that the performance for EKF worsens significantly and so for scenarios with large σ_θ, EKF should be used only with extreme caution. An explanation for the rapid degeneration of performance by the EKF may be found by considering a rule of thumb

for performance of the EKF in [65]. For the parameters used in the simulations here, this rule of thumb predicts good performance of EKF for $\sigma_\theta < 1^o$. This agrees with results here, where the EKF matches the CRLB at 1^o but degenerates soon after. The performance of UKF also worsens as σ_θ increases, though less dramatically, and as its computational load is low by comparison with SMC and the grid method, it should be considered a viable alternative to EKF. The performance of SMC as compared to the CRLB did not worsen as σ_θ increases, and the grid method approximately attained the CRLB for all tests, as expected.

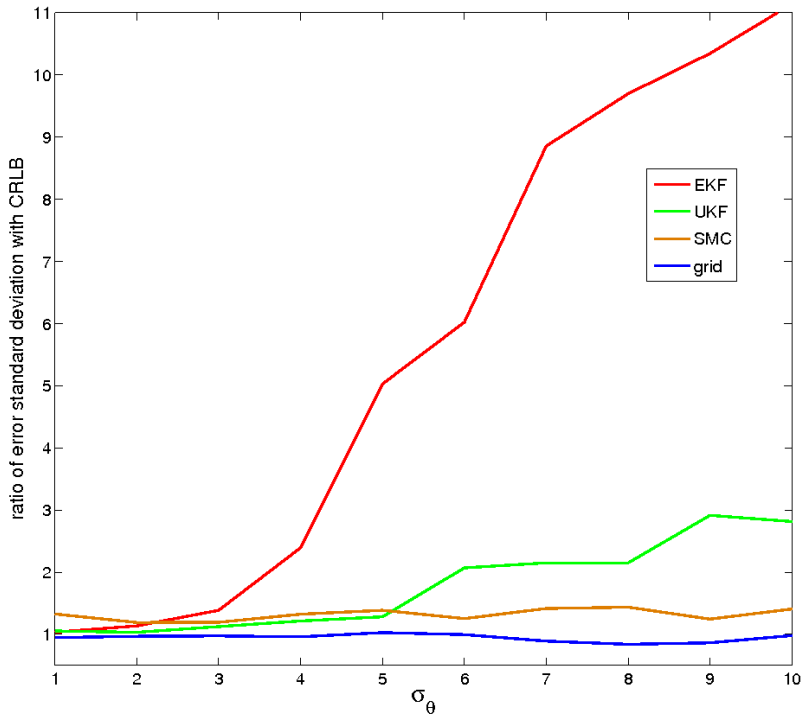


Figure 2.5: The ratio of error standard deviation to the corresponding CRLB for various σ_θ

2.2.3 Conclusions from Performance Comparison

The methodology presented here in evaluating the nonlinear filters for tracking took a different approach to the one taken in the author's transfer thesis [132], but the conclusions are broadly similar. That is to say, the SMC filter performs better than UKF which, in turn, performs better than EKF and these advantages in performance increase as bearing error in the measurements increases. This might be taken into account in the future filter design for tracking systems. Unlike the more simplistic

presentation in [132], the results here go some way to illuminating the reason for this difference in performance. The EKF is not a very good match at all to the correct posterior probability density; it is observed here assigning zero probability to the region around the maximum point of the true posterior. Given the results presented here, it is all the more striking that EKF is able to perform sufficiently well in general systems that it is considered the de facto standard for nonlinear estimation [95]. By contrast, the advantages of using Monte Carlo methods to approximate non-Gaussian probability densities are clear.

A final thing to bear in mind is that it was only possible to investigate a small number of filtering methods here. A recent survey by Li and Jilkov on tracking methods spans five parts and 179 pages [67], so it is only possible to scratch the surface in this chapter. Other possible methods for computing approximate solutions to the Bayes filter include Markov chain Monte Carlo [98] and the Ensemble Kalman filter [40].

While it will never be possible to investigate every possible combination it is hoped that the selection of methods presented here presents some insight into how well certain categories of methods will be able to approximate the true Bayesian posterior.

2.3 Initialisation

The subject of choosing a prior density for a filter (i.e. directly after first detection of the target) is not given much attention in the tracking literature, despite its obvious importance. In [94, ch. 2], a general suggestion is made for the case of using only the location of the first measurement, $\mathbf{z}_0$ (assuming that this is the target location corrupted by zero mean, Gaussian noise with variance σ^2). Assuming a Gaussian form of the prior with mean $\hat{\mathbf{x}}_0$, and covariance $\hat{P}_0$ and writing this in block form with the position coordinates occupying the top of the target state, and velocity coordinates on the bottom:

$$\hat{\mathbf{x}_0} = \begin{pmatrix} \mathbf{z}_0 \\ \mathbf{0} \end{pmatrix} \quad , \quad \hat{P}_0 = \begin{pmatrix} \sigma^2\mathbb{I} & \mathbf{0} \\ \mathbf{0} & (\frac{1}{2}v_{max})^2\mathbb{I} \end{pmatrix}. \tag{2.67}$$

So the idea is to decouple position and velocity, and to assume that the distribution for velocity is zero mean with variance equal to $(\frac{1}{2}v_{max})^2$ where v_{max} is a known maximum possible velocity.

This is a good starting point, and a simple nonlinear approximation (following the approach in [65]) to this choice of prior has been used in all of the test results

presented here as it is well-known and its Gaussian form permits equal comparison between those filters which can and cannot handle non-Gaussian densities. However, it is somewhat ad hoc and it should be possible to use Bayesian principles to construct a prior more rigorously.

A prior using the first measurement $\mathbf{z}_0$ will be constructed using the assumption that before the measurement arrives, there is a uniform density for target location over some surveillance region. As a result, the density for target location after receiving $\mathbf{z}_0$ is proportional to the likelihood function $f_0(\mathbf{z}_0|\mathbf{x})$. In the example from the testing section, writing $\mathbf{z}_0 = (r_0, \theta_0)^T$,

$$f_0(\mathbf{z}_0|\mathbf{x}) \propto \exp\left(-\frac{1}{2}\Big[\frac{1}{\sigma_r^2}\big((x^2+y^2)^{\frac{1}{2}} - r_0\big)^2 + \frac{1}{\sigma_\theta^2}\big(\arg(x+iy) - \theta_0\big)^2\Big]\right). \quad (2.68)$$

Therefore (2.68) should be used as the prior density (after receiving the first measurement) wherever possible. If using a filter which requires all densities to be Gaussian, the appropriate means and covariances can be computed by

$$\begin{aligned}
f_x(x) &= \int f_0(\mathbf{x}|\mathbf{z}_0)dy, && (2.69)\\
f_y(y) &= \int f_0(\mathbf{x}|\mathbf{z}_0)dx, && (2.70)\\
\text{mean}(x) &= \int x f_x(x)dx, && (2.71)\\
\text{mean}(y) &= \int y f_y(y)dy, && (2.72)\\
\text{Cov}(X,X) &= \int x^2 f_x(x)dx - \text{mean}(x)^2, && (2.73)\\
\text{Cov}(Y,Y) &= \int y^2 f_y(y)dy - \text{mean}(y)^2, && (2.74)\\
\text{Cov}(X,Y) &= \int xy f_0(\mathbf{x}|\mathbf{z}_0)dxdy - \text{mean}(x)\text{mean}(y). && (2.75)
\end{aligned}$$

There are generally no closed form expressions for these quantities but they can be quickly calculated. For the velocity components, use the same approach as in (2.67).

Preliminary Test

A preliminary test was carried out the verify the effectiveness of the choice of prior suggested in (2.68) as opposed to the traditional choice in (2.67). Two SMC filters were used on 100 Monte Carlo runs of the tracking scenario from the test in Section 2.2.2 with $\sigma_\theta = 5^o$ and 30 time steps. One SMC filter used the prior in (2.67) and the other used the prior in (2.68). The filter with the prior in (2.68) gave a 22% reduction

in error standard deviation as compared with the other method. This is a significant improvement in performance from only altering the prior at no extra computational cost. Therefore, the method for prior selection described above is both theoretically well-founded as it is based on a non-informative prior and provides an immediate improvement in results in a simple test.

2.4 Manoeuvre Handling

A manoeuvre is an abrupt change in target dynamics. Examples of manoeuvres include fighter planes making sudden sharp turns or submarines diving. Target dynamics during a manoeuvre cannot be modelled using the same forward model as for normal (for example, straight line) motion. A popular method for handling target manoeuvres is based on jump Markov linear systems (JMLS). It is assumed that target motion is governed by one of a finite set of motion models and that transition between these different motion models is a Markov process.

Let the motion mode at time k be denoted r_k where $r_k \in \{1, 2, ..., N\}$ and N is the finite number of possible motion modes. The Markov assumption on r_k gives

$$p(r_{k+1} = i | r_k = j) = \pi_k^{i,j}, \tag{2.76}$$

where $\pi_k^{i,j}$ is usually assumed known.

Restricting to the linear Gaussian case for simplicity of presentation, the forward and motion models are then

$$\mathbf{x}_{k+1} = F_k(r_k)\mathbf{x}_k + \mathbf{v}_{k+1} \qquad \mathbf{v}_{k+1} \sim \mathcal{N}(\mathbf{0}, Q_{k+1}), \tag{2.77}$$

$$\mathbf{z}_{k+1} = H_{k+1}(r_{k+1})\mathbf{x}_{k+1} + \mathbf{w}_{k+1} \qquad \mathbf{w}_{k+1} \sim \mathcal{N}(\mathbf{0}, R_k). \tag{2.78}$$

The Bayesian solution to the inference problem in this case becomes the process of jointly inferring the motion mode and the target state. Defining $\{M_k^l\}_{l=1}^{N^k}$ as the set of possible sequences of motion modes up to time k, so that for fixed k there are N^k possibilities, the Bayes filter for the target state is:

$$f_{k+1|k+1}(\mathbf{x}_k | Z^k) = \sum_{l=1}^{N_k} f(\mathbf{x}_k | M_k^l, Z^k) f(M_k^l | Z^k). \tag{2.79}$$

The term $f(\mathbf{x}_k | M_k^l, Z^k)$ is the probability density for a given motion mode sequence, and this can easily be calculated using (2.77-2.78) and a Kalman filter. The probability density over motion mode sequences $f(M_k^l | Z^k)$ can be calculated sequentially using

the innovation corresponding to each motion mode and the transition probabilities. This will be discussed in more detail in Section 5.2.2.

It can be seen from the form of (2.79) that in the linear, Gaussian case, the posterior is a Gaussian mixture with an exponentially increasing number of components. In order to retain computational tractability, it is necessary to approximate this. One possible scheme is to merge all motion mode sequences which are the same for the last n time steps. This is known as Generalised Pseudo-Bayes and often is often denoted GPB_n.

A more popular method is using interacting multiple models (IMM). This uses a mixing step to combine probabilities every time step. It has been shown [17] that IMM performs better than GPB_1 but slightly worse than GPB_2 while having equivalent computational complexity to GPB_1. This explains its popularity in the tracking literature. Full details of IMM may be found in [78].

In [132], an example was presented of tracking a target on a trajectory including several coordinated turn manoeuvres. It was demonstrated that it is possible to track the target successfully and maintain a probability density for the motion mode with a good correspondence to the ground truth even in the case where the precise parameters of the manoeuvre are not known. No further numerics will be presented here, but the handling of manoeuvres in the multitarget case is considered in Section 5.2.

Chapter 3

Multitarget Tracking: Background

Multitarget tracking is the process of jointly determining how many targets are present and their states from noisy sets of measurements. At each time step, a set of measurements is received and each measurement in this set is either a false alarm or is generated by a target. Both the target states and the target number may vary with time due to target motion and appearance or disappearance of targets. Furthermore, individual measurements may be corrupted by noise and target measurements may be missed.

The multitarget case is in contrast to the single target case in which a single measurement, corresponding to a particular known target, is received at each time step. To clarify this distinction, consider an example where it is known that exactly one target is present but where at each time step two measurements are received. Suppose that one of these measurements must be generated by the target (i.e. there are no missed detections) and the other must be a false alarm. Even though it is known that only one target is present, this example is still a multitarget tracking problem. The ambiguity about the origin of each of a set of measurements is the key point which characterises a multitarget tracking problem.

Broadly speaking, the aspects of the dynamical system and measurement process which characterise the multitarget tracking problem, and which make it challenging, may be summarised as follows:

- Dynamical System
  - Target dynamics are uncertain.
  - Targets may appear and disappear.
  - Targets may spawn other targets.
- Measurement Process

– Measurements are corrupted by noise.

– Target detections may be missed.

– There may be false alarms, or *clutter*.

The false alarms may be particularly troublesome as it is not unusual for there to be orders of magnitude more false alarms than true target detections in practical systems.

Multitarget tracking is an established area of research. Standard applications of multitarget tracking may be found in surveillance and navigation for sonar [1] [81] [23], radar [14] [141] [41] and infrared [16] [85] data. Further applications have been found in marine biology [61] [139], traffic monitoring [52] [9], navigation for autonomous vehicles and robots [64] [76], remote sensing [83] [112] and biomedical research [42] [46] in addition to the novel applications presented in Chapter 6.

This chapter will begin with an overview of the conventional approach to multitarget tracking and some of its disadvantages in Section 3.1. In Section 3.2 an introduction to random finite set theory is given which covers its aims, concepts and motivation as well as the crucial models without going into detail. More in-depth background of random finite set theory is given in Section 3.3 which contains an explanation of the relationship with point process theory as well as a derivation of the multitarget Bayes filter. The probability hypothesis density (PHD) filter, which propagates only the first-order moment of the posterior densities from the multitarget Bayes filter, is derived in Section 4.1. Finally, in Section 4.2 the details of the schemes for implementation of the PHD filter are given in detail.

3.1 Conventional Approach

Standard tracking methods such as the Kalman filter cannot be directly applied in the multitarget tracking case due to the ambiguity about which of the indistinguishable measurements should be used to update the target state. The conventional approach to handling this ambiguity is to split the problem up into two or three steps:

1. Data Association - deciding which of the measurements received will be used to update a given target.

2. Target update - using a standard single-target filter, usually a Kalman filter or nonlinear approximation.

3. *Track maintenance - some methods also require additional techniques to handle appearance and disappearance of targets.*

The standard textbooks on the conventional methods which are based on the use of data association to resolve the multitarget ambiguity are [6] and [15]. Some of the more popular methods will be briefly discussed below in the context of the Bayesian solution.

3.1.1 Single Target in Clutter

A special case of the multitarget tracking problem is that of a single target in clutter. In this case, as the number of targets is known to be one, it is only necessary to compute the probability density for the location of that target.

Let $Z_k = \{\mathbf{z}_{k,1}, ..., \mathbf{z}_{k,m_k}\}$ be the set of measurements received at time k where m_k is the number of measurements received and let Z^k be all of the sets of measurements received up to time k. Remembering that, at each time step, only one measurement can be due to the target and the rest must be false alarms, the Bayes filter can be used to calculate

$$f(\mathbf{x}_k|Z^k) = \sum_{j=1}^{m_k} \psi_{kj} f_{k,j}(\mathbf{x}_k|\mathbf{z}_{k,j}, Z^{k-1}), \tag{3.1}$$

where ψ_{kj} is the probability that the measurement $\mathbf{z}_{k,j}$ was generated by the target and $f_{k,j}(\mathbf{x}_k|\mathbf{z}_{k,j})$ is the Bayes filter update of $f(\mathbf{x}_{k-1}|Z^{k-1})$ subject to the relevant forward and measurement models and the measurement $\mathbf{z}_{k,j}$ as described in Section 2.1.

For conventional multitarget tracking implementations it is typically assumed that the forward and measurement equations are linear and that all noise is white and Gaussian. In this case, it has been shown in Section 2.1.2 that the Bayes filter reduces to the Kalman filter. This implies that the posterior density $f(\mathbf{x}_k|Z^k)$ will be a Gaussian mixture where the mixture components are easily computed.

Note that in cases with a nonlinear forward/measurement model, the nonlinear approximations of the Kalman filter as described in Section 2.1.3 are often used, but the description here will focus on the linear, Gaussian case only.

Now, $\psi_{kj} = p(\omega_j|Z_k)$ where ω_j is the hypothesis that the measurement $\mathbf{z}_{k,j}$ is the measurement generated by the target. Assuming a uniform prior over ω_j, as there is no prior reason to believe any measurement would be more or less likely than the others, gives:

$$\psi_{kj} \propto p(Z_k|\omega_j). \tag{3.2}$$

This may be calculated by considering the innovation, $\mathbf{v}_{k,j} = \mathbf{z}_{k,j} - H_k \mathbf{x}_{k|k-1}$, where H_k and $\mathbf{x}_{k|k-1}$ are defined as in the Kalman filter equations in Section 2.1.2. The innovation may be interpreted as the distance of the measurement from its expected position and it can be seen that its covariance is $S_k = H_k P_{k|k-1} H_k^T + R_k$, employing the notation from the Kalman filter equations. Therefore ψ_{kj} may be calculated as

$$\psi_{kj} \propto \mathcal{N}(\mathbf{v}_{k,j}; \mathbf{0}, S_k). \tag{3.3}$$

Therefore, the Gaussian mixture can, in principle, be calculated. However, there is an exponential explosion in the number of mixture components so, for practical implementations, there will need to be some approximation to the ideal solution. The most popular conventional methods for multitarget tracking will be described below. The discussion will highlight how although the methods are usually described in terms of a combination of data association with single target tracking, they may also be considered as approximations of (3.1) in order to avoid the exponential explosion.

Nearest Neighbour

The nearest neighbour (NN) filter [6] might be considered the simplest form of data association algorithm. It works by picking the measurement with the highest probability of being the true measurement as calculated by (3.3) and using this directly in the Kalman filter update. This could be interpreted in terms of (3.1) as using the weights $\hat{\psi}_{kj}$ where:

$$\hat{\psi}_{kj} = \begin{cases} 1 & \text{if} \quad j = \arg\max_i \psi_{ki}, \\ 0 & \text{o.w.} \end{cases} \tag{3.4}$$

The method is simple to implement and understand but is known to suffer from poor performance as the level of clutter increases [6] [15]. This should not be surprising as it is clear that as the level of clutter increases, the probability of making an incorrect association increases. A single incorrect association is often enough to lose a track.

Probabilistic Data Association

The probabilistic data association (PDA) [8] filter involves computing the Gaussian mixture in (3.1) directly. This mixture is then approximated using the best fitting single Gaussian density. This strategy prevents exponential explosion of Gaussian mixture components. The full details and derivation may be found in [8].

PDA is also able to take into account the possibility of a missed target detection and number of false alarms, λ, by propagating an additional hypothesis for the possibility that none of the measurements were generated by the target. Like the NN filter, PDA is fairly simple to understand and implement. It has been shown by testing on simulated data that PDA outperforms nearest neighbour [6] [15].

Note that another lesser known technique, known as mixture reduction and introduced in [104], is to approximate the Gaussian mixture in (3.1) by a Gaussian mixture with at most N components. In the case when $N = 1$ this reduces to PDA.

Note also that in order to reduce computational cost, it is standard practise when implementing PDA to reject measurements whose probability of being generated by the target is sufficiently low. This process is known as 'gating'.

3.1.2 Multiple Targets in Clutter

The methods discussed above only permit the tracking of a single target. If there is a known number of targets (greater than 1) present, it is necessary to make some adjustments to the nearest neighbour and probabilistic data association methods. Furthermore, if there is an unknown or time-varying number of targets it will be necessary to also include some method for track maintenance or to use the multiple hypothesis tracker, which will be described below.

Global Nearest Neighbour

Global Nearest Neighbour (GNN) is the simple extension of the NN filter to account for multiple targets. The adjustment is required to prevent more than one target being updated by a single measurement. This is achieved by defining a cost to the association of each track with each measurement and minimising the total cost over all possible sets of assignments. Typically this cost is defined by a statistical distance, d, between the measurement and track where

$$d^2 = \mathbf{v}^T S^{-1} \mathbf{v}, \tag{3.5}$$

and $\mathbf{v}$, S, are the innovation and innovation covariance respectively as defined in Section 3.1.1.

The set of assignments giving the lowest total cost may be computed using a 2D assignment algorithm such as the auction algorithm [12].

Joint Probabilistic Data Association

The joint probabilistic data association algorithm (JPDA) works in the same way as

probabilistic data association except for the computation of the association probabilities [7]. In JPDA, association of a single measurement to more than one target is prevented by using joint association (i.e. over the whole set of assignments) probabilities. The computation increases exponentially with the number of targets and false alarms, making implementation infeasible in general. There exist computationally tractable suboptimal approximations to JPDA for cases with a high number of targets or false alarms [99] [100].

Multiple Hypothesis Tracking

Multiple hypothesis tracking (MHT) [14] is an exhaustive search of measurement to track associations over a number of time steps. Whereas in JPDA the various hypotheses are merged at each time step, in MHT these hypotheses are all propagated forward in anticipation of subsequent data removing the ambiguity. Each of these hypotheses may lead to another set of hypotheses at the next time step and so on for subsequent time steps. It is clear that this process may lead to a combinatorial explosion and so much of the work required to implement MHT involves managing the number of hypotheses using techniques such as clustering, hypothesis and track pruning and track merging [15]. A comprehensive overview of the issues related to implementing MHT may be found in [15, ch. 16].

Blackman and Popoli [15, p. 373] describe the results of a test where the MHT implementation took "around 6 person months to develop" as opposed to only "a few days" for the NN and PDA equivalents. The additional work does give advantages though: it is well known that MHT is able to outperform NN and PDA filters significantly in high levels of clutter [15]. Furthermore, in MHT every measurement is considered to be either due to an existing target, clutter or a new target, so track initialisation and deletion are handled within MHT. This approach permits the tracking of an unknown time-varying number of targets which is not possible using GNN or JPDA alone. However, caution should be exercised in anticipating a certain performance from MHT as Panta et al. [89] note that "the performance of the MHT filter depends heavily upon the particular implementation... techniques that are ad-hoc in general".

3.1.3 Track Management

Of the methods discussed so far, only MHT provides a mechanism for initialisation and deletion of tracks. Thus, for the other methods it is necessary to include a track management method. This usually means starting a new track for every measurement

that is not associated with an existing target. Obviously it will not be useful for the algorithm to output a track for every measurement received, so a method of classifying those tracks which are newly appeared, or 'tentative', and those which are 'confirmed' is necessary.

'M out of N' - This is a basic method where a track is confirmed if it is updated by a new measurement for M out of the last N time steps where M and N are predetermined integers. It may also be desired to delete a track if it has not been updated for M_{del} out of the last N_{del} time steps where M_{del} and N_{del} are also predetermined.

Sequential Likelihood Ratio Test - This method involves computing, at every time step, the likelihood ratio, LR, defined to be

$$LR = \frac{p(\mathbf{z}|H_1)}{p(\mathbf{z}|H_0)}, \tag{3.6}$$

where $\mathbf{z}$ is the measurement that has been associated with the track in question and H_1, H_0 are the hypotheses that this track is and is not a true target respectively. The log of the product of the LRs over the life of the track gives a 'track score' which may be used to determine when to confirm or delete a target. Full details may be found in [15, Ch. 6].

Note that a scheme for a PDA filter with inbuilt track management, known as integrated probabilistic data association (IPDA), has been developed but is only applicable in the case where it is known that there are 0 or 1 targets [80].

3.1.4 Critique of Association Based Multitarget Tracking

In his PhD thesis, B.-T. Vo [128] raises an issue with the consistency of the conventional methods within the Bayesian paradigm. In particular the issue is with the computation of probabilities for association hypotheses using Bayes rule, which is a crucial part in PDA and MHT based multitarget tracking algorithms.

In the single target case, this computation is captured in (3.2). Implicit in this equation is that the discrete posterior density has been computed over $\Omega(Z)$, the set of all possible association hypotheses:

$$p(\omega|Z) = \frac{p(Z|\omega)p(\omega)}{p(Z)} \qquad \forall \omega \in \Omega(Z). \tag{3.7}$$

As Ω is dependent on Z this should be written explicitly as:

$$p(\omega(Z)|Z) = \frac{p(Z|\omega(Z))p(\omega(Z))}{p(Z)} \qquad , \tag{3.8}$$

and from this form, issues emerge. Namely, it is not clear that:
i) $p(\omega(Z))$ is a valid prior, since it includes future information.
ii) $p(Z|\omega(Z))$ is a valid likelihood as Z is conditioned on $\omega(Z)$ which in turn depends on Z.

Thus, the relationship between conventional methods and the Bayesian paradigm is not clear, and this provides part of the motivation for a full Bayesian treatment of the multitarget tracking problem.

3.2 Informal Introduction to Random Finite Sets

To understand the additional difficulty posed in multitarget tracking, as opposed to single target tracking, consider the measurement due to one target with state $\mathbf{x}$. In single target tracking, as discussed in Chapter 2, the measurement has the form of a vector $\mathbf{z}$ generated according to the likelihood function $L_{\mathbf{z}}(\mathbf{x})$. For example, in the case of the Kalman filter $L_{\mathbf{z}}(\mathbf{x}) = \mathcal{N}(\mathbf{z}; \mathbf{x}, R)$ where R is the measurement noise covariance matrix. Three realisations of $\mathbf{z}$ for a fixed $\mathbf{x}$ would be three vectors with various distances from $\mathbf{x}$.

On the other hand, for multitarget tracking, the measurement is a set of vectors $Z = \{\mathbf{z}_1, ..., \mathbf{z}_m\}$. Three realisations of this measurement, for fixed target state $\mathbf{x}$, might be sets containing 30, 40 and 50 vectors respectively, each of which may or may not contain one vector, $\mathbf{z}_i$, whose generation was dependent on $\mathbf{x}$ according due the likelihood $L_{\mathbf{z}}(\mathbf{x})$. All other vectors in the set will be false alarms. It is clear that, in the multitarget case, there is a more general source of uncertainty surrounding the generation of the measurement than in the single target case.

Similarly, there is a more general form of uncertainty in the forward step of multitarget tracking as opposed to single target tracking. In the Kalman filter, for example, the prediction step involves computing the mean and covariance for the state of exactly one target. In multitarget tracking, this is not sufficient as it does not take into account the possibility of the target disappearing or a new target appearing.

In order to handle the multitarget tracking problem rigorously, it is necessary to characterise these more general sources of uncertainty systematically. This is done using random finite sets.

Random Finite Sets

A random finite set is a set with a random number of elements which are themselves random [75]. Concrete examples of random finite sets could include the locations

of raindrops falling on a paving stone in a fixed period of time or stars visible in a randomly selected region of the sky. another example of a random finite set that will come into use in Chapter 6 is a Poisson random finite set with uniform density on some region. To generate realisations from this random finite set, one must first generate a realisation from the Poisson distribution, say n, and then generate n realisations from the uniform distribution.

The concept of random finite sets is useful in modelling the fact that in the multitarget tracking problem it is not known *how many* targets there are or *where* they are. Let $X_k = \{\mathbf{x}_{k,1}, ..., \mathbf{x}_{k,n_k}\}$ be the set of target state vectors and $Z_k = \{\mathbf{z}_{k,1}, ..., \mathbf{z}_{k,m_k}\}$ the set of measurements, both at time k, where n_k and m_k are the number of targets and measurements respectively. These are random finite sets because the size of each of the sets is random and their member vectors are also random. X_k will henceforth be referred to as the multitarget state and Z_k the multitarget measurement. This terminology suggests a different way of thinking about the multitarget tracking problem. Whereas in the conventional methods each individual target state was considered separately, in the random finite set approach the multitarget tracking problem is regarded as attempting to determine X_k, the full set of target states, given Z_k [74]. This shift of approach is illustrated in fig. 3.1.

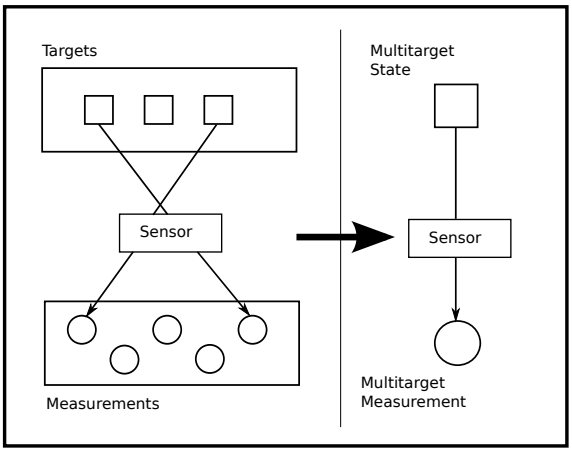


Figure 3.1: Illustration of the concept whereby the multitarget tracking problem is transformed into a single target problem by combining all target states into a 'meta-state' and all measurements into a 'meta-measurement'. The particular example on the left shows a case with three targets generating two measurements and with three false alarms.

There are two conceptual advantages to this approach for modelling the multitarget tracking problem. The first is the aforementioned ability to model the randomness

in the number of targets present directly. The second is that there is no inherent ordering of either targets or measurements, so a set representation is natural. Further to this point, consider the example $X = \{\mathbf{x}_1, \mathbf{x}_2\}$ representing a multitarget state where there are two targets with state vectors $\mathbf{x}_1$ and $\mathbf{x}_2$. If this multitarget state were instead expressed in vector form it could be written as $(\mathbf{x}_1, \mathbf{x}_2)$ or $(\mathbf{x}_2, \mathbf{x}_1)$ with both representing the same underlying physical state. This means that in vector form, there is a redundancy in the underlying multitarget state space [74].

With this formulation of the multitarget tracking problem the goal is to compute, using Bayes' rule, the posterior density for X_k conditioned on $Z^k = (Z_1, ..., Z_k)$. This may be computed sequentially using the Bayes' filter which gives:

$$f_{k+1|k+1}(X|Z^{k+1}) = \frac{f_{k+1}(Z_{k+1}|X) f_{k+1|k}(X|Z^k)}{f_{k+1}(Z_{k+1}|Z^k)}, \tag{3.9}$$

$$f_{k+1|k}(X|Z^k) = \int f_{k+1|k}(X|X') f_{k|k}(X'|Z^k) \delta X', \tag{3.10}$$

where $f_{k+1}(Z_{k+1}|X)$ is the multitarget likelihood function and $f_{k+1|k}(X|X')$ is the multitarget Markov density. Multitarget likelihoods and Markov densities are analogous to their single target equivalents but must also take into account the more general forms of uncertainty of the multitarget case. The statisticians Naylor and Smith [82] warn that "the implementation of Bayesian inference procedures can be made to appear deceptively simple". This is highly relevant here as there is clearly more work to do before the Bayes filter in (3.9-3.10) can be implemented. For example, an integral with respect to a set (i.e. $\int \delta X'$) must be defined and the multitarget Markov and likelihood functions must be determined. The work addressing these issues will be presented in Section 3.3.

Multitarget Forward and Measurement Models

The following are the basic forms of the forward and measurement models taking into account the most fundamental aspects of the multitarget scenario:

$$X_{k+1} = S_k(X_k) \cup B_k(X_k) \cup \Gamma_k, \tag{3.11}$$

$$Z_{k+1} = \Theta_{k+1}(X_{k+1}) \cup K_{k+1}, \tag{3.12}$$

where S_k is the set of surviving targets, B_k is the set of spawned targets, Γ_k is the set of appearing targets, Θ_{k+1} is the set of target generated measurements and K_k is the set of clutter generated measurements. Each of these terms must, in turn, include models for the relevant aspects of the problem. For example, S_k must include a model for the forward motion of individual targets and the probability of

disappearance. Similarly Θ_{k+1} must include a model for the measurement process and the probability of detection. More detail will be given in Section 3.3.4 for the measurement model and Section 3.3.5 for the forward model. The models themselves are simple but they incorporate all of the relevant phenomena which make the multitarget tracking problem difficult, as summarised at the beginning of this chapter. These models will be used in the derivation of the multitarget Bayes filter in Section 3.3.

Approximation and Implementation

Like the standard Bayes filter, the multitarget Bayes filter turns out to be computationally difficult and requires approximation in general (see the discussion in Section 3.3.6). A classical method for approximating the Bayes filter is the moment matching approach and so this is the approach taken in approximating the multitarget Bayes filter. The first moment of the multitarget posterior, the probability hypothesis density (PHD), is propagated in the PHD filter [73] which will be presented in Section 4.1. The PHD filter is also computationally challenging in general. Approximations based on sequential Monte Carlo (SMC-PHD) [124] and Gaussian mixtures (GM-PHD) [122] have been developed and their convergence to the PHD filter is established in [30] and [26] respectively. The SMC-PHD and GM-PHD filters are presented in Section 4.2. This summarises the basics of implementation of the multitarget Bayes filter, though a number of extensions will be discussed in Chapter 5.

Why Random Finite Sets?

It is worth considering briefly before moving on what are the benefits of rejecting the conventional multitarget tracking methods in favour of a formulation using random finite sets. Some of the conventional methods described in Section 3.1 rely on a series of approximations. For example, the replacement of association probabilities (3.2) by (3.4), the use of EKF instead of the Bayes filter for nonlinear tracking or the 'M out of N' criteria for track management. The more complex multitarget tracking methods, such as MHT, require pruning and merging strategies to allow manageable levels of computation. Following these approximations, the relationship between the practical tracking solution and the optimal filtering solution may be obscure and performance difficult to predict. It may also be difficult to diagnose poor performance. Furthermore, as discussed in Section 3.1.4, the computation of association probabilities required in most data association algorithms does not have a clear relationship with the desired Bayesian framework.

In contrast, the random finite sets framework permits explicit modelling of the fact that the number of targets and their states are unknown and includes explicit statistical models for all of the phenomena relevant to the multitarget tracking problem as summarised in (3.11-3.12). Approximations are still necessary, but the approximations made are principled, following the well-known moment-matching approach, and have well-understood convergence properties.

3.3 The Multitarget Bayes Filter

The background to random finite sets and the derivation of the multitarget Bayes filter will be presented in a way which emphasizes the parallels between multitarget and conventional probability theory. There will also be an emphasis on the connections between random finite set theory and point process theory. The presentation here is broadly based upon Chapters 11-14 in the textbook by Mahler [75]. For ease of reference, the textbook [75] will be referred to as M1 for the rest of this chapter.

3.3.1 Connection to Point Process Theory

In [72], Mahler writes that "[Random finite set theory] is essentially a judicious, engineering-oriented distillation of point processes and related concepts...". In order to better understand the tools used in the derivation of the multitarget Bayes filter, the relevant elements of point process theory will be briefly presented.

In the language of point process theory, a random finite set is a simple finite point process (defined below). The following conditions which must be satisfied by a finite point process are given in [32, p. 121]:

1. The points are located in a complete separable metric space $\mathcal{X}$ (accordingly, it may be assumed henceforth that $\mathcal{X} = \mathbb{R}^d$).

2. There exists a probability density $\{p_n\}_{n\in\mathbb{N}}$ for the number of points in the process.

3. There exists a family of probability mass functions Π_n for $n = 1, 2, ...$ on the Borel sets of $\mathcal{X}^n = \underbrace{\mathcal{X} \times ... \times \mathcal{X}}_{n}$ for the joint distribution of the position of the points of the process given that the total number of points in the process is n.

A *simple* finite point process also requires the condition [32, p. 44]:

4. The number of points of the process at any location in $\mathcal{X}$ is either 0 or 1 with probability 1.

This final condition is relevant for random finite sets because the number of targets at one point in space must be either 0 or 1.

Janossy Densities

The joint probability distribution Π_n is defined on the Borel sets of $\mathcal{X}^n$ and so is determined by the values of $\Pi_n(A_1 \times ... \times A_n)$ where $A_i \subseteq \mathcal{X}$ for $i = 1, ..., n$. Noting that point processes apply to unordered sets, it is generally necessary to work with a symmetrised form of this distribution. Hence, the Janossy measure, J_n, is defined by [32, p. 122]:

$$J_n(A_1 \times ... \times A_n) = p_n \sum_{\sigma} \Pi_n(A_{\sigma(1)} \times ... \times A_{\sigma(n)}), \tag{3.13}$$

where the sum is taken over permutations $\sigma : \{1, ..., n\} \to \{1, ..., n\}$. This ensures that J_n is symmetric with respect to permutations of the points $(\mathbf{x}_1, ..., \mathbf{x}_n)$. Thus, an interpretation of the Janossy measure in the case where $A_1, ..., A_n$ are pairwise disjoint is that $J_n(A_1 \times ... \times A_n)$ is the probability that there are exactly n points in the process and that one point of the process lies in each of the sets A_i. Assuming that J_n admits a density with respect to the Lebesgue measure on $\mathcal{X}^n$, this density is referred to as the Janossy density and denoted j_n. The Janossy density is such that

$$\int_{A_1 \times ... \times A_n} j_n(\mathbf{x}_1, ..., \mathbf{x}_n) d\mathbf{x}_1 ... d\mathbf{x}_n = J_n(A_1 \times ... \times A_n). \tag{3.14}$$

Due to the $n!$ possible permutations of the elements of $\{\mathbf{x}_1, ..., \mathbf{x}_n\}$

$$\int_{\mathcal{X}^n} j_n(\mathbf{x}_1, ..., \mathbf{x}_n) d\mathbf{x}_1 ... d\mathbf{x}_n = p_n n!. \tag{3.15}$$

For simple finite point processes, the Janossy measure allots zero probability to events where $\mathbf{x}_i = \mathbf{x}_j$ for $i \neq j$ [32, p. 134]. Hence, the convention is taken that $j_n(\mathbf{x}_1, ..., \mathbf{x}_n) = 0$ whenever $\{\mathbf{x}_1, ..., \mathbf{x}_n\}$ are not distinct.

Random Finite Sets vs Conventional Point Process Theory

In [72], Mahler offers three reasons for using random finite set theory rather than conventional point process theory. Firstly, random finite set theory is explicitly geometric and amenable to visualisation as the random variables in question are actual sets of observations rather than abstract integer-valued measures. Secondly, random

finite set theory allows a unification of multitarget filtering with expert systems theory (see [M1] for more on this). Thirdly, the random finite sets perspective results in a formulation of point process theory directly analogous to conventional probability theory, which has long been established as the methodology of choice for single target tracking (see Chapter 2).

To elaborate on this final point, this section will go on to define multitarget analogues for the probability mass and density functions. In Sections 3.3.4 and 3.3.5, multitarget analogues to the likelihood function and Markov density will be derived permitting the calculation of a multitarget Bayes filter. Hence, a multitarget tracking methodology which parallels the single target tracking methodology will be developed.

3.3.2 Definitions and Preliminary Results

Before proceeding with the derivation of the multitarget Bayes filter some definitions are needed along with some key results linking the probability theory of random finite sets with well known results from conventional probability theory. Further technical details on the relationship between finite set statistics and measure theoretic probability theory may be found in Appendix A.

Multitarget Mass and Density Functions

The *belief mass function* of a random finite set X is [M1, p. 360]

$$\beta_X(S) := \Pr(X \subseteq S), \tag{3.16}$$

where S is a general subset of the underlying state space for targets, $\mathcal{X}$ (i.e. unlike X it is not necessarily finite). A function $f(X)$ on finite set variable X is said to be a *multitarget probability density function* if [M1, p. 362]

- The unit of measurement of $f(X)$ is $u^{-|X|}$ where u is the unit of measurement in $\mathcal{X}$ and $|X|$ denotes the cardinality (number of elements) of X.
- $f(X) \geq 0$ for all X.
- $\int_{\mathcal{X}} f(X)\delta X = 1$.

where the set integral $\int_S f(X)\delta X$ of a function over a general subset, S, will be defined shortly.

The *probability density function* for a random finite set X, if it exists, is the function $f_X(X)$ such that [M1, p. 363]

$$\int_S f_X(X)\delta X = Pr(X \subseteq S) = \beta_X(S), \tag{3.17}$$

for all $S \subseteq \mathcal{X}$. Note that if $S_1 \subseteq S_2$ then $\beta_X(S_1) \leq \beta_X(S_2)$ and so β_X, the belief mass function is analogous to the cumulative distribution function from standard probability, and f_X is analogous to the standard probability density function.

The probability density function, $f_X(X)$ of a random finite set, X, must fully characterise the uncertainty in the cardinality of X and the location of all of the elements of X. It does so in the following way:

$$f(\{\mathbf{x}_1, ..., \mathbf{x}_n\}) := f_n(\mathbf{x}_1, ..., \mathbf{x}_n) = j_n(\mathbf{x}_1, ..., \mathbf{x}_n) \tag{3.18}$$

where j_n is the Janossy density of the equivalent simple finite point process. Note that as $j_n(\mathbf{x}_1, ..., \mathbf{x}_n)$ is symmetric in its arguments, the fact that f is a function of an unordered set causes no problems.

An example of the construction of such a probability density function and associated belief mass function is given in Section 3.3.3.

Set Integrals and Derivatives

The *set integral* of a given real-valued function $f(X)$ of a finite set variable X over the region $S \subseteq \mathcal{X}$ is [M1, p.361]:

$$\int_S f(X)\delta X := \sum_{n=0}^{\infty} \frac{1}{n!} \int_{S^n} f(\{\mathbf{x}_1, ..., \mathbf{x}_n\}) \mathrm{d}\mathbf{x}_1 ... \mathrm{d}\mathbf{x}_n \tag{3.19}$$

$$= f(\emptyset) + \int_S f(\{\mathbf{x}\}) \mathrm{d}\mathbf{x} + \frac{1}{2} \int_{S \times S} f(\{\mathbf{x}_1, \mathbf{x}_2\}) \mathrm{d}\mathbf{x}_1 \mathrm{d}\mathbf{x}_2 + ... \tag{3.20}$$

If f is a probability density function, then by (3.18) and (3.15)

$$\int_S f(X)\delta X = f(\emptyset) + \sum_{n=1}^{\infty} \frac{1}{n!} \int_{S^n} f_n(\mathbf{x}_1, ..., \mathbf{x}_n) \mathrm{d}\mathbf{x}_1 ... \mathrm{d}\mathbf{x}_n \tag{3.21}$$

$$= p_0 + \sum_{n=1}^{\infty} p_n. \tag{3.22}$$

Therefore, it can be seen that $\int f(X)\delta X = 1$ automatically whenever f is a probability density function having the form prescribed in (3.18).

Note that $f(\emptyset)$ must be a dimensionless probability. If the unit of measurement in $\mathcal{X}$ is u, then $\mathrm{d}\mathbf{x}_1 ... \mathrm{d}\mathbf{x}_n$ has units u^n. Therefore $f(X)$ must have a form such that $f_n(\mathbf{x}_1, ..., \mathbf{x}_n)$ has units u^{-n} for (3.19) to be well defined. This is automatically satisfied whenever f is a probability density function.

Note also that set integrals originate from the theory of statistical mechanics (see, for example, [47, pp. 234,266]).

Set derivatives and functional derivatives will be briefly defined. Further discussion of the link between the two is given in [M1, ch. 11] The functional derivative of a functional $F[h]$ is defined to be the gradient derivative in the direction of the Dirac delta function $\delta_{\mathbf{y}}$ given as [M1, p. 375]

$$\frac{\delta F}{\delta \mathbf{y}}[h] := \lim_{\epsilon \downarrow 0} \frac{F[h + \epsilon \delta_{\mathbf{y}}] - F[h]}{\epsilon}, \tag{3.23}$$

assuming this limit exists. The functional derivative with respect to the finite set Y is defined as [M1, p.376]

$$\frac{\delta F}{\delta Y}[h] := \begin{cases} F[h] & \text{if } Y = \emptyset, \\ \dfrac{\delta^n F}{\delta \mathbf{y}_1 ... \delta \mathbf{y}_n} & \text{if } Y = \{\mathbf{y}_1, ..., \mathbf{y}_n\}, \end{cases} \tag{3.24}$$

where the iterated derivative is defined in the obvious way as

$$\frac{\delta^n F}{\delta \mathbf{y}_1 ... \delta \mathbf{y}_n} := \frac{\delta}{\delta \mathbf{y}_n} \frac{\delta^{n-1} F}{\delta \mathbf{y}_1 ... \delta \mathbf{y}_{n-1}}, \tag{3.25}$$

assuming that these derivatives exist and that this derivative is independent of the order of differentiation so it is indeed dependent on the set Y.
N.B.1 The concept of a functional derivative defined in this way originated in quantum field theory [102, pp. 173-174].
N.B.2 The notation employed in (3.23) is non-standard [75, p. 382]. There are two possible standard notations for this functional derivative given by $\delta F / \delta h(\mathbf{y})[h]$ and $\delta F / \partial \delta_{\mathbf{y}}[h]$. The notation in (3.23) is an abbreviation of these.

The set derivative of a set function $\phi(S)$ is defined as [M1, p. 381]:

$$\frac{\delta \phi}{\delta \mathbf{y}}(S) = \lim_{|E_{\mathbf{y}}| \downarrow 0} \frac{\phi(S \cup E_{\mathbf{y}}) - \phi(S)}{|E_{\mathbf{y}}|}, \tag{3.26}$$

where $E_{\mathbf{y}}$ is a small neighbourhood of $\mathbf{y}$ with (hyper)volume denoted $|E_{\mathbf{y}}|$ assuming this limit exists and is independent of the shape of $E_{\mathbf{y}}$. The general set derivative is

$$\frac{\delta \phi}{\delta Y}(S) := \begin{cases} \phi(S) & \text{if } Y = \emptyset, \\ \dfrac{\delta^n \phi}{\delta \mathbf{y}_1 ... \delta \mathbf{y}_n} & \text{if } Y = \{\mathbf{y}_1, ..., \mathbf{y}_n\}, \end{cases} \tag{3.27}$$

when this exists with the iterated derivative defined in the same obvious way as in (3.25).

Results Needed to Derive the Multitarget Bayes Filter

This will be a brief presentation of the results required to derive the multitarget likelihood functions and Markov densities. A comprehensive list of basic rules for handling set derivatives is available in [M1, ch. 11] and a detailed derivation of the results is available in [43].

The *fundamental theorem of multitarget calculus* states that the set integral and set derivative as defined in (3.19) and (3.27) are inverse operations. Thus [M1, p. 384],

$$\phi(S) = \int_S \frac{\delta\phi}{\delta Y}(\emptyset)\delta Y, \tag{3.28}$$

$$\left[\frac{\delta}{\delta Y}\int_S f(W)\delta W\right]_{S=\emptyset} = f(Y). \tag{3.29}$$

The *multitarget Radon-Nikodým theorem* states that [M1, p. 385]

$$\int_S \frac{\delta\beta_\Psi}{\delta Y}(\emptyset)\delta Y = \beta_\Psi(S), \tag{3.30}$$

and hence that the probability density function for a random set Ψ can be constructed as

$$f_X(Y) = \frac{\delta\beta_X}{\delta Y}(\emptyset). \tag{3.31}$$

The *fundamental convolution rule* states that if $X = X_1 \cup ... \cup X_n$ where $X_1, ..., X_n$ are statistically independent subsets then the probability density of X is given by [M1, p. 385]

$$f_X(Y) = \sum_{W_1 \oplus ... \oplus W_n = Y} f_{X_1}(W_1) \cdots f_{X_n}(W_n), \tag{3.32}$$

where the summation is over all partitions of Y into pairwise disjoint subsets $W_1, ..., W_n$.

Other results concerning set derivatives are summarised below:

1. *Chain rule:* Let f be a real valued function on $\mathcal{X}$ and ϕ a set function. Then [M1,p. 390]
$$\frac{\delta}{\delta \mathbf{y}} f(\phi(S)) = f'(\phi(S)) \cdot \frac{\delta\phi}{\delta \mathbf{y}}(S). \tag{3.33}$$

2. *Sum rule:* Let a_1, a_2 be real numbers and ϕ_1, ϕ_2 set functions. Then [M1, p. 388]
$$\frac{\delta}{\delta Y}\big(a_1\phi_1 + a_2\phi_2\big) = a_1\frac{\delta\phi_1}{\delta Y} + a_2\frac{\delta\phi_2}{\delta Y}. \tag{3.34}$$

3. *Constant rule:* Let ϕ be a constant set function. Then [M1, p. 387]

$$\frac{\delta}{\delta Y}\phi = \begin{cases} \phi \text{ if } Y = \emptyset, \\ 0 \text{ o.w.} \end{cases} . \tag{3.35}$$

4. *Linear rule:* Let f be a real valued function on $\mathcal{X}$ and let $p_f(S) = \int_S f(\mathbf{y})d\mathbf{y}$ for all S. Then [M1, p. 387]

$$\frac{\delta}{\delta Y}p_f(S) = \begin{cases} p_f(S) \text{ if } & Y = \emptyset, \\ f(\mathbf{y}) \text{ if } & Y = \{\mathbf{y}\} \text{ and } \mathbf{y} \notin S, \\ 0 \text{ if } & o.w. \end{cases} \tag{3.36}$$

5. *General product rule:* Let $\phi_1, ..., \phi_n$ be set functions. Then [M1, p. 389]

$$\frac{\delta}{\delta Y}(\phi_1(S)...\phi_n(S)) = \sum_{W_1 \oplus ... \oplus W_n = Y} \frac{\delta \phi_1}{\delta W_1}(S)...\frac{\delta \phi_n}{\delta W_n}(S), \tag{3.37}$$

where the summation is over all partitions of Y into pairwise disjoint subsets $W_1, ..., W_n$.

3.3.3 Example Calculation of a Multitarget Bayes Filter

This will be a presentation of the calculation of the multitarget Bayes filter in a simple scenario to help illustrate some of the concepts in the preceding section. In order to permit a simple closed form solution, the following assumptions will be made:

- The target state space is $\mathbb{R}$.
- The Markov transition density for a target with state w is given by $f(x|w) = \mathcal{N}(x; w, q)$.
- There is no target disappearance.
- At each time step a single new target appears with probability p_B. The location of the new target has probability density $b(x)$. No more than one target can appear during a single time step.
- The likelihood function for a measurement z made of a target with state x is $L_z(x) = \mathcal{N}(z; x, r)$.
- There are no missed detections.

- At each time step a single false alarm is received with probability p_{FA}. The location of the false alarm has probability density $\kappa(z)$. No more than one false alarm may be received during a single time step.

These assumptions give a tracking scenario which is simple enough that it will be possible to write down the multitarget likelihood and Markov densities and use these to calculate the multitarget Bayes filter using (3.9)-(3.10).

The Bayes filter will be calculated for the following data:

- *Time step 0:* It is known that there are no targets present.
- *Time step 1:* A set, Z_1, containing one measurement, $\{z_1\}$ is received.
- *Time step 2:* A set, Z_2, containing three measurements, $\{z_2, z_3, z_4\}$ is received.

Time Step 1

It is possible to use the assumptions to write down the prior probability density $f_{0|0}(X)$, the multitarget Markov density $f_M(X|W)$ in the case where $W = \emptyset$ and the multitarget likelihood $f_L(Z|X)$ in the case where $Z = \{z_1\}$. Note that throughout this section, the notation $X = \{x_1, ..., x_n\}$ should be taken to imply that $|X| = n$.

Firstly, it is clear that the prior is given by

$$f_{0|0}(X) = \begin{cases} 1 \text{ if } X = \emptyset, \\ 0 \text{ if } |X| > 0. \end{cases} \tag{3.38}$$

Using the assumptions about target appearance, it can also been seen that

$$f_M(X|\emptyset) = \begin{cases} 1 - p_B & \text{if } X = \emptyset, \\ p_B b(x) & \text{if } X = \{x\}, \\ 0 & \text{if } |X| > 1. \end{cases} \tag{3.39}$$

Similarly, using the assumptions about the measurement process

$$f_L(\{z_1\}|X) = \begin{cases} p_{FA}\kappa(z_1) & \text{if } X = \emptyset, \\ (1 - p_{FA})\mathcal{N}(z_1; x, r) & \text{if } X = \{x\}, \\ 0 & \text{if } |X| > 1. \end{cases} \tag{3.40}$$

Therefore, using (3.10) the predicted multitarget density, $f_{1|0}(X)$, is:

$$f_{1|0}(X) = \int f_M(X|W) f_{0|0}(W) \delta W \tag{3.41}$$

$$= f(X|\emptyset) f_{0|0}(\emptyset) \tag{3.42}$$

$$= \begin{cases} 1 - p_B & \text{if } X = \emptyset, \\ p_B b(x) & \text{if } X = \{x\}, \\ 0 & \text{if } |X| > 1. \end{cases} \tag{3.43}$$

In order to compute the posterior, using (3.9), first calculate

$$f_L(\{z_1\}|\emptyset)f_{1|0}(\emptyset) = p_{FA}(1-p_B)\kappa(z_1) \tag{3.44}$$

$$f_L(\{z_1\}|\{x\})f_{1|0}(\{x\}) = p_B(1-p_{FA})\mathcal{N}(z_1;x,r)b(x). \tag{3.45}$$

Using these, the normalising constant in (3.9) can be calculated as

$$K = \int f_L(\{z_1\}|X)f_{1|0}(X)\delta X \tag{3.46}$$

$$= p_{FA}(1-p_B)\kappa(z_1) + \int p_B(1-p_{FA})\mathcal{N}(z_1;x,r)b(x)dx. \tag{3.47}$$

Putting these together gives the posterior density, $f_{1|1}(X)$:

$$f_{1|1}(X|Z_1) = \begin{cases} K^{-1}p_{FA}(1-p_B)\kappa(z_1) & \text{if } X = \emptyset, \\ K^{-1}p_B(1-p_{FA})\mathcal{N}(z_1;x,r)b(x) & \text{if } X = \{x\}, \\ 0 & \text{if } |X| > 1. \end{cases} \tag{3.48}$$

Concrete Calculation

In order to allow a visualisation of $f_{1|1}(X)$ let $p_B = p_{FA} = 0.5$, $q = r = 0.01$ and

$$b(x) = \begin{cases} 2x & \text{if } x \in [0,1], \\ 0 & \text{o.w.} \end{cases} \tag{3.49}$$

$$\kappa(z) = \begin{cases} 2-2z & \text{if } z \in [0,1], \\ 0 & \text{o.w.} \end{cases} \tag{3.50}$$

These choices of $b(x)$ and $\kappa(z)$ mean that all false alarms and new targets are in the region $[0,1]$ with false alarms more likely towards 0 and new target more likely towards 1. If $z_1 = 0.5$ then K is within 10^{-6} of 0.5, so approximating $K = 0.5$ gives:

$$f_{1|1}(X) = \begin{cases} 0.5 & \text{if } X = \emptyset. \\ x\mathcal{N}(0.5;x,0.1) & \text{if } X = \{x\} \ , \ x \in [0,1]. \\ 0 & o.w. \end{cases} \tag{3.51}$$

It can easily be verified that $\int f_{1|1}(X)\delta X = 1$. Therefore, $f_{1|1}(X)$ may be interpreted as saying there is a single target present with probability 0.5 and that if there is a target, the probability density for its location is as shown in fig. 3.2.

As an aside, note that if $z_1 = 0.1$, $f_{1|1}(\emptyset) = 0.893$ while for $z_1 = 0.9$, $f_{1|1}(\emptyset) = 0.12$. This shows that varying z_1 has the intuitive effect on the probability of there being a target present. If z_1 is closer to 0 (i.e. in the region where there are more false alarms) the probability of there being a target is lower, and vice versa.

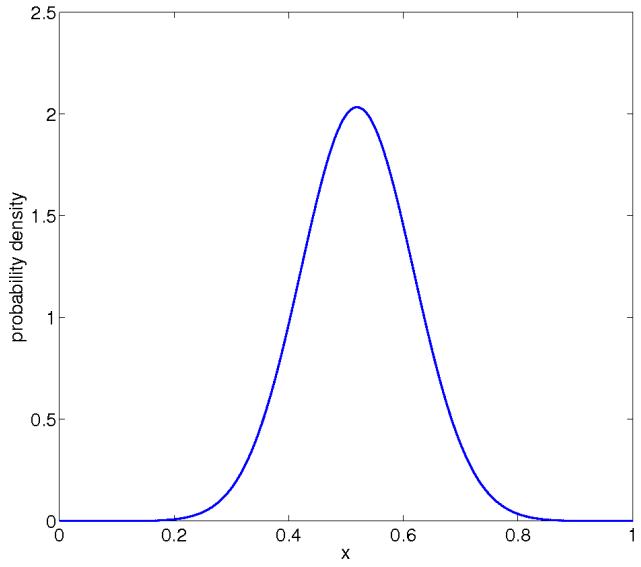


Figure 3.2: The probability density $f_{1|1}(X)$ for $|X| = 1$

Time Step 2

In order to calculate the update for the second time step, it will be necessary to calculate the multitarget Markov density $f_M(X|W)$ for $W = \{w\}$ and the multitarget likelihood $f_L(Z|X)$ for $Z = \{z_2, z_3, z_4\}$. Stating with the Markov density, note that there will be a non-zero probability that $X = \{x_1, x_2\}$. As this is an unordered set, there should be an equal probability of either x_1 or x_2 being the new/existing target. Using the technique in (3.13) to ensure that the Markov density is symmetric, it can be seen from the assumptions on the forward model that

$$f_M(X|\{w\}) = \begin{cases} 0 & \text{if } X = \emptyset, \\ (1 - p_B)\mathcal{N}(x; w, q) & \text{if } X = \{x\}, \\ p_B b(x_1)\mathcal{N}(x_2; w, q) + p_B b(x_2)\mathcal{N}(x_1; w, q) & \text{if } X = \{x_1, x_2\}. \end{cases} \tag{3.52}$$

Similarly, the multitarget likelihood is

$$f_L(\{z_2, z_3, z_4\}|X) \tag{3.53}$$

$$= \begin{cases} 0 & \text{if } |X| < 2, \\ \sum_\sigma p_{FA}(\mathcal{N}(z_{\sigma(1)}; x_1, r)\mathcal{N}(z_{\sigma(2)}; x_2, r)\kappa(z_{\sigma(3)}) & \text{if } X = \{x_1, x_2\}, \\ \sum_\sigma (1 - p_{FA})\times & \\ \mathcal{N}(z_{\sigma(1)}; x_1, r)\mathcal{N}(z_{\sigma(2)}; x_2, r)\mathcal{N}(z_{\sigma(3)}; x_3, r) & \text{if } X = \{x_1, x_2, x_3\}, \\ 0 & \text{if } |X| > 3, \end{cases} \tag{3.54}$$

where the sums are over the permutations $\sigma : \{1,2,3\} \to \{1,2,3\}$. Using the same method as for time step 1, the predicted probability density, $f_{2|1}(X|Z_1)$ and the posterior density, $f_{2|2}(X|Z_1, Z_2)$ may be computed.

$$f_{2|1}(X|Z_1) \tag{3.55}$$

$$= \int f_M(X|W) f_{1|1}(W) \delta W \tag{3.56}$$

$$= \begin{cases} f_M(\emptyset|\emptyset) f_{1|1}(\emptyset) + \int f(\emptyset|\{w\}) f_{1|1}(\{w\}) dw & \text{if } X = \emptyset, \\ f_M(\{x\}|\emptyset) f_{1|1}(\emptyset) + \int f(\{x\}|\{w\}) f_{1|1}(\{w\}) dw & \text{if } X = \{x\}, \\ f_M(\{x_1, x_2\}|\emptyset) f_{1|1}(\emptyset) + \int f(\{x_1, x_2\}|\{w\}) f_{1|1}(\{w\}) dw & \text{if } X = \{x_1, x_2\}, \end{cases} \tag{3.57}$$

$$= \begin{cases} K^{-1}(1-p_B)^2 p_{FA} \kappa(z_1) & \text{if } X = \emptyset, \\ K^{-1} p_B (1-p_B) p_{FA} b(x) \kappa(z_1) & \\ + \int K^{-1} p_B (1-p_B)(1-p_{FA}) \mathcal{N}(x; w, q) \mathcal{N}(z_1; w, r) b(w) dw & \text{if } X = \{x\}, \\ \int K^{-1} p_B^2 (1-p_{FA}) \mathcal{N}(z_1; w, r) \times & \\ (b(x_1)\mathcal{N}(x_2; w, q) + b(x_2)\mathcal{N}(x_1; w, q)) dw & \text{if } X = \{x_1, x_2\}. \end{cases} \tag{3.58}$$

Using the same assumptions as in the section on concrete calculations for time step 1, the predicted density gives $Pr(X = \emptyset) = 0.25$, $Pr(|X| = 1) = 0.5$ and $Pr(|X| = 2) = 0.25$. Visualisations of the predicted density for $|X| = 1$ and $|X| = 2$ are given in figs. 3.3 and 3.4 respectively.

Using (3.9),

$$f_{2|2}(X|Z_1, Z_2) \propto f_{2|1}(X|Z_1) f_L(\{z_2, z_3, z_4\})|X). \tag{3.59}$$

However, it can be seen from (3.54) that $f_L(\{z_2, z_3, z_4\})|X) = 0$ for $|X| < 2$ while (3.58) shows that $f_{2|1}(X|Z_1) = 0$ for $|X| > 2$. Therefore, $f_{2|2}(X|Z_1, Z_2)$ only takes non-zero values when $|X| = 2$ and hence

$$f_{2|2}(\{x_1, x_2\}|Z_1, Z_2) = L^{-1} f_{2|1}(\{x_1, x_2\}|Z_1) f_L(\{z_2, z_3, z_4\})|\{x_1, x_2\}) \tag{3.60}$$

where L^{-1} is the normalising constant. This can be easily computed using (3.54) and (3.58).

Using the same assumptions as previously and taking the measurement $Z_2 = \{0.1, 0.5, 0.9\}$, visualisations of both the multitarget likelihood and posterior density are shown in figs. 3.5 and 3.6 respectively. The six peaks in fig. 3.5 correspond to the six different permutations of the measurement set in $f_L(\{z_2, z_3, z_4\})|X)$. The two main peaks in fig. 3.6 may be interpreted as there being a high probability, based on the measurements received, that one of the targets is near 0.5 and the other is near 0.8.

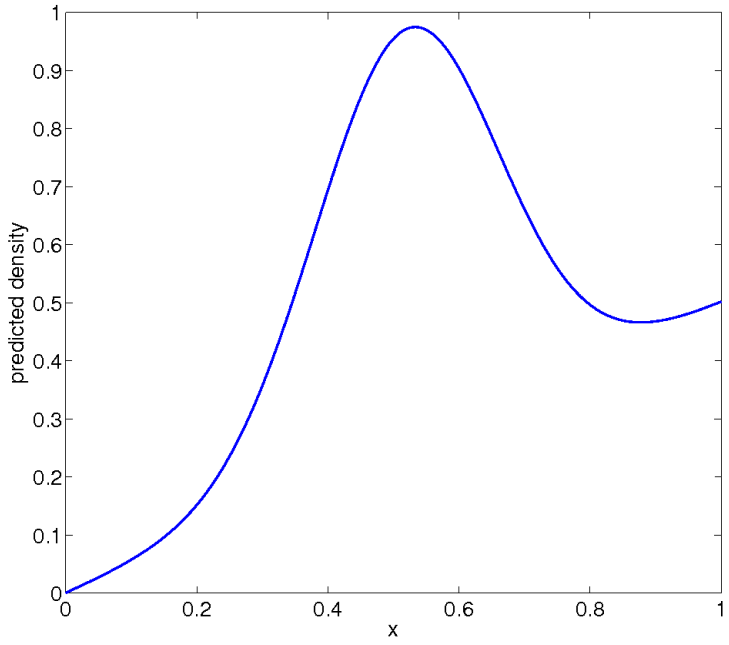


Figure 3.3: The probability density $f_{2|1}(X)$ for $|X| = 1$

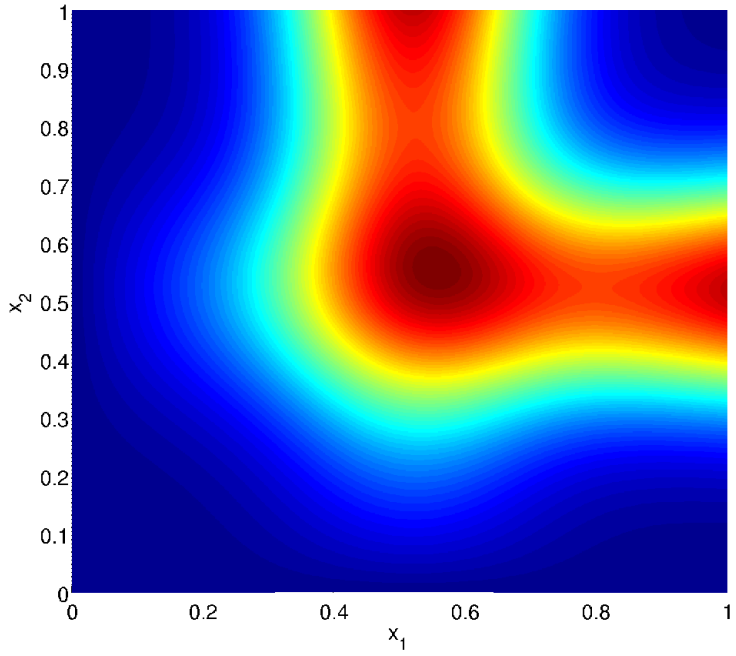


Figure 3.4: The probability density $f_{2|1}(X)$ for $|X| = 2$

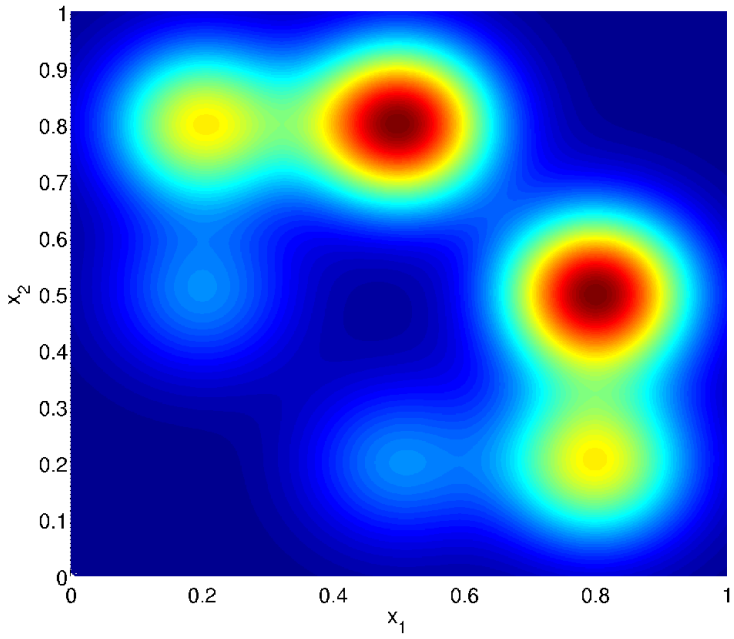


Figure 3.5: The multitarget likelihood $f_L(\{z_2, z_3, z_4\}|X)$ for $|X| = 2$.

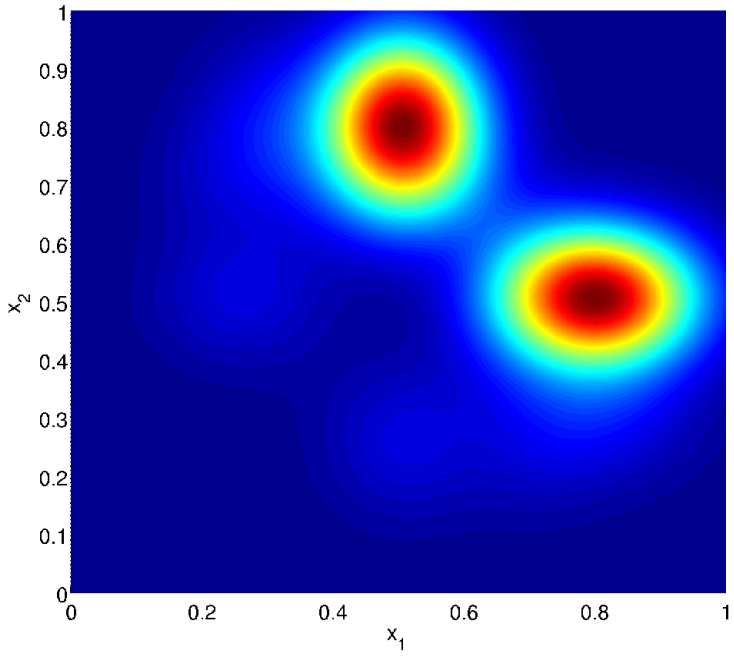


Figure 3.6: The posterior density $f_{2|2}(X)$ for $|X| = 2$

3.3.4 The Multitarget Likelihood Function

In the previous section, a simple scenario was presented where it was possible to write down the multitarget likelihood and Markov density directly from the assumptions. This will not generally be possible so the next two subsections will present the calculation of the multitarget likelihood and Markov density in a more general case.

The multitarget likelihood function $f_{k+1}(Z_{k+1}|X)$ from (3.9) will be derived using the background in Section 3.3.2. The process for doing so is as follows:

- Determine the conditional belief mass function $\beta_{Z_{k+1}}(T|X)$ using the simple multitarget model (3.12)
- Construct $f_{k+1}(Z_{k+1}|X) = \frac{\delta\beta_{Z_{k+1}}}{\delta Z_{k+1}}(\emptyset|X)$ using the multitarget Radon-Nikodým theorem (3.31).

The belief mass function and the likelihood function contain precisely the same information as one can be derived from the other using (3.31) and (3.17). Thus, the likelihood function derived is the true multitarget likelihood function.
N.B. The time indices (i.e $k+1$) will be omitted for the rest of this subsection for notational convenience.

In addition to the multitarget measurement model (3.12), the following assumptions are made [M1, p. 408]:

- Each measurement, $\mathbf{z}$, made due to a true target with state $\mathbf{x}$ has the (single-target) likelihood function $f(\mathbf{z}|\mathbf{x})$.
- A single target can generate at most one measurement.
- A target with state $\mathbf{x}$ generates a measurement with probability $p_D(\mathbf{x})$ and fails to generate a measurement (a missed detection) with probability $1 - p_D(\mathbf{x})$.
- The number of false alarms in a given time step has Poisson distribution with mean λ and the individual false alarms are drawn independently according to the probability density $\kappa(\mathbf{z})$.
- The false alarm process and target measurement process are statistically independent.

Lemma 3.1: The multitarget likelihood in the case where there are no targets present is given by [M1, p. 412]:

$$f(Z|\emptyset) = e^{-\lambda} \prod_{\mathbf{z}\in Z} \lambda\kappa(\mathbf{z}). \tag{3.61}$$

Proof: Using (3.12), the conditional belief mass function is

$$\begin{aligned}
\beta_Z(T|X=\emptyset) &= P(K\subseteq T|\emptyset) && (3.62)\\
&= \sum_{m=0}^{\infty} P(K\subseteq T,|K|=m|\emptyset) && (3.63)\\
&= \sum_{m=0}^{\infty} P(|K|=m|\emptyset)\cdot P(K\subseteq T|\emptyset,|K|=m) && (3.64)\\
&= e^{-\lambda}\sum_{m=0}^{\infty}\frac{\lambda^m}{m!}p(\mathbf{z}\subseteq T)^m && (3.65)\\
&= e^{\lambda p_\kappa(T)-\lambda}, && (3.66)
\end{aligned}$$

where $p_\kappa(T)$ is the probability mass function corresponding to κ over the set T.

$$\begin{aligned}
\frac{\delta\beta_Z}{\delta Z}(T|\emptyset) &= \frac{\delta^m}{\delta\mathbf{z}_m...\delta\mathbf{z}_1}e^{\lambda p_\kappa(T)-\lambda} && (3.67)\\
&= \frac{\delta^{m-1}}{\delta\mathbf{z}_m...\delta\mathbf{z}_2}e^{\lambda p_\kappa(T)-\lambda}\frac{\delta}{\delta\mathbf{z}_1}\big[\lambda p_\kappa(T)-\lambda\big] && (3.68)\\
&= \frac{\delta^{m-1}}{\delta\mathbf{z}_m...\delta\mathbf{z}_2}e^{\lambda p_\kappa(T)-\lambda}\Big(\lambda\frac{\delta}{\delta\mathbf{z}_1}\big[p_\kappa(T)\big]-\frac{\delta\lambda}{\delta\mathbf{z}_1}\Big) && (3.69)\\
&= \lambda\kappa(\mathbf{z}_1)\frac{\delta^{m-1}}{\delta\mathbf{z}_m...\delta\mathbf{z}_2}e^{\lambda p_\kappa(T)-\lambda} && (3.70)\\
&= e^{\lambda p_\kappa(T)-\lambda}\prod_{i=1}^{m}\kappa(\mathbf{z}_i), && (3.71)
\end{aligned}$$

where (3.68) is due to (3.33), (3.69) is due to (3.34), (3.70) is due to (3.35) and (3.36) and (3.71) is obtained from repeating the process iteratively. Since $p_\kappa(\emptyset)=0$, it is clear that

$$f(Z|\emptyset)=\frac{\delta\beta_Z}{\delta Z}(\emptyset|\emptyset)=e^{-\lambda}\prod_{\mathbf{z}\in Z}\kappa(\mathbf{z}_i).\quad\square \tag{3.72}$$

Lemma 3.2 - The multitarget likelihood in the case where there are no false alarms and where $|X|=n$ and $|Z|=m$ with $m\leq n$ is given by [M1, p. 418]:

$$f(Z|X)=\begin{cases}\prod_{\mathbf{x}\in X}(1-p_D(\mathbf{x}_i)) & \text{if } Z=\emptyset,\\ f(\emptyset|X)\sum_{1\leq i_1\neq...\neq i_m\leq n}\prod_{j=1}^{m}\frac{p_D(\mathbf{x}_{i_j})\cdot f(\mathbf{z}_j|\mathbf{x}_{i_j})}{1-P_D(\mathbf{x}_{i_j})} & \text{if } Z\neq\emptyset.\end{cases} \tag{3.73}$$

Proof: Let $\theta(\mathbf{x})$ denote the set containing either the measurement generated by the

target with state $\mathbf{x}$ or $\emptyset$ if the detection is missed. Then,

$$\begin{aligned}\beta_Z(T|X) &= P(Z \subseteq T|X) && (3.74)\\ &= \prod_{\mathbf{x}\in X} P(\theta(\mathbf{x}) \subseteq T|\mathbf{x}) && (3.75)\\ &= \prod_{i=1}^{n} p_i(T), && (3.76)\end{aligned}$$

where $p_i(T) = P(\theta(\mathbf{x}_i) \subseteq T)$. Now,

$$\frac{\delta\beta_Z}{\delta Z}(T|X) = \sum_{W_1\oplus...\oplus W_n=Z} \frac{\delta p_1}{\delta W_1}(T)...\frac{\delta p_n}{\delta W_n}(T), \tag{3.77}$$

by (3.37) and hence

$$f(Z|X) = \sum_{W_1\oplus...\oplus W_n=Z} \frac{\delta p_1}{\delta W_1}(\emptyset)...\frac{\delta p_n}{\delta W_n}(\emptyset), \tag{3.78}$$

but as $p_i(T) = P(\theta(\mathbf{x}_i) \subseteq T)$, (3.36) gives

$$\frac{\delta p_i}{\delta Z} = \begin{cases} 1 - p_D(\mathbf{x}_i) & \text{if } Z = \emptyset, \\ p_D(\mathbf{x}_i) f(\mathbf{z}|\mathbf{x}_i) & \text{if } Z = \{\mathbf{z}\}, \\ 0 & \text{if } |Z| > 1. \end{cases} \tag{3.79}$$

From which the result follows immediately. □

Proposition 3.3: The multitarget likelihood for the full multitarget measurement model (3.12) is given by [M1, p. 420]:

$$f(Z|X) = e^{\lambda} f_K(Z) \cdot f(\emptyset|X) \cdot \sum_{\Omega} \prod_{i:\Omega(i)>0} \frac{p_D(\mathbf{x}_i) \cdot f(\mathbf{z}_{\Omega(i)}|\mathbf{x}_i)}{(1 - p_D(\mathbf{x}_i)) \cdot \lambda\kappa(\mathbf{z}_{\Omega(i)})}, \tag{3.80}$$

where the summation is over all association hypotheses $\Omega : \{1, ..., n\} \to \{0, 1, ..., m\}$.

Proof: The belief mass function can be written in the notation of (3.12) as

$$\begin{aligned}\beta_Z(T|X) &= P(\Theta(X) \cup K \subseteq T|X) && (3.81)\\ &= P(\Theta(X) \subseteq T|X) \cdot P(K \subseteq T|X) && (3.82)\\ &= \beta_\Theta(T|X) \cdot \beta_K(T|X) && (3.83)\\ \Rightarrow \frac{\delta\beta_Z}{\delta Z}(T|X) &= \sum_{W\subseteq Z} \frac{\delta\beta_\Theta}{\delta W}(T|X) \cdot \frac{\delta\beta_K}{\delta(Z - W)}(T|X) && (3.84)\end{aligned}$$

by (3.37). Therefore,

$$f(Z|X) = \frac{\delta \beta_Z}{\delta Z}(\emptyset|X) \tag{3.85}$$

$$= \sum_{W \subseteq Z} f_\Theta(W|X) \cdot f_K(Z-W|X), \tag{3.86}$$

where f_K and f_Θ are as derived in Lemmas 3.1 and 3.2 respectively. All that remains is to rewrite (3.86) in a more friendly form. Now,

$$f(Z|X) = e^{-\lambda} f_\Theta(\emptyset|X) \sum_{W \subseteq Z:|W| \leq n} \left(\sum_{1 \leq i_1 \neq \ldots \neq i_m \leq n} \prod_{j=1}^{m} \frac{p_D(\mathbf{x}_{i_j}) \cdot f(\mathbf{z}_j|\mathbf{x}_{i_j})}{1 - p_D(\mathbf{x}_{i_j})} \right) \cdot \left(\prod_{\mathbf{z} \in Z-W} \lambda \kappa(\mathbf{z}) \right) \tag{3.87}$$

$$= f_K(Z) f_\Theta(\emptyset|X) \sum_{W \subseteq Z:|W| \leq n} \sum_{1 \leq i_1 \neq \ldots \neq i_m \leq n} \prod_{j=1}^{m} \frac{p_D(\mathbf{x}_{i_j}) \cdot f(\mathbf{z}_j|\mathbf{x}_{i_j})}{(1 - p_D(\mathbf{x}_{i_j})) \lambda \kappa(\mathbf{z})}. \tag{3.88}$$

Note that the double sum in (3.88) is the sum over all possible combinations of the measurements being clutter or due to true targets, and then within each such association the sum over the possible associations between measurements and target states. Thus, it is the sum over all possible association hypotheses and can be written:

$$f(Z|X) = f_K(Z|X) f_\Theta(\emptyset|X) \cdot \sum_{\Omega} \prod_{i:\Omega(i)>0} \frac{p_D(\mathbf{x}_i) \cdot f(\mathbf{z}_{\Omega(i)}|\mathbf{x}_i)}{(1 - p_D(\mathbf{x}_i)) \cdot \lambda \kappa(\mathbf{z}_{\Omega(i)})}, \tag{3.89}$$

where the summation is over all $\Omega : \{1, ..., n\} \to \{0, 1, ..., m\}$. The equivalence of these sums is established in more detail in [75, pp. 332-335]. Finally, noting that $f(\emptyset|X) = f_K(\emptyset) \cdot f_\Theta(\emptyset|X)$ and that $f_K(\emptyset) = e^{-\lambda}$ from equations (3.86) and (3.61) respectively gives

$$f(Z|X) = e^{\lambda} f_K(Z) \cdot f(\emptyset|X) \cdot \sum_{\Omega} \prod_{i:\Omega(i)>0} \frac{p_D(\mathbf{x}_i) \cdot f(\mathbf{z}_{\Omega(i)}|\mathbf{x}_i)}{(1 - p_D(\mathbf{x}_i)) \cdot \lambda \kappa(\mathbf{z}_{\Omega(i)})}. \quad \square \tag{3.90}$$

3.3.5 Multitarget Markov Density

The multitarget Markov density, $f_{k+1|k}(X|X')$, will now be derived using the same method as in Section 3.3.4. The assumptions required in addition to the simple multitarget forward model in (3.11) are [M1, p. 466]:

- The probability density for a target state $\mathbf{x}$ at time $k+1$ given that it had state $\mathbf{x}'$ at time k is $f_{k+1|k}(\mathbf{x}|\mathbf{x}')$.

- A target with state $\mathbf{x}'$ at time k has probability $p_S(\mathbf{x}')$ of surviving to time $k+1$ and probability $1 - p_S(\mathbf{x}')$ of disappearing.
- The number of targets appearing in a given time step has Poisson distribution with mean μ and the appearing target states are independent of each other and distributed according to the birth density $b(\mathbf{x})$.
- The target birth process and target forward process are statistically independent.
- For ease of presentation, it will be assumed that there are no spawning targets.

N.B. The time indices (i.e k, $k+1$) will again be omitted for the rest of this subsection for notational convenience.

It may be seen from closer inspection that these assumptions and the assumptions on the multitarget measurement process at the beginning of Section 3.3.4 are mathematically identical with the substitution of:

- Single target forward process, $f(\mathbf{x}|\mathbf{x}')$, for the single target measurement process, $f(\mathbf{z}|\mathbf{x})$.
- Probability of survival, p_S for probability of detection p_D.
- Target birth process $b(\mathbf{x})$ for clutter process $\kappa(\mathbf{z})$.

Therefore, it is possible to reuse the work in Section 3.3.4 in deriving the multitarget Markov density.

Proposition 3.4 - The multitarget Markov density is given by [M1, p. 472]

$$f(X|X') = e^{\mu} f_B(X) \cdot f(\emptyset|X') \cdot \sum_{\Omega} \prod_{i:\Omega(i)>0} \frac{p_S(\mathbf{x}'_i) \cdot f(\mathbf{x}_{\Omega(i)}|\mathbf{x}'_i)}{(1 - p_D(\mathbf{x}'_i)) \cdot \mu b(\mathbf{x}_{\Omega(i)})}, \tag{3.91}$$

where the summation is over all association hypotheses $\Omega : \{1, ..., n\} \to \{0, 1, ..., m\}$ and where

$$f_B(X) = e^{-\mu} \prod_{\mathbf{x} \in X} \mu b(\mathbf{x}), \tag{3.92}$$

$$f(\emptyset|X') = e^{-\mu} \prod_{\mathbf{x}' \in X'} (1 - p_S(\mathbf{x}')). \tag{3.93}$$

Proof - Using the mathematical equivalence of the multitarget measurement and multitarget forward models and Proposition 3.3. □

3.3.6 Implementing the Multitarget Bayes Filter

Using (3.80) and (3.91) it is theoretically possible to implement the multitarget Bayes filter (3.9-3.10). There are however, two main issues with practical implementations: computational tractability and extracting estimates.

Computation - It is clear that the high dimension of the multitarget state space makes computation implausible in general. Even in the case where the number of targets is restricted, the sum over permutations in (3.80) and (3.91) leads to computational intractability in most cases.

For example, consider a grid approximation similar to Section 2.1.3. Mahler [M1, ch. 15] computes the computational complexity in the special case where the number of targets, n, is assumed fixed and known, there are no false alarms or missed detections and where target motions are independent. Let ν be the number of grid cells, a the number of operations required to evaluate $f_{k+1|k}(\mathbf{x}|\mathbf{x}')$ and b the number of operations required to evaluate $f_{k+1}(\mathbf{z}|\mathbf{x})$. In this special case, the number of operations required for one iteration of the multitarget Bayes filter is $n\nu^n(\nu^n a + n!b)$ [M1, p. 545].

For implementations using sequential Monte Carlo (SMC), the number of computations required is $n(a + n!b)\Pi$ where Π is the number of multitarget particles [M1, p. 546]. The number of multitarget particles Π may need to be large and whether or not real-time computation will be possible will depend on the scenario. Furthermore, the factor of $n!$ will cause problems for implementation with large numbers of targets.

Estimation - Though the full posterior, $f(X)$, is computed by the Bayes filter, most practical implementations at some point seek to summarise this information using state estimators. However, the most common state estimators from standard probability theory, the maximum a posteriori (MAP) and the expectation, or expected a posteriori (EAP), are not defined for multitarget posteriors unless the target state space is dimensionless.

To see why, consider that the straightforward MAP estimator would be defined as

$$\hat{X}^{MAP} = \arg\sup_{X} f(X), \tag{3.94}$$

and note that the dimension of $f(X)$ is $u^{-|X|}$ (as discussed in Section 3.3.1) where u is the dimension of the target state space. Thus, the comparison required for (3.94) is undefined [M1, p. 493].

Similarly, the EAP estimator is defined as [M1, p. 495]

$$\hat{X}^{EAP} = \int X \cdot f(X)\delta X = \emptyset \cdot f(\emptyset) + \int \{\mathbf{x}\} \cdot f(\{\mathbf{x}\})d\mathbf{x} + ... \tag{3.95}$$

It is clear from the first two terms that this sum is not defined.

Therefore it is necessary to construct new estimators for the multitarget posterior. One such estimator, presented in [43] is the Marginal Multitarget (MaM) estimator defined by [M1, p. 497]

$$\hat{X}^{MaM} = \arg \sup_{\mathbf{x}_1,...,\mathbf{x}_{\hat{n}}} f(\{\mathbf{x}_1, ..., \mathbf{x}_{\hat{n}}\}), \tag{3.96}$$

where

$$\hat{n} = \arg \sup_n \frac{1}{n!} \int f(\{\mathbf{x}_1, ..., \mathbf{x}_n\})\mathrm{d}\mathbf{x}_1...\mathrm{d}\mathbf{x}_n. \tag{3.97}$$

Note that $\hat{n}$ is the standard MAP of the discrete cardinality distribution. The MaM estimator is shown to be Bayes optimal in [43].

Implementations of the multitarget Bayes filter based on sequential Monte Carlo methods have been presented for a number of different multitarget tracking problems such as vehicle tracking in terrain [110], bearing only tracking [121] and passive acoustic tracking [70]. Each of these implementations was for a multitarget tracking problem with low (≤ 2) or fixed target number as necessary in order to permit manageable computation.

Chapter 4

The Probability Hypothesis Density Filter

4.1 Deriving the Probability Hypothesis Density Filter

4.1.1 Background, Definitions and Preliminary Results

Propagating the full Bayes filter in the single target case was seen in Section 2.2 to be generally impractical. It should be unsurprising, then, that the same is true of the multitarget Bayes filter, as discussed in Section 3.3.6. Mahler notes in [73] that "the multitarget Bayes filter will have no practical utility without drastic but intelligent approximation strategies."

In the single target case, a common approach to simplifying the calculation of the Bayes filter is to attempt to propagate only one or two moments of the posterior density. For example, the Kalman filter may be regarded as the propagation of the first two statistical moments of the posterior whereas the alpha-beta filter, or constant gain Kalman filter, propagates only the first moment [75]. This is exactly the approach taken in the probability hypothesis density (PHD) filter which seeks to propagate only the first moment of the multitarget posterior.

It is not immediately clear what the first moment of the multitarget posterior should be. It was noted in Section 3.3.6 that the calculation of the traditional expected value is not defined. Instead, following common practise in the point process literature [32], the first moment of the multitarget posterior for random finite set X, is defined not as $\mathbb{E}[X]$ but $\mathbb{E}\big[\delta_X(\mathbf{x})\big]$ where

$$\delta_X(\mathbf{x}) = \sum_{\mathbf{w}\in X} \delta_{\mathbf{w}}(\mathbf{x}), \tag{4.1}$$

and $\delta_{\mathbf{w}}$ is the Dirac delta centred at $\mathbf{w}$. Therefore, the first moment of the multitarget posterior, denoted D_X, and referred to hereafter as the *probability hypothesis density* (PHD) is defined as:

$$D_X(\mathbf{x}) = \int \delta_X(\mathbf{x}) \cdot f_X(X)\delta X. \tag{4.2}$$

The presentation of the PHD filter here will be based largely upon the derivation given in [73]. For ease of reference [73] will be referred to as M2 for the rest of this chapter.

Properties of the PHD

1. A statistical first order moment - From the point of view of point process theory, D_X is the standard first statistical moment [M2, Sec. D], also known as the first-order moment density [32].

2. A density of expected targets -

$$\int_S D_X(\mathbf{x})d\mathbf{x} = \mathbb{E}[|X \cap S|] \tag{4.3}$$

   So the integral over the PHD over a region S is equal to the *expected number of targets* in that region [M1, Thm. 2].

3. An Information-Theoretic Best Approximation - The multitarget density f is a *multitarget Poisson density* if, for some intensity function $I(\mathbf{x})$ with $N = \int I(\mathbf{x})\mathrm{d}\mathbf{x}$,

$$f(X) = e^{-N} \prod_{\mathbf{x}\in X} I(\mathbf{x}). \tag{4.4}$$

   Consider approximating a general multitarget density by a multitarget Poisson density. This approximation has minimal Kullback-Leiber divergence when $I(\mathbf{x})$ is the PHD [M2, Thm. 2].

Probability Generating Functionals

A concise derivation of the PHD filter requires the concept of a probability generating functional (p.g.fl.) of a random finite set X, denoted $G_X[h]$. This is another concept borrowed from point process theory (see Moyal [79]). This is defined analogously to the probability generating function from standard probability theory as [M2, Eq. 44]

$$G_X[h] = E\big[h^X\big] = \int h^X f(X)\delta X, \tag{4.5}$$

where $h^X = \prod_{\mathbf{x}\in X} h(\mathbf{x})$. Some properties of p.g.fl.s which will be needed in the derivation of the PHD filter equations are listed below:

1. The p.g.fl. is an integral transform and, as such, an advantage comes from its ability to transform convolutions into products. Thus, the convolution inherent in the probability density function for a union of variables, as in (3.32) becomes a product of p.g.fl.s. Let $X = X_1 \cup ... \cup X_n$ where $X_1, ..., X_n$ are statistically independent, then [79]

$$G_X[h] = G_{X_1}[h] \cdots G_{X_n}[h]. \tag{4.6}$$

2. Let $G_X[h]$ be the p.g.fl. of X and $D(\mathbf{x})$, the PHD of X, then [M2, Thm. 1]

$$D(\mathbf{x}) = \left.\frac{\delta G_X[h]}{\delta \mathbf{x}}\right|_{h=1}. \tag{4.7}$$

3. Let $\phi[h]$ be a functional s.t. $\phi[1] = 1$, $G_X[h]$ be the p.g.fl. and D_X the PHD of X. The PHD corresponding to the p.g.fl. $G_X[\phi[h]]$, denoted $D(\mathbf{x})$, is [M2, Prop. 3]

$$D(\mathbf{x}) = \int \left.\frac{\delta \phi}{\delta \mathbf{x}}[h]\right|_{h=1} (\mathbf{w}) D_X(\mathbf{w}) \mathrm{d}\mathbf{w}. \tag{4.8}$$

4. Let $G[h] = \int h(\mathbf{x}) f(\mathbf{x}) \mathrm{d}\mathbf{x}$ then [M2, Eq. 54]

$$\frac{\delta G}{\delta \mathbf{x}}[h] = f(\mathbf{x}). \tag{4.9}$$

5. Let β_X and G_X be the belief mass function and p.g.fl. for X respectively, then a generalisation of the multitarget Radon-Nikodým theorem given in [75, ch. 11] is

$$\frac{\delta G_X}{\delta X}[h] = \int h^W \cdot f_X(X \cup W) \delta W \ , \ \frac{\delta \beta_X}{\delta X}(S) = \int_S f_X(X \cup W) \delta W, \tag{4.10}$$

   from which it is clear that

$$\frac{\delta G_X}{\delta X}[\mathbf{1}_S] = \frac{\delta \beta_X}{\delta X}(S). \tag{4.11}$$

6. The p.g.fl. of a random finite set X which is Poisson distributed with mean cardinality λ and probability density $I(\mathbf{x})$ is given by [M2, Prop. 5]:

$$G_I[h] = e^{\lambda I[h] - \lambda}, \tag{4.12}$$

   where $I[h] = \int I(\mathbf{x}) h(\mathbf{x}) \mathrm{d}\mathbf{x}$.

4.1.2 PHD Predictor

The PHD predictor (forward) step will now be derived using p.g.fl.s. The method employed will be to transform the predicted probability density using the p.g.fl. in order to simplify the convolution due to the set unions in (3.11), and then recover the PHD using (4.7). The PHD at time k is denoted $D_{k|k}$ and the PHD predicted to time step $k+1$ is denoted $D_{k+1|k}$. The following assumptions will be made in addition to (3.11):

- The probability density for a target state $\mathbf{x}$ at time $k+1$, given that it had state $\mathbf{w}$ at time k, is $f_{k+1|k}(\mathbf{x}|\mathbf{w})$.
- A target with state $\mathbf{w}$ at time k has probability $p_S(\mathbf{w})$ of surviving to time $k+1$ and probability $1-p_S(\mathbf{w})$ of disappearing.
- The appearance of new targets between time steps k and $k+1$ is captured by the Poisson distributed random finite set $\Gamma_{k+1|k}$. The PHD of $\Gamma_{k+1|k}$ is denoted $b_{k+1|k}(\mathbf{x})$ (i.e. $b_{k+1|k}$ integrates to the expected number of total targets appearing, denoted μ and $b_{k+1|k}/\mu$ is a probability density for appearing target locations).
- The target birth process and target forward process are statistically independent.
- For ease of presentation, it will be assumed that there are no spawning targets.

Proposition 4.1 - The PHD predictor equation is given by [M2, Thm. 5]

$$D_{k+1|k}(\mathbf{x}) = b_{k+1|k}(\mathbf{x}) + \int p_S(\mathbf{w}) f_{k+1|k}(\mathbf{x}|\mathbf{w}) D_{k|k}(\mathbf{w}) \mathrm{d}\mathbf{w}. \tag{4.13}$$

Proof - Denote the p.g.fl. of the predicted probability density $f_{k+1|k}(X|Z^k)$ by $G_{k+1|k}[h]$. Then

$$\begin{aligned} G_{k+1|k}[h] &= \int h^X f_{k+1|k}(X|Z^k)\delta X & (4.14)\\ &= \int \Big(\int h^X f_{k+1|k}(X|W)\delta X \Big) f_{k|k}(W|Z^k)\delta W, & (4.15) \end{aligned}$$

where $f_{k+1|k}(X|W)$ is the multitarget Markov density. Therefore,

$$G_{k+1|k}[h] = \int G_{k+1|k}[h|X] f_{k|k}(X|Z^k)\delta X, \tag{4.16}$$

where $G_{k+1|k}[h|X]$ is the p.g.fl. of the multitarget Markov density.

Now, if $X_k = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ then, by (3.11), $X_{k+1} = S(\{\mathbf{x}_1\}) \cup ... \cup S(\{\mathbf{x}_n\}) \cup \Gamma_{k+1|k}$ where $S(\{\mathbf{x}\})$ is the set containing the new target state if the target survives and the empty set if it does not. Due to the statistical independence of individual targets and the birth process, (4.6) gives:

$$G_{k+1|k}[h|X] = G_\Gamma[h] \prod_{\mathbf{x}\in X} G_{k+1|k}[h|\mathbf{x}]. \tag{4.17}$$

It is possible to evaluate $G_{k+1|k}[h|\mathbf{x}]$ as follows:

$$\begin{aligned} G_{k+1|k}[h|\mathbf{x}_i] &= \int h^Y f_{k+1|k}(Y|\mathbf{x}_i)\delta Y && (4.18)\\ &= f(\emptyset|\mathbf{x}_i) + \int h(\mathbf{y}) f_{k+1|k}(\{\mathbf{y}\}|\mathbf{x}_i)\mathrm{d}\mathbf{y} && (4.19)\\ &= 1 - p_S + p_S p[h], && (4.20) \end{aligned}$$

where $p[h] = \int h(\mathbf{y}) f_{k+1|k}(\mathbf{y}|\mathbf{x}_i)\mathrm{d}\mathbf{y}$. So $G_{k+1|k}[h|X]$ may be written as

$$G_{k+1|k}[h|X] = G_\Gamma[h] \cdot (1 - p_S + p_S p[h])^X. \tag{4.21}$$

Substituting this expression into (4.16) gives

$$\begin{aligned} G_{k+1|k}[h] &= \int G_\Gamma[h]\big(1 - p_S + p_S p[h]\big)^X f_{k|k}(X|Z^k)\delta X && (4.22)\\ &= G_\Gamma[h] G_{k|k}\big[1 - p_S + p_S p[h]\big]. && (4.23) \end{aligned}$$

Writing $\phi[h] = 1 - p_S + p_S p[h]$ and using (4.7), the predicted PHD is given by:

$$\begin{aligned} D_{k+1|k}(\mathbf{x}) &= \left.\frac{\delta G_{k+1|k}[h]}{\delta \mathbf{x}}\right|_{h=1} && (4.24)\\ &= \left.\frac{\delta G_\Gamma[h]}{\delta \mathbf{x}}\right|_{h=1} \cdot G_{k|k}(\phi[1]) + G_\Gamma(1) \cdot \left.\frac{\delta G_{k|k}[\phi(h)]}{\delta \mathbf{x}}\right|_{h=1}. && (4.25) \end{aligned}$$

Now, $\delta G_\Gamma[1]/\delta \mathbf{x}$ is the PHD of Γ which is denoted $b_{k+1|k}$. Since $\phi[1] = 1$ it follows that $G_{k|k}[\phi[1]] = 1$ and (4.8) gives:

$$\left.\frac{\delta G_{k|k}[\phi[h]]}{\delta \mathbf{x}}\right|_{h=1} = \int \left.\frac{\delta \phi}{\delta \mathbf{x}}[h]\right|_{h=1} (\mathbf{w}) D_{k|k}(\mathbf{w})\mathrm{d}\mathbf{w}, \tag{4.26}$$

and

$$\begin{aligned} \frac{\delta \phi}{\delta \mathbf{x}}[h] &= p_S(\mathbf{w})\frac{\delta p[h]}{\delta \mathbf{x}}(\mathbf{w}) && (4.27)\\ &= p_S(\mathbf{w}) f_{k+1|k}(\mathbf{x}|\mathbf{w}), && (4.28) \end{aligned}$$

by (4.9). Hence, it follows that

$$D_{k+1|k}(\mathbf{x}) = b_{k+1|k}(\mathbf{x}) + \int p_S(\mathbf{w}) f_{k+1|k}(\mathbf{x}|\mathbf{w}) D_{k|k}(\mathbf{w})\mathrm{d}\mathbf{w}. \quad \square \tag{4.29}$$

4.1.3 PHD Corrector

In this section, the derivation of the PHD filter will be completed by the derivation of the PHD corrector. The approach will be similar to the method used for the derivation of the PHD predictor, using the p.g.fl. of the multitarget posterior density to simplify the convolutions due to set unions in (3.12) and then applying (4.7) to obtain the PHD. The corrected PHD based on measurements up to time $k+1$ is denoted $D_{k+1|k+1}(\mathbf{x})$. The assumptions required in addition to (3.12) are:

- Each measurement, $\mathbf{z}$, made due to a true target with state $\mathbf{x}$ has the (single-target) likelihood function denoted $L_{\mathbf{z}}(\mathbf{x})$.
- A single target can generate at most one measurement.
- A target with state $\mathbf{x}$ generates a measurement with probability $p_D(\mathbf{x})$ and fails to generate a measurement (a missed detection) with probability $1 - p_D(\mathbf{x})$.
- The set of false alarms at time step k, K_k has a multitarget Poisson distribution with mean cardinality λ and probability density function $\kappa(\mathbf{z})$.
- The false alarm process and target measurement process are statistically independent.
- The predicted multitarget density $f_{k+1|k}(X|Z^k)$ is assumed to be Poisson distributed. Thus,

$$f_{k+1|k}(X|Z^k) \approx e^{-\mu}\mu^n s(\mathbf{x}_1)...s(\mathbf{x}_n) \tag{4.30}$$

  for some mean cardinality μ and probability density $s(\mathbf{x})$. Note that $D_{k+1|k}(\mathbf{x}) = \mu \cdot s(\mathbf{x})$.

Proposition 4.2 - The PHD Corrector equation is given by [M2, Thm. 6]:

$$D_{k+1|k+1}(\mathbf{x}) = F_{k+1}(Z_{k+1}|X)D_{k+1|k}(\mathbf{x}), \tag{4.31}$$

where

$$F_{k+1|k}(Z|\mathbf{x}) = 1 - p_D(\mathbf{x}) + \sum_{\mathbf{z}\in Z_{k+1}} \frac{p_D(\mathbf{x})L_{\mathbf{z}}(\mathbf{x})}{\lambda\kappa(\mathbf{z}) + D_{k+1|k}[p_D L_{\mathbf{z}}]}, \tag{4.32}$$

and $s[h]$ is defined as $\int s(\mathbf{x})h(\mathbf{x})\mathrm{d}\mathbf{x}$.

Proof - The multitarget posterior $f_{k+1|k+1}(X|Z^{k+1})$ is given by

$$f_{k+1|k+1}(X|Z^{k+1}) = C^{-1} \cdot f_{k+1}(Z_{k+1}|X) \cdot f_{k+1|k}(X|Z^k) \tag{4.33}$$

where $f_{k+1}(Z_{k+1}|X)$ is the multitarget likelihood function, $f_{k+1|k}(X|Z^k)$ the predicted multitarget density and

$$C = f_{k+1}(Z_{k+1}|Z^k) = \int f_{k+1}(Z_{k+1}|X) \cdot f_{k+1|k}(X|Z^k)\delta X. \tag{4.34}$$

Define the two variable p.g.fl. by

$$\begin{aligned} F[g,h] &= \int\int h^X g^Z f_{k+1}(Z|X) f_{k+1|k}(X|Z^k)\delta X \delta Z & (4.35)\\ &= \int h^X G_{k+1}[g|X] f_{k+1|k}(X|Z^k)\delta X, & (4.36) \end{aligned}$$

where

$$G_{k+1}[g|X] = \int g^Z f_{k+1}(Z|X)\delta Z, \tag{4.37}$$

which is the p.g.fl. of the multitarget likelihood. It is clear from (4.11) that

$$\left.\frac{\delta G_{k+1}[g|X]}{\delta Z}\right|_{g=0} = \left.\frac{\delta \beta_Z}{\delta Z}(S|X)\right|_{S=\emptyset} = f_{k+1}(Z|X), \tag{4.38}$$

using (3.28). Therefore,

$$C = f_{k+1}(Z_{k+1}|Z^k) = \frac{\delta F}{\delta Z}[0,1]. \tag{4.39}$$

Similarly,

$$\frac{\delta}{\delta \mathbf{x}}\frac{\delta F}{\delta Z} = \frac{\delta}{\delta \mathbf{x}} \int h^X \frac{\delta G_{k+1}[g|X]}{\delta Z} \cdot f_{k+1|k}(X|Z^k)\delta X, \tag{4.40}$$

so that,

$$\left.\frac{\delta}{\delta \mathbf{x}}\frac{\delta F}{\delta Z}\right|_{g=0} = \frac{\delta}{\delta \mathbf{x}} \int h^X f_{k+1}(Z|X) \cdot f_{k+1|k}(X|Z^k)\delta X, \tag{4.41}$$

where the integral is the p.g.fl. of $f_{k+1}(Z|X)f_{k+1|k}(X|Z^k)$, denoted $G_{k+1|k+1}$. Therefore

$$\left.\frac{\delta}{\delta \mathbf{x}}\frac{\delta F}{\delta Z}\right|_{[g,h]=[0,1]} = \left.\frac{\delta G_{k+1|k+1}}{\delta \mathbf{x}}\right|_{h=1} = C D_{k+1|k+1}. \tag{4.42}$$

By direct analogy with (4.17-4.20), it is clear that it is possible to express $G_{k+1}[g|X]$ as

$$G_{k+1}[g|X] = G_K[g] \prod_{\mathbf{x}\in X} G[g|\mathbf{x}], \tag{4.43}$$

where $G[g|\mathbf{x}] = (1 - p_D + p_D p_g)$, $p_g = \int g(\mathbf{z}) L_{\mathbf{z}}(\mathbf{x}) \mathrm{d}\mathbf{z}$ and $G_K[g] = e^{\lambda\kappa[g]-\lambda}$ by (4.12). Letting $h_0 = (1 - p_D + p_D p_g)$,

$$\begin{aligned} F[g,h] &= \int h^X h_0^X e^{\lambda\kappa[g]-\lambda} f_{k+1|k}(X|Z^k)\delta X && (4.44)\\ &= e^{\lambda\kappa[g]-\lambda} \int (h_0 h)^X f_{k+1|k}(X|Z^k)\delta X && (4.45)\\ &= e^{\lambda\kappa[g]-\lambda} G_{k+1|k}[h_0 h]. && (4.46) \end{aligned}$$

Using the assumption that $f_{k+1|k}$ is Poisson distributed with mean cardinality μ and probability density $s(\mathbf{x})$ along with (4.12) gives

$$G_{k+1|k}[h(1 - p_D + p_D p_g)] = \exp(\mu s[h(1 - p_D)] + \mu s[h p_D p_g] - \mu), \tag{4.47}$$

and therefore

$$F[g,h] = \exp(\lambda\kappa[g] - \lambda + \mu s[h(1 - p_D)] + \mu s[h p_D p_g] - \mu). \tag{4.48}$$

Using (4.39) and (4.42) the PHD can be computed as:

$$D_{k+1|k+1}(\mathbf{x}) = \frac{\frac{\delta}{\delta\mathbf{x}}\frac{\delta F}{\delta Z}\big|_{[g,h]=[0,1]}}{\frac{\delta F}{\delta Z}\big|_{[g,h]=[0,1]}}. \tag{4.49}$$

This can be evaluated with the help of two additional results:

$$\text{(i)} \quad \left.\frac{\delta F}{\delta Z}\right|_{[g,h]=[0,1]} = e^{-\lambda-\mu s[p_D]} \cdot \prod_{\mathbf{z}\in Z} (\lambda\kappa(\mathbf{z}) + \mu s[p_D L_{\mathbf{z}}]), \tag{4.50}$$

$$\begin{aligned} \text{(ii)} \quad \left.\frac{\delta}{\delta\mathbf{x}}\frac{\delta F}{\delta Z}\right|_{[g,h]=[0,1]} &= e^{-\lambda-\mu s[p_D]} \cdot \mu(1 - p_D(\mathbf{x}))s(\mathbf{x}) \prod_{z\in Z} (\lambda\kappa(\mathbf{z}) + \mu s[p_D L_{\mathbf{z}}])\\ &\quad + e^{-\lambda-\mu s[p_D]} \prod_{\mathbf{z}\in Z} (\lambda\kappa(\mathbf{z}) + \mu s[p_D L_{\mathbf{z}}]) \cdot \sum_{\mathbf{z}\in Z} \frac{\mu p_D(\mathbf{x}) L_{\mathbf{z}}(\mathbf{x}) s(\mathbf{x})}{\lambda\kappa(\mathbf{z}) + \mu s[p_D L_{\mathbf{z}}]}. \end{aligned} \tag{4.51}$$

The proofs of these results are given in Appendix B.1. In [73], Mahler suggests proving the two results above by induction without giving the details of the proof. As an alternative, the proof in Appendix B.1 is an original proof using a direct method.

Using (4.50) and (4.51) in (4.49) while remembering that $D_{k+1|k}(\mathbf{x}) = \mu s(\mathbf{x})$ gives:

$$D_{k+1|k+1}(\mathbf{x}) = (1 - p_D(\mathbf{x}))D_{k+1|k}(\mathbf{x}) + \sum_{\mathbf{z}\in Z_{k+1}} \frac{p_D(\mathbf{x}) L_{\mathbf{z}}(\mathbf{x}) D_{k+1|k}(\mathbf{x})}{\lambda\kappa(\mathbf{z}) + D_{k+1|k}[p_D L_{\mathbf{z}}]}. \quad \square \tag{4.52}$$

4.1.4 Example Calculation of the PHD Filter

In order the elucidate the PHD filter methodology and the interpretation of the PHD, a closed form solution of the PHD filter will be presented for the same simple scenario from Section 3.3.3.

In the scenario from Section 3.3.3 it was assumed that there was a 0.5 probability of a single target birth at a given time step and a 0.5 probability of zero target births. Due to the Poisson assumption required for the PHD filter it is necessary to approximate the birth process by a Poisson distribution with mean 0.5. The same is true of the false alarm process. In the notation of this section, the assumptions made are:

- $p_S = p_D = 1$.
- The PHD of the birth process is given by $b(x) = 0.5 \times \hat{b}(x)$ where $\hat{b}(x)$ is the probability density for target appearance.

$$b(x) = \begin{cases} x & \text{if } x \in [0,1], \\ 0 & \text{o.w.} \end{cases} \tag{4.53}$$

- The mean number of false alarms, $\lambda = 0.5$ and

$$\kappa(z) = \begin{cases} 2 - 2z & \text{if } z \in [0,1], \\ 0 & \text{o.w.} \end{cases} \tag{4.54}$$

- $L_z(x) = \mathcal{N}(z; x, 0.01)$.
- $f(x|w) = \mathcal{N}(x; w, 0.01)$.

The knowledge that there are no targets at time 0 gives the prior PHD, $D_{0|0} = 0$. Using (4.13), the predicted PHD, $D_{1|0}$ is given by

$$D_{1|0}(x) = \begin{cases} x & \text{if } x \in [0,1], \\ 0 & \text{o.w.} \end{cases} \tag{4.55}$$

The posterior PHD, $D_{1|1}$ when the measurement received is $Z_1 = \{0.5\}$ may be calculated using (4.31) as

$$D_{1|1}(x) = \begin{cases} x\mathcal{N}(0.5; x, 0.1) & \text{if } x \in [0,1], \\ 0 & \text{o.w.} \end{cases} \tag{4.56}$$

This is similar to the full multitarget Bayes filter calculation in Section 3.3.3 as given in (3.51) and is the function visualised in fig. 3.2 but note that whereas for the

multitarget Bayes filter a full cardinality distribution for X could be inferred, with the PHD only the mean of the cardinality distribution is calculated.

Applying (4.13) again to calculate the predicted PHD $D_{2|1}$ for time step 2 gives

$$D_{2|1}(x) = x + \int_0^1 \mathcal{N}(x; w, 0.1) w \mathcal{N}(0.5; w, 0.1) dw \tag{4.57}$$

Again, there is a close link to the equivalent multitarget Bayes filter calculation from Section 3.3.3. In fact, $D_{2|1} = 2 \times f_{2|1}(\{x\})$ where $f_{2|1}$ is the multitarget predicted density assuming that one target is present, which is visualised in fig. 3.3. Furthermore, the mean of the cardinality distribution is equal to 1, the same as for the full multitarget Bayes filter but, as before, the full cardinality distribution is not available for the PHD filter.

Finally, using (4.31) the posterior PHD for time step 2, $D_{2|2}$ is

$$D_{2|2}(x) = D_{2|1}(x) \sum_{i=2}^{4} \frac{\mathcal{N}(z_i; x, 0.1)}{(1 - z_i) + K_i}, \tag{4.58}$$

$$K_i = \int \left(x + \int_0^1 \mathcal{N}(x; w, 0.1) w \mathcal{N}(0.5; x, 0.1) dw \right) \mathcal{N}(z_i; x, 0.1) dx. \tag{4.59}$$

Figure 4.1 shows $D_{2|2}$ for the case when the measurements are $\{0.1, 0.5, 0.9\}$, as in Section 3.3.3. Note that the expected number of targets at time step 2, $N_{2|2} = \int D_{2|2}(x) dx = 1.507$. This demonstrates that the PHD filter is not able to take advantage of the information in the multitarget Bayes filter using the full cardinality distribution to infer that exactly two targets must be present. Regardless, the summary made of $D_{2|2}$ by considering fig. 4.1, might be that there are two target present in the vicinities of 0.5 and 0.8. This is the same summary that would be drawn from the full multitarget Bayes filter. One might conclude from this that despite the approximations, the important information has been preserved.

4.2 Implementing the PHD filter

Implementing the PHD filter using a grid-based method is no more feasible for practical purposes than using such a method for a Bayes filter as discussed in Section 2.1.3. Fortunately, two alternative methods have been established using a Sequential Monte Carlo approach (SMC-PHD) [124] or using Gaussian mixtures (GM-PHD) [122]. Aspects of these approaches will be discussed extensively in Chapters 4 and 5 and so the details for each is presented with brief explanations in Sections 4.2.1 and 4.2.2 for SMC-PHD and GM-PHD respectively. Additional considerations required

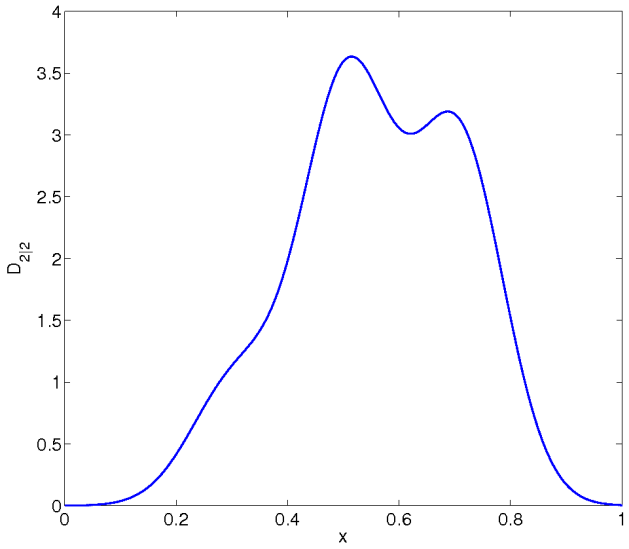


Figure 4.1: The posterior PHD at time step 2

for implementation such as state extraction and track continuity are briefly discussed in Section 4.2.3.

For ease of reference, the PHD filter equations from Section 4.1 are:

$$D_{k+1|k}(\mathbf{x}) = b_{k+1|k}(\mathbf{x}) + \int p_S(\mathbf{w}) f_{k+1|k}(\mathbf{x}|\mathbf{w}) D_{k|k}(\mathbf{w}) \mathrm{d}\mathbf{w}, \tag{4.60}$$

$$D_{k+1|k+1}(\mathbf{x}) = (1 - p_D(\mathbf{x})) D_{k+1|k}(\mathbf{x}) + \sum_{\mathbf{z} \in Z_{k+1}} \frac{p_D(\mathbf{x}) L_{\mathbf{z}}(\mathbf{x}) D_{k+1|k}(\mathbf{x})}{\lambda \kappa(\mathbf{z}) + D_{k+1|k}[p_D L_{\mathbf{z}}]}. \tag{4.61}$$

Here $b_{k+1|k}$ is the PHD of target birth density, p_S is the probability of a target surviving to the next time step, $f_{k+1|k}$ is the single target Markov transition, p_D is the probability of detection, $L_{\mathbf{z}}$ is the single target likelihood function for measurement $\mathbf{z}$, κ is the clutter density, λ is the average number of false alarms, and $f[g] = \int f(\mathbf{x}) g(\mathbf{x}) \mathrm{d}\mathbf{x}$.

Some Remarks on the PHD Filter Equations

1. It will be seen below that the computational complexity of implementations of the PHD filter is *linear in the number of targets and the number of measurements* [75]. Thus, PHD filter implementations have favourable computational characteristics with comparison to multitarget Bayes filter or MHT implementations which have combinatoric complexity.

2. The birth and clutter processes, given by $b_{k+1|k}(\mathbf{x})$ and $\lambda\kappa(\mathbf{z})$ are assumed to be Poisson distributed, so the variance of their cardinality is constrained by its mean. The effect of this constraint is discussed in a special case in Section 4.1.4.

3. The clutter density term, $\lambda\kappa(\mathbf{z})$, in the denominator of (4.61) moderates the impact of the component due to each measurement in the updated PHD. For example, if the measurement $\mathbf{z}$ is such that $\lambda\kappa(\mathbf{z})$ is large, this means that the measurement fell in a region of high clutter. Accordingly, the component in the updated PHD will receive a lower weighting. It may be difficult to accurately determine $\lambda\kappa(\mathbf{z})$ in real data. A strategy for determining this term could involve averaging known clutter points over a large set of past data to estimate the mean cardinality and the distribution of clutter.

4.2.1 The Sequential Monte Carlo PHD Filter

A Sequential Monte Carlo PHD (SMC-PHD) filter was presented in [124]. The form of SMC-PHD presented here is based on this version and might be considered a standard or 'vanilla' SMC-PHD implementation, whereas more recent developments will be presented in Section 5.1. A convergence result for the SMC-PHD filter will be briefly summarised at the end of this subsection.

The main difference between the SMC-PHD filter and the more familiar SMC filter discussed in Section 2.1.3 is that the sum of the particle weights for the SMC-PHD filter is not equal to 1 but rather to the expected number of targets present. As a result, extra attention is required at the prediction and resampling steps to manage the number of particles and their weights. The principles followed in obtaining the SMC-PHD filter equations presented are:

1. Allow the number of particles at time k, $N_{k|k}$ to vary with the expected number of targets, $E_{k|k}$, so that $N_{k|k} = \rho E_{k|k}$ where ρ is 'particles per expected target'.

2. Use standard importance sampling techniques for each term in the PHD filter equations (4.60-4.61).

3. The standard SMC approximation means that for particles $\{\mathbf{x}_k^{(i)}\}_i$ with weights $\{w_k^{(i)}\}_i$ at time k, then $D_{k|k}(\mathbf{x}) \approx \sum_i w_k^{(i)} \delta_{\mathbf{x}_k^{(i)}}(\mathbf{x})$.

Initialisation

Assume that the initial PHD, $D_{0|0}$, is known and the expected number of targets is $E_{0|0}$. The number of particles required is $N_{0|0} = \rho E_{0|0}$ and the particles are drawn from q_0 and given weights $w_0^{(i)}$ such that

$$\mathbf{x}_0^i \sim q_0(\cdot) = \frac{D_{0|0}(\cdot)}{E_{0|0}}, \tag{4.62}$$

$$w_0^i = \frac{1}{\rho}. \tag{4.63}$$

Prediction

Let $\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^{N_{k|k}}$ be the set of particles and associated weights from time step k. The predicted particle set $\{\mathbf{x}_{k+1|k}^{(i)}, w_{k+1|k}^{(i)}\}_{i=1}^{N_{k+1|k}}$ is obtained by treating the surviving targets and birth targets separately, using $N_{k|k}$ particles for the surviving targets and M particles for the birth targets so that $N_{k+1|k} = N_{k|k} + M$.

For the surviving targets, $i = 1, ..., N_{k|k}$:

$$\mathbf{x}_{k+1|k}^i \sim q(\cdot|\mathbf{x}_k^i, Z_{k+1}), \tag{4.64}$$

$$w_{k+1|k}^i = \frac{p_S(\mathbf{x}_k^i) f_{k+1|k}(\mathbf{x}_{k+1}^i, \mathbf{x}_k^i)}{q(\cdot|\mathbf{x}_k^i, Z_{k+1})} w_k^i. \tag{4.65}$$

Note that the choice $q(\cdot|\mathbf{x}_k^i, Z_{k+1}) = f_{k+1|k}(\cdot|\mathbf{x}_k^{(i)})$ (if this can be sampled from) gives $w_{k+1|k}^i = p_S(\mathbf{x}_k^{(i)}) w_k^i$.

Similarly, for birth targets, $i = N_{k|k} + 1, ..., N_{k|k} + M$:

$$\mathbf{x}_{k+1|k}^i \sim \hat{q}(\cdot|Z_{k+1}), \tag{4.66}$$

$$w_{k+1|k}^i = \frac{b(\mathbf{x}_{k+1|k}^i)}{M\hat{q}(\cdot|Z_{k+1})}. \tag{4.67}$$

The choice $\hat{q}(\cdot|Z_{k+1}) = b_{k+1|k}(\cdot)/B_{k+1|k}$ where $B_{k+1|k} = \int b_{k+1|k}(\mathbf{x})\mathrm{d}\mathbf{x}$, which is the expected number of birth targets, gives $w_{k+1|k}^i = B_{k+1|k}/M$.

Update

For $i = 1, ..., N_{k+1|k}$ the updated weights $\{\hat{w}_{k+1|k+1}^i\}$ are computed using (4.61) as:

$$\hat{w}_{k+1|k+1}^i = \left[1 - p_D(\mathbf{x}_{k+1|k}^i) + \sum_{\mathbf{z} \in Z_{k+1}} \frac{p_D(\mathbf{x}_{k+1|k}^i) L_z(\mathbf{x}_{k+1|k}^i)}{\kappa(\mathbf{z}) + \langle w_{k+1|k}, p_D L_{\mathbf{z}} \rangle}\right] w_{k+1|k}^i, \tag{4.68}$$

where the inner product $\langle w, f \rangle = \sum_i f(\mathbf{x}^i) w^i$.

Resample

The number of expected targets after the update is given by
$E_{k+1|k+1} = \sum_i w_{k+1|k+1}^i$ and hence the number of particles required after resampling is $N_{k+1|k+1} = int(\rho E_{k+1|k+1})$ where $int(P)$ is the nearest integer to P. Therefore, the weights $\{\hat{w}_{k+1|k+1}^i\}$ are normalised to form a discrete probability density, and $N_{k+1|k+1}$ particle selections are made from the set of particles with individual selections proportional to $\{\hat{w}_{k+1|k+1}^i\}$. The weights after resampling are $w_{k+1|k+1}^i = 1/\rho \quad \forall i$.

The full process is given in pseudocode form in Table 4.1 and an overview is given in fig. 4.2.

Convergence Result

It has been shown in [26] and [53], that under some weak assumptions of continuity and boundedness on $f_{k+1|k}$, $L_{\mathbf{z}}$ and the importance ratios in (4.65) and (4.67) that:

$$\mathbb{E}\Big[\Big(\int D_{k|k}(\mathbf{x})\psi(\mathbf{x})\mathrm{d}\mathbf{x} - \sum_{i}^{N} w_{k|k}^{(i)}\psi(\mathbf{x}_k^{(i)})\delta_{\mathbf{x}_k^{(i)}}(\mathbf{x})\Big)^2\Big] \leq \frac{c_k||\psi||^2}{N}, \tag{4.69}$$

for all bounded ψ, where c_k is independent of N. Therefore the mean squared error in the particle approximation to the true PHD converges to zero as the number of particles, $N \to \infty$.

4.2.2 The Gaussian Mixture PHD Filter

The Gaussian mixture PHD (GM-PHD) filter, originally presented in [122] is an alternative method which approximates the PHD by a Gaussian sum similarly to the approach in [111] for standard probability densities. The GM-PHD filter provides an exact closed form solution to the PHD recursion (4.60-4.61) in the case where forward and measurement models are linear, p_D and p_S are constant and all noise is Gaussian and the birth density is a Gaussian sum. Note that the latter requirement can automatically be satisfied as any probability density may be approximated with arbitrary accuracy by a Gaussian sum [111]. The GM-PHD filter recursion bears a similarity to the Kalman filter equations given in Section 2.1.2 and hence might be readily understood by readers already familiar with the Kalman filter.

Formally, the assumptions required are:

1. $f_{k+1|k}(\mathbf{x}|\mathbf{x}_k) = \mathcal{N}(\mathbf{x}; F_k\mathbf{x}_k, Q_k)$.
2. $L_{\mathbf{z}}(\mathbf{x}) = \mathcal{N}(\mathbf{z}; H_k\mathbf{x}, R_k)$.
3. p_S and p_D are constant.
4. $b_{k+1|k}(\mathbf{x}) = \sum_{i=1}^{J_{\gamma,k}} w_{\gamma,k}^i \mathcal{N}(\mathbf{x}; m_{\gamma,k}^i, P_{\gamma,k}^i)$.

Here, F_k and H_k are the forward and measurement matrices and Q_k and R_k are the process noise and measurement noise covariance matrices respectively, as in Section 2.1.2.

Prediction

If the PHD at time k is a Gaussian mixture of the form $D_{k|k}(\mathbf{x}) = \sum_{i=1}^{J_k} w_k^i \mathcal{N}(\mathbf{x}; m_k^i, P_k^i)$, then the predicted PHD is given by

$$D_{k+1|k}(\mathbf{x}) = D_{S,k+1|k}(\mathbf{x}) + b_{k+1|k}(\mathbf{x}), \tag{4.70}$$

where $b_{k+1|k}(\mathbf{x})$ is as given in assumption 4. Then

$$\begin{aligned} D_{S,k+1|k}(\mathbf{x}) &= p_S \sum_{i=1}^{J_k} w_k^i \mathcal{N}(\mathbf{x}; m_{S,k+1|k}, P_{S,k+1|k}), && (4.71)\\ m_{S,k+1|k}^i &= F_k m_k^i, && (4.72)\\ P_{S,k+1|k}^i &= Q_k + F_k P_k^i F_k^T. && (4.73) \end{aligned}$$

This result is proved using (2.19) and the proof is given in Appendix B.2.

Update

If the predicted PHD at time $k+1$ is a Gaussian mixture of the form $D_{k+1|k}(\mathbf{x}) = \sum_{i=1}^{J_{k+1|k}} w_{k+1|k}^i \mathcal{N}(\mathbf{x}; m_{k+1|k}^i, P_{k+1|k}^i)$, then the updated PHD is given by

$$D_{k+1|k+1}(\mathbf{x}) = (1 - p_D) D_{k+1|k}(\mathbf{x}) + \sum_{\mathbf{z} \in Z_{k+1}} D_{D,k}(\mathbf{x}; \mathbf{z}), \tag{4.74}$$

where

$$\begin{aligned} D_{D,k}(\mathbf{x}; \mathbf{z}) &= \sum_{i=1}^{J_{k+1|k}} w_k^i(\mathbf{z}) \mathcal{N}(\mathbf{x}; m_{k+1|k+1}^i(\mathbf{z}), P_{k+1|k+1}^i), && (4.75)\\ w_k^i(\mathbf{z}) &= \frac{p_D w_{k+1|k}^i q_k^i(\mathbf{z})}{\kappa(\mathbf{z}) + p_D \sum_{j=1}^{J_{k+1|k}} w_{k+1|k}^j q_k^j(\mathbf{z})}, && (4.76)\\ q_k^i(\mathbf{z}) &= \mathcal{N}(\mathbf{z}; H_k m_{k+1|k}^i, R_k + H_k P_{k+1|k}^i H_k^T), && (4.77)\\ m_{k+1|k+1}^i(\mathbf{z}) &= m_{k+1|k}^i + K_k^i(\mathbf{z} - H_k m_{k+1|k}^i), && (4.78)\\ P_{k+1|k+1}^i &= [\mathbb{I} - K_k H_k] P_{k+1|k}^i, && (4.79)\\ K_k^i &= P_{k+1|k}^i H_k^T (H_k P_{k+1|k}^i H_k^T + R_k)^{-1}. && (4.80) \end{aligned}$$

This result is proved using (2.19) and the proof is given in Appendix B.2.

Pruning and Merging

It can be seen from (4.74) and (4.75) that the number of components in the Gaussian mixture will explode exponentially with time. This is due to the fact that there is a Gaussian component for every measurement/component from predicted PHD pair

and is similar to the problem of MHT in associating every track with every measurement (see Section 3.1.2). However, whereas for the MHT filter it is necessary to apply ad hoc pruning schemes, for the GM-PHD filter it is possible to use established techniques for reducing the number of components in the Gaussian mixture. In [122], it is suggested to use some combination of the following three methods:

- Pruning - deleting all Gaussian components with associated weight below a threshold T.
- Merging - merge Gaussian components with means m_1, m_2 and covariances P_1, P_2 whenever $(m_1 - m_2)P_1^{-1}(m_1 - m_2) < U$ for some threshold U. Merging is performed using the joining algorithm described in [103].
- Limiting - limit the number of Gaussian components by choosing only the V components with the highest associated weights.

The details of these processes are given in more detail in the GM-PHD pseudocode at the end of this section and an overview is given in fig. 4.3. Clark and Vo [30] show how to choose T and U so as to maintain L_1 convergence to the true PHD filter.

Nonlinear Forward and Measurement Models

In the single target case, if the deterministic part of the forward or measurement model is nonlinear but all of the noise is Gaussian it may be desirable to use an Extended or Unscented Kalman filter. Similarly, in the multitarget case there are extended and unscented variations on the GM-PHD filter for cases with nonlinear forward or measurement models.

Consider the equations for the means and covariances (4.72-4.73) in the prediction step and (4.78-4.80) in the update step. These values are equivalent to Kalman filter computations. By replacing these values with their respective Extended Kalman Filter (EKF) and Unscented Kalman filter (UKF) computations given in Section 2.1.3, it is possible to obtain the EKF-GM-PHD and UKF-GM-PHD. The full details of the schemes may be found in the paper by Vo and Ma [122] who describe the details as "conceptually straightforward but notationally cumbersome" (which is why they are omitted here).

Remark: The discussion in Section 2.2 about the relative merits of EKF, UKF and the particle filter for nonlinear filtering should be considered when deciding to apply EKF-GM-PHD, UKF-GM-PHD or SMC-PHD.

4.2.3 State Extraction and Track Continuity

Sections 4.2.1 and 4.2.2 gave two possible methods for calculating a PHD recursion. However, most tracking systems are required to output estimates of target location, possibly with a method for monitoring trajectories of tracks present continuously for a number of time steps. The standard methods for performing these tasks will be briefly summarised.

State Extraction

It is possible to extract target location estimates from the PHD by looking for 'peaks' in the PHD. The number of peaks to select may be chosen based on the expected number of targets present, which is the integral of the PHD.

SMC-PHD - In SMC-PHD the integral of the PHD is simply the sum of the particle weights, denoted $\hat{N}$ here. An obvious choice for the number of peaks to extract is $N = int(\hat{N})$. Extracting N peaks from the particle set representing the PHD requires the clustering of the particles in N partitions, each of which is taken to represent an individual target [27] and declared. The thesis by Clark [24] contains a detailed comparison of various clustering methods for state extraction in SMC-PHD and concludes that k-means clustering performs the best. The details of the k-means algorithm may be found in [59]. The clustering step partitions the particles and, from each partition, a mean and covariance may be computed. The sum of the weights of particles in each partition represents the expected number of true targets associated with the partition, and this may be used as a measure of confidence in the hypothesis that the partition represents a true target. State extraction and track continuity for the SMC-PHD filter will be discussed further in Section 5.1.

GM-PHD - For the GM-PHD filter, it is noted that the weight of each Gaussian component is the number of expected targets corresponding to the Gaussian. Thus, the components with weights high enough to be considered true targets are declared.

Track Continuity

Methods for track continuity are presented in [25] and [88] for the SMC-PHD filter and [87] for the GM-PHD filter. Both methods are based on assigning labels to individual particles or Gaussian components which are maintained through the prediction step in order to allow association of tracks over time steps. This will be discussed further in the SMC-PHD case in Section 5.1.

Initialise
for $i = 1, ..., N_{0|0}$
  sample $\mathbf{x}_0^i \sim q_0(\cdot)$
  $w_0^i = 1/\rho$
end

for $k = 0, 1, ...$

  Predict
  [surviving targets]
  for $i = 1, ..., N_{k|k}$
    sample $\mathbf{x}_{k+1|k}^i \sim q_k(\cdot|\mathbf{x}_k^i, Z_{k+1})$
    $w_{k+1|k}^i = (p_S(\mathbf{x}_k^i) f_{k+1|k}(\mathbf{x}_{k+1}^i, \mathbf{x}_k^i) w_k^i)/(q_k(\mathbf{x}_{k+1|k}^i|\mathbf{x}_k^i, Z_{k+1}))$
  end
  [birth targets]
  for $i = N_{k|k} + 1, ..., N_{k|k} + M$
    sample $\mathbf{x}_{k+1|k}^i \sim \hat{q}_k(\cdot|Z_{k+1})$
    $w_{k+1|k}^i = b(\mathbf{x}_{k+1|k}^i)/M\hat{q}_k(\mathbf{x}_{k+1}^i|Z_{k+1})$
  end

  Update
  for $\mathbf{z} \in Z_{k+1}$
    $C_k(\mathbf{z}) = \sum_{i=1}^{N_{k|k}+M} w_{k+1|k}^i p_D(\mathbf{x}_{k+1|k}^i) L_{\mathbf{z}}(\mathbf{x}_{k+1|k})$
    for $i = 1, ..., N_{k|k} + M$
      $w_{k+1|k+1}^i(\mathbf{z}) = \left[1 - p_D(\mathbf{x}_{k+1|k}^i) + \sum_{z \in Z_{k+1}} \frac{p_D(\mathbf{x}_{k+1|k}^i) L_{\mathbf{z}}(\mathbf{x}_{k+1|k}^i)}{\kappa(\mathbf{z}) + C_k(\mathbf{z})}\right] w_{k+1|k}^i$
    end
  end

  Resample
  • *compute* $N_{k+1|k+1} = \rho \sum_{\mathbf{z} \in Z_{k+1}} \sum_{i=1}^{N_{k|k}+M} w_{k+1|k+1}^i(\mathbf{z})$
  • *resample* $\{\mathbf{x}_{k+1|k}^i, w_{k+1|k+1}^i(\mathbf{z})\}_{i,\mathbf{z}}$ $N_{k+1|k+1}$ times to give the set $\{\mathbf{x}_{k+1|k+1}^i, 1/\rho\}_i$

  State Extraction
  • *compute* $T_k = int(\sum_i^{N_{k+1|k+1}} w_{k+1|k+1})$
  • *partition* $\{\mathbf{x}_{k+1|k+1}^i\}_i$ into T_k sets using k-means algorithm
  • *declare* the mean and covariance of each partition as a separate target estimate

end

Table 4.1: Pseudocode for the SMC-PHD filter with state extraction

Given $\{w_0^i, m_0^i, P_0^i\}_{i=1}^{J_0}$

for $k = 0, 1, ...$

Predict

for $i = 1, ..., J_k$

$w_{k+1|k}^i = p_S w_k^i \;\;,\;\; m_{k+1|k}^i = F_k m_k^i \;\;,\;\; P_{k+1|k}^i = Q_k + F_k P_k^i F_k^T$

end

for $i = 1, ..., J_\gamma$

$w_{k+1|k}^{i+J_k} = w_{\gamma,k}^i \;\;,\;\; m_{k+1|k}^{i+J_k} = m_{\gamma k}^i \;,\; P_{k+1|k}^{i+J_k} = P_{\gamma,k}^i$

end

$J_{k+1|k} = J_k + J_\gamma$

Update

for $i = 1, ..., J_{k+1|k}$

$S_k^i = H_k P_{k+1|k}^i H_k^T + R_k \;,\; K_k^i = P_{k+1|k}^i H_k^T (S_k^i)^{-1}$

$P_{k+1}^i = [\mathbb{I} - K_k^i H_k] P_{k+1|k}^i$

for $\mathbf{z} \in Z_{k+1}$

$q_k^i(\mathbf{z}) = \mathcal{N}(\mathbf{z}; H_k m_{k+1|k}^i, S_k^i) \;,\; C_k(\mathbf{z}) = \sum_{j=1}^{J_{k+1|k}} w_{k+1|k}^j q_k^j(\mathbf{z})$

$w_{k+1}^i(\mathbf{z}) = p_D w_{k+1|k}^i q_k^i(\mathbf{z}) / (\kappa(\mathbf{z}) + C_k(\mathbf{z}))$

$m_{k+1}^i(\mathbf{z}) = m_{k+1|k}^i + K_k^i(\mathbf{z} - H_k m_{k+1|k}^i)$

end

end

[Reorganise components]

for $i = 1, ..., J_{k+1|k}$

$j = 0$

for $\mathbf{z} \in Z_{k+1}$

$j = j + 1$

$w_{k+1}^{(i-1)(|Z_{k+1}|+1)+j} = w_{k+1}^i(\mathbf{z}) \;,\; m_{k+1}^{(i-1)(|Z_{k+1}|+1)+j} = m_{k+1}^i(\mathbf{z})$

$P_{k+1}^{((i-1)(|Z_{k+1}|+1)+j} = P_{k+1}^i$

end

$w_{k+1}^{i\cdot(|Z_{k+1}|+1)} = (1 - p_D) w_{k+1|k}^i \;,\; m_{k+1}^{i\cdot(|Z_{k+1}|+1)} = m_{k+1|k}^i \;,\; P_{k+1}^{i\cdot(|Z_{k+1}|+1)} = P_{k+1|k}^i$

end

$J_{k+1} = J_{k+1|k} \cdot (|Z_{k+1}| + 1)$

Prune/Merge

$I = \{i = 1, ..., J_{k+1} | w_{k+1}^i > T\}$

• *replace* $\{w_{k+1}^i, m_{k+1}^i, P_{k+1}^i\}_{i=1}^{J_{k+1}}$ by $\{w_{k+1}^i, m_{k+1}^i, P_{k+1}^i\}_{i \in I}$

$l = 0$

while $I \neq \emptyset$

$l = l + 1$

$j = argmax_{i \in I}(w_{k+1}^i)$

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

$\hat{w}_{k+1}^l = \sum_{i \in L} w_{k+1}^i \;,\; \hat{m}_{k+1}^l = \sum_{i \in L} w_{k+1}^i m_{k+1}^i$

$\hat{P}_{k+1} = \frac{1}{w_{k+1}^l} \sum_{i \in L} w_{k+1}^i (P_{k+1}^i + (\hat{m}_{k+1}^l - m_{k+1}^i)(\hat{m}_{k+1}^l - m_{k+1}^i)^T)$

$I = I \setminus L$

end

• *replace* $\{w_{k+1}^i, m_{k+1}^i, P_{k+1}^i\}_{i=1}^{J_{k+1}}$ by $\{\hat{w}_{k+1}^i, \hat{m}_{k+1}^i, \hat{P}_{k+1}^i\}_{i=1}^{L}$

end

Table 4.2: Pseudocode for the GM-PHD filter with pruning and merging

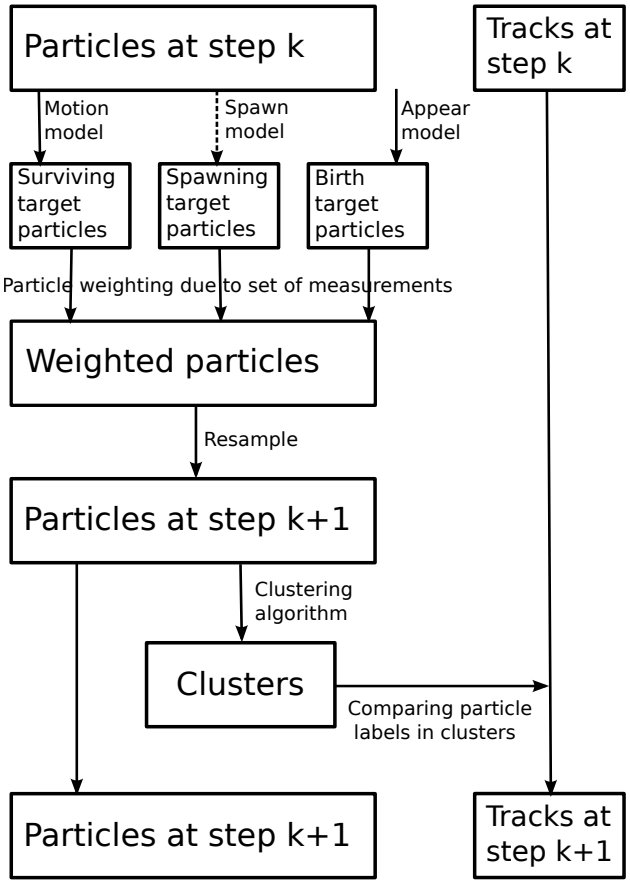


Figure 4.2: Flowchart illustrating the implementation of the SMC-PHD filter.

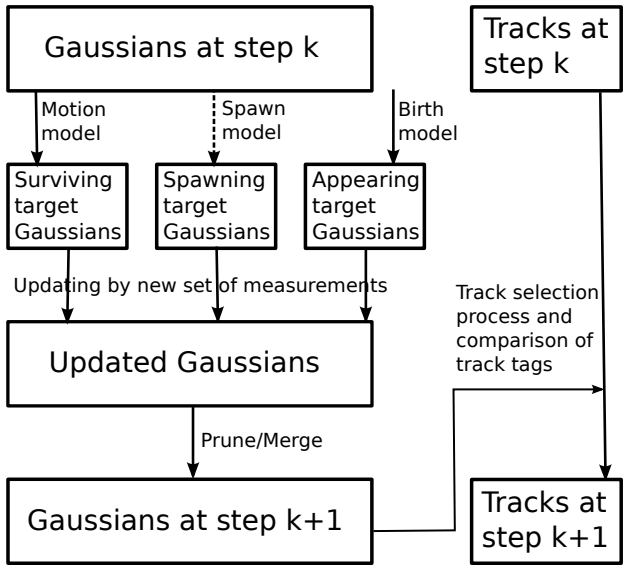


Figure 4.3: Flowchart illustrating the implementation of the GM-PHD filter.

Chapter 5

Multitarget Tracking: Analysis and Extensions

In Chapters 3 and 4, the random finite set method for multitarget tracking was described. A full analytical solution to the multitarget tracking problem in the form of the multitarget Bayes filter was derived, but it was seen that this would be computationally intractable in any realistic tracking scenario. The PHD filter provides a principled approximation to the multitarget Bayes filter and opens the possibility of implementation through sequential Monte Carlo and Gaussian mixture methods.

This chapter will focus on analysis and extensions to the PHD filter. The aim is to permit PHD filter implementations to tracking scenarios beyond the 'vanilla' case considered in Chapter 4 and to understand these implementations. This understanding will be drawn from observing equivalence or similarity between the PHD filter and other established solutions, particularly those which relate to Bayes filtering.

The issues considered will be manoeuvring targets within the jump Markov linear system framework as in Section 2.4, inclusion of target amplitude information, extensions to the SMC-PHD filter and the elaboration of a multitarget miss distance to aid performance analysis.

In Section 5.1, there is a discussion of some important problems with state extraction and target birth within the standard SMC-PHD filter which make the implementation described in Section 4.2 generally implausible. Extensions correcting these problems are presented leading to the improved SMC-PHD filter with track continuity. A general framework for manoeuvre handling within the PHD filter is presented in Section 5.2. In Section 5.3 a scheme for the inclusion of target amplitude information is presented, showing close correspondence with the track-before-detect methodology. Section 5.4 develops the concept of a miss distance for multitarget states, the existence of which is fundamental to estimation theory.

5.1 SMC-PHD Filter

There are two issues which cause significant difficulties in implementing the standard SMC-PHD filter: clustering and target birth. This section will elucidate the nature of these problems and explore some new methodologies for alleviating them. This eventually gives rise to the improved SMC-PHD filter with track continuity [96] [136].

5.1.1 Issues with the Standard SMC-PHD Filter

Clustering

As discussed in Section 4.2, the standard SMC-PHD filter propagates a particle representation of the PHD. In order to extract state estimates from the particle representation it is necessary to cluster the particle set into partitions with each partition representing a separate target. Each partition's associated mean, covariance and weight can then be used for target declaration. The clustering methods that have been employed in [124] [25] [38] [88] [117], among others, take the number of clusters as an input parameter. This parameter is typically chosen as the nearest integer to the expected target number. It was first observed in [124] that this form of clustering might lead to unreliable results when the expected target number is incorrect. The existence of problems with the clustering step is well known; in [90], for example, "unreliability of clustering techniques for extracting state estimates" is given as the reason for eschewing the SMC implementation of the PHD filter altogether.

Clustering Methods - This will be a brief overview of the clustering methods employed in the literature for the SMC-PHD filter including their computational complexity and convergence properties.

A possible clustering method is hierarchical clustering, which returns a hierarchy of clusters from a single cluster containing all particles to a set of clusters each containing one particle. Letting N be the number of particles, the complexity of the algorithm is $O(N^2 logN)$, which may make it an impractical choice for large N.

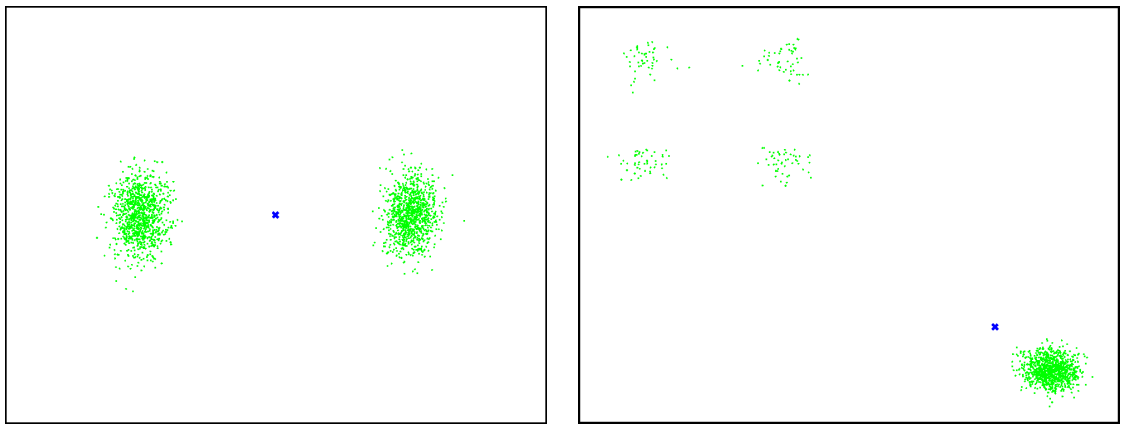
The articles [88] and [25] among others mention the use of the expectation maximisation (EM) algorithm [34] to fit a Gaussian Mixture to the data. For this algorithm, if the number of partitions is k, and t is the number of iterations, the computational complexity is $O(Nk^2t)$. An alternative to EM is k-means clustering which is investigated in depth in Daniel Clark's PhD thesis [24]. Clark finds that k-means clustering is able to outperform EM while also having a lower complexity of $O(Nkt)$. K-means clustering is widely used: a 2002 survey on data mining techniques states that it is

"by far the most popular clustering algorithm used in scientific and industrial applications". For these reasons, k-means clustering will be the algorithm of choice for clustering for the rest of this section.

There are some disadvantages to using k-means. It is an ad-hoc method and convergence to the optimal clustering is not guaranteed. In fact, there are examples for which the algorithm generates arbitrarily bad clusterings by getting stuck in a local minimum. Clark et al. [28] suggest using the extension to k-means proposed by Kanungo et al. [59] guaranteeing convergence arbitrarily close to the optimal solution. However, this comes at a cost of a complexity proportional to N^3 making it slower than hierarchical clustering and hence impractical for many applications. Alternatively, a newer method by developed by Arthur and Vassilvitskii [4], given the name `k-means++`, is able to guarantee an upper bound on the expectation of the ratio between the k-means clustering and the optimal clustering with no significant computational burden. This is achieved simply by a careful probabilistic choice of initial centres in the k-means algorithm. Due to its favourable computational characteristics, this is the method that will be employed. Full details of k-means with the initial centres chosen as in [4] are available in pseudo-code in Table 5.2.

Choosing the 'k' in k-means - Vo et al. [124] point out that "when the estimated [target] number is incorrect, the clustering output becomes unreliable". To see why this is the case consider the scenario depicted in fig. 5.1(a). A casual inspection of the particle distribution suggests two targets. However, the expected target number falls slightly below 1.5 and so rounds to 1. K-means clustering of the particle distribution with $k = 1$ gives the unhelpful cluster centre depicted.

It is possible to go further than the problematic case already identified and present an example where the estimated target number is correct but the clustering output is still doomed to become unreliable. Fig. 5.1(b) shows a scenario where there is one group of particles in the bottom right corner with total associated weight 1 and four other groups of particles in the top left hand corner each corresponding to a false alarm and having total associated weight 0.05. In this case, the particle distribution is reasonable and the rounded expected target number is correct but the centre returned by clustering is useless and in a tracking context would be likely to lead to the loss of the track. In short, inevitable false alarms will generate low weight components of the PHD which will not be allocated separate partitions but instead simply skew the centre away from the true target.



(a) (b)

Figure 5.1: Two examples of particle representation of PHDs. Particles are denoted by green dots and the resulting k-means centres are denoted by blue crosses.
a) Two targets each with associated weight 0.7.
b) One target with weight 1 and four false alarms with weight 0.05.

The problem of choosing the 'k' in k-means is considered by Clark et al. in [27] who suggest that k-means should be applied for all values of k with the average distortion (average of the distances of each particle to the nearest centre) computed. It is clear that as k increases, distortion will decrease and so Clark et al. suggest using the rate of change of distortion with respect to k and 'some judgement' to determine the correct value of k. A potential problem with such an approach is highlighted in [140] which points out that in such a method "the problem of determining the number of clusters is converted into a parameter selection problem, and the resulting number of clusters is largely dependent on parameter tweaking".

Target Birth

In general, the target birth density will cover the entire surveillance region whereas the likelihood due to single measurements will only have non-negligible values for a comparatively small region. Thus, in order to permit target births in the SMC-PHD filter, it will be necessary to sample particles covering the entire surveillance region with a sampling sufficiently dense as to be accurate within the localisation of a single target likelihood function. This may require a large number of samples to be allocated to target birth and could lead to impractical levels of computation.

A simple numerical simulation was carried out to illustrate this point. The simulation concerned a sensor with region of surveillance $[0, 40] \times [0, 40]$. The model parameters for the PHD recursion were $p_D = 1$, $p_B = 0.1$, $\lambda = 10$ and $\kappa = constant$

on the measurement space. Measurements were taken of the x and y location of the target with zero-mean, white Gaussian noise with standard deviation $\sigma = 0.1$ added to each component. A single target appeared in the centre of the surveillance region and the target was considered to be identified if the expected target number was greater than 0.75 after 5 time steps. 100 Monte Carlo runs of the test were performed with a GM-PHD filter with 25 Gaussian components assigned to approximating the birth density, and an SMC-PHD filter with various numbers of particles assigned to target birth. The results are given in Table 5.1. It can be seen from the results that it requires a very large number of birth particles to approach the results obtained easily by the GM-PHD filter. The computation for the SMC-PHD filter with 100000 birth particles took over 6000 times longer than the GM-PHD filter.

	GM	SMC-1000	SMC-10000	SMC-100000
% tracked	100	8	30	76

Table 5.1: Percentage of the tests in which the target passed a declaration threshold of 0.75 for various PHD filter implementations. SMC-N denotes the SMC-PHD filter with N particles assigned to target birth

5.1.2 Improved Clustering for the SMC-PHD Filter

In order to alleviate problems with the clustering step in the SMC-PHD filter, an improved clustering step is proposed. The improved clustering step involves jointly determining the number of clusters and the clusters themselves. The improved clustering step follows on from the ideas of Clark et al. [27] in attempting a range of values of k, the number of clusters, but the extra judgement required is replaced by a criterion which exploits existing information about the tracking scenario.

The following assumptions are made:

1. $L_{\mathbf{z}}(\mathbf{x}) = \mathcal{N}(\mathbf{z}; h(\mathbf{x}), R)$ (where $\mathcal{N}(\mathbf{a}; \mathbf{b}, C)$ is the normal distribution with variable $\mathbf{a}$ which has mean $\mathbf{b}$ and covariance C).

2. The length scale of change in $L_{\mathbf{z}}(\mathbf{x})$ is short by comparison with the length scale of change for $b_{k+1|k}(\mathbf{x})$ so that the particles corresponding to a single target upon first detection are drawn approximately from $L_{\mathbf{z}}(\mathbf{x})$.

3. The distribution of particles corresponding to a given target are most diffuse upon first detection of that target (as subsequent updates will tend to make the distribution more "peaked").

Considering the above assumptions, which are reasonable for many tracking applications, it can be seen that clustering in this context has more structure than a general clustering problem. The particles corresponding to a single target should be distributed either like $L_{\mathbf{z}}(\mathbf{x})$ in the measurement space or possibly more "peaked" than $L_{\mathbf{z}}(\mathbf{x})$ for established targets. A group of particles more diffuse than $L_{\mathbf{z}}(\mathbf{x})$ in the measurement space should not correspond to a single target and hence should not be a cluster.

The expression,

$$d^2 = \mathbf{v}^T R^{-1} \mathbf{v}, \tag{5.1}$$

where $\mathbf{v} = \mathbf{z} - h(\mathbf{x})$ is a statistical distance of a realisation of the likelihood function from its mean. It is typically to approximate [15, p. 337]

$$d^2 \sim \chi^2_{dim(\mathbf{z})}. \tag{5.2}$$

Hence, using the cumulative distribution function for χ^2_k it can be seen that if a proportion p of realisations of $L_{\mathbf{z}}(\mathbf{x})$ were to satisfy $d^2 \leq \gamma$ then

$$p = P\big(\frac{dim(\mathbf{z})}{2}, \frac{\gamma}{2}\big), \tag{5.3}$$

where $P(s,x) = \frac{1}{\Gamma(s)} \int_0^x e^{-t} t^{s-1} \mathrm{d}t$ is the regularised gamma function. This suggests a criteria to determine if a cluster of particles corresponds to a single target under the assumptions above. Table 5.3 gives the pseudocode for the improved clustering algorithm which tries increasing the number of clusters until a clustering can be found which satisfies the condition that all clusters are at least as "peaked" as $L_{\mathbf{z}}(\mathbf{x})$ in the measurement space and hence correspond to a single target. Numerical simulations to test the performance of the SMC-PHD filter with improved clustering algorithm are given in Section 5.1.4.

5.1.3 Improved SMC-PHD filter with Track Continuity

The improved SMC-PHD filter was first presented in [96] and was extended in [136] to permit track continuity. The author's contribution to the jointly authored paper [136] is outlined in Section 1.4.1. The improved SMC-PHD filter avoids issues with clustering by performing state extraction without the use of clustering and dramatically reduces the number of particles required for target birth by only sampling particles close to measurements. The latter may potentially cause bias in the cardinality of the PHD filter, so it is necessary to re-formulate the PHD filter, keeping the birth

Given $\{x_i\}_{i=1}^N, k$

Initialisation

• *Choose* m_1^1 randomly from $\{x_i\}_{i=1}^N$

for $j = 2, ..., k$

• *Compute* $D(\mathbf{x}_i)$, the distance from $\mathbf{x}_i$ to the nearest centre $\forall i$

• *Choose* m_j^1 from $\{x_i\}_{i=1}^N$ with probability $D(x_i)^2 / \sum_{i=1}^N D(x_i)^2$

$P_i^1 = 1 \quad \forall i$

end

Recursion

for $l = 2, 3, ...$

for $i = 1, ...N$

$P_i^l = \arg\min_j ||x_i - m_j^{l-1}||$

end

for $j = 1, ..., k$

$N_j = \{i : P_i^l = j\}$

$m_j^l = \sum_{N_j} x_i / |N_j|$

end

if $|\sum_{i=1}^N ||x_i - m_{P_i^l}^l|| - \sum_{i=1}^N ||x_i - m_{P_i^{l-1}}^{l-1}||| \leq \epsilon$

break;

end

end

Output - $P_i = P_i^l \quad \forall i \quad m_j = m_j^l, \ , C_j = \sum_{i:P_i=j} (x_i - m_j)^2 \quad \forall j$

Table 5.2: Pseudocode for the `k-means++` algorithm

Given $\{x_i\}_{i=1}^N$ and p, γ satisfying (5.3)

for k = 1,2,...

$[m_j, C_j, P_i] = kmeans++(\{x_i\}_{i=1}^N, k)$

for $j = 1, ..., k$

$N_j = \{i : P_i = j\}$, $D_j = 0$

for $i = N_j$

$v_i = h(m_j) - h(x_i), d_i^2 = v_i^T R^{-1} v_i$

if $d_i^2 > \gamma$

$D_j = D_j + 1$

end

end

end

if $D_j / |N_j| \leq p$

break;

end

end

Output- $[m_j, C_j, P_i]$

Table 5.3: Pseudocode for the improved clustering algorithm

process separate from the persisting targets.

Measurement Dependent Birth Density

A formulation of the PHD filter is derived which propagates separate PHDs for persistent and newborn targets. These are recombined at the end of each time step as the newborn targets cease to be new. Sampling from the newborn PHD is dependent on the measurements resulting in a dramatic reduction in the number of particles required for target birth.

Include a label, β, in the target state vector to distinguish newborn targets from persistent targets so that the augmented state vector is $\hat{\mathbf{x}} = (\mathbf{x}, \beta)$ where

$$\beta = \begin{cases} 0 & \text{for a persistent object,} \\ 1 & \text{for a newborn object.} \end{cases} \tag{5.4}$$

As the birth density only applies to newborn targets, the birth PHD $\gamma_{k+1|k}$ is:

$$\gamma_{k+1|k}(\hat{\mathbf{x}}) = \begin{cases} \gamma_{k+1|k}(\mathbf{x}) & \beta = 1, \\ 0 & \beta = 0. \end{cases} \tag{5.5}$$

Similarly, because a newborn target becomes a persistent target after one time step, but a persistent target cannot become a newborn target, the single target transition density $f_{k+1|k}$ is:

$$f_{k+1|k}(\hat{\mathbf{x}}|\hat{\mathbf{x}}') = f_{k+1|k}(\mathbf{x}, \beta|\mathbf{x}', \beta') \tag{5.6}$$

$$= \begin{cases} 0 & \beta = 1, \\ f_{k+1|k}(\mathbf{x}|\mathbf{x}') & \beta = 0. \end{cases} \tag{5.7}$$

As p_S is independent of β, substituting (5.5-5.7) into the PHD predictor equation (4.60) gives:

$$D_{k+1|k}(\mathbf{x}, \beta) = \begin{cases} \gamma_{k+1|k}(x) & \beta = 1, \\ \langle D_{k|k}(\cdot, 1) + D_{k|k}(\cdot, 0), p_S f_{k+1|k}(x|\cdot)\rangle & \beta = 0. \end{cases} \tag{5.8}$$

Using this formulation, the PHD update equation (4.61) is:

$$D_{k+1|k+1}(\mathbf{x}, \beta) = (1 - p_D(\mathbf{x}, \beta)) D_{k+1|k}(\mathbf{x}, \beta) + \sum_{\mathbf{z} \in Z_k} \frac{p_D(\mathbf{x}, \beta) L_{\mathbf{z}}(\mathbf{x}, \beta) D_{k+1|k}(\mathbf{x}, \beta)}{\lambda \kappa(\mathbf{z}) + \sum_{\beta=0}^{1} \langle p_D(\cdot, \beta) L_{\mathbf{z}}(\cdot, \beta), D_{k+1|k}(\cdot, \beta)\rangle}. \tag{5.9}$$

Assuming that new targets are always detected (i.e. $p_D(\mathbf{x}, 0) = 1$) and that the measurement does not depend on β gives:

$$D_{k+1|k+1}(\mathbf{x}, 0) = (1 - p_D(\mathbf{x}))D_{k+1|k}(\mathbf{x}, 0) + \sum_{\mathbf{z}\in Z_k} \frac{p_D(\mathbf{x})L_{\mathbf{z}}(\mathbf{x})D_{k+1|k}(\mathbf{x}, 0)}{\lambda\kappa(\mathbf{z}) + \langle L_{\mathbf{z}}, \gamma_{k+1|k}\rangle + \langle p_D L_{\mathbf{z}}, D_{k+1|k}(\cdot, 0)\rangle}, \tag{5.10}$$

$$D_{k+1|k+1}(\mathbf{x}, 1) = \sum_{\mathbf{z}\in Z_k} \frac{L_{\mathbf{z}}(\mathbf{x})\gamma_{k+1|k}(\mathbf{x})}{\lambda\kappa(\mathbf{z}) + \langle L_{\mathbf{z}}, \gamma_{k+1|k}\rangle + \langle p_D L_{\mathbf{z}}, D_{k+1|k}(\cdot, 0)\rangle}. \tag{5.11}$$

The importance sampling for the persistent targets may be carried out in the same way as in (4.64-4.65) and (4.68) taking care to note that the denominator in the measurement terms has a slightly different form in (5.10). Thus, given $\{\mathbf{x}_k^i, w_k^i\}_{i=1}^{N_k}$

$$\mathbf{x}_{k+1|k}^i \sim f_{k+1|k}(\cdot|\mathbf{x}_k^i), \tag{5.12}$$

$$w_{k+1|k}^i = p_S(\mathbf{x}_k^i)w_k^i, \tag{5.13}$$

and the weights are updated by

$$w_{k+1}^i = \left[1 - p_D(\mathbf{x}_{k+1|k}^i) + \sum_{\mathbf{z}\in Z_{k+1}} \frac{p_D(\mathbf{x}_{k+1|k}^i)L_z(\mathbf{x}_{k+1|k}^i)}{\mathcal{L}(\mathbf{z})}\right] w_{k+1|k}^i. \tag{5.14}$$

Here, $\mathcal{L}(\mathbf{z})$ is the denominator of (5.11).

The form of (5.11) permits a different strategy for newborn targets with a proposal density dependent on the measurements. For each $\mathbf{z} \in Z_{k+1}$, generate η random samples from the probability density

$$\mathbf{x}_{\mathbf{z},k+1|k}^i \sim \frac{L_{\mathbf{z}}(\cdot)\gamma_{k+1|k}(\cdot)}{\nu_{\mathbf{z},k+1|k}}, \tag{5.15}$$

$$w_{\mathbf{z},k+1}^i = \frac{\nu_{\mathbf{z},k+1|k}}{\eta}, \tag{5.16}$$

where $\nu_{\mathbf{z},k+1|k}$ is the normalising constant $\langle L_{\mathbf{z}}(\cdot)\gamma_{k+1|k}(\cdot)\rangle$ which corresponds to the expected number of newborn targets due to the measurement $\mathbf{z}$. The weights of the newborn particles after the update step are then

$$w_{\mathbf{z},k+1}^i = \frac{w_{\mathbf{z},k+1|k}^i}{\mathcal{L}(\mathbf{z})}. \tag{5.17}$$

Note that the importance sampling scheme has been adjusted slightly from the one presented in [96] to allow for different weightings of particles due to different measurements as may be required if the birth density is non-constant. In [96], an average is taken giving an average number of expected targets per measurement in place of

$\nu_{\mathbf{z},k+1|k}$. Full details of the improved SMC-PHD filter including state extraction and track continuity are given in Table 5.4.

State Extraction

State extraction will only be performed for the persisting targets; this should reduce the number of false alarms. Writing $Z_{k+1} = \{\mathbf{z}_{1,k+1}, ..., \mathbf{z}_{m_{k+1},k+1}\}$ and considering the form of (5.14) it can be seen that

$$w^i_{k+1} = 1 - p_D(\mathbf{x}^i_{k+1|k}) w^i_{k+1|k} + \sum_{j=1}^{m_{k+1}} w^{i,j}_{k+1}, \tag{5.18}$$

where

$$w^{i,j}_{k+1} = \left[\frac{p_D(\mathbf{x}^i_{k+1|k}) L_z(\mathbf{x}^i_{k+1|k})}{\mathcal{L}(\mathbf{z})} \right] w^i_{k+1|k}. \tag{5.19}$$

This can be interpreted as each particle having a component due to each measurement. These components are used to extract a state due to each measurement as follows:

$$W^j_{k+1} = \sum_i w^{i,j}_{k+1}, \tag{5.20}$$

$$\mathbf{X}^j_{k+1} = \sum_i w^{i,j}_{k+1} \mathbf{x}^i_{k+1|k} / W^j_{k+1}, \tag{5.21}$$

$$\hat{P}^j_{k+1} = \sum_i w^{i,j}_{k+1} (\mathbf{x}^i_{k+1|k} - \mathbf{X}^j_{k+1})(\mathbf{x}^i_{k+1|k} - \mathbf{X}^j_{k+1})^T / W^j_{k+1}. \tag{5.22}$$

Hence, the form of the PHD update equation (4.61) has been exploited to identify components of the PHD due to individual measurements within the SMC-PHD filter and this has removed the need for particle clustering to perform state extraction. A disadvantage is that, as no clustering is performed, labelling of particles is not possible and hence track continuity in the sense introduced in [25] is not possible. Furthermore, the particles due to missed detections in (5.19) are not used for state extraction and hence it will not be possible to extract states from targets with missed detections. These issues will be resolved using a stratified resampling technique.

Stratified Resampling For Track Continuity

In stratified resampling, a separate resampling is conducted for each measurement (and additionally for the "missed detection" case). Define $w^{i,0}_{k+1} = (1 - p_D(\mathbf{x}^i_{k+1|k})) w^i_{k+1|k}$, the weights corresponding the missed detections in (5.19) and also define

$$W^0_{k+1} = \sum_i w^{i,0}_{k+1}. \tag{5.23}$$

Given ρ, the number of particles per target in the SMC-PHD filter implementation, the number of particles corresponding to each measurement (and the missed detection case) is thus

$$N_{j,k+1} = int(W_{k+1}^{j} \cdot \rho) \quad \text{for} \quad j = 0, ..., m_{k+1}. \tag{5.24}$$

Finally, $N_{j,k+1}$ particles are resampled from the set $\{\mathbf{x}_{k+1|k}^{n}, w_{k+1}^{n,j}\}$, for $j = 0, ..., m_{k+1}$, to give $\{\mathbf{x}_{k+1}^{i}, w_{k+1}^{i}\}_{i=1}^{N_{k+1}}$ where $N_{k+1} = \sum_{j=0}^{m_{k+1}} N_{j,k+1}$. The particles due to each resampling set are now considered as individual clusters. In order to enable track continuity each persisting particle carries a track I.D. label which is maintained through the prediction and update steps. In addition to the track I.D. label, each particle is also now assigned a label due to its cluster from the resampling step.

Track Association - Next, the clusters are either associated with existing tracks or used to begin new tracks. Finally, all of the particles in the cluster will be given the track label I.D. for the appropriate track. Let,

$$\begin{aligned}
C(i) \quad & i = 1, ..., N_{k+1} \quad && -\text{Cluster label for each particle} \\
L(i) \quad & i = 1, ..., N_{k+1} \quad && -\text{Track id label for each particle} \\
M(j) \quad & j = 1, ..., m_{k+1} \quad && -\text{Labels for each cluster} \\
T(\ell) \quad & \ell = 1, ..., T_k \quad && -\text{Existing track I.D.s}
\end{aligned}$$

where T_k is the number of track I.D.s from the previous time step. Define:

$$A_{j,\ell} = \sum_{i=1}^{N} \delta_{C(i),M(j)} \delta_{L(i),T(\ell)} \quad \text{for} \quad j = 1, ..., m_{k+1}, \quad l = 1, ..., T_k \tag{5.25}$$

where $\delta_{p,q}$ is the Kronecker delta. Therefore $A_{j,\ell}$ represents the number of particles with cluster label $M(j)$ and track id label $T(\ell)$.
Also, let

$$B_j = \sum_{i=1}^{N} \delta_{C(i),M(j)} \delta_{L(i),0} \quad \text{for} \quad j = 1, ..., m_{k+1} \tag{5.26}$$

so that B_j represents the number of particles with cluster label $M(j)$ and no track id label.

The association between clusters and track I.D.s is performed by maximising the number of particles which keep their track labels, while recognising the benefit of beginning a new track if a cluster mostly contains particles with no track I.D. label. An optimisation is performed where the "reward" for associating cluster label $M(j)$ with

track label $T(\ell)$ is equal to $A_{j,\ell}$. The reward for starting a new track is proportional to B_j.

To understand better the choice of constant of proportionality, denoted θ, associated with the reward for starting a new track. Consider a case with a cluster comprised of 499 particles with track id '1' and 500 particles with no track I.D.. In most cases, it would be considered preferable for this cluster to be associated with track I.D. '1' rather than starting a new track so θ should usually be chosen as less than 1.

The optimal set of associations between clusters and tracks, which maximises the rewards from maintaining particle track I.D.s and starting new tracks, is computed by performing an auction-type algorithm (see [12]) on the matrix:

$$C = [A, \theta \mathbb{I}_B], \tag{5.27}$$

where $\mathbb{I}_B$ is the identity matrix with the vector B for a diagonal and A is the matrix computed in (5.25) .

After the assignment has taken place, state extraction may be calculated by

$$\hat{\mathbf{x}}_{k+1,J} = \sum_{\mathbf{x}_{k+1}^i:\text{track id}=J} w_{k+1}^i \mathbf{x}_{k+1}^i, \tag{5.28}$$

$$P_{k+1,J} = \sum_{\mathbf{x}_{k+1}^i:\text{track id}=J} w_{k+1}^i (\mathbf{x}_{k+1}^i - \hat{\mathbf{x}}_{k+1,J})(\mathbf{x}_{k+1}^i - \hat{\mathbf{x}}_{k+1,J})^T, \tag{5.29}$$

$$W_{k+1,J} = \sum_{\mathbf{x}_{k+1}^i:\text{track id}=J} w_{k+1}^i, \tag{5.30}$$

for each J, belonging to the set of track labels present.

Remark: Under this formulation of track continuity, particles which are in the "no measurement" cluster have their track id labels unchanged. This allows for state extraction of targets for which no measurement is received, which is not possible in the state extraction method outlined in [96].

Gating

The bulk of the computation for the SMC-PHD filter is due to (5.14) which has complexity $N_k \times m_k$. The value of $L_{\mathbf{z}}(\mathbf{x}_{k+1|k}^i)$ is likely to be very small for most particle/measurement pairs and so it would make sense to only compute those values not too small to contribute significantly to the weights. This can be achieved by gating particle/measurement pairs using track I.D. labels. At the beginning of the update step, the means and covariances for each track can be estimated using (5.28-5.29) and

the forward model. For each measurement, $\mathbf{z}_j$ and each track, $\hat{\mathbf{x}}^l_{k|k-1}$, the statistical distance, $d_{j,l}$ can be computed by:

$$d_{j,l} = (pr(\mathbf{z}_j) - \hat{\mathbf{x}}^l_{k|k-1})P^{-1}_{k,j}(pr(\mathbf{z}_j) - \hat{\mathbf{x}}^l_{k|k-1})^T \tag{5.31}$$

where $pr(\mathbf{z}_i)$ is the projection of $\mathbf{z}_i$ onto the single target state space. If $d_{j,l}$ is above some suitably high threshold then it may be assumed that all particles with the corresponding track label are sufficiently far from the $\mathbf{z}_i$ that no significant contribution to the particle weight will come from that measurement. Thus, the contribution to (5.14) is not calculated but set to 0.

Given $\{\mathbf{x}^i_k, w^i_k\}_{i=1}^{N_k}$

Prediction/Update

for $i = 1, ..., N_k$

- *Sample* $\mathbf{x}^{b,i}_{k+1|k} \sim f_{k+1|k}(\cdot, \mathbf{x}^i_k)$
- *Compute* $w^{b,I}_{k+1|k} = p_S(\mathbf{x}^i_k)w^i_k$
- *Compute* $w^{b,i}_{k+1}$ according to (5.14)

end

for $z \in Z_{k+1}$

for $i = 1, ..., \eta$

- *Sample* $\mathbf{x}^i_{z,k+1|k}$ according to (5.15)
- *Compute* $w^i_{z,k+1}$ according to (5.16)

end

•Combine $\{\hat{\mathbf{x}}^i_{k+1}, \hat{w}^i_{k+1}\}_i = \{\mathbf{x}^{b,i}_{k+1}, w^{b,i}_{k+1}\}_i \cup \bigcup_{\mathbf{z}\in Z_{k+1}} \{\mathbf{x}^i_{\mathbf{z},k+1|k}, w^i_{\mathbf{z},k+1}\}_i$

end

State Extraction/Track Continuity

for $j = 0, ..., m_{k+1}$

- •*Compute* $w^{i,j}_{k+1}$ according to (5.19) and (5.23)
- •*Compute*$N_{j,k+1} = int(W_{k+1}\rho)$
- *Resample* $\{\mathbf{x}^i_{k|k+1}, w^{i,j}_{k+1}\}$ to give $\{\mathbf{x}^i_{j,k+1}, w^i_{j,k+1}\}$ and give each resampled particle the cluster label j

end

• Find optimal association between clusters and tracks using (5.25-5.27).

for all existing track i.d.s, J

- •*Compute* $\hat{\mathbf{x}}_{k,J}, \hat{W}_{k,J}, \hat{P}_{k,J}$ using (5.28-5.30)

end

Table 5.4: Pseudocode for one iteration of the improved SMC-PHD filter.

5.1.4 SMC-PHD Performance Assessment with Simulated Data

Simple Test of the Effectiveness of the Improved Clustering Step

A simple test was carried out to determine if the use of improved clustering alleviates the state estimation problem in the SMC-PHD filter. The improved SMC-PHD filter was used for comparison and a linear, Gaussian scenario was selected so that the GM-PHD filter is the closed form solution to the PHD recursion, providing a further point of comparison.

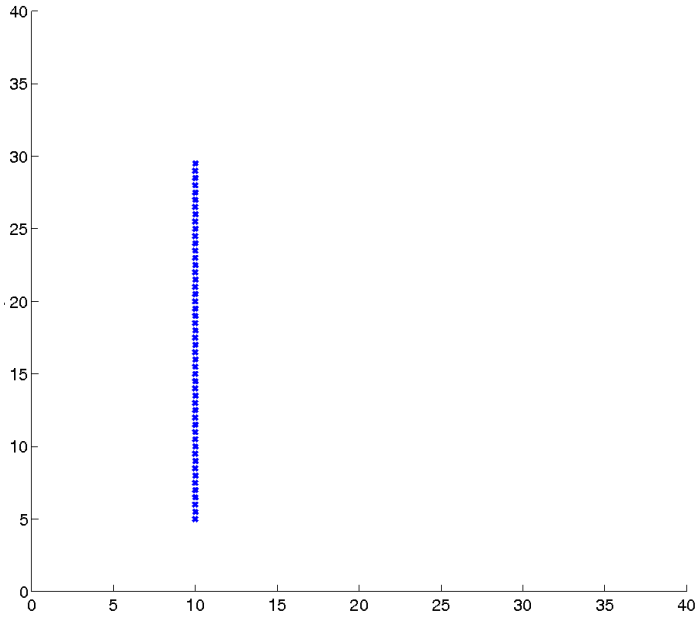


Figure 5.2: Simple scenario with one target.

The tracking scenario is shown in fig. 5.2. Measurements of (x, y) position are received and each measurement component is corrupted by zero-mean, white Gaussian noise with standard deviation $\sigma = 0.3$. The measurement dimension is two so (5.3) reduces to $p = 1 - e^{\frac{\gamma}{2}}$ and so γ can be calculated for a given value of p as $\gamma = -2\log(1-p)$. For this test, $p = 0.99$ which gives $\gamma = 9.21$. The birth model used in all of the PHD filter forward models is a uniformly distributed Poisson birth process with mean 0.1. The clutter process is also uniformly distributed and Poisson with mean λ.

100 Monte Carlo tests were run for a variety of choices of the clutter level, λ. The criterion for success was a "track maintenance" measure of effectiveness (see Section 5.4 for further discussion of measures of effectiveness). A track was automatically

initialised and was considered to have been maintained if a track with weight > 0.2 was declared within one unit of the true target location for at least $\frac{2}{3}$ of the time steps in the test. Both SMC filters used 1000 particles per target. For birth targets, the improved SMC-PHD filter used 100 particles per target whereas the other SMC-PHD filter implementations used 50000 birth particles. The results are presented in Table 5.5.

clutter	SMC-PHD	impr. clustering	impr. SMC-PHD	GM-PHD
10	0	1	1	1
20	0	1	1	1
50	0	1	1	1
100	0	1	1	1
200	0	1	1	1
500	0	0.97	0.97	0.99

Table 5.5: The proportion of successful tracks maintained from 100 Monte Carlo tests using standard SMC-PHD, SMC-PHD with improved clustering, improved SMC-PHD and GM-PHD filters.

The results show that improved clustering fixes the state estimation problem associated with clustering within the SMC-PHD filter, giving significantly improved results despite the fact that the underlying SMC-PHD filter recursion is unchanged. The results for the SMC-PHD filter are brought into line with the results for a similar scenario using the GM-PHD filter as presented in Section 6.4. However, the improved SMC-PHD filter performs just as well but generally requires far fewer particles for the birth process as well as not requiring any computation for the clustering step. Hence, although the improved clustering solves the state estimation problem, it is superseded by the improved SMC-PHD filter which also solves the target birth problem.

Effect of Gating in the improved SMC-PHD filter

The negligible effect of gating on results is illustrated using a simple test. In this scenario, illustrated in fig. 5.2, there is one target observed by a sensor located at the origin which measures range and bearing with measurement noise standard deviation 0.1 and 1^o respectively. Clutter is uniformly distributed over the surveillance region with the number of false detections being Poisson distributed with mean λ. The probability of detection is constant, $P_D = 0.95$. The forward transition $\phi(\mathbf{x}|\mathbf{x}')$ is based on constant velocity with a small amount of process noise. 1000 particles per target are used, with 100 particles for each newborn target. 1000 Monte Carlo tests were run.

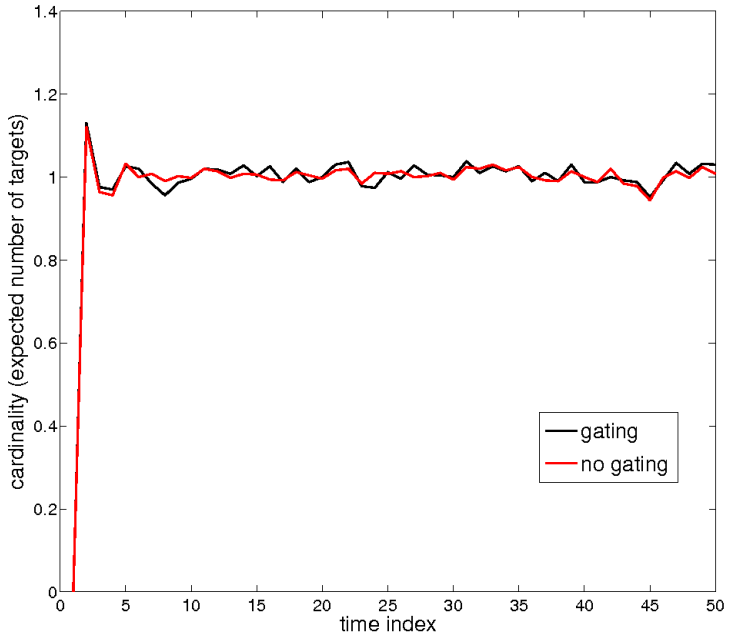


Figure 5.3: Cardinality with and without gating for $\lambda = 10$.

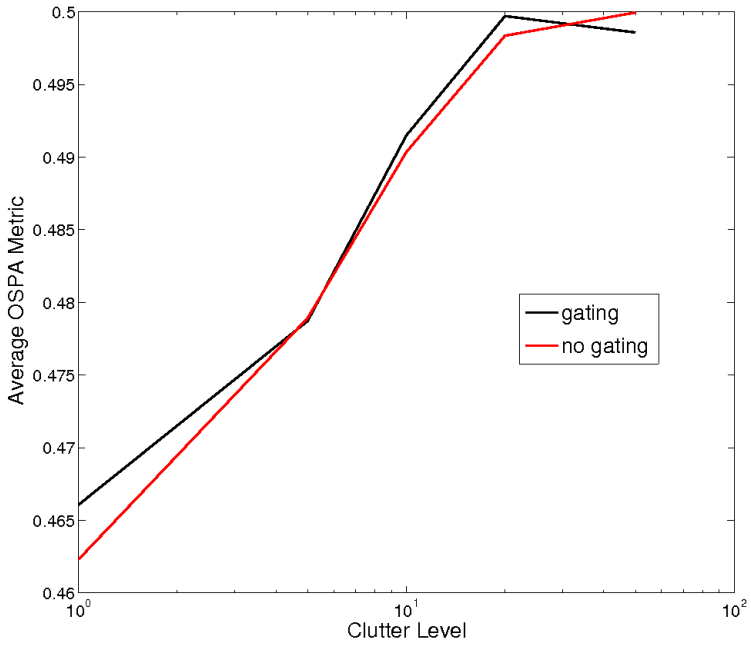


Figure 5.4: OSPA metric with and without gating for a range of λ.

Fig. 5.3 shows the number of targets declared at each time step, averaged over the Monte Carlo trials, illustrating the equivalence of performance with or without gating. More generally, we consider the optimal sub-pattern assignment (OSPA) metric [107], as discussed in Section 5.4. This metric is chosen because it captures the error in both localisation and cardinality. The OSPA parameter c is set as 1 throughout.

In fig. 5.4, the OSPA metric averaged over the set of Monte Carlo trials and the length of the each run is shown for a range of λ. Similarity of performance is again apparent, with the non-gating case having an average OSPA metric only 0.2% lower than with gating.

Fig. 5.5 shows the computing time taken both with and without gating for a range of clutter levels. This demonstrates that using gating offers a significant speed-up without a corresponding loss of performance.

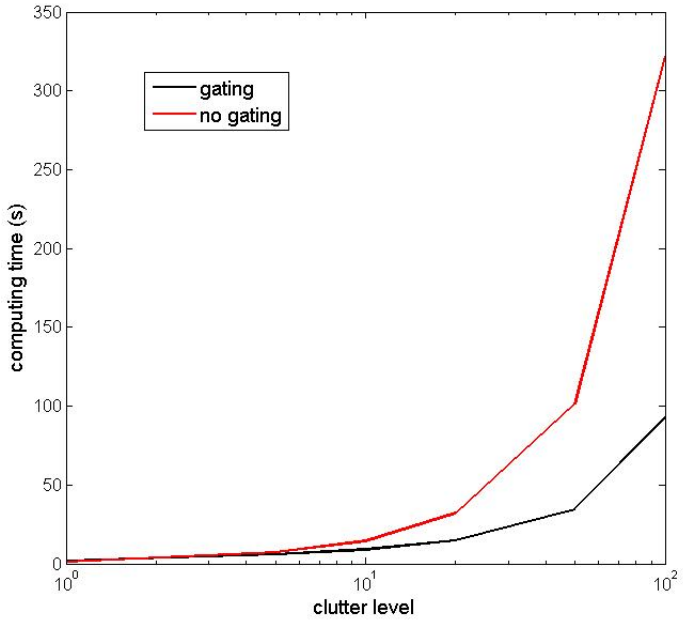


Figure 5.5: Computing times with and without gating for a range of λ.

Performance Comparison of the improved SMC-PHD and EKF-GM-PHD filters

Prior to the development of the improved SMC-PHD filter, the GM-PHD filter and its nonlinear approximations were considered the most plausible methods for multi-target tracking. Both the improved SMC-PHD and EKF-GM-PHD filters were tested on the scenario shown in fig. 5.6, where 8 targets appear at various times. All other

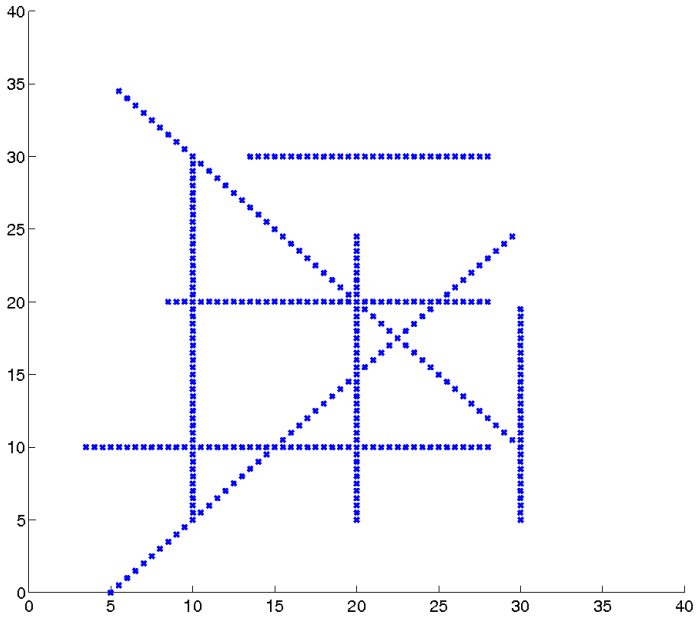


Figure 5.6: Scenario with 8 targets

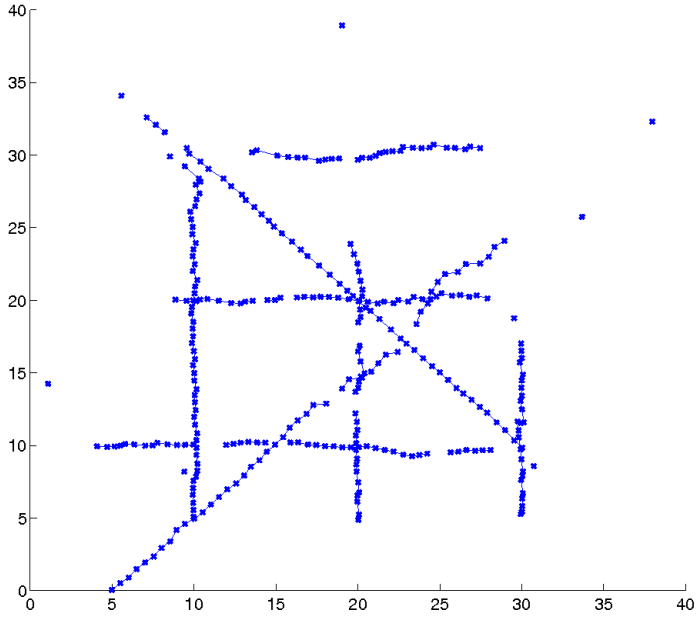


Figure 5.7: Example tracker output with $\lambda = 10$.

scenario parameters were the same as those described in the previous test for gating. The success of the improved SMC-PHD filter with track labelling is demonstrated in one incidence with $\lambda = 10$ in fig. 5.7. The filter output shown in the figure includes both false alarm track declarations and missing declarations of tracks due to missed detections, however it is broadly clear that the targets have been tracked and that track continuity has been maintained.

Fig. 5.8 shows the average number of targets declared at each time step for $\lambda = 10$ for both the SMC-PHD and EKF-GM-PHD filters. It can be seen that the SMC-PHD filter is able to initialise new tracks more quickly which gives the SMC-PHD method an average cardinality error 72% lower than the EKF-GM-PHD filter. On the other hand, fig. 5.9 shows that the OSPA metric with $c = 1$ for $\lambda = 10$ demonstrating that the OSPA distance is clearly lower for the EKF-GM-PHD filter apart from recently after appearance of new targets. This suggests that the localisation for the EKF-GM-PHD filter is superior. Over a set of tests for various clutter levels, the OSPA metric was found to be 9-11% lower for the EKF-GM-PHD filter for $\lambda = 0, 5, 10, 20, 50$. This shows that the EKF-GM-PHD filter performs slightly better in this sense but that performance of the two is comparable.

Note that the bearing error standard deviation for all of the tests here was 1^o. The results from Section 2.2 suggest that for bearing error any higher than this, the EKF approximation will break down so that the improved SMC-PHD filter would be expected to perform better in comparison.

The larger error in localisation appears to be due to restriction on the number of particles per target in the SMC-PHD filter. For a smaller test with 100 Monte Carlo trials, 10000 particles per target (as opposed to the usual 1000) were used with $\lambda = 10$. In this case, the OSPA metric was only 0.5% lower for the EKF-GM-PHD filter. This demonstrates that increasing the number of particles per target effectively eliminates the difference in performance between the two methods. However, this increases the computational cost of the SMC-PHD filter ten-fold, so that the computational cost increases from 40% higher than the EKF-GM-PHD filter to 14 times higher. This demonstrates the well-known computational disadvantage of particle methods.

Track continuity of the two methods was compared by calculating the average length of a "correct" track. This was 18-21% higher for the EKF-GM-PHD filter for each of the clutter levels $\lambda = 0, 5, 10, 20, 50$ demonstrating that track continuity is comparable but slightly worse for the SMC-PHD filter than the established GM-PHD track continuity method as presented in [87].

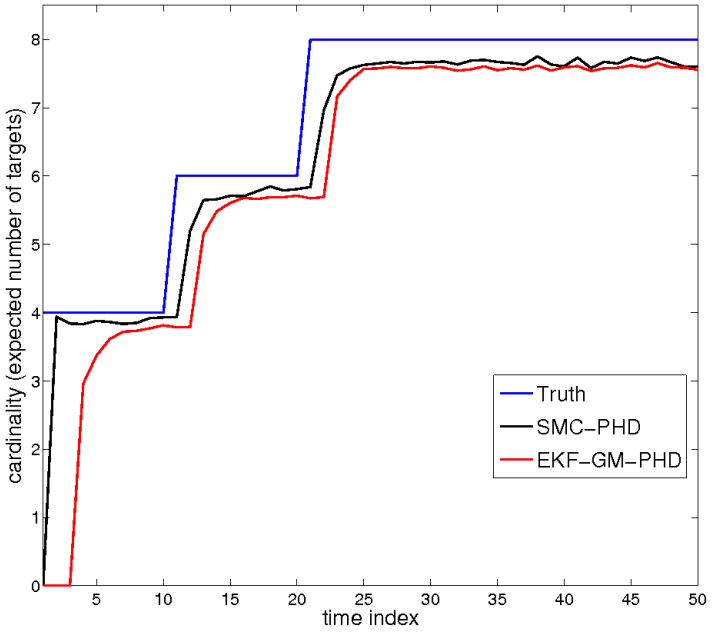


Figure 5.8: Average cardinality for SMC-PHD and EKF- GM-PHD filters for $\lambda = 10$.

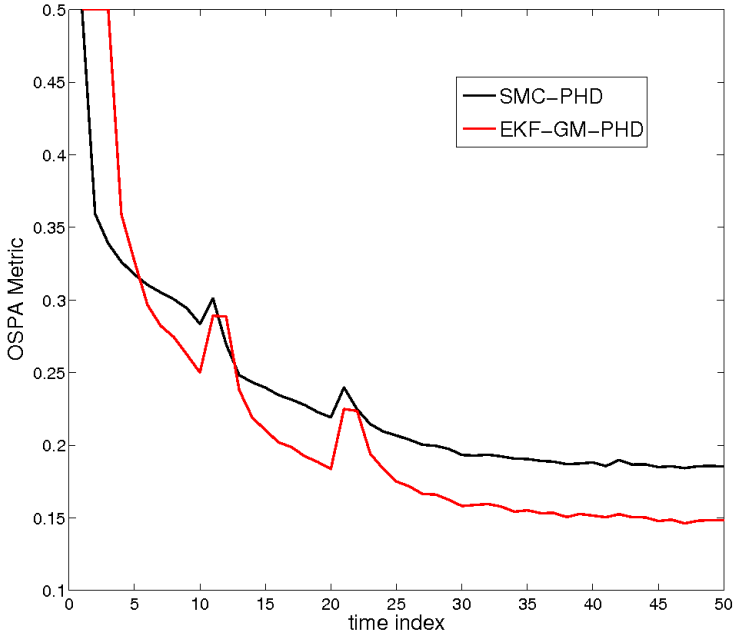


Figure 5.9: Average OSPA metric with $\lambda = 10$.

5.2 Manoeuvre Handling

Section 2.4 considered the problem of handling target manoeuvres in the single target case using jump Markov linear systems (JMLS). Methods for applying the PHD filter to tracking scenarios with JMLS structure were first proposed in [123] and [93] for the GM-PHD and SMC-PHD filters respectively. The former paper leaves an open question about how to incorporate "interacting" multiple models.

As an explanation for the interest in interacting multiple models for the PHD filter it should be noted that the interacting multiple model (IMM) filter is the de facto algorithm of choice for dealing with manoeuvring targets in a single target context. In the tracking literature it is deemed to be "cost effective" [78] and "very efficient" [15] which, practically speaking, means that there is a good trade off between accuracy and efficiency. The prevalence of IMM in single target tracking naturally leads to questions about the relationship between the IMM methodology and the PHD filter for multiple target tracking.

This section will address the open question, deriving the PHD filter with JMLS manoeuvre handling (JMLS-PHD) from the point of view of interacting multiple models without any need for assumptions on target dynamics. The filter derived is shown to be equivalent to the existing implementation described in [90] in the linear Gaussian case.

The resulting JMLS-PHD filter is then shown to reduce to the optimal Bayes solution for jointly estimating target state and motion mode in the single target case with linear and Gaussian motion and measurement models. This work is currently in press [131].

5.2.1 JMLS-PHD Filter

Assume that there are N motion modes, denoted r_i for $i = 1, ..., N$, and that transition between these modes is a Markov process with transition probabilities given by

$$p(r_k = i | r_{k-1} = j) = \pi_k^{i,j}, \tag{5.32}$$

where $\pi_k^{i,j}$ is known.

The method for interacting the motion modes is proposed by Punithakumar et al. [93] in the context of the SMC-PHD filter. In the IMM filter, two moments (mean and covariance) of the single target state space are being propagated; it is these two moments that are mixed by weighted combinations. Similarly, in the PHD filter,

where the first moment of the evolving multitarget density is propagated, so it is these first moments that are mixed by weighted combination.

The JMLS-PHD method is to compute a separate PHD for each motion mode. These are mixed at the beginning of the time step, and predicted and updated according to the relevant mode-dependent PHD equations. Thus, if the initial PHDs at time step $k-1$ are $D^r_{k-1|k-1}$ then the mixed PHDs, $\hat{D}^r_{k|k-1}$, are:

$$\hat{D}^r_{k|k-1}(\mathbf{x}) = \sum_{q=1}^{N} D^q_{k-1|k-1}(\mathbf{x})\pi^{r,q}_k. \tag{5.33}$$

After the mixing step, the prediction and update steps for the PHDs proceed as standard mode-dependent PHD prediction and update equations [73]. This is analogous to the standard Kalman filter prediction and update for each mode in the IMM filter.

Thus, the predicted PHD for mode r, $D^r_{k|k-1}$, as given in (4.60) is:

$$D^r_{k|k-1}(\mathbf{x}) = \gamma^r_k(\mathbf{x}) + \int (p^r_S(\mathbf{w}) f^r_{k|k-1}(\mathbf{x}|\mathbf{w}) + b^r_{k|k-1}(\mathbf{x}|\mathbf{w}))\hat{D}^r_{k-1|k-1}(\mathbf{w})d\mathbf{w} \tag{5.34}$$

where $f^r_{k|k-1}$ is the single target Markov transition density, $p^r_S(\mathbf{x})$ is the probability that a target with state $\mathbf{x}$ at time step $k-1$ will survive to time step k, $b_{k|k-1}$ describes the spawning of new targets from existing ones and γ_k describes the appearance of new targets. Note that all of these processes are potentially mode-dependent.

Similarly, the updated PHD for mode r is given by:

$$D^r_{k|k}(\mathbf{x}) = F^r_k(Z_k|\mathbf{x})D^r_{k|k-1}(\mathbf{x}), \tag{5.35}$$

where F^r_k is the mode-dependent multitarget likelihood given by

$$F^r_k(Z_k|\mathbf{x}) = (1 - p^r_D(\mathbf{x})) + \sum_{\mathbf{z}\in Z_k} \frac{p^r_D(\mathbf{x})L^r_{\mathbf{z}}(\mathbf{x})}{\lambda\kappa(\mathbf{z}) + D^r_{k|k-1}[P^r_D L^r_{\mathbf{z}}]}, \tag{5.36}$$

and where $D^r_{k|k-1}[h] = \int h(\mathbf{x})D^r_{k|k-1}(\mathbf{x})d\mathbf{x}$. Also, $p^r_D(\mathbf{x})$ is the probability of detecting a target with state $\mathbf{x}$ and $L^r_{\mathbf{z}}(\mathbf{x})$ is the mode-dependent single target likelihood function and $\lambda\kappa(\mathbf{z})$ is the clutter density. This is a mode-dependent equivalent of the multitarget likelihood given in (4.61).

The "total" PHD may be computed as $D_{k|k} = \sum_r D^r_{k|k}$. Unlike in IMM, no weighting in this sum is necessary as PHDs are inherently weighted.

Relationship to the GM-PHD filter for JMLS of Pasha et al. (2009)

The most complete description of the Gaussian mixture PHD filter for jump Markov

linear systems is given in Pasha et al. [90]. Prediction and update equations for the JMLS-GM-PHD filter are derived in detail using linear, Gaussian assumptions on the birth process, target dynamics and measurement process as well as the assumption that individual targets follow a JMLS motion model.

Here, it is shown that the PHD prediction and update equations derived in [90] are the linear Gaussian special case of the JMLS-PHD filter as given in (5.33)-(5.35). The JMLS-GM-PHD filter of [90] will be denoted $\tilde{D}_{k|k}$ to distinguish it from the JMLS-PHD filter defined in (5.33)-(5.35) until it has been shown that these are equivalent.

Proposition 5.1(a): *The predicted PHD computed in [90] is equivalent to $D^r_{k|k-1}$ as computed in (5.33)-(5.34).*

Proof: The predicted PHD computed in [90] is given by

$$\tilde{D}^r_{k|k-1}(\mathbf{x}) = \gamma^r_k(\mathbf{x}) + D^r_{f,k|k-1}(\mathbf{x}) + D^r_{\beta,k|k-1}.(\mathbf{x}), \tag{5.37}$$

The birth density, γ^r_k, is assumed given and

$$D^r_{f,k|k-1}(\mathbf{x}) = \sum_q \sum_{i=1}^{J_{k-1}(q)} w^{(i)}_{f,k|k-1}(r,q)\mathcal{N}(\mathbf{x};\Theta^{(i)}_{f,k|k-1}(r,q)), \tag{5.38}$$

$$w^{(i)}_{f,k|k-1}(r,q) = p^q_{S,k}\pi^{r,q}_k w^{(i)}_{k-1}(q), \tag{5.39}$$

$$D^r_{\beta,k|k-1}(\mathbf{x}) = \sum_q \sum_{i=1}^{J_{k-1}(q)} \sum_{j=1}^{J_{\beta,k}(r,q)} w^{(i,j)}_{\beta,k|k-1}(r,q)\mathcal{N}(\mathbf{x};\Theta^{(i,j)}_{\beta,k|k-1}(r,q)), \tag{5.40}$$

$$w^{(i,j)}_{\beta,k|k-1}(r,q) = \pi^{r,q}_k w^{(j)}_{\beta,k|k-1}(r,q) w^{(i)}_{k-1}(q), \tag{5.41}$$

where $\Theta^{(i)}_{f,k|k-1}(r,q)$, $\Theta^{(i,j)}_{\beta,k|k-1}(r,q)$ contain the means and covariances computed due to the mode-dependent PHD prediction equation, $w^{(i)}_{k-1}$ and $w^{(j)}_{\beta,k|k-1}$ are assumed known from the previous time step and spawning process respectively. Similarly, $J_{k-1}(q)$ and $J_{\beta,k}(r,q)$ are the (known) number of Gaussian components from the previous time step and due to spawned targets respectively.

In [122] it is shown that, in the linear Gaussian case

$$\int p_{S,k}(\mathbf{w}) f_{k|k-1}(\mathbf{x}|\mathbf{w}) D_{k-1|k-1}(\mathbf{w}) d\mathbf{w} = p_{S,k} \sum_{i=1}^{J_{k-1}} w^{(i)}_{k-1}\mathcal{N}(\mathbf{x};\Theta^{(i)}_{f,k|k-1}), \tag{5.42}$$

$$\int b_{k|k-1}(\mathbf{x}|\mathbf{w}) D_{k-1|k-1}(\mathbf{w}) d\mathbf{w} = \sum_{i=1}^{J_{k-1}} \sum_{j=1}^{J_{\beta,k}} w^{(i)}_{k-1} w^{(i,j)}_{\beta,k|k-1}\mathcal{N}(\mathbf{x};\Theta^{(i)}_{\beta,k|k-1}). \tag{5.43}$$

Hence it can be seen that (5.37) may be written as:

$$\tilde{D}^r_{k|k-1}(\mathbf{x}) = \gamma^r_k(\mathbf{x}) \quad + \quad \sum_q \pi^{r,q}_{k|k-1} \int p_S(\mathbf{w}) f_{k|k-1}(\mathbf{x}|\mathbf{w}) D^q_{k-1|k-1}(\mathbf{w}) d\mathbf{w}$$
$$+ \quad \sum_q \pi^{r,q}_{k|k-1} \int b_{k|k-1}(\mathbf{x}|\mathbf{w}) D^q_{k-1|k-1}(\mathbf{w}) d\mathbf{w}. \tag{5.44}$$

By changing the order of summation and integration, it is clear that (5.44) is equivalent to (5.33) and (5.34). □

Proposition 5.1(b): *The PHD update computed in [90] is equivalent to $D^r_{k|k}$ as computed in (5.35).*
Proof: The update step in [90] is calculated directly as the mode-dependent PHD update for each of the mode-dependent PHDs computed in the prediction step. This is the same as (5.35) so there is nothing to prove. □

Thus, the JMLS-PHD filter which is derived using only an IMM-style mixing step and is valid for general PHDs is shown to reduce to the carefully derived JMLS-GM-PHD filter of [90] in the linear Gaussian case.

5.2.2 Bayes Optimality in the Single Target Case

In the single target case, as discussed in Section 2.4 there is a set of measurements corresponding to a single target. The joint distribution for target state and motion mode, conditioned on the measurements received, must be computed. Letting $\mathbf{x}_{0:k}$ be the set of target states up to time k and similarly letting $r_{1:k}$ and $\mathbf{y}_{1:k}$ be the set of motion modes and measurements up to time k respectively, the full probability density that must be computed is $f_k(\mathbf{x}_{0:k}, r_{1:k}|\mathbf{y}_{1:k})$. This subsection begins with some preliminary probability results obtained by simple application of conditional probability.

Lemma 5.2:

$$f_k(\mathbf{x}_{0:k}, r_{1:k}|\mathbf{y}_{1:k}) = f_k(\mathbf{x}_{0:k}|r_{1:k}, \mathbf{y}_{1:k}) f_k(r_{1:k}|\mathbf{y}_{1:k}) \tag{5.45}$$

Proof: Using conditional probability rule:

$$f(\mathbf{x}_{0:k}, r_{1:k}|\mathbf{y}_{1:k}) = \frac{f(\mathbf{x}_{0:k}, r_{1:k}, \mathbf{y}_{1:k})}{f(\mathbf{y}_{1:k})}, \tag{5.46}$$

$$\text{and} \qquad f(\mathbf{x}_{0:k}|r_{1:k}, \mathbf{y}_{1:k}) = \frac{f(\mathbf{x}_{0:k}, r_{1:k}, \mathbf{y}_{1:k})}{f(r_{1:k}, \mathbf{y}_{1:k})}. \tag{5.47}$$

Putting these together gives,

$$f(\mathbf{x}_{0:k}, r_{1:k}|\mathbf{y}_{1:k}) = \frac{f(\mathbf{x}_{0:k}|r_{1:k}, \mathbf{y}_{1:k})f(r_{1:k}, \mathbf{y}_{1:k})}{f(\mathbf{y}_{1:k})}. \quad (5.48)$$

Also by conditional probability:

$$f(r_{1:k}, \mathbf{y}_{1:k}) = f(r_{1:k}|\mathbf{y}_{1:k})f(\mathbf{y}_{1:k}), \quad (5.49)$$

and combining (5.48) and (5.49) gives

$$f(\mathbf{x}_{0:k}, r_{1:k}|\mathbf{y}_{1:k}) = f(\mathbf{x}_{0:k}|r_{1:k}, \mathbf{y}_{1:k})f(r_{1:k}|\mathbf{y}_{1:k}). \quad \square \quad (5.50)$$

Note that $f_k(\mathbf{x}_{0:k}|r_{1:k}, \mathbf{y}_{1:k})$ may be computed using a standard Kalman filter and hence only $f_k(r_{1:k}|\mathbf{y}_{1:k})$ remains to be calculated.

It can be seen from (5.33)-(5.34) that in the JMLS-PHD filter there will be one Gaussian component for each possible mode history. Therefore, it is only necessary to show that the JMLS-PHD filter calculated weight for the Gaussian corresponding to a particular history is equal to $f_k(r_{1:k}|\mathbf{y}_{1:k})$ for that history.

Lemma 5.3: If it is assumed that

1. Measurements are conditionally independent of one another.
2. $f(r_k|r_{1:k-1}, \mathbf{y}_{1:k-1}) = f(r_k|r_{1:k-1})$ (i.e. all past dependence of r_k is taken into account by the mode history).

then:

$$f_k(r_{1:k}|\mathbf{y}_{1:k}) \propto f_k(\mathbf{y}_k|r_{1:k})f_k(r_k|r_{1:k-1})f_k(r_{1:k-1}|\mathbf{y}_{1:k-1}). \quad (5.51)$$

Proof: Using the conditional probability rule:

$$f(r_{1:k}|\mathbf{y}_{1:k}) = \frac{f(r_{1:k}, \mathbf{y}_{1:k})}{f(\mathbf{y}_{1:k})}, \quad (5.52)$$

$$\text{and} \qquad f(\mathbf{y}_k|r_{1:k}, \mathbf{y}_{1:k-1}) = \frac{f(r_{1:k}, \mathbf{y}_{1:k})}{f(r_{1:k}, \mathbf{y}_{1:k-1})}. \quad (5.53)$$

Putting these together gives,

$$f(r_{1:k}|\mathbf{y}_{1:k}) = \frac{f(\mathbf{y}_k|r_{1:k}, \mathbf{y}_{1:k-1})f(r_{1:k}, \mathbf{y}_{1:k-1})}{f(\mathbf{y}_{1:k})}, \quad (5.54)$$

and by assumption 1, this may be written as

$$f(r_{1:k}|\mathbf{y}_{1:k}) = \frac{f(\mathbf{y}_k|r_{1:k})f(r_{1:k}, \mathbf{y}_{1:k-1})}{f(\mathbf{y}_{1:k})}. \quad (5.55)$$

Also, using conditional probability again,

$$f(r_{1:k}, \mathbf{y}_{1:k-1}) = f(r_k, r_{1:k-1}, \mathbf{y}_{1:k-1}), \tag{5.56}$$

$$= f(r_k|r_{1:k-1}, \mathbf{y}_{1:k-1}) f(r_{1:k-1}, \mathbf{y}_{1:k-1}), \tag{5.57}$$

and by assumption 2 and another application of conditional probability

$$f(r_{1:k}, \mathbf{y}_{1:k-1}) = f(r_k|r_{1:k-1}) f(r_{1:k-1}, \mathbf{y}_{1:k-1}), \tag{5.58}$$

$$= f(r_k|r_{1:k-1}) f(r_{1:k-1}|\mathbf{y}_{1:k-1}) f(\mathbf{y}_{1:k-1}). \tag{5.59}$$

Combining (5.55) and (5.59):

$$f(r_{1:k}|\mathbf{y}_{1:k}) = \frac{f(\mathbf{y}_k|r_{1:k}) f(r_k|r_{1:k-1}) f(r_{1:k-1}|\mathbf{y}_{1:k-1})}{\frac{f(\mathbf{y}_{1:k})}{f(\mathbf{y}_{1:k-1})}}, \tag{5.60}$$

from which it follows that

$$f(r_{1:k}|\mathbf{y}_{1:k}) \propto f(\mathbf{y}_k|r_{1:k}) f(r_k|r_{1:k-1}) f(r_{1:k-1}|\mathbf{y}_{1:k-1}). \qquad \square \tag{5.61}$$

Proposition 5.4:
For a given motion mode history $r_{1:k}$

$$f_k(r_{1:k}|\boldsymbol{y}_{1:k}) = w^{r_{1:k}}$$

where $w^{r_{1:k}}$ is the weight of the Gaussian component corresponding to that mode history as calculated in the JMLS-PHD filter.
Proof: The mixture weights in the JMLS-PHD update equation (equation (44) in [90]) are:

$$w_k^{(i)}(r, \mathbf{y}_k) = \frac{p_{D,k}(r) w_{k|k-1}^{(i)}(r) q_k^{(i)}(r; \mathbf{y}_k)}{\lambda \kappa_k(\mathbf{y}_k) + \sum_r p_{D,k}(r) \sum_{i=1}^{J_{k|k-1}(r)} w_{k|k-1}^{(i)}(r) q_k^{(i)}(r; \mathbf{y}_k)}, \tag{5.62}$$

where $p_{D,k}$ is the probability of detection, $\lambda\kappa_k$ is the distribution of clutter, $q_k^{(i)}(r; \mathbf{y}_k)$ is the likelihood function for receiving $\mathbf{y}_k$ given the motion mode r and $w_{k|k-1}^{(i)}$ are the predicted weights of the Gaussian mixture components from the previous time step.

Now, for the single target case $p_{D,k} = 1$, $\kappa_k = 0$ and since there is one mixture component for each possible mode history, we may write $w_k^{(i)}(r)$ as $w^{r_{1:k}}$. Thus, (5.62) becomes:

$$w^{r_{1:k}} = \frac{w_{k|k-1}^{r_{1:k-1}}(r_k) q_k^{r_{1:k}}(r_k; \mathbf{y}_k)}{\sum_{s_{1:k}} w_{k|k-1}^{s_{1:k-1}}(s_k) q_k^{s_{1:k}}(s_k; \mathbf{y}_k)}, \tag{5.63}$$

where $\sum_{s_{1:k}}$ is the sum over all possible mode histories, $s_{1:k}$, and is a normalising factor.

Therefore, $w^{r_{1:k}} \propto w_{k|k-1}^{r_{1:k-1}}(r_k) q_k^{r_{1:k}}(r_k; \mathbf{y}_k)$ but from (5.39)

$$w_{k|k-1}^{r_{1:k-1}}(r_k) = p_S w^{r_{1:k-1}} \pi^{r_k, r_{k-1}}. \tag{5.64}$$

It can be seen that $\pi^{r_k, r_{k-1}}$ and $w^{r_{1:k-1}}$ are equivalent to $f_k(r_k|r_{1:k-1})$ and $f_k(r_{1:k-1}|\mathbf{y}_{1:k-1})$ respectively. Additionally, $q^{r_{1:k-1}}(r_k|\mathbf{y}_k)$ is the likelihood of measurement $\mathbf{y}_k$ conditioned on the mode history $r_{1:k}$ and hence may be written as $f_k(\mathbf{y}_k|r_{1:k})$. Finally, in the single target case $p_S = 1$.

Thus, substituting (5.64) into (5.63) gives:

$$w^{r_{1:k}} \propto f_k(\mathbf{y}_k|r_{1:k}) f_k(r_k|r_{1:k-1}) f_k(r_{1:k-1}|\mathbf{y}_{1:k-1}) \tag{5.65}$$

and the results follows from (5.51). □.

In summary, the JMLS-PHD filter is proposed as a method for using interacting multiple models within the PHD filter, removing the need for assumptions on target dynamics. This method is shown to be equivalent to the JMLS-GM-PHD filter derived by Pasha et al. [90] in the linear Gaussian case. It is also shown that in the single target linear Gaussian case, the JMLS-PHD filter is equal to the optimal Bayesian filter for jointly estimating the target state and motion mode. Both of these results give some support to the use of (5.33)-(5.35) as the general PHD filter in the presence of multiple motion models with JMLS structure. However, as with the GM-PHD filter claims of optimality are hampered by an exponentially increasing number of Gaussian mixture components with time. Some approximation will be necessary for computational tractability but fortunately the methods, such as pruning and merging, which are well-known within the context of the GM-PHD filter may be applied.

5.3 Comparison of PHD Filter with Track-Before-Detect Methods

For real tracking applications, the full data set for a given time step typically includes an amplitude value determined for each point in the (discretised) region of surveillance. The set of measurements used for multitarget tracking is obtained by thresholding the amplitude data, a process known as 'detection'. For high signal-to-noise ratio (SNR) targets, this is an adequate strategy as there will be a high probability of detecting any targets present and a low level of false alarms for a well chosen threshold. On the other hand, for low SNR targets the information loss in

thresholding may be large. It might be preferable to retain some, or in fact all, of the information lost at the detection step.

Track-before-detect (TBD) is an alternative method to the conventional multitarget tracking methods discussed in Section 3.1 and is particularly used for low SNR tracking. TBD rejects the use of a detection step and works on the full data set. It is well known that using the full set of amplitudes in this way allows targets to be detected more quickly [118]. There has been a recent extension of the PHD filter due to Clark et al. [29], referred to as the SNR-PHD filter hereafter, permitting incorporation of amplitude information through the detection step. This section will show that TBD methods are closely related to a special case of the SNR-PHD filter and give some illustrative results for simulated data.

Track-Before-Detect

The main idea of TBD, which first appeared in [118], is the incorporation of ideas from the sequential likelihood ratio test (see [15]) in order to permit computations of a track score which is updated at each time step,k, by multiplication by the likelihood ratio $(LR)_k$:

$$LR_k = \frac{p(\mathbf{Z}_k|\lambda_k = 1)}{p(\mathbf{Z}_k|\lambda_k = 0)}, \tag{5.66}$$

where $\mathbf{Z}_k$ is the data set at time k and λ_k is the random variable such that

$$\begin{cases} \lambda_k = 1 & \text{if target present at time } k, \\ \lambda_k = 0 & \text{if no target present at time } k. \end{cases} \tag{5.67}$$

Compare this with (3.6) but note that $\mathbf{Z}_k$ now represents the full data set for a given time step rather than the set of detections. The general goal is to infer λ_k and the unknown target state $\mathbf{x}_k$ at each time step k based on pixel data $\mathbf{Z}_k = \{z_k^i\}$ for $i = 1, ..., D$ where D is the total number of pixels.

Implementation by Salmond and Birch [105] - In [105] an equation of the same form as (5.66) is derived by considering a formal Bayes solution to computing the joint posterior $f_{k|k}(\mathbf{x}_k, \lambda_k|\mathbf{Z}^k)$ where $\mathbf{Z}^k = \{\mathbf{Z}_1, ..., \mathbf{Z}_k\}$ the full data set at each time step. For simplicity the presentation here concerns the special case where the target only affects the value of one pixel, although the implementation in [105] permits more general observation models.

The predicted density for $\mathbf{x}_{k+1}$ is given by:

$$f_{k+1|k}(\mathbf{x}_{k+1}, \lambda_{k+1}|\mathbf{x}_k, \lambda_k) = f(\mathbf{x}_{k+1}|\mathbf{x}_k, \lambda_k, \lambda_{k+1}) \cdot f(\lambda_{k+1}|\mathbf{x}_k, \lambda_k), \tag{5.68}$$

where $f(\lambda_{k+1}|\mathbf{x}_k, \lambda_k)$ is determined by the target birth and death models. If $\lambda_{k+1} = 0$ then $\mathbf{x}_{k+1}$ is undefined. If $\lambda_{k+1} = 1$ then

$$f(\mathbf{x}_{k+1}|\mathbf{x}_k, \lambda_k, 1) = \begin{cases} f_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{x}_k) & \text{if } \lambda_k = 1, \\ \gamma_k(\mathbf{x}_{k+1}) & \text{if } \lambda_k = 0. \end{cases} \tag{5.69}$$

where γ_k is the target birth density.

The likelihood for the Bayes update with pixel data z_k^i is based on the model:

$$z_k^i \sim \begin{cases} p_1(\cdot) & \text{if target is present affecting pixel } i, \\ p_0(\cdot) & \text{o.w.} \end{cases} \tag{5.70}$$

It is shown in [105] that the likelihood function for the Bayes update is:

$$f_{k+1}(\mathbf{Z}_{k+1}|\mathbf{x}_{k+1}, \lambda_{k+1}) \propto \begin{cases} \frac{p_1(z_k^i)}{p_0(z_k^i)} & \lambda_{k+1} = 1, \\ 1 & \lambda_{k+1} = 0. \end{cases} \tag{5.71}$$

Implementation by Billon [13] - The implementation presented in [13] is a more straightforward interpretation of (5.66) applied to a passive sonar tracking problem. In order to give extra clarity, a one dimensional simplification of the work in [13] is given.

The passive sonar data is of the form $\mathbf{Z}_k(f)$ where f represents frequency which, like time, is discretised. It is required to infer the target state $\mathbf{x}_k = f_k$ (the rate of change of frequency may also be required). Thus, the single target system has the form of a hidden Markov model (HMM) as discussed in Section 2.1.3.

Fig. 5.10 gives an example of such a data set which includes one obvious target frequency with $\mathbf{x}_k = 200$ for all k. Passive sonar data will be studied in more detail in Section 6.2. In [13] a 'track score' is computed for each point in the discretised frequency domain and at each time. The track score $\Lambda_k(f_i)$, for $i = 1, , , F$ (the states in the discretised frequency space), is calculated recursively by:

$$\Lambda_{k+1}(f_i) = \frac{p_1(\mathbf{Z}_{k+1}(f_i))}{p_0(\mathbf{Z}_{k+1}(f_i))} \sum_j f_{k+1|k}(f_i|f_j)\Lambda_k(f_j), \tag{5.72}$$

where $f_{k+1|k}(X|X')$ comes from the forward model.

Multi-Bernoulli Implementation by Vo et al. [126]

Vo et al. [126] propose a multitarget filter for TBD applications based on a multi-Bernoulli model which will be described briefly here. A Bernoulli random finite set, X, with associated probability r has a probability $(1-r)$ of being empty and a probability r of having exactly one element which is distributed according to the probability

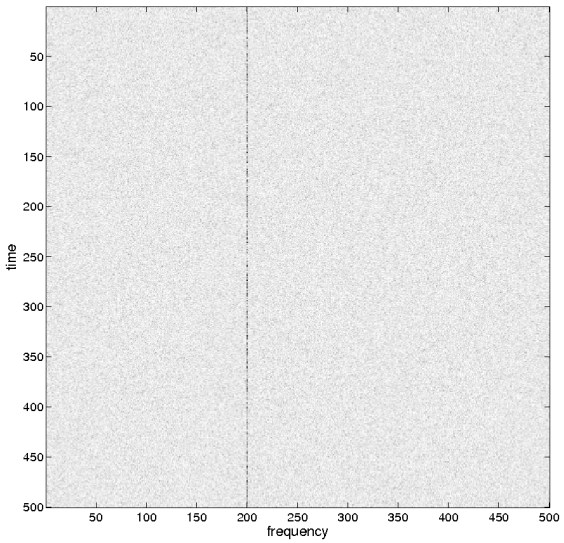


Figure 5.10: Example passive sonar data with one prominent target frequency.

density function p. The multi-Bernoulli model assumes that the multitarget state is a union of a fixed number of Bernoulli random finite sets.

Using this model, a multi-Bernoulli filter is derived for a varying number of targets in pixel data of the form described above. An SMC implementation of this filter is also presented in [126]. In the update step of the SMC filter, the particles weights are updated by multiplication by $p_1(z)/p_0(z)$ where z is the pixel data associated with the given particle. The is a clear similarity with the particle weight update in (5.71) although the multi-Bernoulli TBD (MB-TBD) filter is different in detail to the other TBD filters because of the multi-Bernoulli model.

Like the TBD implementation by Salmond and Birch [105], the observation model in the multi-Bernoulli approach permits targets which affect multiple pixels. However, in this section only the special case of targets affecting exactly one pixel is considered.

Full details of the MB-TBD filter may be found in [126] and the performance of the filter will be considered in Section 5.3.3.

SNR-PHD filter

The SNR-PHD approach of Clark et al. [29] is different in that it relies on the detection step, but retains the amplitude information for each measurement. The target state vector is then thought of as $\mathbf{x}_k = [\hat{\mathbf{x}}, d]$ where $\hat{\mathbf{x}}$ is the traditional target state and d contains the parameter(s) relating to the SNR of the target. Similarly, each measurement vector is appended with amplitude information, labelled a so that $\mathbf{z}_k = [\hat{\mathbf{z}}, a]$.

It is assumed that the amplitude return is independent of state measurement and so the likelihood including amplitude information, $L_{\mathbf{z}}$, and the clutter likelihood, $\kappa(\mathbf{z})$, are given by:

$$L_{\mathbf{z}}(\mathbf{x}) = L_{\hat{\mathbf{z}}}(\hat{\mathbf{x}}) \cdot L_a(d), \tag{5.73}$$

$$\kappa(\mathbf{z}) = \kappa(\hat{\mathbf{z}}) \cdot \kappa(a), \tag{5.74}$$

where $L_a(d)$ and $\kappa(a)$ are the amplitude likelihood functions for a target with parameter d and clutter respectively.

Because the detection step requires only accepting measurements with amplitudes greater than some threshold τ, it is necessary to define $L_a^\tau(d)$ and $\kappa^\tau(a)$, the amplitude likelihood functions for target and clutter given that the threshold has been exceeded, as:

$$L_a(d) = L_a^\tau(d) p_D^\tau(d), \tag{5.75}$$

$$\kappa(a) = \kappa^\tau(a) p_{FA}^\tau, \tag{5.76}$$

where

$$p_D(d)^\tau = \int_\tau^\infty L_a(d) \mathrm{d}a, \tag{5.77}$$

$$p_{FA}^\tau = \int_\tau^\infty \kappa(a) \mathrm{d}a, \tag{5.78}$$

are the probabilities of detection and false alarm, given the threshold τ, respectively.

This leads to a PHD pseudo-likelihood including amplitude information, as derived in [22] given by:

$$L_{\mathbf{z}}(\mathbf{x}) = (1 - p_D^\tau(d)) + \sum_{z \in Z} \frac{p_D^\tau(d) L_a^\tau(d) L_{\hat{\mathbf{z}}}(\hat{\mathbf{x}})}{\lambda p_{FA}^\tau \kappa^\tau(a) \kappa(\mathbf{z}) + \langle D_{k+1|k}, p_D^\tau(d) \cdot L_a^\tau(d) L_{\hat{\mathbf{z}}}(\hat{\mathbf{x}}) \rangle}, \tag{5.79}$$

where $D_{k+1|k}$ is the predicted PHD and $\langle f, g \rangle$ is the inner product $\int f(x) g(x) \mathrm{d}x$.

5.3.1 Comparison of SNR-PHD with HMM Implementation

By restricting the SNR-PHD filter to the HMM case, it will be shown that in the special case when the threshold $\tau = 0$ there is a close relationship between the SNR-PHD filter and the HMM track-before-detect (HMM-TBD) filter as given in (5.72).

First, the SNR-PHD filter for the HMM case will be derived. This requires three things: filter initialisation, a forward step and an update step. As in the HMM-TBD filter, it will be assumed that the initialisation of the filter is given. This will require

the PHD to have constant value for each of the finite number of target states, denoted $\{y^i\}_i$.

$$D_{0|0}(y^i) = w^i_{0|0} \quad \forall i. \tag{5.80}$$

The predicted PHD will be prescribed by the HMM, giving the forward step:

$$D_{k+1|k}(y^i) = \sum_j f_{k+1|k}(y^i|y^j) D_{k|k}(y^j) \tag{5.81}$$

Note that this step of the recursion is equivalent to the last two terms in (5.72).

The update step will be the multiplication of $D_{k+1|k}$ by the PHD pseudo-likelihood given in (5.79). But, some simplifications can be made to (5.79) in the special case when $\tau = 0$. This is when the measurement corresponding to each state in the discretised space is declared, The simplifications are:

- $p_D^\tau = 1$
- $L_a^\tau(d)$ and $\kappa^\tau(a)$ reduce to simply the likelihoods for the amplitude in the presence and absence of a target respectively. Hence, in the notation of this section, they may be written as $p_1(\mathbf{z})$ and $p_0(\mathbf{z})$.
- By assuming, without loss of generality, that the volume of each cell is unit, the non-amplitude likelihood is:

$$L_{\hat{\mathbf{z}}}(y^i) = \begin{cases} 1 & \text{if } \hat{\mathbf{z}} \text{ corresponds to the state } y^i, \\ 0 & \text{o.w.} \end{cases} \tag{5.82}$$

- $\kappa(z) = 1/\lambda p_{FA}$. Note that integrating $\kappa(z) \times \lambda p_{FA}$ over a region gives the expected number of false alarms in that region. Hence, this choice reflects the fact that when $\tau = 0$ it is expected that there will be a false alarm in almost all cells.

Using the above simplifications, the PHD pseudo-likelihood may be written as

$$L_{\mathbf{z}}(y^i) = \frac{p_1(\mathbf{Z}_{k+1}(f_i))}{p_0(\mathbf{Z}_{k+1}(f_i)) + D_{k+1|k}(y^i) p_1(\mathbf{Z}_{k+1}(f_i))}, \tag{5.83}$$

and hence

$$D_{k+1|k+1}(y^i) = \frac{D_{k+1|k}(y^i) p_1(\mathbf{Z}_{k+1}(f_i))}{p_0(\mathbf{Z}_{k+1}(f_i)) + D_{k+1|k}(y^i) p_1(\mathbf{Z}(k+1, f_i))} \tag{5.84}$$

$$= \frac{\frac{D_{k+1|k} p_1}{p_0}}{1 + \frac{D_{k+1|k} p_1}{p_0}}. \tag{5.85}$$

It can be seen from (5.72) that $D_{k+1|k}p_1/p_0$ is the result of the recursion using the HMM-TBD method and so denoting $\tilde{D}_{k+1|k+1}$ as the result of propagating $D_{k|k}$ under the HMM-TBD recursion,

$$D_{k+1|k+1} = \frac{\tilde{D}_{k+1|k+1}}{1 + \tilde{D}_{k+1|k+1}}. \tag{5.86}$$

Therefore, it is clear that there is a close relationship between this special case of SNR-PHD filter and the HMM-TBD filter. This information may give a practitioner of track-before-detect methods reason to use the PHD filter to provide an alternate 'score function' to the traditional TBD score function. The PHD provides some obvious advantages over the existing score function. Firstly, the PHD is always less than 1 and has a physical interpretation as a density of expected targets. Furthermore, as the score can increase indefinitely, a track with high SNR for a large number of time steps may remain a confirmed track for many time steps after it has disappeared by virtue of having accumulated a high score while it was present. Such inconsistencies are not a problem for the PHD filter.

Perhaps more advantageous is the ability to use the SNR-PHD filter as the basis for a unified multitarget tracker incorporating target initialisation, track management and association, whereas standard TBD techniques leave the multitarget aspects of the problem unresolved and require additional heuristics, such as those described in [13].

5.3.2 Comparison with Particle Filter Track-Before-Detect

In [105], Salmond and Birch derive prediction and update steps for a Bayesian inference taking amplitude returns into account in the case where there are either 0 or 1 targets. Their results can be re-derived in terms of the multitarget Bayes filter using a similar derivation to that presented in Section 3.3. Thus, the Bayesian TBD filter in [105] is equivalent to a multitarget Bayes filter. This result is proved in Appendix B.3.

However, as the multitarget Bayes filter is not computationally tractable in general, we also consider an SMC-PHD implementation. Due to the assumptions related to Poisson distributions, which are necessary in the derivation of the PHD filter but are contrary to the assumptions in [105], the two methods cannot be equivalent. There is, however, a similarity regarding the update step in the special case where the declaration threshold τ, is equal to zero.

Most of the analysis in Section 5.3.2 holds here apart from a difference in the calculation of the inner product in the denominator of the pseudo-likelihood (5.79). Whereas for the HMM implementation the PHD was constant on each grid cell, here the PHD has a particle representation. This means that the term $\langle D_{k+1|k}, L_{\hat{\mathbf{z}}} p_1(\mathbf{z}) \rangle$ gives $p_1(\mathbf{z}) p_{\mathbf{z}}^{sum}$ where $p_{\mathbf{z}}^{sum}$ is the sum of the weights of the particles in the predicted PHD in the grid cell corresponding to the measurement $\mathbf{z}$.

Therefore, the particle weightings in the update step of the SMC-PHD filter depend only on the measurement in their grid cell due to the assumed form on the likelihood function in (5.82) giving:

$$w_{k+1}^{j} = \frac{p_1(\mathbf{z}_{k+1}^{i})}{p_0(\mathbf{z}_{k+1}^{i}) + p_{\mathbf{z}_{k+1}^{i}}^{sum} p_1(\mathbf{z}_{k+1}^{i})}, \tag{5.87}$$

where i is the cell number that the jth particle falls in. Comparing this with (5.71) it can be seen that there is a relationship between the update step for the two methods.

5.3.3 Illustrative Results

Single Target

Following the approach in [105] a simple tracking scenario was simulated to illustrate the performance of the SMC implementation of the SNR-PHD filter with comparison to the particle filter TBD. In the first simulation, a sequence of 40 frames, each consisting of 20x20 pixels was simulated. For each pixel corresponding to 'no target', amplitudes were drawn from a Rayleigh distribution with $\sigma = 1$. A target appears at time step 10 and disappears at time step 30. Pixels corresponding to a present target have amplitudes drawn from a Rayleigh distribution with $\sigma = 2.5119$ giving a 8dB SNR as calculated using equation (5) in [29]. The data is displayed in [105] by showing six selected frames from the simulated data, but it is more clear to see the nature of the data by viewing the full video. This can be viewed at `http://people.maus.ox.ac.uk/woodtm/8db1target.avi`. In the author's opinion it is difficult to spot the target with the human eye. A set of data with 20dB SNR was also generated to help the reader see the target if they could not spot it in the first video: `http://people.maths.ox.ac.uk/woodtm/20db1target.avi`.

For both the track before detect filter and the SNR-PHD filter, p_B in the birth model was set to 0.1 and p_S to 0.9, the target birth density is taken to be uniform on the surveillance region. The number of particles used was 10000 for the track before detect filter and 10000 particles per target for the SMC-PHD filter.

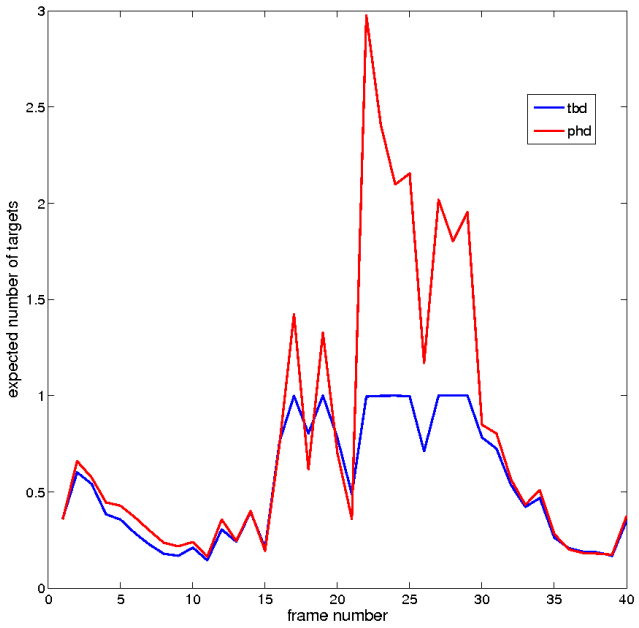


Figure 5.11: Expected number of present targets for the SNR-PHD and TBD filters. There is a target present for frames 10-30.

Fig. 5.11 shows the expected number of targets present for both the SNR-PHD and track before detect filters. This may be interpreted as the 'probability that a target is present' for the track before detect. It is clear that the SNR-PHD filter is more unstable and reactive to the data. This could be due to the 'short memory' effect of the PHD filter as discussed in [38]. Furthermore, it is clear that the SMC-PHD filter has a tendency to overestimate the total number of targets present. However, in standard implementations of the PHD filter, the particles are clustered into sets of particles representing individual targets, with an associated weight. These are then taken to be individual possible tracks with weights subject to thresholding. This prevents lots of low weight tentative tracks from contributing to an overestimation of the total target number causing the overestimation seen here.

The improved clustering method as outlined in Section 5.1.2 is applied to obtain tentative track estimates from the particle PHD representations. Track estimates are considered "correct" and corresponding to the target if they lie within 1 pixel of the true target pixel. Fig. 5.12 shows the expected number of targets given by the particle TBD filter as compared with the expected number of targets corresponding to "correct" track estimates in the SNR-PHD filter, showing a close correspondence. If, as suggested in [105] a target is declared if it has probability greater than 0.6, then the target would be declared for 14 out of 21 time steps for the track before detect

filter and 13 out of 21 for SMC-PHD.

It is also interesting to note that the particle clustering techniques are beneficial for the track before detect algorithm. The standard technique for state estimation in the track before detect filter is an average over all of the particles. However, some particles which do not correspond to the target, but instead to other tentative target positions, may bias the mean away from the true target location. Using a clustering method would prevent this. The distance of the estimate from the true target location with and without clustering is shown in fig. 5.13. This figure shows a significant improvement in state estimation in the particle TBD algorithm obtained by employing techniques from the SMC-PHD filter (see Section 5.1). Thus, clustering techniques used in PHD implementations may be a valuable addition to the particle filter TBD toolbox.

Some final points of comparison: the mean localisation error (in pixels) for particle TBD (after clustering) was 0.39 whereas for SNR-PHD it was 0.46. There were 2 false declarations for SNR-PHD compared with 1 false declaration for particle TBD (where a false declaration is a state estimate with a weight above the 0.6 threshold but more than one pixel away from the true target pixel).

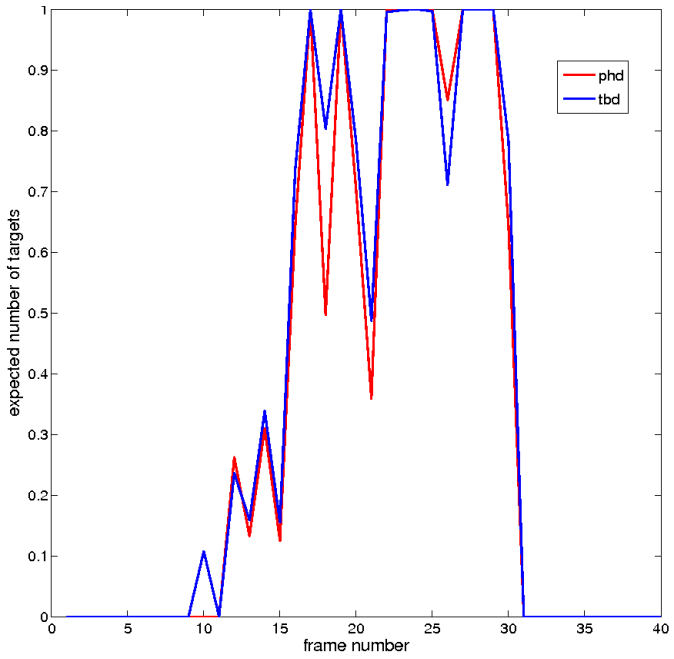


Figure 5.12: Expected number of present targets for the SNR-PHD and track before detect filters after clustering techniques are applied. There is a target present for frames 10-30.

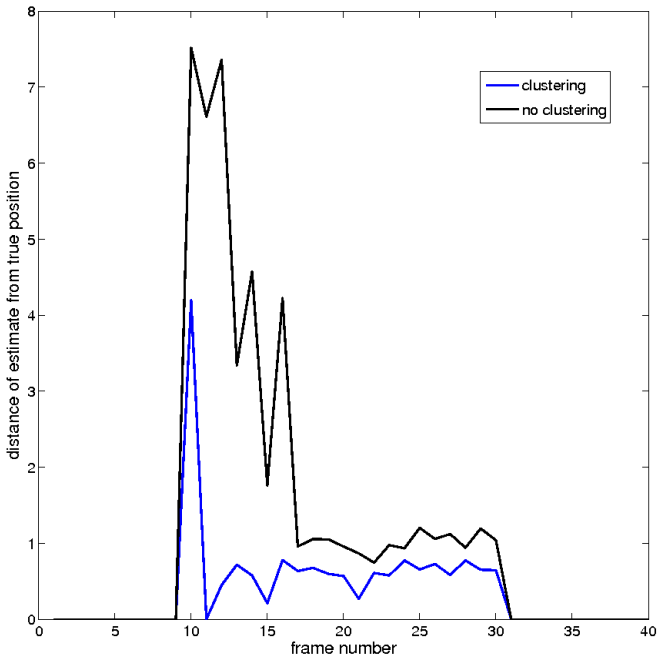


Figure 5.13: Distance of particle TBD estimate from true target position (in pixels) with and without the use of clustering techniques.

Multiple Targets

As an illustration of the ability of the SNR-PHD filter to work in the case with multiple targets, a scenario was simulated with 3 targets appearing and disappearing at different times over 60 time steps. 100 Monte Carlo simulations were run and one instance of the video data can be viewed at
`http://people.maths.ox.ac.uk/woodtm/8db3target.avi` and, as previously, with a higher SNR at
`http://people.maths.ox.ac.uk/woodtm/20db3target.avi`. For all tests Target 1 is present for frame numbers 10-30, target 2 for frames 20-40 and target 3 for frames 20-50.

No adjustments were made to the algorithm deployed in the single target case.

In order to help evaluate the performance of the SNR-PHD filter in multitarget tracking scenario, an SMC implementation of the MB-TBD filter [126] was also computed for comparison. The same parameters and models used in the SNR-PHD filter as described above were used for the MB-TBD filter where possible. For target birth in the MB-TBD filter, the model is the appearance of one target per time step with probability 0.1. The number of particles used in the MB-TBD filter is 10000 particles per possible target (i.e. for each independent Bernoulli random finite set).

Fig. 5.14 shows the expected number of targets (averaged over the 100 runs) in the vicinity of the three target locations for both the SNR-PHD filter and the MB-TBD filter, broadly demonstrating that both filters establish tracks on all three targets.

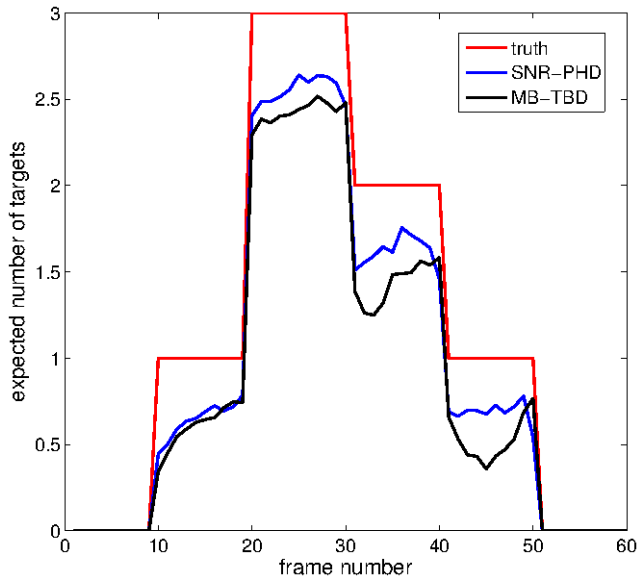


Figure 5.14: Expected target number for the SNR-PHD and MB-TBD filters in the vicinity of the multiple targets and the true target number.

The average cardinality error for the SNR-PHD filter is 0.26 whereas for MB-TBD it was 0.34. Furthermore, the SNR-PHD filter returned an average of 1.5 false alarms over the 60 time steps of the test while the MB-TBD filter returned 2.8 false alarms on average. These results demonstrate good performance in detecting and locating multiple targets by the SNR-PHD filter with performance slightly better than the established MB-TBD filter in this simple test. It should be noted however, that the MB-TBD should be expected to demonstrate better performance in scenarios where targets affect more than one pixel, as this cannot be modelled in the SNR-PHD filter.

In summary, an implementation of the SNR-PHD filter analogous to two contrasting track-before-detect implementations is derived. For the HMM implementation, it is shown that in the special case when the declaration threshold is set equal to zero there is a close link between the SNR-PHD and TBD methodologies. The particle filter implementation of TBD is shown to be equivalent to a special case of a multi-target Bayes implementation, and there is a similarity between the update steps of particle TBD and SNR-PHD in the special case when the declaration threshold is set to zero. Illustrative results are given comparing the performance of particle TBD with

SNR-PHD showing similar performance in the case where there is only one target, but the flexibility of the SNR-PHD filter permits tracking of multiple targets with no adjustments.

5.4 Multitarget Miss Distance and Extensions

The concept of a miss distance between a quantity and its estimated value is well-understood and fundamental to the theory of filtering. Miss distances are taken for granted in single target filtering systems but cannot be in multitarget systems. Possession of a multitarget miss distance is crucial to the performance evaluation of multitarget filters as well as parameter tuning for such filters.

There are a large number of potential measures of effectiveness of a multitarget tracker. Blackman and Popoli [15, p. 903], for example, suggest a score calculated by awarding points for early track declaration, track label continuity and covariance reliability while subtracting points for false tracks or early track deletion. While such measures help to condense all of the aspects of multitarget tracking into a single value and hence aid comparison, a rigorously derived distance satisfying the conditions of a metric is preferable from a mathematical point of view.

Schuhmacher et al. [107] concisely summarise the properties that a multitarget miss distance should have as follows:

- a metric on the space of finite sets;
- have a natural (meaningful) physical interpretation;
- capture cardinality and localisation errors meaningfully;
- be easily computed.

Let $\mathcal{X}$ be the space of finite sets. A metric on the space of finite sets is a function $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ which satisfies:

1. $d(x, y) > 0 \quad \forall x, y \in \mathcal{X}$ *[positivity]*
2. $d(x, y) = 0 \quad$ iff $x = y$ *[identity]*
3. $d(x, y) = d(y, x) \quad \forall x, y \in \mathcal{X}$ *[symmetry]*
4. $d(x, y) \leq d(x, z) + d(y, z) \quad \forall x, y, z \in \mathcal{X}$ *[triangle inequality]*

For multitarget tracking purposes it will be sufficient to consider $\mathcal{X}$ a finite subset of a bounded window $W \subseteq \mathbb{R}^n$.

In this section, the existing work on multitarget metrics is summarised and a new multitarget distance is proposed which takes track confidence information (such as the track weights in the GM-PHD or SMC-PHD filter) into account. To see why this is desirable, consider two filters, filter A and filter B. Both filters return one track estimate at a point in the state space, but the filter A assigns track confidence 0.9 to the track whereas filter B assigns track confidence 0.5 (where these may be thought of something like 'the probability that these is a track associated with this state estimate'). In the case where there is a target present, it is clear that the performance of filter A is somewhat better, however in the case where there is no target present, the performance of filter B is better. A metric which does not take target confidence information into account effectively states that the performance of both filters is identical in both cases.

It is shown that the proposed multitarget distance including track confidence satisfies the conditions of a metric.

5.4.1 Existing Multitarget Distance Metrics

A first candidate for a finite set metric is the Hausdorff metric, d_H defined for $X, Y \in \mathcal{X}$:

$$d_H(X,Y) = \max\big\{ \max_{x\in X} \min_{y\in Y} d(x,y), \max_{y\in Y} \min_{x\in X} d(x,y)\big\}. \tag{5.88}$$

It is observed in [49] that the Hausdorff metric is insensitive to cardinality differences, which makes it unsuitable as a multitarget miss distance.

Hoffman and Mahler [49] proposed a new metric for finite sets based on the p-th order Wasserstein metric between the empirical distributions of the point patterns for $X = \{x_1, ..., x_m\}$ and $Y = \{y_1, ..., y_n\}$ given by:

$$d_p(X,Y) = \min_C \big(\sum_{i=1}^{m} \sum_{j=1}^{n} C_{i,j} d(x_i, y_j)^p \big)^{\frac{1}{p}}. \tag{5.89}$$

Here, $d(x,y)$ is a distance metric in the underlying space (typically a Euclidean distance) and the minimum is taken over all $m \times n$ transportation matrices, $C_{i,j}$. An $m \times n$ matrix is a transportation matrix if its entries are non-negative and if

$$\sum_{j=1}^{n} C_{i,j} = \frac{1}{m} \quad \forall i, \quad \sum_{j=1}^{m} C_{i,j} = \frac{1}{n} \quad \forall j. \tag{5.90}$$

Optimal Sub-pattern Assignment (OSPA) Metric

Schuhmacher et al. [107] subsequently demonstrated inconsistency within the Hoffman-Mahler metric particularly in the handling of cardinality error by the metric. Using a similar construction, also based on the p-th order Wasserstein metric, they introduced the optimal sub-pattern assignment (OSPA) metric. The OSPA metric eliminates the inconsistencies within the Mahler-Hoffman metric and results in a metric with a more natural physical interpretation of the handling of localisation and cardinality errors.

Define

$$d^c(x, y) = \min(c, d(x, y)), \tag{5.91}$$

where $d(x, y)$ is a metric in the underlying space. $d^c(x, y)$ may be interpreted as the distance between x and y cut off at c. Let $X = \{x_1, ..., x_m\}$, $Y = \{y_1, ..., y_n\}$ and $\Pi_{m,n}$ be the set of maps $\pi : \{1, ..., m\} \to \{1, ..., n\}$ where $m \leq n$. The OSPA metric, $d_p^c(X, Y)$ is defined as:

$$d_p^c(X, Y) = \left(\frac{1}{n} \left(\min_{\pi \in \Pi_{m,n}} \sum_{i=1}^{m} d^c(x_i, y_{\pi(i)})^p + c^p(n - m) \right) \right)^{\frac{1}{p}}, \tag{5.92}$$

if $m \leq n$ and $d_p^c(X, Y) = d_p^c(Y, X)$ is $m > n$. If $m = n = 0$ the distance is set to zero. The fact that $d_p^c(X, Y)$ is a metric is not obvious from inspection, but is proved in the appendix of [107].

The interpretation of the OSPA metric becomes clear when one considers the method for computing it:

1. Calculate $d^c(x, y)^p$ for each pair of points $x \in X$ and $y \in Y$.
2. Compute the optimal assignment using, for example, an auction algorithm (see [12]).
3. Add the total distance from the optimal assignment to $c^p(n - m)$.
4. Multiply by $1/n$ and take the pth root

From this, it is clear that the OSPA metric is composed of an optimal localisation error added to a cardinality error. Furthermore, the penalisation for each increment to the cardinality error is equal to the maximum value of the distance metric $d^c(x, y)^p$. Therefore, the parameter c corresponds to the value of localisation distance equivalent to each unit of cardinality error. The parameter p corresponds to how much outliers are penalised much like in a Euclidean L_p norm. The parameters c and p may

be chosen to determine the relative importance of localisation and cardinality errors and of outliers as required for the tracking scenario. Larger values of c place a higher weighting on cardinality errors and lower values place a higher weight on localisation errors.

OSPA Metric for Tracks

Ristic et al. [97] describe an adaptation of the OSPA metric to include tracks. The distinction between a multitarget state and a multitarget track is that the track carries a label permitting the continuous monitoring of trajectories due to single targets. Tracks which carry a label continuously are preferred and so a switch of track label or an incorrect assignment of track label should be penalised by a higher metric value. This is achieved by introducing an extra component into the single-target metric d_c in the case of an incorrect track label. The resulting metric is shown in [97] to have a good correspondence with intuition for multitarget tracking errors.

5.4.2 Multitarget Miss Distance with Target Weights

As discussed in Section 4.2, the PHD filter outputs tracks with associated weights. Similarly, the MHT filter discussed in Section 3.1.2 also includes weights associated with tracks while the track management methods described in Section 3.1.3 also associate a level of confidence with each track. These levels of confidence that each track corresponds to a true target will be referred to generically as 'track weights' throughout this section.

The track weights, as well as track locations, are obviously relevant to the performance of a filter. For example, a filter with higher track weights corresponding to its correct track estimates and lower track weights corresponding to false alarms is a better filter than one with equal weightings for each.

Track weights can only taken into account in the OSPA metric by a thresholding process which declares only those tracks which have weights above a threshold, τ. Those tracks with weights above the threshold are implicitly declared as "full confidence" tracks whereas those below the threshold are not tracks at all. This can lead to a certain form of inconsistency within the OSPA metric. To see this, consider making a small variation in the model probability of detection parameter p_D within a PHD filter. A small change in this parameter can result in a small change in the weight associated with a given track. If this small change in weight moves the track over the threshold, τ, it will lead to a large change in the OSPA miss distance for the filter output. This observation does not challenge the fact that OSPA is a metric. OSPA is

a metric over the space of finite sets, not multitarget filters. However, from the point of view of performance analysis a metric which alleviated this inconsistency might be preferable. As a further example, consider a filter which declares a number of false alarms each with weight 0.245. Changing the threshold from 0.25 to 0.24 would lead to a large change in the OSPA metric, so the choice of threshold itself may have a large impact on the performance analysis of a filter, which is also undesirable.

A distance metric which takes target weights into account in order to alleviate these problems should have the following properties:

- a metric on the space of finite sets including weights;
- higher weight "true tracks" are better (i.e. lead to lower distances);
- lower weight "false tracks" are better;
- equivalent to OSPA when all weights are 1.

In this section, a distance metric satisfying the above properties will be proposed. The distance will be defined between two sets with weights, X and Y where $|X| = l$, $|Y| = m$ and

$$X = \{(x_1, u_1), ..., (x_l, u_l)\} = \{\mathbf{x}_1, ..., \mathbf{x}_l\}, \tag{5.93}$$

$$Y = \{(y_1, v_1), ..., (y_m, v_m)\} = \{\mathbf{y}_1, ..., \mathbf{y}_m\}. \tag{5.94}$$

Note that u_i is the weight associated with the state x_i and so on. Note that the notational convention used is that target states with associated weight are bold whereas conventional target states (i.e. without weights) are not bold.

Simple Incorporation of Weight Into Single Target Distance

Following the approach of Ristic et al. [97] the track weight error will be incorporated into the base distance. Let:

$$\hat{d}(\mathbf{x}, \mathbf{y}) = d(x, y) + \alpha|u - v|, \tag{5.95}$$

where α may be chosen to determine how important weight errors are compared to localisation errors. This distance with weights errors included obviously satisfies the positivity, identity and symmetry conditions for a metric. To show that it also

satisfies the triangle inequality, let $\mathbf{z} = (z, w)$.

$$\begin{aligned}
\hat{d}(\mathbf{x}, \mathbf{y}) &= d(x, y) + \alpha|u - v| && (5.96)\\
&\leq d(x, z) + d(y, z) + \alpha(|u - w| + |v - w|) && (5.97)\\
&= (d(x, z) + \alpha|u - w|) + (d(y, z) + \alpha|v - w|) && (5.98)\\
&= \hat{d}(\mathbf{x}, \mathbf{z}) + \hat{d}(\mathbf{y}, \mathbf{z}). && (5.99)
\end{aligned}$$

Therefore $\hat{d}$ is a metric and using $\hat{d}$ as the base metric in the OSPA metric will give a multitarget metric which takes into account differences in track weight.

When using the metric for performance evaluation of filters, the track weights in the known ground truth will all be set to 1. Therefore, using $\hat{d}$ as the base metric will mean that higher weight true tracks are better, as required. This satisfies the first three conditions required for a multitarget metric with track confidence but there is no obvious mechanism to allow for the converse: the preference for lower weight false tracks.

Alternative Base Distance Incorporating Track Weight

An alternative base distance will be introduced which allows a natural extension to a multitarget metric incorporating target weight and satisfying the criteria listed earlier in this section. The distance is given by:

$$\tilde{d}_c(\mathbf{x}, \mathbf{y}) = \min(u, v) d_c(x, y) + \alpha|u - v|. \quad (5.100)$$

While $\tilde{d}$ obviously satisfies positivity and symmetry and $x = y \Rightarrow \tilde{d}(x, y) = 0$ it is possible to have $\tilde{d} = 0$ for distinct x and y if $u = v = 0$. Technically, this means that $\tilde{d}$ is at best a pseudo-metric [113]. This will not be important in practise as tracks with zero confidence would not be declared by a practical multitarget filter. Furthermore, it could be argued that the locations of zero weight tracks is irrelevant, thus any two points with weight zero might reasonably be taken to be non-distinct. In order to show that $\tilde{d}$ is in fact a (pseudo)metric, the triangle inequality must be shown for $\mathbf{x} = (x, u), \mathbf{y} = (y, v), \mathbf{z} = (z, w)$.

Proof of triangle inequality for d_c:

Assume w.l.o.g. that $u > v$ and consider two distinct cases: i) w is the smallest weight. ii) w is not the smallest weight.

Case i)

$$\begin{aligned}
\tilde{d}_c(\mathbf{x}, \mathbf{y}) &= v d_c(x, y) + \alpha(u - w + w - v) && (5.101)\\
&= v d_c(x, y) + \alpha|u - w| - \alpha|v - w| && (5.102)\\
&= v d_c(x, y) - 2\alpha|v - w| + \alpha|u - w| + \alpha|v - w|. && (5.103)
\end{aligned}$$

Now, it is necessary to assume that $\alpha \geq c/2$ which gives:

$$\begin{aligned} d_c(x,y) &\leq 2\alpha, && (5.104)\\ \Rightarrow (v-w)d_c(x,y) &\leq 2\alpha(v-w), && (5.105)\\ \Rightarrow v d_c(x,y) - 2\alpha|v-w| &\leq w d_c(x,y). && (5.106) \end{aligned}$$

Substituting this result into (5.103) gives:

$$\begin{aligned} \tilde{d}_c(\mathbf{x},\mathbf{y}) &\leq w d_c(x,y) + \alpha|u-w| + \alpha|v-w| && (5.107)\\ &\leq w d_c(x,z) + w d_c(y,z) + \alpha|u-w| + \alpha|v-w| && (5.108)\\ &= \min(u,w) d_c(x,z) + \alpha|u-w| + \min(v,w) d_c(y,z) + \alpha|v-w| && (5.109)\\ &= \tilde{d}_c(\mathbf{x},\mathbf{z}) + \tilde{d}_c(\mathbf{y},\mathbf{z}). && (5.110) \end{aligned}$$

Proving the triangle inequality for case ii) is easier:

$$\begin{aligned} \tilde{d}_c(\mathbf{x},\mathbf{y}) &\leq \min(u,v)(d_c(x,z) + d_c(y,z)) + \alpha(|u-w| + |v-w|) && (5.111)\\ &= \min(u,v,w)(d_c(x,z) + d_c(y,z)) + \alpha(|u-w| + |v-w|) && (5.112)\\ &\leq \min(u,w) d_c(x,z) + \min(v,w) d_c(y,z)) + \alpha(|u-w| + |v-w|) && (5.113)\\ &= \tilde{d}_c(\mathbf{x},\mathbf{z}) + \tilde{d}_c(\mathbf{y},\mathbf{z}). \quad \square && (5.114) \end{aligned}$$

So $\tilde{d}_c$ is a (pseudo)metric for $\alpha \geq c/2$.

The distance $\tilde{d}_c$ may be interpreted as the sum of a localisation error and a weight error (which may be regarded as a form of cardinality error). Localisation error are penalised at the "localisation error rate" as far as the weights match up while the mismatch is charged at the "cardinality error rate", α. The restriction on α ensures that cardinality errors are penalised at a sufficiently high rate.

For example, consider a track with weight p being compared to a true target in a known ground truth. As true targets must have weight equal to one, the distance $\tilde{d}_c$ will be $p\times$'localisation error' $+(1-p)\times\alpha$. In order to ensure that higher weight matches to true tracks are preferred by $\tilde{d}_c$, it will be necessary to impose the stricter condition of $\alpha \geq c$.

Multitarget Distance

The inclusion of a distinction between localisation and cardinality errors in the base distance suggests a method for calculating a multitarget distance with cardinality errors handled using track weights according to the criteria listed at the beginning of this section. Schuhmacher et al. [107] remark that the cardinality error in OSPA

may be considered as "filling up the point pattern X, which has smaller cardinality l with $m - l$ 'dummy points' located at distance $\geq$ c from any points in Y". Their proof that OSPA is a metric uses this construction.

Following this approach, consider filling X, the set with smaller cardinality, with dummy points with weight zero (making their location irrelevant). Denoting such a dummy point $\mathbf{s}$, it can be seen from (5.100) that $\tilde{d}(\mathbf{y}_i, \mathbf{s}) = \alpha v_i$ for any $\mathbf{y}_i \in Y$. This suggests the following definition for a multitarget distance:

$$\tilde{D}_c(X, Y) = \min_{\pi \in \Pi} \sum_{i=1}^{l} \tilde{d}_c(\mathbf{x}_i, \mathbf{y}_{\pi(i)}) + \alpha \sum_{j \in \Omega} v_j, \tag{5.115}$$

for $|X| \leq |Y|$ and $D_c(X, Y) = D_c(Y, X)$ for $|X| > |Y|$. The minimisation is over Π, the set of association hypotheses $\Pi : \{1, ..., l\} \to \{1, ..., m\}$ and $\Omega = \{j : \pi(i) \neq j \quad \forall i\}$.

Some properties of this multitarget distance metric include:

- Higher weight "true tracks" are preferred. This is taken into account in $\tilde{d}_c$ as described above.
- False (unmatched) tracks are penalised at a rate proportional to their weight, so lower weight false tracks are preferred. Note that unmatched "true tracks" or missed tracks are also penalised by their weight, which is equal to one, so missed tracks are always penalised at the full rate.
- In the case when all weights are equal to one and $\alpha = c$ the above metric clearly reduces to the L_1 version of the OSPA metric (without the factor of $1/m$).

Hence, the only property of the multitarget distance which remains to be shown is that it is a metric.

Proof that $\tilde{D}_c$ is a Metric

Firstly, it will be checked that the process of adding dummy points cannot cause inconsistencies by changing the optimal assignment within the metric. Let $X = \{(x, u)\}, Y = \{(y, v)\}, X' = \{(x, u), \mathbf{s}\}$ and $Y' = \{(y, v), \mathbf{t}\}$ where $\mathbf{s}, \mathbf{t}$ are dummy points. The distance between two dummy points is zero, so for consistency it is required that $\tilde{D}_c(X, Y) = \tilde{D}_c(X', Y')$. In order to show this, it must be demonstrated that:

$$\tilde{d}_c(x, y) \leq \tilde{d}_c(x, t) + \tilde{d}_c(s, y), \tag{5.116}$$

from which it follows that the assignment of x to y and s to t is the optimal one and hence $\tilde{D}_c(X,Y) = \tilde{D}_c(X',Y')$.

Proof of (5.116):

$$\begin{aligned} d_c(x,y) &= \min(u,v)d_c(x,y) + \alpha|u-v| && (5.117)\\ &= \min(u,v)(d_c(x,y)-\alpha) + \alpha\max(u,v) && (5.118)\\ &\leq \min(u,v)(2\alpha-\alpha) + \alpha\max(u,v) && (5.119)\\ &= \alpha u + \alpha v && (5.120)\\ &= \tilde{d}_c(x,t) + \tilde{d}_c(s,y), && (5.121) \end{aligned}$$

where (5.119) uses $\alpha \geq c/2$. $\square$.

Positivity, identity (allowing for equality of all zero weight tracks) and symmetry of $\tilde{D}_c$ are obvious, so all that remains to be shown is that $\tilde{D}_c$ satisfies the triangle inequality.

Proof of triangle inequality for $\tilde{D}_c$:
Let

$$X = \{(x_1,u_1),...,(x_l,u_l)\} = \{\mathbf{x}_1,...,\mathbf{x}_l\}, \quad (5.122)$$
$$Y = \{(y_1,v_1),...,(y_l,v_l)\} = \{\mathbf{y}_1,...,\mathbf{y}_m\}, \quad (5.123)$$
$$Z = \{(z_1,w_1),...,(z_n,w_n)\} = \{\mathbf{z}_1,...,\mathbf{z}_n\}, \quad (5.124)$$

with $l \leq m$ w.l.o.g. First consider the case where $l \leq m \leq n$.

Let $S = \{\mathbf{s}_i\}_{i=1}^{n-l}$ and $T = \{\mathbf{t}_i\}_{i=1}^{n-m}$ be sets of zero weight dummy points, and append these to the end of X and Y respectively so that each set (including the dummy points) has cardinality n.

Also let $\pi = \arg\min_\rho \sum_{i=1}^n \tilde{d}_c(\mathbf{x}_i, \mathbf{y}_{\rho(i)})$, $\sigma = \arg\min_\rho \sum_{i=1}^n \tilde{d}_c(\mathbf{x}_i, \mathbf{z}_{\rho(i)})$, $\tau = \arg\min_\rho \sum_{i=1}^n \tilde{d}_c(\mathbf{y}_i, \mathbf{z}_{\rho(i)})$. Now,

$$\begin{aligned} \tilde{D}_c(X,Y) &= \sum_{i=1}^n \tilde{d}_c(\mathbf{x}_i, \mathbf{y}_{\pi(i)}) && (5.125)\\ &\leq \sum_{i=1}^n \tilde{d}_c(\mathbf{x}_i, \mathbf{z}_{\sigma(i)}) + \tilde{d}_c(\mathbf{y}_{\tau(\sigma(i))}, \mathbf{z}_{\sigma(i)}) && (5.126)\\ &= \tilde{D}_c(X,Z) + \tilde{D}_c(Y,Z). && (5.127) \end{aligned}$$

The proof for the case where $l, n \leq m$ is almost identical. $\square$

Chapter 6

Multitarget Tracking: Applications

6.1 Active Sonar Data with Synthetic Target

A set of active sonar data has been made available by Thales Underwater Systems. The dataset contains no targets but does feature dense, locally structured clutter with amplitude information. The aim of this section will be to inject a simulated target into the real active sonar data in order to gain indicative results about the performance of the multitarget filters described in Chapters 4 and 5 in active sonar data.

In the dataset, for each time step there are around 1000 detections, each corresponding to a local maximum of signal amplitude in range-bearing, or (r, θ), space. Fig. 6.1 shows the detections for a single time step. Note that due to the confidential nature of the data displayed, the axis labelling has been suppressed for several of the figures in this section. It can be seen that the clutter is not uniformly distributed in the region of surveillance but is instead denser near the origin, which is in the middle of the bottom of the figure. Each detection has an associated amplitude. Fig. 6.2 shows an estimate of the probability density of the amplitudes in the sonar data calculated using kernel density estimation with Gaussian kernels using Silverman's rule of thumb to calculate the bandwidth [19]. It can be seen from fig. 6.2 that the distribution of the amplitudes has a fat tail.

The results presented aim to provide answers to the following questions:

- Is it possible to track targets through the densest clutter using the GM-PHD filter?
- How do the results compare to a "first attempt" ad hoc method?
- Can using the target amplitudes within the GM-PHD filter in the manner described in Section 5.3 improve performance for high/low SNR targets?

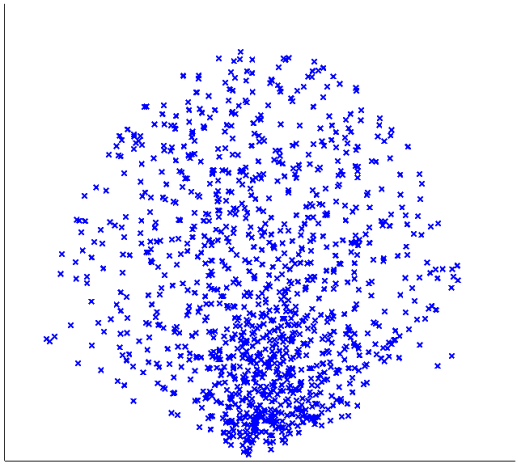


Figure 6.1: The locations of the detections in one time step of the active sonar data.

Simulation Details

As discussed in Section 5.3, for data with amplitude information a common technique is to threshold based on amplitude before applying a multitarget tracker. The choice of threshold will depend on the amplitude values expected for a true target. For a 'loud' target with high amplitude values it will be possible to choose a high threshold. This will eliminate many of the clutter detections without risking missed detections of the target. On the other hand, for a 'quiet' target with low amplitude values, a low threshold must be chosen in order to ensure detection of the target.

The amplitudes of the simulated targets will be restricted to being drawn from Rayleigh distributions with parameter σ, which is a standard simple model for target amplitude returns [66]. Various choices of the parameter σ will be used. The threshold will be selected to ensure that the probability of detecting the simulated target, p_D, is equal to 0.95. The trajectory, shown alongside combined clutter for all time steps is shown in fig. 6.3. The trajectory begins in an area of moderate clutter, enters the region of densest clutter and performs a change of course in the middle. The simulated target measurements of r and θ are made, these being corrupted with white Gaussian noise with magnitude equivalent to that reported in the active sonar data.

Tracking will be attempted using a standard GM-PHD filter as described in Section 4.2 (using the EKF implementation as the measurements are (r, θ)). A Gaussian mixture implementation of the SNR-PHD filter discussed in Section 5.3 will also be used. Finally, a global nearest neighbour filter with 'M out of N' track maintenance scheme as described in Section 3.1 will be employed as a dummy 'first attempt' solution to verify that the tracking scenarios are non-trivial.

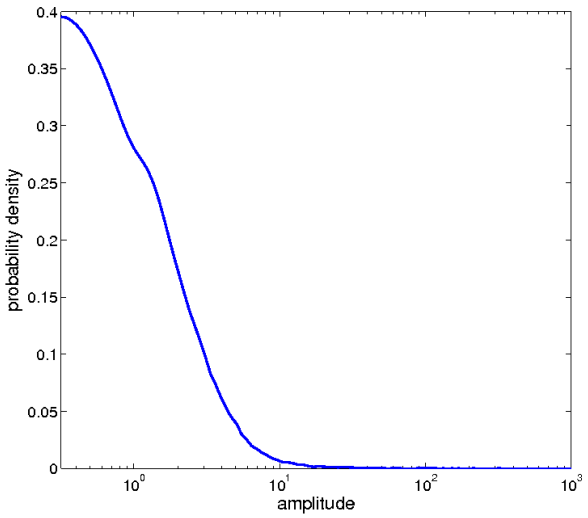


Figure 6.2: Probability density for the amplitudes in active sonar data

Amplitude Distributions

The values of σ used will be $\sigma = 150, 30, 10, 5$ referred to as 'very high', 'high', 'moderate' and 'low' SNR respectively. In order to give some meaning to these labels consider the standard definition of signal to noise ratio in the decibel scale, SNR_{dB} (see, for example [92]):

$$SNR_{dB} = 10\log_{10}\left(\frac{A_{signal}}{A_{noise}}\right)^2 = 20\log_{10}\left(\frac{A_{signal}}{A_{noise}}\right), \tag{6.1}$$

where A_{signal} and A_{noise} are the root mean squares of the amplitude of the signal (i.e. the simulated target) and the noise (i.e. the active sonar clutter) respectively. A_{noise} can be calculated directly from the data whereas A_{signal} can be estimated by generating a large number of realisations from $Rayl(\sigma)$. Table 6.1 shows the SNR_{dB} for each of the simulated targets. Note that the fat tail visible in fig. 6.2 skews A_{noise} upward in the root mean square. Thus, the 'adjusted SNR_{dB}' listed in Table 6.1 is calculated using only those amplitude values of the clutter below the 0.1 percentile of the corresponding Rayleigh distribution. Above this point, clutter points have amplitude *too high* to be target-related, which is counter-intuitive for a signal-to-noise ratio calculation, so ignoring them gives a more reasonable SNR_{dB} value. The adjusted values are presented here to avoid giving misleading values of SNR.

The discussion above raises an issue with the widespread use of SNR to characterise the difficulty of a tracking scenario: there are many cases where SNR gives a misleading characterisation of this difficulty. In the example here, a fat tail on the

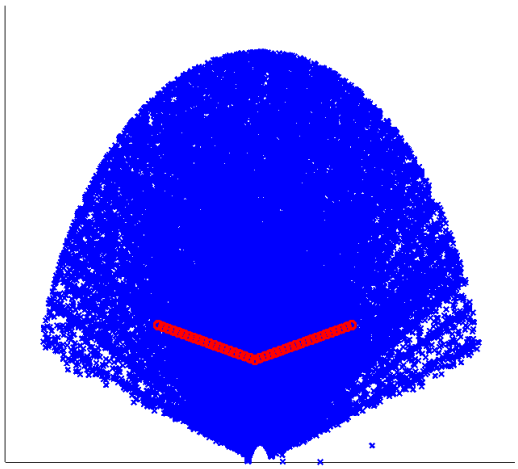


Figure 6.3: Simulated trajectory (red) overlayed onto active sonar clutter (blue).

noise distributions makes the raw SNR values misleading. Another example could be where the amplitude for a target is known with high confidence to within a small number of values so that most noise detections could be immediately rejected. The target in this case may not be 'high SNR' but the tracking scenario would be relatively easy.

σ	SNR_{dB}	Adjusted SNR_{dB}
150	22.4dB	24.9dB
30	8.4dB	15.6dB
10	-1.2dB	9.4dB
5	-7.2dB	5.0dB

Table 6.1: Signal to noise ratio (decibel scale) for the simulated targets.

Another relevant concept is the receiver operating characteristic, or ROC curve which plots the probability of detecting the target, p_D against the probability of declaring a false alarm, p_{FA} over the range of possible thresholds. Figure 6.4 illustrates the concept. The dotted black line is p_D against p_{FA} in the case where proportion p of the data points are chosen randomly for $p \in [0, 1]$. In this case $p_D = p$ and $p_{FA} = p$. As this method can almost always be improved upon, all ROC curves should lie above the dotted line. The closer to the 'perfect' data case, indicated by the dotted red line, in which all detections can be made without any risk of false alarms, the better. Figure 6.5 shows the ROC curve for each of the targets which illustrates the increasing difficulty of the scenarios.

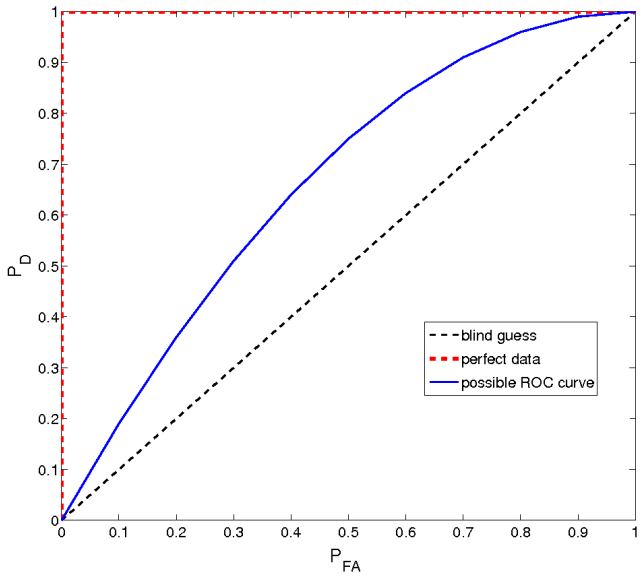


Figure 6.4: Example illustrating the concept of a ROC curve. The dotted black line corresponds to guessing blindly which detections to declare. The dotted red line is only possible where the data is 'perfect'. All other ROC curves must lie between these two.

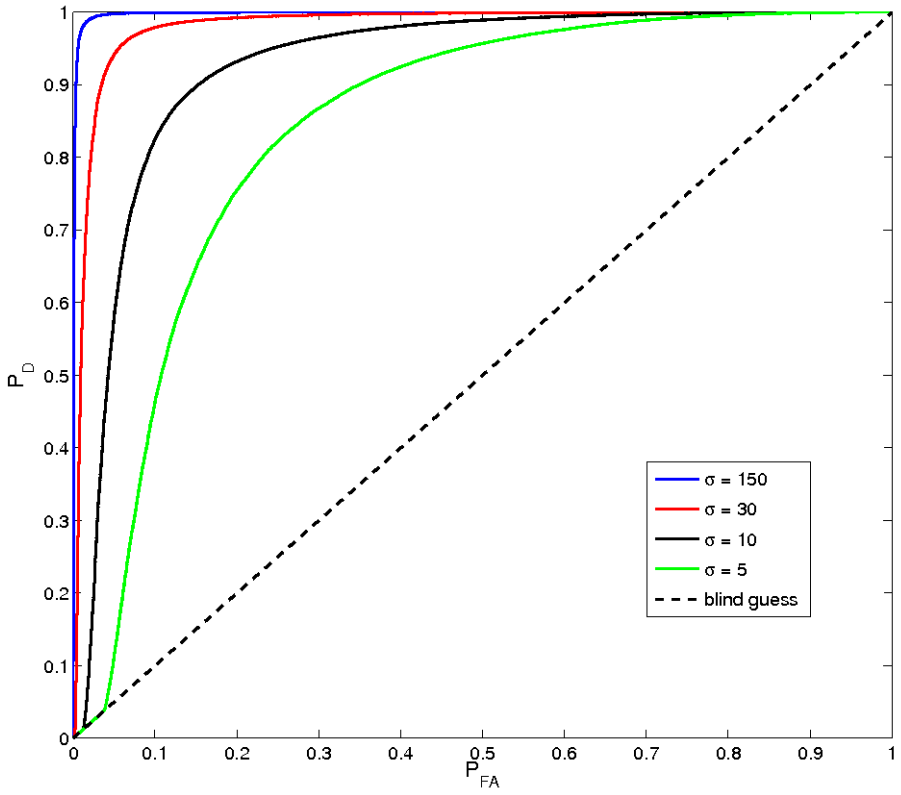


Figure 6.5: ROC curves for the four simulated targets.

Very High SNR ($\sigma = 150$)

Fig. 6.6 shows the output of the GNN, GM-PHD and SNR-GM-PHD filters along with all of the detections with amplitudes greater than the threshold. All of the filters are able to identify the target, though the GNN declares several false targets in the denser region of clutter. The GM-PHD filter also declares an additional track which the SNR-GM-PHD filter rules out as being a true target.

High SNR ($\sigma = 30$)

The results for the high SNR target are shown in fig. 6.7. It can be seen that even for this high SNR target, the level of false detections passing the threshold overwhelms the GNN filter. It can also be seen that the SNR-GM-PHD filter is able to declare the target earlier while declaring fewer false alarms.

Moderate SNR ($\sigma = 10$)

Fig. 6.8 shows the results for the moderate SNR target. Both the GM-PHD filter and SNR-GM-PHD filter are able to identify the target without declaring any false alarms, the SNR-GM-PHD filter doing so more quickly and consistently. The missing detections in the middle of the track, visible in fig. 6.8(b),(c), are due to the target falling below the declaration threshold within the (SNR)-GM-PHD filter. A realistic tracking system might wish to 'hold onto' established tracks and hence apply a more lenient declaration threshold to established high confidence tracks. Applying a simple version of such an approach to the SNR-GM-PHD filter permits the recovery of an uninterrupted track as shown in fig. 6.8(d) for no significant extra computational cost.

Low SNR ($\sigma = 5$)

Finally, fig. 6.9 shows the tracks for the low SNR target. The GM-PHD filter without amplitude information is unable to track targets with Rayleigh parameters much less than $\sigma = 10$ whereas SNR-GM-PHD is able to track down to $\sigma = 5$ although it takes 10 time steps to locate the target at this level. As for the moderate SNR level test, fig. 6.9(c) shows the effect of employing a simple track label dependent declaration criteria.

Conclusions

These tests demonstrate that using the SNR-GM-PHD filter permits the tracking of low SNR targets in real active sonar data without significant levels of false target declarations. There is a significant improvement in performance compared with not

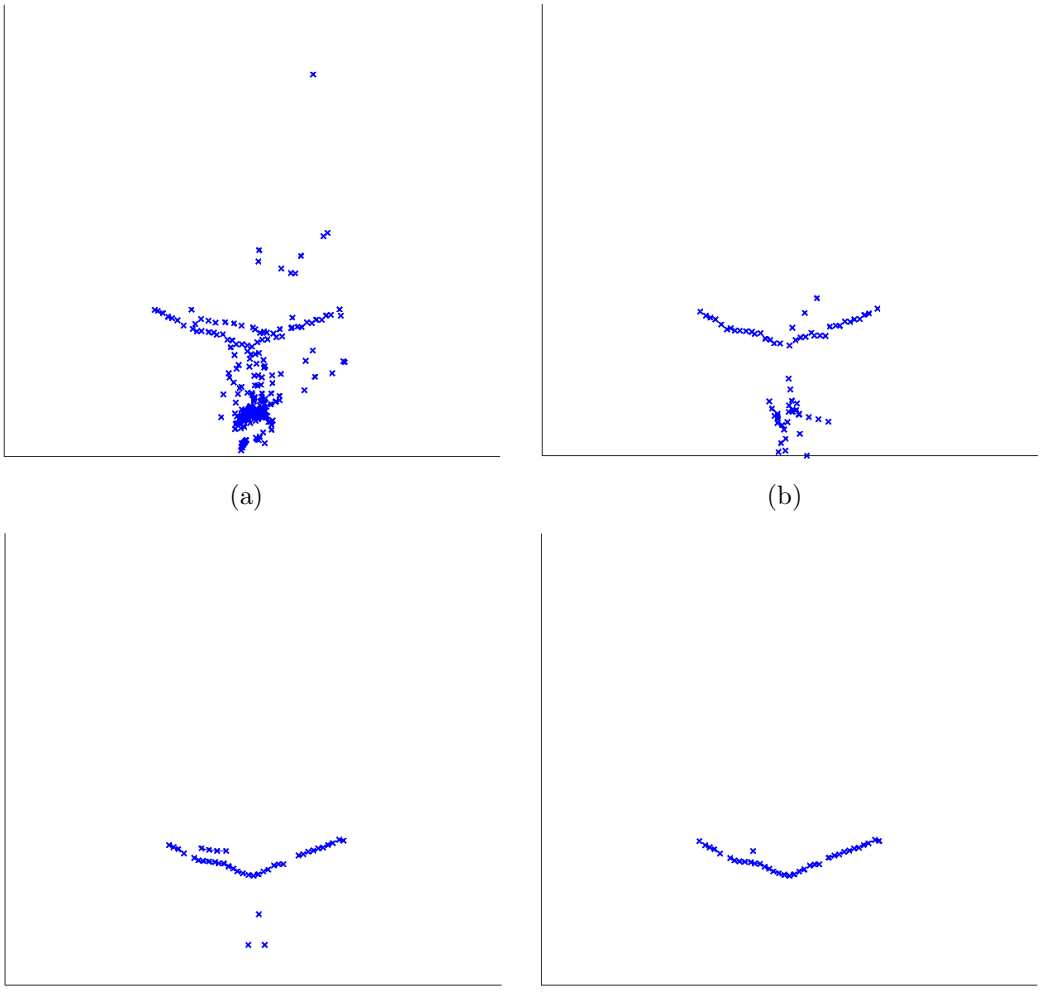


Figure 6.6: Results for tracking a very high SNR ($\sigma = 150$) target.
(a) All detections passing threshold.
(b) Tracks from GNN.
(c) Tracks from GM-PHD
(d) Tracks from SNR-GM-PHD

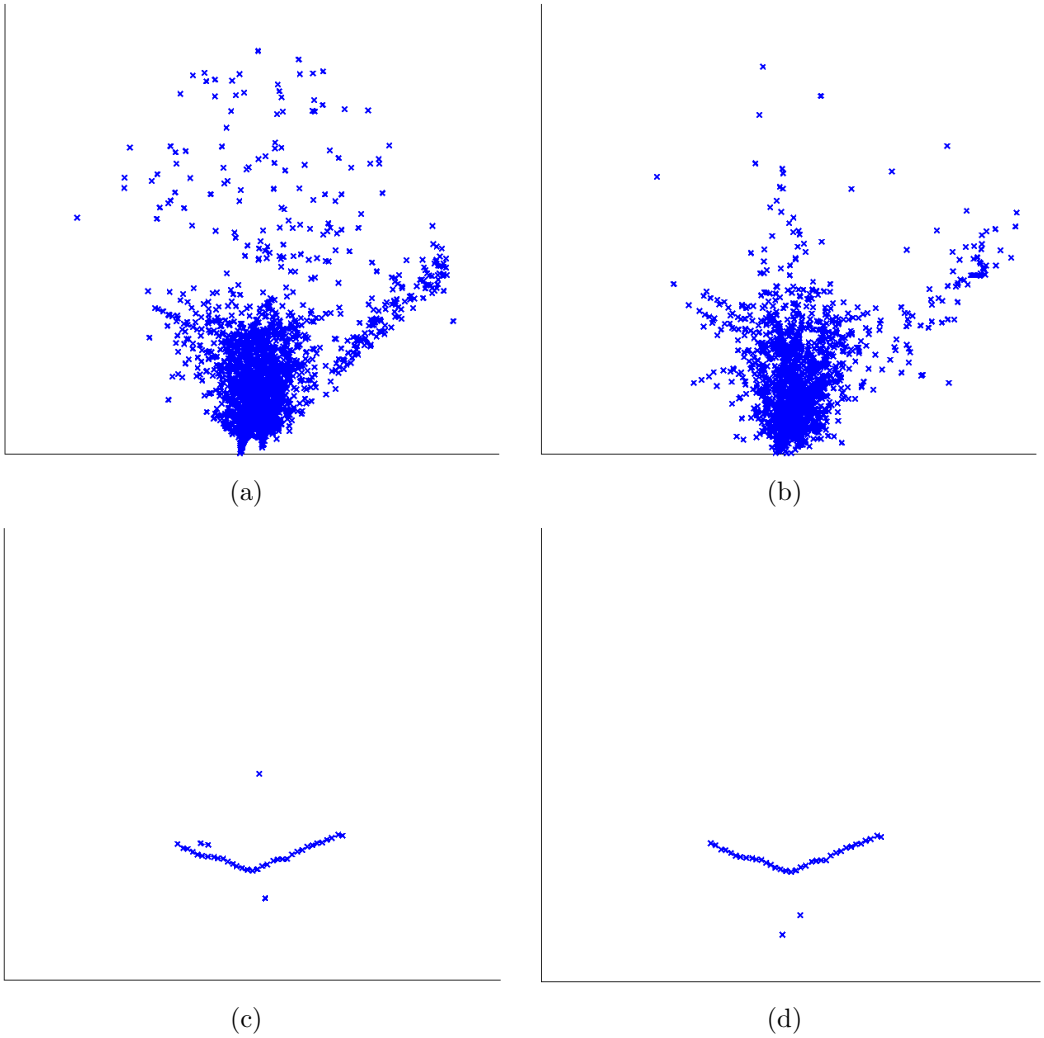


Figure 6.7: Results for tracking a high SNR ($\sigma = 30$) target.
(a) All detections passing threshold.
(b) Tracks from GNN.
(c) Tracks from GM-PHD
(d) Tracks from SNR-GM-PHD

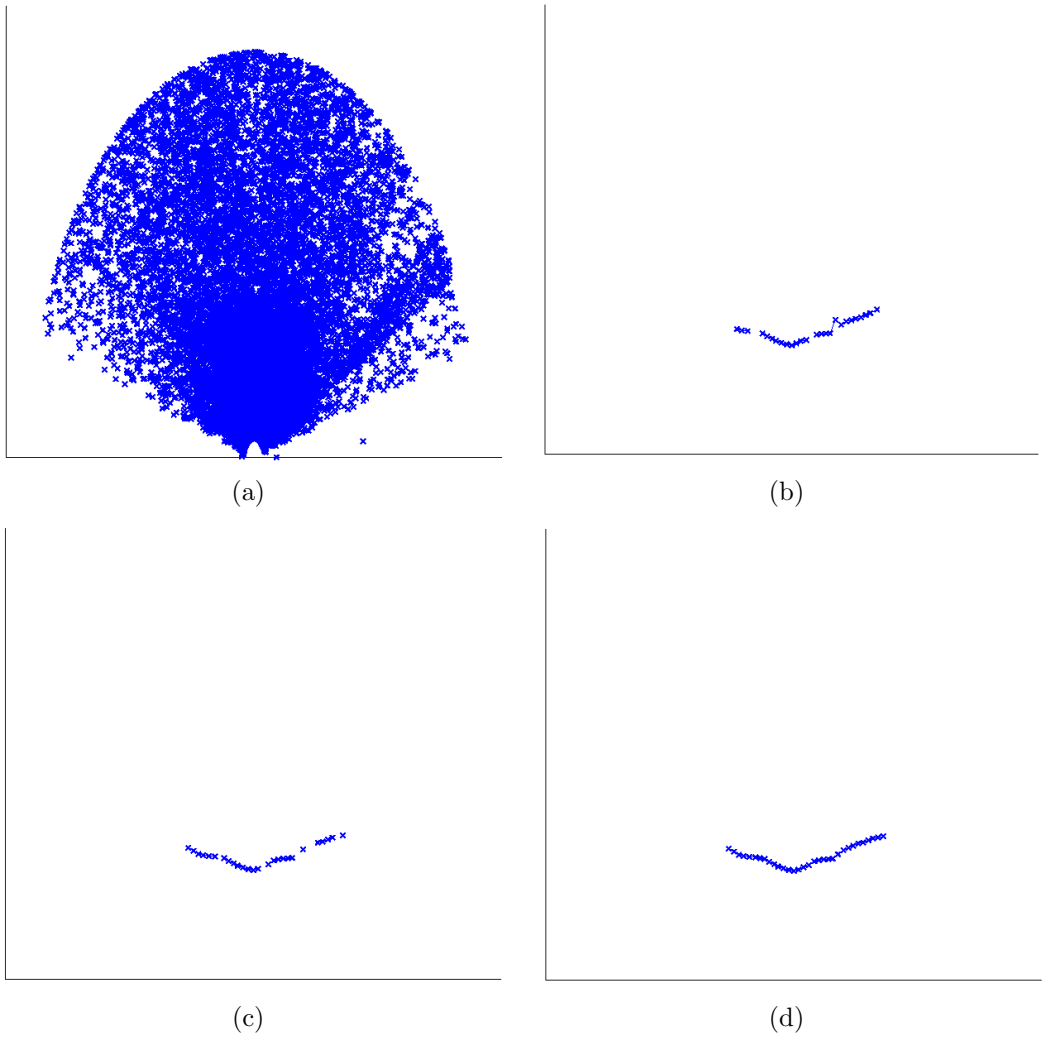


Figure 6.8: Results for tracking a moderate SNR ($\sigma = 10$) target.
(a) All detections passing threshold.
(b) Tracks from GM-PHD.
(c) Tracks from SNR-GM-PHD
(d) Tracks from SNR-GM-PHD with label dependent declaration criteria.

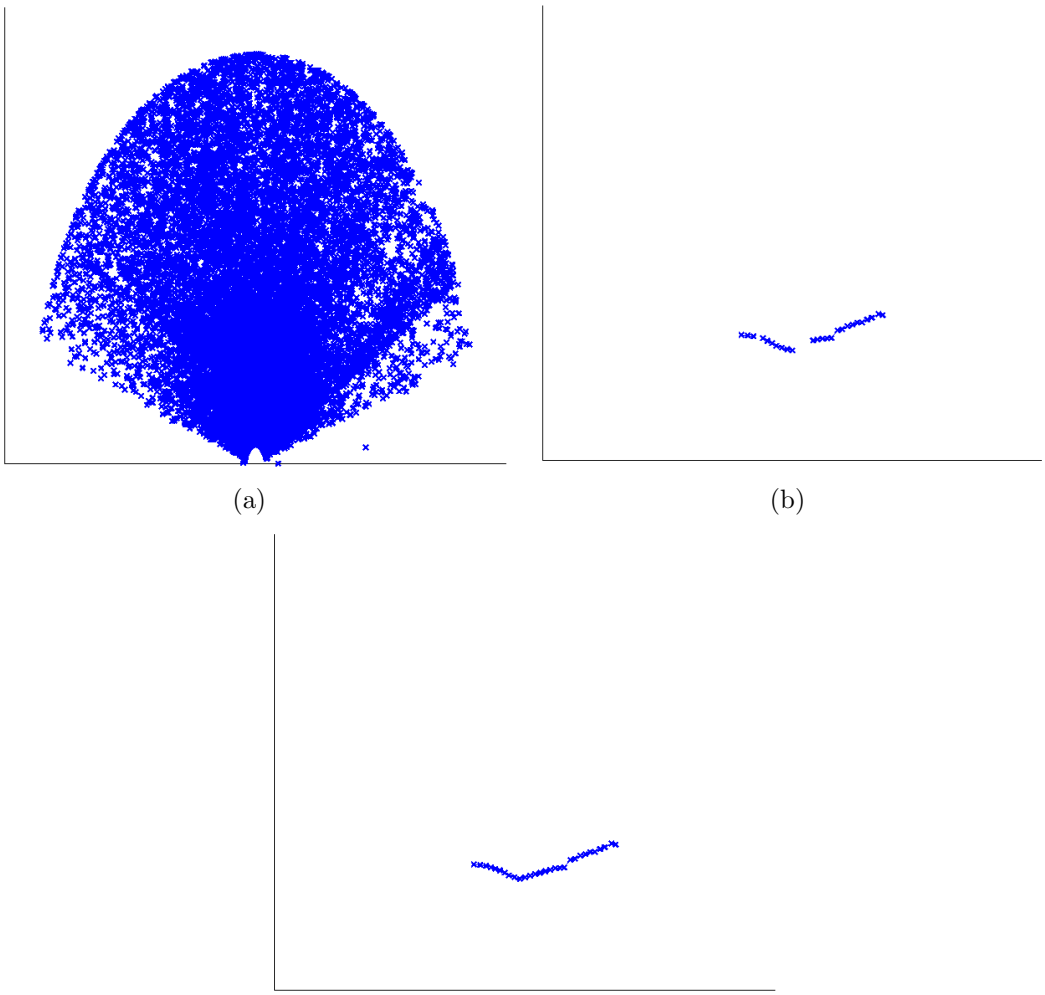


Figure 6.9: Results for tracking a low SNR ($\sigma = 5$) target.
(a) All detections passing threshold.
(b) Tracks from SNR-GM-PHD
(c) Tracks from SNR-GM-PHD with label dependent declaration criteria.

only the 'first attempt' solution but also with the standard GM-PHD filter. This provides evidence that taking SNR information into account directly in the multitarget filter could aid the tracking of quiet or stealthy targets. For an in-depth performance comparison of the GM-PHD filter with other tracking methods see Section 6.4.

Although the results presented here have been for a simulated target, comparable performance has been achieved for tracking in active sonar data with real targets.

6.2 Passive Sonar

Passive sonar involves listening to, without transmitting, acoustic signals in order to detect other vessels. The goal is to identify the audio spectrum of any vessels present (localisation will not be considered). This audio spectrum is comprised of important frequencies (or tones) which correspond to, for example, rotating machinery in the engine of the detected vessel. It is hoped that automatic detection and monitoring of the tones present in the signal will ease the workload of the operator and could potentially aid target classification.

Passive sonar data in its raw form is a sound wave. Throughout this section, the representation of this data in the time-frequency domain is considered. This representation of the data is achieved by taking a spectrogram (see, for example, [84, pp. 713-718]). This involves splitting the time-series up into smaller time windows and calculating the FFT for each window, and hence estimating the spectrum at each time step. This section will be an exposition of a new method for processing passive sonar data in the time-frequency domain. This new method was first presented in [135]. The author's contribution to this jointly authored paper is outlined in Section 1.4.1.

Firstly, a pre-processing step is introduced in order to denoise the data to enable easier detection of tones. The relevant techniques are the Radon transform and wavelet denoising.

The second step is the extraction and monitoring of tones present in the signal. The number of tones present together with their frequencies, which may not be constant, must be jointly determined from the set of potential detections returned from the data. The data may include some combination of: unpredictable tone shifts, appearance and disappearance of tones, missed detections of tones, false detections, imprecision in the frequency values detected for tones and crossing tones. Due to these factors, extraction and monitoring is non-trivial and is formulated here as a multitarget tracking problem in the frequency-frequency rate domain.

6.2.1 Pre-Processing

The aim of the pre-processing step is to denoise the data to allow accurate extraction of potential detected tones. This process is aided by two linear transformations, the Radon and wavelet transforms, motivated by the idea that separation of signal from noise will be easier in the transform domain. The denoising process consists of transforming the data, denoising the data in the transform domain and then inverting the transform. An optimal solution to this filtering problem would integrate the detection of tones with the Bayes filter using appropriate models for noise and signal characteristics rather than exploit these in a separate pre-processing step. However, the use of a heuristic such as a denoising step will not always degrade results and may be valuable in reducing the required computation.

Radon Transform

Let f be a continuous function vanishing outside of a disc in $\mathbb{R}^2$. The Radon transform [68, pp.42-45], denoted Rf, and defined on the set of lines in $\mathbb{R}^2$, is given by:

$$Rf(L) = \int_L f(x)\mathrm{d}\sigma(x) \tag{6.2}$$

where the integration is performed with respect to the arclength measure $d\sigma(x)$ on the line L. Stated concisely, the Radon transform of a function evaluated on a particular line is the integral of the function along that line. Thus, the Radon transform has large values for lines along which the function has high values. Put differently, the Radon transform has high values for lines which are prominent in the data. Applying a Radon transform to the data exploits the fact that tones are generally approximately straight lines in frequency-time space. As a result, the signal energy will be more concentrated in the Radon transform domain whereas noise should not be, making separation of signal and noise easier. Some examples are presented below.

It should be noted that previous work by Sun and Willett [114] considered use of the Hough transform, which is closely related to the Radon transform, for detecting tones in passive data. The key difference is that the Radon transform permits use of amplitude information whereas the Hough transform is discrete and forces thresholding first. Use of the Radon transform might be considered a natural extension of the previous work using the Hough transform.

Wavelet Transform

A wavelet basis is chosen for the data, with the basis divided into levels representing different scales of detail. Elements of the wavelet basis which correspond to smaller scale detail are then subject to thresholding. This is analogous to expressing a function in terms of its Fourier series and then thresholding the higher frequency coefficients to remove high frequency noise. The wavelet basis used is 'sym4'. The method for choosing threshold levels is given in [35]. 'A Wavelet Tour of Signal Processing' [77] is a good general purpose reference book for this subject.

Examples of Pre-Processing for Simple Simulated Data

Two examples will be presented; the first demonstrates the advantage of employing the Radon transform and wavelet denoising (Radon+wavelets) as compared to only using wavelet denoising. The second example shows that the Radon transform together with a less sophisticated thresholding method will run into problems.

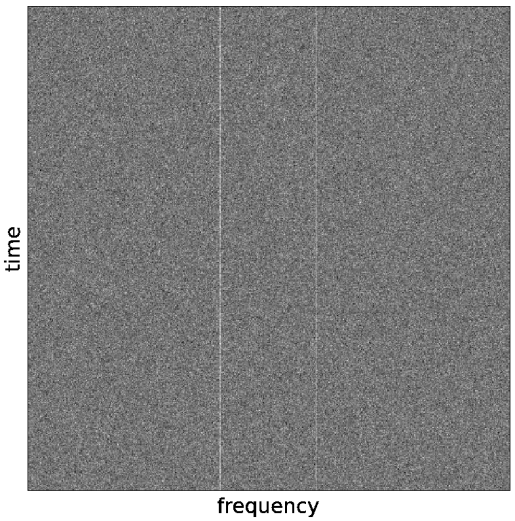


Figure 6.10: Raw simulated passive sonar data with two constant tones plus noise

For the first example, fig. 6.10 shows two constant tones in the presence of additive white Gaussian noise. This data is denoised in two ways; firstly, using the Radon transform followed by wavelet denoising (and finally an inverse Radon transform) and secondly, using only wavelet denoising.

Fig. 6.11 shows the Radon transform of the raw data. The two lines visible in the raw data are now represented by two points in the radon transform domain with angle of projection 0/180 degrees due to the vertical slopes of the lines in the frequency time domain. Viewing the data in both the standard and Radon domains (as in fig.

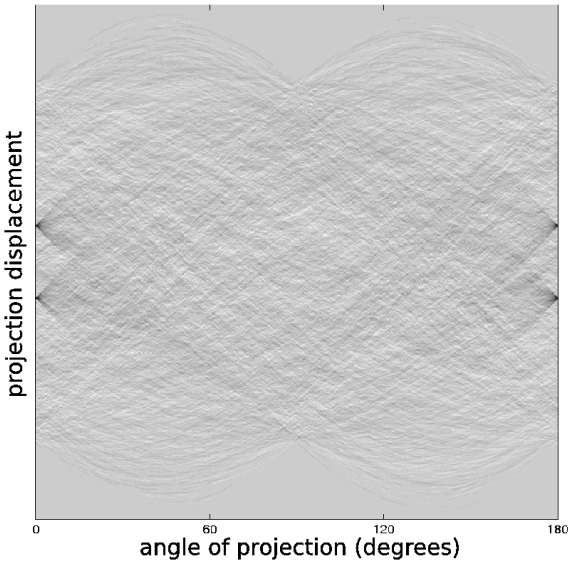


Figure 6.11: Radon transform of data shown in fig. 6.10

6.10 and fig. 6.11 respectively), it becomes clear that separating signal from noise might be easier in the latter domain.

In this example the signal to noise ratio (SNR) in the decibel scale, as defined in equation (6.1), for the original data was 2.8dB. After processing using only wavelets this had increased to 4.4dB, but after processing with Radon+wavelets there was a more substantial increase to 15.4dB. This simple example demonstrates the benefit of using both Radon and wavelet transforms.

A more complicated example is shown in fig. 6.12. Here, there are four tones; two of them shifting, one constant and one unstable (this is the wavering tone third from the left). In this example, we compare the results of Radon+wavelets against a method using the Radon transform with naive thresholding, rather than with wavelet thresholding, in order to ascertain if wavelets are necessary at all.

Fig. 6.13 shows the output of the Radon transform with naive thresholding. This method is not able to pick up some wrinkles of important detail, such as the briefly present tone on the far right. Furthermore, if a line is present for part of the image, the Radon transform with naive thresholding tends to declare this line for the whole image, resulting in artefacts. By contrast, fig. 6.14 shows the output of Radon+wavelets in which there is a perceptible improvement in the image with respect to the contrast of the tones against the noise.

From these preliminary results, we expect the best pre-processing to be due to the use of both Radon and wavelet transforms. This claim will be tested on less simplistic

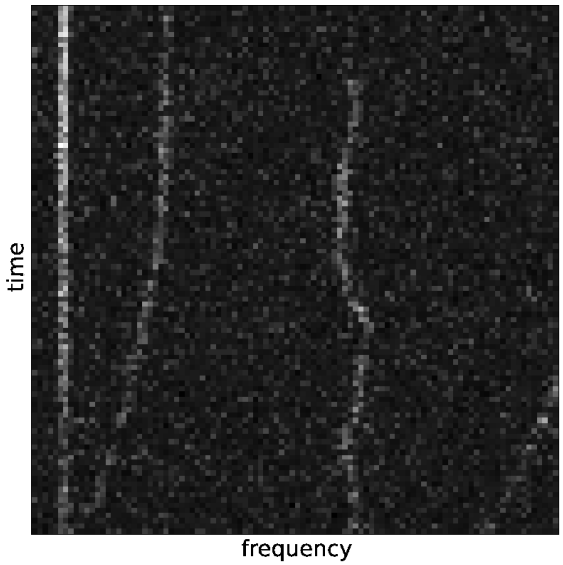


Figure 6.12: Raw data with four important frequencies plus noise

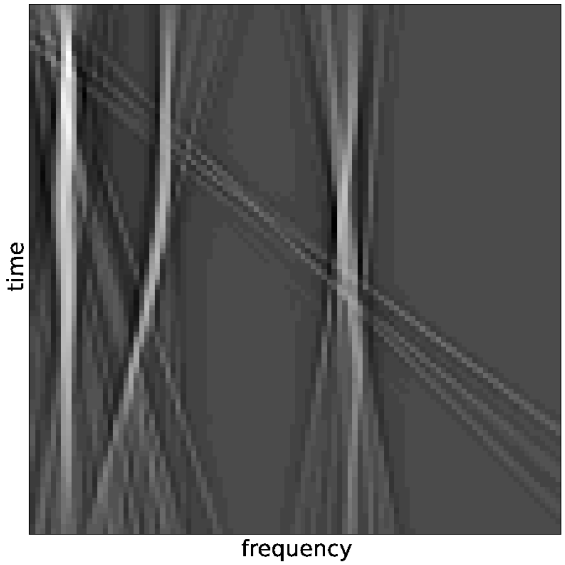


Figure 6.13: Data from Fig. 6.12 after Radon transform and naive thresholding

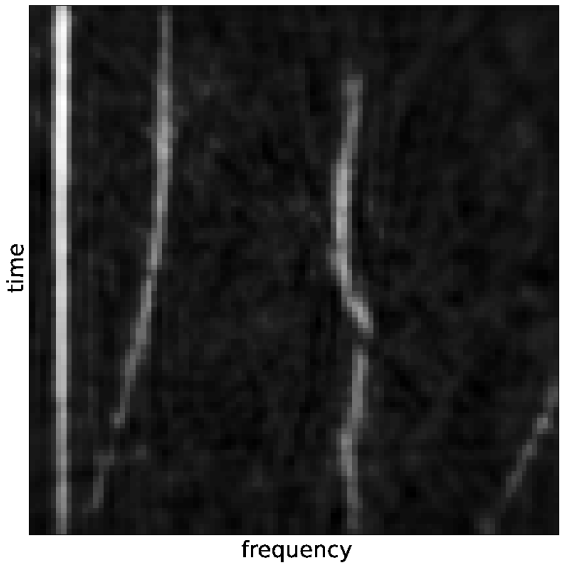


Figure 6.14: Data from Fig. 6.12 after Radon and wavelet denoising

data in Section 6.2.3.

6.2.2 Tone Tracking

After pre-processing the data, the aim is to establish, at each time step, how many tones are present in the data and the frequency of each tone. Associations between frequencies at different time steps are also required to enable, for example, detection of Doppler shifts.

Potential detections for tones are extracted from the processed data. Despite the denoising step, there are still likely to be some imperfections. Some features of the problem include:

- Uncertain dynamics of true tones (as frequencies may unexpectedly shift due to Doppler or engine effects).
- Possible appearance or disappearance of tones as vessels come into or out of range.
- Missed detections of true tones due to imperfection of data or other factors, such as the Lloyd's mirror effect which is caused by constructive and destructive interference between sound waves propagating by the direct path and reflected paths.
- False detections (possibly many more of these than detections of true tones).

- Imprecise frequency measurements for the detected tones (due to imprecision in equipment, diffuse tones, problems with Fourier transforms on noisy signal).

The problem is formulated as a multitarget tracking problem using the framework of FISST as presented in Chapter 3. Let X_k be the set of tones present at time step k, Z_k be the set of detections at that time step. Thus,

$$\begin{aligned} X_k &= \{\mathbf{x}_k^{(1)}, ..., \mathbf{x}_k^{(M_k)}\}, && (6.3) \\ Z_k &= \{z_k^{(1)}, ..., z_k^{(N_k)}\}, && (6.4) \\ \mathbf{x}_k^{(j)} &= (f_k^{(j)}, \dot{f}_k^{(j)})^T, && (6.5) \end{aligned}$$

where $f_k^{(j)}$ and $\dot{f}_k^{(j)}$ are the frequency and frequency rate of tone $\mathbf{x}_k^{(j)}$ and M_k, N_k are the number of tones and detections at time k respectively.

The forward and measurement models for the random finite sets X_k and Z_k follow (3.11) and (3.12) respectively. The following further assumptions are made in order to permit use of the GM-PHD filter:

- The probability density for the state $\mathbf{x}$ at time $k+1$ for a target which had state $\mathbf{x}'$ at time k is

$$f_{k+1|k}(\mathbf{x}|\mathbf{x}') = \mathcal{N}(\mathbf{x}; F_k\mathbf{x}', Q_k), \tag{6.6}$$

  where

$$F_k(\mathbf{x}) = \begin{bmatrix} 1 & dt \\ 0 & 1 \end{bmatrix}, \quad Q_k = \begin{bmatrix} \frac{1}{3}dt^3 & \frac{1}{2}dt^2 \\ \frac{1}{2}dt^2 & 1 \end{bmatrix}, \tag{6.7}$$

  and dt is the time step in the data. These represent "straight line motion" and "white noise" respectively as outlined in the single target case in Section 2.2.

- The probability of target survival, p_S is known and is constant.

- There is no spawning (i.e. tones cannot split or give rise to new tones).

- The number of birth targets is Poisson distributed with mean p_B and the target birth density is constant with an appropriate Gaussian sum approximation.

- A target with state $\mathbf{x}$ generates a measurement $\mathbf{z}$ with likelihood

$$L_{\mathbf{z}}(\mathbf{x}) = \mathcal{N}(\mathbf{z}; H_k\mathbf{x}, \sigma^2), \tag{6.8}$$

  where

$$H_k = [1 \quad 0], \tag{6.9}$$

  and σ is the standard deviation of frequency measurement error.

- The probability of detecting a target, p_D, is known and is constant.
- The number of false alarms is Poisson distributed with mean λ and the false alarm density, κ is constant.

The advantage of using multitarget tracking methods in this context is that it enables incorporation of available information about the data in the form of explicit statistical models for all of the relevant phenomena. Tracking methods are robust to noise in the data and the set of tracks returned is a concise expression of the important information contained in the original data. As an example, fig. 6.15 shows the output of the GM-PHD filter for the data from fig. 6.12.

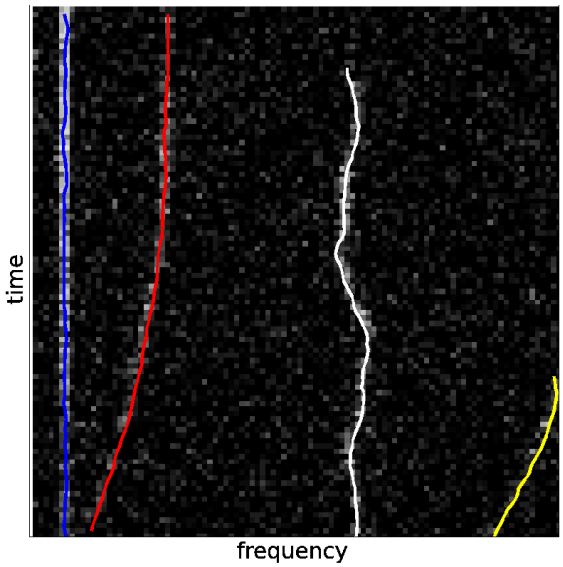


Figure 6.15: Frequency tracks extracted from the data shown in fig. 6.12 using the GM-PHD filter.

6.2.3 Results

The new method was tested on two sets of data. The first is a test scenario designed for the performance assessment of existing algorithms made available by Thales Underwater Systems. The scenario includes many aspects which might be present in real passive sonar data including:

- Doppler and non-Doppler related frequency shifts.
- Crossing tones.

- Intermittently fading contacts.
- Unstable tones.
- Diffuse tones.

The scenario is based on a target trajectory hence the Doppler shifts will be representative of an actual target motion. The crossing of tones is achieved by ensuring that a tone with a non-Doppler related shift crosses a tone which is Doppler related. For example, speed changes will cause an engine related tone to undergo significant changes in frequency. Such frequency changes could cause some of the tones from the same vessel, but from a different source within that vessel, to cross.

The raw data is shown in fig. 6.16 and the data after pre-processing is shown in fig. 6.17 for comparison. Tones that are visible but varying in strength in the raw data become more consistent after pre-processing, while other tones that were difficult to detect by sight become more pronounced. It can also be seen that noise level has generally been reduced though the Lloyd's mirror effect in the top left corner of the image is still visible.

The second set of data is taken for a real passive sonar recording of a Diesel engine. The raw data is shown in fig. 6.18 and the pre-processed data in fig. 6.19. In contrast with the first example, the noise in this data is less uniform making this a more realistic test of the method. Fig. 6.19 shows that the strong noise on the right hand side of the image is not eliminated by pre-processing. However, overall the denoising has been successful and many tones not visible in the raw data become apparent despite the remaining noise.

Effect of Pre-Processing Step

As discussed in Section 6.2.1, the process of extracting detections of tones from the data may be expected to lead to false detections as well as missed detections of real tones. The levels of both false detections and missed detections are monitored as a test of the effectiveness of the pre-processing techniques. For the test, a subset of the data from fig. 6.16 was chosen, and the true tones present were identified manually. Detections were extracted using various thresholds and the proportion of true tones correctly reported was recorded along with the level of false detections for each threshold. This test was repeated for three cases: i) raw data, ii) wavelet thresholding and iii) Radon+wavelet thresholding. The results were used to measure the trade off between the probability of missing true tones and the probability of detecting false

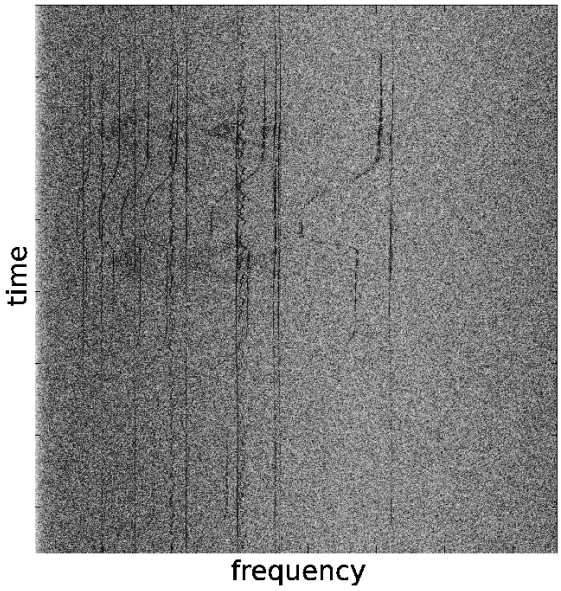


Figure 6.16: Simulated Test Scenario

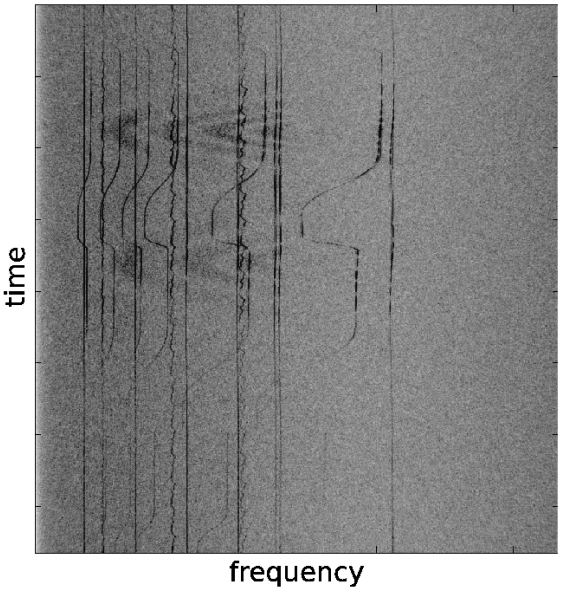


Figure 6.17: Simulated Test Scenario after pre-processing

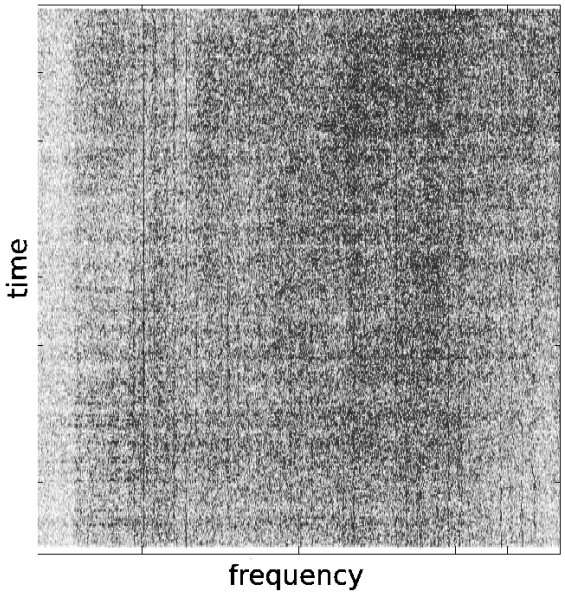


Figure 6.18: Diesel engine passive data

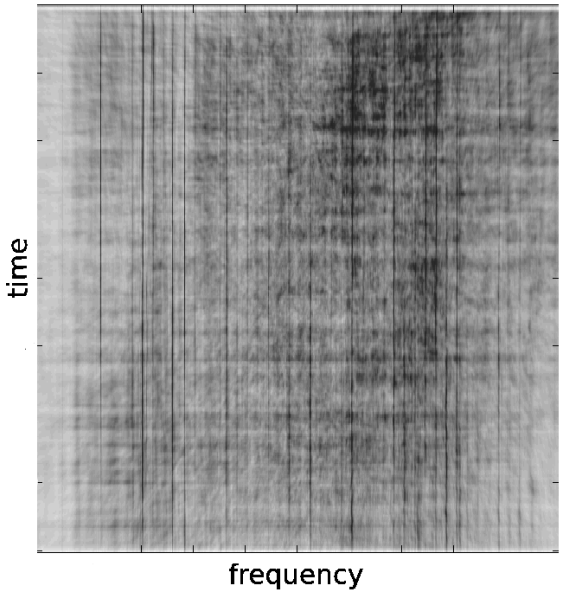


Figure 6.19: Diesel engine data after pre-processing

ones in each case. This trade off between probability of detection and probability of false alarm is closely linked to the idea of receiver operating characteristics (ROC) as described in more detail in Section 6.1.

Fig. 6.20 shows this trade-off and is analogous to a ROC curve. It shows that a significant improvement in the level of false detections for any probability of detection required can be achieved by using Radon+wavelet pre-processing. The results also confirm the superiority of Radon+wavelets as opposed to wavelets alone.

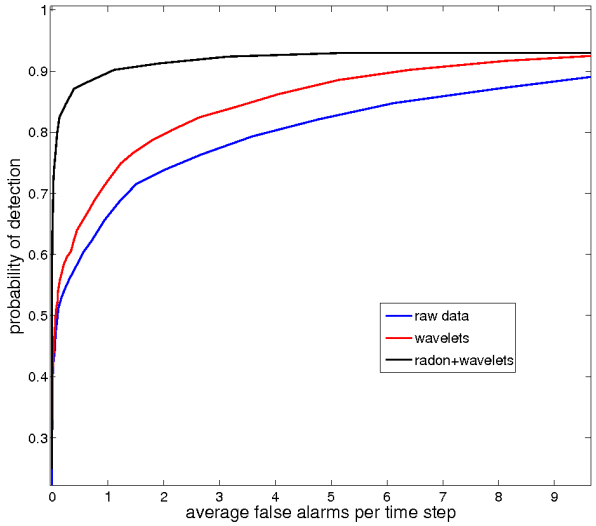


Figure 6.20: Average level of false alarms vs probability of detection for various pre-processing methods for a subset of the data from fig. 6.16

Tracking step

Figs. 6.21 and 6.22 show the output of the tracking process for the simulated and real data respectively.

For the simulated data, all 16 tones known to be present are detected and largely maintained despite extended sections of fading and crossovers. However, the output is not perfect and there are some sections where fading combined with crossovers has resulted in mistaken track association.

For the real data, 21 tones are identified and tracks maintained successfully on all of them. As the data is real, the 'ground truth' is not known but the tracks reported all correspond to peaks in the FFT of the raw data (a potential check only in this special case in which the frequencies appear constant for the whole time period). Furthermore, a close inspection of the data confirms the tones and the periods for which they are declared are plausible.

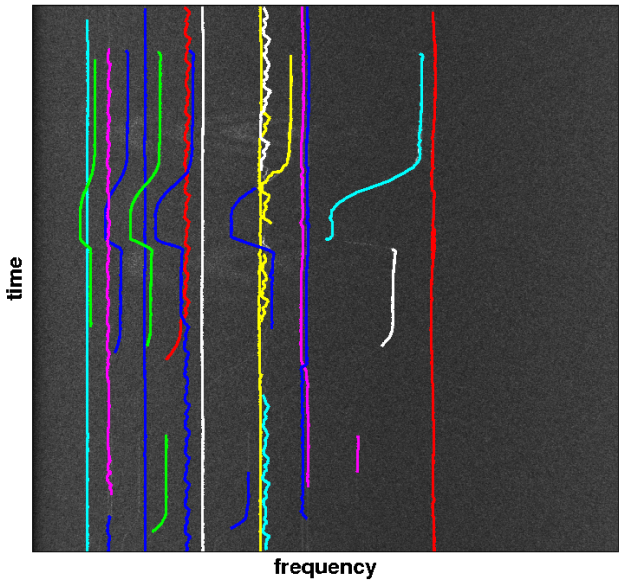


Figure 6.21: Tracker output for the simulated data

In summary, a new method for processing passive sonar data has been presented. An initial implementation on both real data and a simulated complex scenario shows promising results. The Radon and wavelet transforms as well as the tracking techniques from the theory of finite set statistics are potentially valuable tools for the processing of passive sonar data. Future work comparing the new method to existing methods for frequency detection in passive sonar data would be valuable. A rigorous linking between passive and active sonar with both using finite set statistical methods would also be of interest.

Use of the Radon transform exploits the fact that tones tend to move in straight lines in frequency-time space. Future implementations could improve performance by using more general information about the behaviour of tones, incorporated by using a generalised form of the Radon transform.

Using this method, phenomena, such as Doppler shifts, can be automatically detected. A natural extension to this work would be to link this information back to the tracking step, so that detection of Doppler shifts can be used to inform beliefs about the target's dynamics and vice versa.

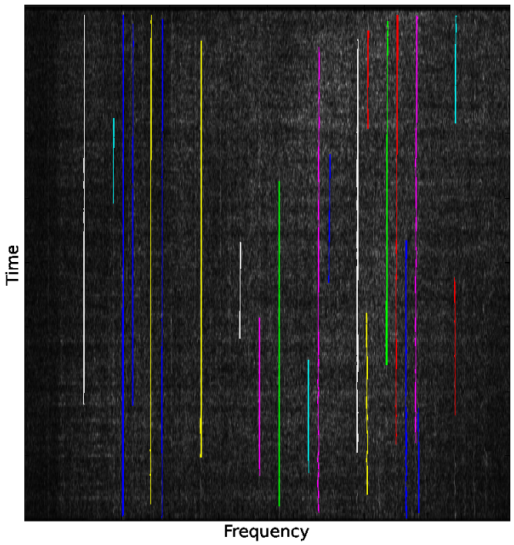


Figure 6.22: Tracker output for the diesel data

6.3 Tracking Multiple Bacteria in Microscopic Video Data

6.3.1 Context

Bacteria such as *Rhodobacter sphaeroides* are capable of constructing and living within surface associated multicellular communities known as biofilms. The formation of biofilms has huge implications for medical and industrial environments. Biofilms have been found to cause chronic infections, to grow on medical implants and to form plaque on teeth. In industry they have been found to block pipes, corrode metal, and to contaminate clean water distribution systems. For these reasons, bacterial behaviour near surfaces is a topic of current research interest [60] [109] [120].

Bacteria were first successfully tracked in microscopic visual data in 1971 by Berg [10] using a specially designed tracking microscope which adjusted its position and focus in real time to follow bacterial movement. Berg poetically described the results: "*The scene through the binocular is extraordinary. The bacterium being tracked seems to be stuck to the centre of the field, turning this way and that trying to free itself, while the other bacteria drift in and out of focus, then to and fro, in apparent synchrony*". In the same passage, Berg also describes a telling incident: "*the organism was seen to collide with one which was not swimming; the microscope locked onto this bacterium and the other swam out of focus*". This highlights the difficulty of multi-target tracking as opposed to single target tracking and is the reason why multitarget

tracking methods are necessary for the problem. In Berg and Brown's widely-cited paper [11] presenting results from the tracking process, 59 trajectories are extracted from over 25 minutes of tracking time.

More recently, some analysis of bacterial behaviour has been carried out for 'tethered cells' [21], where cells are stuck to a microscopic slide cover by their flagella. This provides some information about characteristics of the cell motor but is an unnatural state in which to observe bacteria, which may influence results. Furthermore, parameters such as speed, distance and turn rate cannot be measured using this method.

Taboada et al. [116] appreciate that "*quantification of the different features of bacterial swimming requires the analysis of a large number of free-swimming cells*". In [116], Taboada et al. use a basic digital tracking method relying on cell overlaps between frames for track continuity. This method will encounter difficulties for crossing cell paths, just as Berg did, and as cells often move more than 200 times their own body length per second [3] a high frame rate is required. Consequently, Taboada et al. [116] are limited to obtaining 50 tracks.

In this section, an implementation of the PHD filter for tracking cells in video data is presented in the context of existing work on PHD filters for visual tracking. This method makes it possible to track a far larger number of cell trajectories than was previously possible. The large number of cell trajectories permits some analysis of cell behaviour which matches predictions from fluid dynamical models for bacterial motion. The work in this section is based on [137]. The author's contribution to this jointly authored paper is outlined in Section 1.4.1.

It should be noted that the application of the PHD filter to microscopic bacterial data has been attempted before by Juang et al. [54]. However, the primary purpose for their implementation was for monitoring cell lineage. Problems were found in the course of the implementation with regard to consistently tracking moving cells. Juang et al. state, for example, that "cells that moved too quickly became untracked". The implementation that will be described below focuses on maintaining continuous tracks for cells in motion.

6.3.2 Tracking

Prior Work

Multitarget tracking for general visual data using the PHD filter has been presented by Maggio et al. [71] and Wang et al. [129], both of which show good results on real data. Both [71] and [129] use an SMC-PHD implementation closely following the

approach in [124] but this introduces some additional complexities into the tracking process:

1. *Clustering for state extraction* - In order to extract individual target states, particles must be partitioned using a clustering method. For this purpose, [71] and [129] choose the expectation maximisation and k-means algorithms respectively. Problems that arise in clustering for state extraction are discussed in detail in Section 5.1.1.

2. *Data Association* - Associating tracks between frames in order to give continuous tracks may be important, and is particularly so with regards to the tracking of bacterial motion. In [71], data association is performed externally to the PHD filter using a graph based method. The graph based method used does not permit clutter and hence may produce unreliable results in tracking scenarios with clutter.

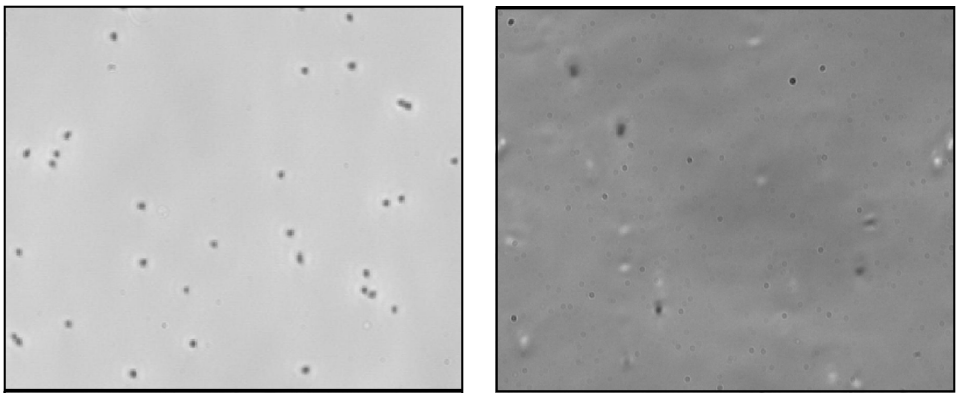
3. *Birth process* - In scenarios with no prior localisation information on the birth process, an unmanageable number of particles may be required for the birth process, as discussed in Section 5.1.1. (This process is avoided in an ad hoc manner in [71] by only sampling particles in an arbitrary vicinity of measurements).

The above issues can be resolved by using the GM-PHD filter (as described in Section 4.2.2) in situation where the motion and measurement models are linear and Gaussian or using the improved SMC-PHD filter (as described in Section 5.1.2) otherwise.

The Bacteria Tracking Scenario

In order to record bacterial motion, cells were placed within a flat glass capillary which was sealed at both ends using silicone grease. Swimming was visualised using phase contrast microscopy at 40x magnification and videos were captured at 50 frames per second (fps). The species *Rhodobacter sphaeroides* was chosen as it is of research interest [91]. Videos were taken at, and away from, the surface of the capillary as differences in swimming behaviour of the bacteria between these regions are of particular interest with regard to biofilm formation.

Figure 6.23 shows frames from two different kinds of microscopic video. Figure 6.23(a) is taken at the surface of the capillary. The cells are quite clear, so that occlusion is the issue most likely to cause a missed detection of a cell. Figure 6.23(b) is taken away from the surface and deeper within the fluid. In this case the cells are more faint and difficult to detect, some are darker than the background while



(a) (b)

Figure 6.23: a) Frame from microscopic video taken at the surface. b) Frame from microscopic video taken away from the surface. ©IEEE 2012

others are lighter due to differences in depth. To the author's knowledge, it has not previously been possible to track multiple bacteria in this scenario.

Linear, Gaussian models are sufficient to capture the target dynamics and measurement models and only positions and velocities of individual cells are of interest. Hence single target states $\mathbf{x}_k$ have the form

$$\mathbf{x}_k \quad = \quad [x_k, \dot{x}_k, y_k, \dot{y}_k]^T, \tag{6.10}$$

and the following standard assumptions are made:

- The probability density for the state $\mathbf{x}$ at time $k+1$ for a target which had state $\mathbf{x}'$ at time k is

$$f_{k+1|k}(\mathbf{x}|\mathbf{x}') = \mathcal{N}(\mathbf{x}; F_k\mathbf{x}', Q_k), \tag{6.11}$$

  where

$$F_k(\mathbf{x}) = \begin{bmatrix} 1 & dt & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & dt \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad Q_k = q \begin{bmatrix} \frac{dt^3}{3} & \frac{dt^2}{2} & 0 & 0 \\ \frac{dt^2}{2} & dt & 0 & 0 \\ 0 & 0 & \frac{dt^3}{3} & \frac{dt^2}{2} \\ 0 & 0 & \frac{dt^2}{2} & dt \end{bmatrix}, \tag{6.12}$$

  where dt is the time step in the data and q is the process noise scaling parameter. These represent "straight line motion" and "white noise" respectively as outlined in the single target case in Section 2.2.

- The probability of target survival, p_S is known and constant.

- There is no spawning (i.e. cell division happens too infrequently to be important).

- The number of appearing targets is Poisson distributed with mean p_B. The birth density, γ_k, will depend on the scenario but should be chosen to localise around the edges of the image in the case, as in Fig. 6.23(a), where the cells are always clearly visible and must emerge from the sides. In the case, as in Fig. 6.23(b), where cells may move into the field of view by changing depth the birth density must cover the whole image.

- A target with state $\mathbf{x}$ generates a measurement $\mathbf{z}$ with likelihood

$$L_{\mathbf{z}}(\mathbf{x}) = \mathcal{N}(\mathbf{z}; H_k\mathbf{x}, R), \tag{6.13}$$

  where

$$H_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad R_k = \sigma_w^2 \mathbb{I}_2, \tag{6.14}$$

  and σ_w is the standard deviation measurement noise.

- The probability of detecting a target, p_D is known and is constant.

- The number of false alarms is Poisson distributed with mean λ and the false alarm density, κ is constant.

Finally, it is well-known that some cells, including *Rhodobacter sphaeroides*, have a run-stop behaviour. This might be modelled as a jump Markov linear system, meaning that there are two different motion modes and switching between these two is a Markov process. Handling jump Markov linear systems within the PHD filter is presented in Section 5.2. This method may be used to improve handling of rapid changes in behaviour while tracking but is omitted here for simplicity as the bacterial strain used does not display stopping behaviour.

Handling Occlusions

Occlusions are an additional issue in video data containing a high concentration of cells. In such high cell concentrations, there are frequent occasions when one cell occludes another for several consecutive time steps and this often leads to the temporary loss of one of the tracks, breaking the desired track continuity. Two heuristic options for alleviating this problem are suggested in lieu of a comprehensive solution.

1. Set the model probability of detection to a lower level for cells in the vicinity of other cells, thereby taking into account the probability of occlusion.

2. "Join up" broken tracks by seeking pairs of tracks where one track ends and another begins close by a short number of time steps later, especially if the birth target is in an area with low birth density.

6.3.3 Testing

Testing the output of the tracker is a difficult task as no 'ground truth' is available. A comparison of the tracker output with a manual analysis was carried out on a selection of tracks. This process is difficult and time consuming. Taboada et al. [116] go as far as to suggest that "simple eye analysis would be extremely difficult or impossible". However, there is some merit in confirming correspondence between the tracker output and the tracks as identified by the human eye. In addition, a simulation was undertaken to test the performance of the tracker in the presence of a known ground truth.

Manual Analysis

100 tracks were randomly selected, 50 of which were from videos captured at the surface and 50 of which were captured away from the surface. A careful manual analysis was performed comparing the track output with the video in slow motion. Occlusions and faint tracks made identifying the track by eye particularly difficult and several viewings were sometimes necessary. 96 out of the 100 tracks chosen were deemed to match the video. A random sample of 20 of these are available for the reader's scrutiny, 10 from the surface and 10 away from the surface at the URLs:
`http://people.maths.ox.ac.uk/yatesc/ieee_movies/tracksurf1.avi` to
`http://people.maths.ox.ac.uk/yatesc/ieee_movies/tracksurf10.avi` and
`http://people.maths.ox.ac.uk/yatesc/ieee_movies/track1.avi` to
`http://people.maths.ox.ac.uk/yatesc/ieee_movies/track10.avi` for tracks at, and away from the surface respectively. An example of a successful track is given in figs. 6.24(a)-6.24(c).

The reader may be more interested in the 4 incidents where the tracker output did not match up to the manual analysis. There were two at the surface and two away from the surface and the videos are available at the URLs:
`http://people.maths.ox.ac.uk/yatesc/ieee_movies/tracksurffail1.avi`,
`http://people.maths.ox.ac.uk/yatesc/ieee_movies/tracksurffail2.avi` and

http://people.maths.ox.ac.uk/yatesc/ieee_movies/trackfail1.avi,
http://people.maths.ox.ac.uk/yatesc/ieee_movies/trackfail2.avi respectively.

All four incidents contain a rare occurrence. One of the track failures at the surface includes a cell occluded three times in quick succession and the other an incident where two cells with similar trajectories are both simultaneously occluded by a third cell as they cross (see figs. 6.24(d)-6.24(f)). Away from the surface, one of the track failures is a case where a cell disappears off of the edge of the image at the same time as a new cell appears at a similar point. The other is of a large cell, which appears to be about to divide, spinning rapidly. A track is declared but its position is erratic.

In a tracking scenario with a large number of varied targets, some of which are faint and which frequently occlude each other, such incidents might be expected to give a low level of unreliable tracks, but the overall results give confidence in the match between manual analysis and the tracker. It is also clear from some of the videos that the tracker allows detection of some cells which might have been too faint to detect by eye, and there were also incidents with occlusions where the tracker at first appeared to be incorrect but upon a more careful viewing was shown to be correct. In these cases the performance of the tracker may be better than that of manual analysis. For self-containment purposes, we have included stills from two of the videos described above to give the reader an insight into what these events look like. The stills may be found in fig. 6.24.

6.3.4 Performance Comparison Using Simulated Data

It is desired to test the PHD filter against other recent methods from microscopic visual tracking. A recent review on the topic which gives a "good picture of the state-of-the-art" [106] for live-cell tracking is given by Jaqaman et al. [50] which broadly characterises tracking methods in terms of defining a cost between cell detections in subsequent frames and then performing a deterministic total cost minimisation. Examples of tracking methods based on cost minimisation may be found in the papers by Ouellette et al. [86] and Xie et al. [138] for tracking in particle image velocimetry (PIV) and tracking *Escherichia coli* respectively. There is a fundamental difference between this class of methods and the PHD filter because the cost minimisation approach is deterministic as opposed to the statistical approach of the PHD filter as described above.

In order to obtain a reliable quantitative comparison between the PHD filter and alternative methods, it will be necessary to use simulated data. This is due

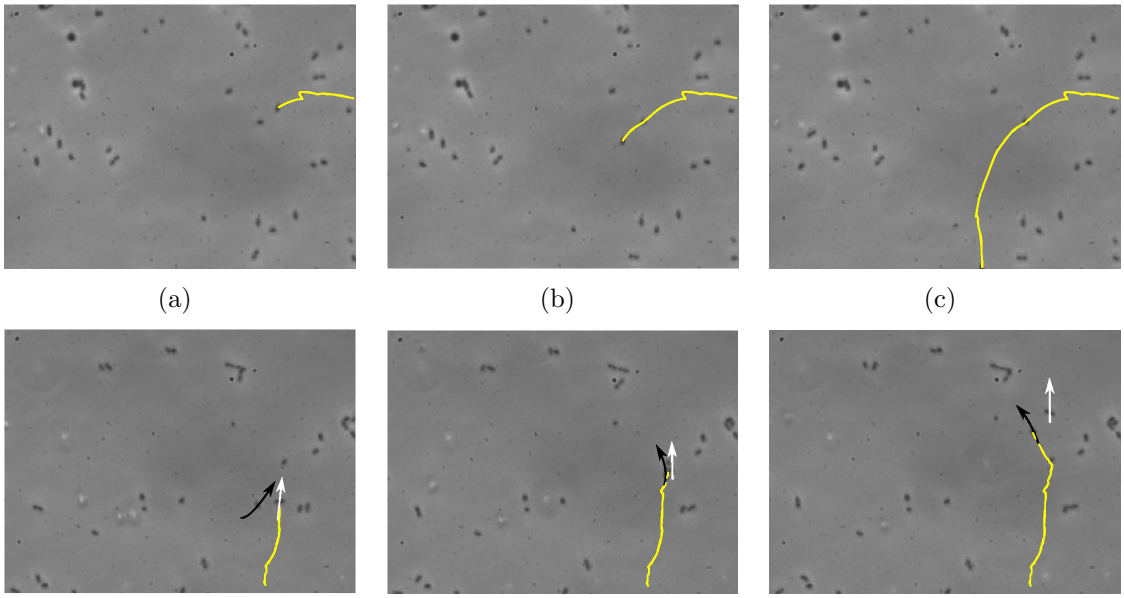


Figure 6.24: Illustrative video frames to complement the above linked videos. Stills from videos illustrating a well-tracked cell (a)-(c) and a track which was lost (d)-(f) due to an occlusion with a moving cell and a stationary cell at the same time (an unusual event). (a) The track (yellow) is initiated almost from the moment the cell arrives on the screen. (b) The tracker manages to deal with an occlusion event as the tracked cell passes over a stationary cell. (c) The track continues until the cell leaves the screen. (d) A new cell is tracked as it appears on the screen and its motion path indicated with a white arrow through its centre. A cell which will occlude the tracked cell also has its motion path marked in a similar way with a black arrow. (e) The three cells occlude each other. (f) The path of the track is swapped to the cell marked with the black arrow as the cells separate. ©IEEE 2012

to the absence of a known 'ground truth' in the real data. Simulating images that accurately replicate experimentally-acquired data is extremely challenging. Different optical contrast-enhancement methods (i.e.. Differential Interference Contrast (DIC), Phase Contrast (PC) or Dark-Field (DF) microscopy), as well as chemical staining techniques can be used to visualise swimming bacteria, and therefore the appearance of each image will be dependent on many experimental factors. Furthermore, different bacterial species will have different shapes, sizes and refractive properties and the buffers that they require for swimming will potentially have different refractive indices. Therefore biological factors will also affect the appearance of the image data both in terms of image contrast and image noise. For these reasons, instead of simulating full images, the set of detections for each frame was simulated taking into account phenomena such as false alarms, occlusions and measurement noise. The dynamics of

the bacteria were simulated in a large region using the model described in [101] and a smaller 'field of view' was created inside the large region. This allowed simulated bacteria to swim in and out of the field of view as in a real recording.

The performance of the GM-PHD filter was compared with two methods based on cost minimisation. The first is the nearest neighbour (NN) method as described in [86] which is used as a benchmark. The cost function in this case is the distance between the estimated position of the tracked cell and the position of the newly detected measurement. This tracker is equivalent to the implementation in, for example, [39]. The second alternative method is the tracker from [138] denoted XIE hereafter. The cost function for this tracker is the sum of the distance between the predicted cell position and the new detection, and the difference between the predicted direction of motion of the cell and the observed direction of motion given the new detection. Information relating to image histograms is not included as it is not available in this simulated data. In order to permit equivalent computational complexity to the GM-PHD filter, the sliding window size for XIE was chosen as two.

Performance Comparison Using OSPA Metric

Evaluation in multitarget tracking is often performed using the optimal sub-pattern assignment (OSPA) metric [107], a consistent metric for the distance between two sets. In multitarget tracking performance evaluation, OSPA is used to measure the distance between the true set of tracks and the estimate returned by a given tracker. The OSPA metric takes two factors into account; the difference in localisation and the difference in cardinality (the number of targets). The relative weighting given to each of these components depends on a parameter, the 'cut-off' parameter, c. For the test here, the parameter c is chosen as $c = 10$ pixels. This may be interpreted as errors in cardinality (e.g. targets not tracked) being penalised at the same rate as a localisation error of 10 pixels, the maximum localisation error considered. OSPA itself may be interpreted as an average error per target and hence has a maximum value of 10 pixels.

A significant problem with the NN and XIE trackers is the way in which false alarms are handled. In the NN implementation by Etgar et al. it is stated that "the bright peaks that were not used... initialize new trajectories". This means that any false alarms not assigned to existing trajectories always initialise new tracks. Similarly, in [108] on which the XIE algorithm is based, it is stated that "a track is either composed of sensor responses from a single world point or it is composed of points occurring from noise only". Therefore, it can be seen that both methods will

produce track sets which contain a high level of tracks corresponding to noise. This results in noisy track sets and also means that cardinality error will be large whenever there are false alarms.

As an example of this problem, GM-PHD, NN and XIE trackers were evaluated on a simple scenario with an average of three cells in the field of view, no missed detections and some false alarms, uniformly distributed over the field of view where the number of false alarms in each frame has a Poisson distribution with mean λ. 4000 time step simulations were computed for a range of λ and the average OSPA distances are shown in fig. 6.25. This figure demonstrates the consistent strong performance of the GM-PHD filter while the other trackers have close to the highest possible OSPA distance even for low false alarm levels.

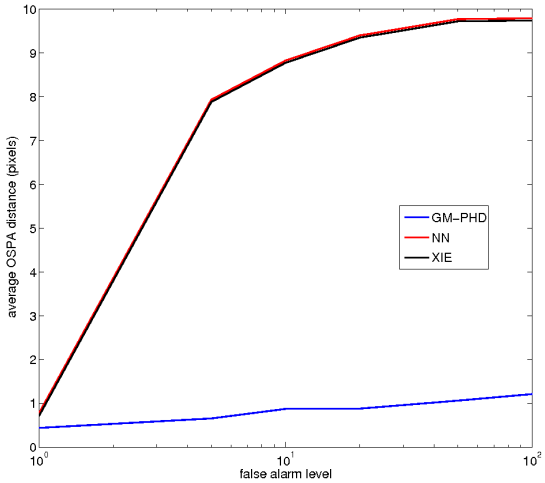


Figure 6.25: Average OSPA distance for each tracker for a range of λ, the average number of false alarms. ©IEEE 2012

The results shown in fig. 6.25 highlight the inability of existing methods based on cost minimisation to handle the multitarget aspect of the tracking problem, which is to jointly determine the number of targets and their locations. The PHD filter's statistical approach to handling varying target number and false alarms allows it to differentiate between false alarms and true targets directly, which results in less noisy track outputs.

Single Established Trajectory Performance Comparison

A further set of comparisons were implemented to determine the ability of the trackers to maintain continuous and accurate tracks on individually identified cell trajectories. In order to test performance under challenging tracking conditions the simulations

included phenomena relevant to microscopic visual tracking which make the scenario more difficult. The phenomena included were uniformly distributed false alarms, spatially and temporally correlated false alarms and occlusions. Although these do not capture every aspect of microscopic visual data, they are broadly representative of the complications which make the problem difficult.

In all of the tests there were an average of 25 cells in the field of view unless otherwise stated. This is representative of the real data as shown in Section 6.3.2. For each test, 100 individual trajectories were randomly selected. A track was initialised on the cell location in the first frame of its appearance and the average distance (in pixels) between the tracker's estimated location and the true location were measured. Distances above 10 pixels were considered lost tracks and penalised as 10 pixel errors, as in the OSPA metric with c =10 pixels.

Test 1: Uniform False Alarms - In this test, the data was corrupted by false alarms uniformly distributed in the field of view with the number of false alarms given by a Poisson distribution with mean λ. Simulations were carried out for a range of λ and results are shown in fig. 6.26. These results show that performance for GM-PHD and XIE are better than NN but broadly similar to each other for all but the highest levels of false alarms. The false alarm levels at which GM-PHD gains a significant advantage over XIE are unrealistically high for real data. Therefore, in realistic tracking conditions for uniformly distributed false alarms, the performance of GM-PHD and XIE is similar.

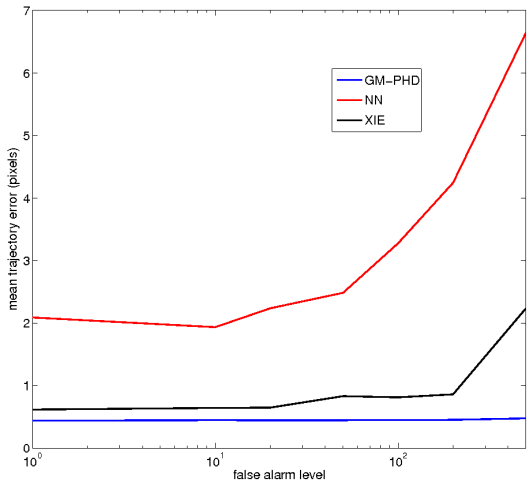


Figure 6.26: Average trajectory localisation error (in pixels) for each tracker for a range of λ ©IEEE 2012

Test 2: Spatially and Temporally Correlated False Alarms - In order to represent the kind of false alarms seen in the real data, a number of false alarms were generated

in the vicinity of each true cell, achieving spatial and temporal correlation in a manner similar to the false alarms observed the real data discussed in Section 6.3.2 as a result of the thresholding process. The number of false alarms generated by each cell was chosen to be Poisson distributed with mean γ. Simulations were carried out for a range of γ and the results are given in fig. 6.27. The results show that the advantage of the GM-PHD is greater in the case of correlated clutter than uniform clutter. An explanation for this is that the spatially correlated false alarms present conflicting options about which detection should be used to update the trajectory. Whereas the cost minimisation methods must choose one measurement, the PHD filter is able to propagate weighted hypotheses for each possible association. The results show that the statistical approach of the PHD filter is more suited to tracking a trajectory in data containing correlated false alarms.

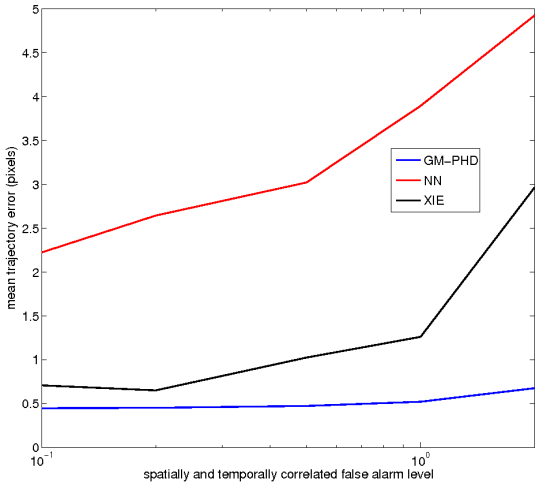


Figure 6.27: Average trajectory localisation error (in pixels) for each tracker for a range of γ ©IEEE 2012

Test 3: Occlusions - In this test, if the centroids of two simulated cells are within five pixels of each other, an occlusion is deemed to have occurred, and a single detection is recorded for the two cells at the midpoint of their two centroids. It is well known [38] that the PHD filter has a short memory with regard to targets for which there are missed detections (i.e. the weight of these targets drops very quickly). In order to alleviate this problem, the cardinalised PHD filter was proposed by Mahler [75]. However, a simpler approach to alleviating the problem is taken here. The existence of track labels permits a trajectory dependent target declaration threshold so that established targets with low weights can continue to be declared despite missed detections. Furthermore, occlusions can be taken into account explicitly by reducing

the probability of detection, p_D for targets in the vicinity of other targets (and hence in danger of being occluded), as described in Section 6.3.2.

In order to test the effect of occlusions on the different tracking algorithms, the cell density in the simulations was varied, as occlusions become more of a severe problem when there are more cells in the field of view. The results are given in fig. 6.28 which shows comparable performance between GM-PHD and XIE with significantly worse performance for NN.

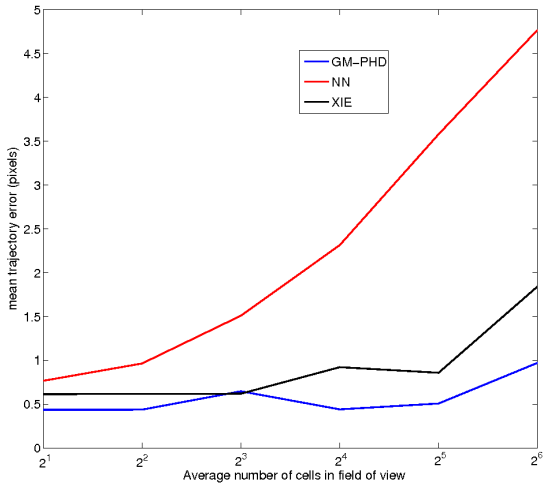
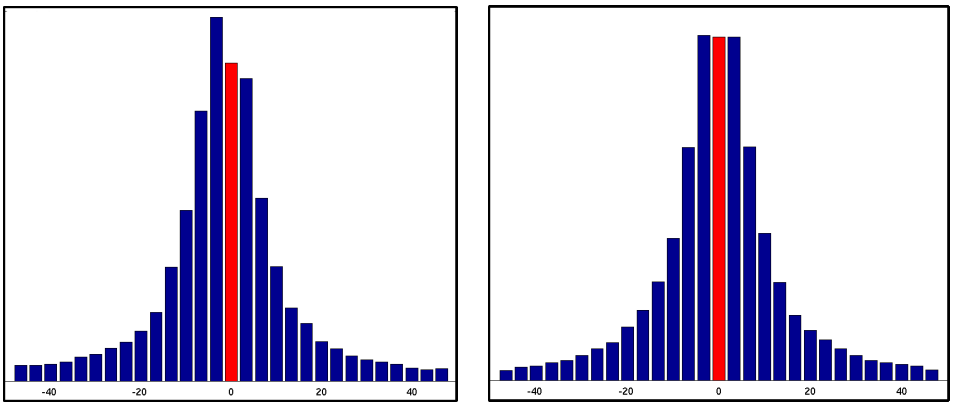


Figure 6.28: Average trajectory localisation error (in pixels) for each tracker for a range of cell densities with occlusions ©IEEE 2012

In summary, the performance of the GM-PHD filter is significantly better than traditional nearest neighbour based algorithms in a wide variety of tracking conditions. The tracking algorithm of Xie et al. is also able to outperform NN. The GM-PHD has a moderate performance advantage over XIE in tracking single trajectories, particularly in the presence of correlated false alarms. The real advantage of using GM-PHD is its ability to distinguish false alarms from true targets. In scenarios where there are a significant number of false alarms, the NN and XIE trackers should be expected to output many tracks containing mostly or entirely false alarms whereas the GM-PHD does not, as is clearly demonstrated in fig. 6.25.

6.3.5 Results

An initial data set contained 78 minutes of video footage. Many of the tracks returned from this footage were immobile cells (which may have been, for example, dead or stuck to the surface). Tracks of immobile cells provide no useful information, so only tracks with a minimum level of movement were used. 4182 tracks were returned from



(a) (b)

Figure 6.29: Distribution of framewise angle changes: a) at the surface. b) away from the surface. ©IEEE 2012

78 minutes of footage, which is nearly two orders of magnitude more than the 59 and 50 tracks returned in [11] and [116] respectively. The large number of tracks returned makes it possible to build up accurate probability densities of parameters of interest from bacterial motion, an example of which is given below.

A bias towards spiralling in one orientation is consistent with fluid dynamical models for bacterial motion near a surface [109]. A direct comparison of this bias at the surface as compared with behaviour away from the surface was not previously possible. Figure 6.29 gives the distribution of angle changes (per time step) over the whole data set, showing a clear bias towards clockwise turns at the surface which is not present for cells away from the surface. This verifies the prediction from fluid dynamical models and is an example of how the large number of tracks extracted permits in-depth, quantitative analysis of a range of features of bacterial motion.

Computational Speed

Results were obtained using non-optimised MATLAB code on a machine with an AMD Phenom(tm) II X3 720 2.8GHz Processor. Performing the GM-PHD filter on 9000 frames of measurements with an average of 15 targets present took 543 seconds, giving a processing rate of about 16.5fps. The detection step may take longer depending on the image resolution and method used. It should be noted that computational complexity of GM-PHD is linear in both the number of measurements and the number of targets, so the speed will depend on both the number of cells present and the level of false alarms.

A similar implementation on the same data set using the improved SMC-PHD

filter with 2000 particles per target and 500 particles for new targets (chosen to enable direct comparison with the computational results presented in [71]) was carried out. This implementation took 6807 seconds, giving a rate of around 1.3 fps. The closest equivalent calculation in [71] is for 12 targets and the processing time given corresponds to around 0.7 fps. There are several factors which might explain the two-fold decrease in computation time as compared with the results in [71]. The improved SMC-PHD does not need clustering or data association, which reduces the computation. Furthermore, the presence of track labels within the PHD filter enables a gating process, as described in Section 5.1, which may have increased computational efficiency. Finally, some of the difference may be due to hardware, although the 3.2GHz processor used in [71] should be comparable to the 2.8GHz processor here.

The difference in performance between the standard SMC-PHD filter and improved SMC-PHD gives a roughly two-fold increase in performance, but the striking difference is that gained by using GM-PHD which gives a 25-fold increase in speed by comparison with the computational results reported in [71]. This 25-fold decrease in computation may provide the most compelling practical argument for adopting GM-PHD for visual tracking.

6.4 Performance Comparison in Simulated Data with Dense Clutter

In this section, simulations are used to allow a comparison, over a large set of Monte Carlo trials, of the GM-PHD filter with some of the conventional multitarget tracking methods described in Section 3.1. This will follow the work presented in [133]. The comparison will focus on high clutter scenarios which are relevant to, for example, active sonar in shallow water highly reverberant environments.

The examples used in the simulations here will represent 'vanilla' multitarget tracking scenarios. This means that the forward and measurement models will be linear, target SNR information will not be included and there will be no target manoeuvres. Restriction to this vanilla case reduces complication by additional factors. It should be noted that all of the aforementioned complications may be handled rigorously within the framework of the PHD filter. It might, therefore, be expected that the appropriate PHD implementations should deal with these complications at least as well as any ad hoc equivalent used with the conventional methods.

The Simulations

The performance of the GM-PHD filter will be compared with the global nearest neighbour (GNN) and probabilistic data association (PDA) methods combined with both 'M out of N' and sequential probability ratio test (SPRT) track maintenance, all of which are described in Section 3.1. The multiple hypothesis tracker (MHT) has been omitted from these tests due to the reported difficulty in implementing such a filter (Blackman and Popoli state that their implementation took "around six person-months to develop [15, p. 373]).

The first two tests will focus on well-known measures of effectiveness (MoE) for a multitarget tracker: track maintenance, initialisation time and level of false alarms. These MoEs will be determined in simple scenarios. It was pointed out in Section 5.4 that these MoEs are not rigorous metrics and so should be applied with caution. However, the use of familiar MoEs does have the advantage that it permits an analysis alongside previously published performance comparisons. Using the previously published performance comparisons it might also be possible to cautiously extrapolate a relationship between GM-PHD and MHT performance. The performance comparisons in both [8] and [15] lead to the conclusion that PDA gives equivalent performance to GNN in "several times" the clutter level. Similarly, in [15], MHT is deemed to give equivalent performance to GNN in 10-100 times the clutter level. GM-PHD will be tested against these benchmarks.

The final test will compare the OSPA metric for the various filters in a more complex tracking scenario. For this test the OSPA metric as described in Section 5.4 will be used. The use of track weights as described in Section 5.4.2 is not deployed as there is no immediate comparison between the 'M out of N' or SPRT scores and the PHD weights. If an analysis taking track confidence into account were desired, this could be achieved by mapping the 'M out of N' or SPRT scores onto [0,1] in a manner giving the desired equivalence between scores and PHD track weights.

The following scenario parameters are true for all of the tests:

- The single target state $\mathbf{x}$ is given by $\mathbf{x} = [x, \dot{x}, y, \dot{y}]$.
- Targets follow straight line motion and the (white, zero-mean Gaussian) process noise model parameter, q, is set to $q = 10^{-4}$ for all filters.
- The region of surveillance is $[-40, 40] \times [0, 40]$ where the units may be thought of as km.

- The measurement matrix H is given by

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{6.15}$$

  and the (white, zero-mean Gaussian) measurement noise has covariance matrix $R = \sigma^2 \mathbb{I}_2$ where $\sigma = 0.3$.

- Clutter is uniformly distributed over the surveillance region and the number of clutter points has a Poisson distribution with mean λ.

6.4.1 Track Maintenance

This will test the ability of the filters to maintain a track on a single established target moving through clutter. Track maintenance is a common measure of effectiveness in the literature [8] [62] [33]. However, the actual criteria for having maintained a track is often not given explicitly. In [8] the criteria is for the track to be sufficiently close to the truth for at least one of the last 20 time steps of a run. It seems a little generous to conclude that tracks fulfilling this criteria are all 'maintained tracks'. No explicit criteria are given in either [62] or [33].

For the tests here, the criteria that will be employed is to insist that for a maintained track, there is a track declared within 1 unit distance of the true target location for 80% of the time steps of the 50 time step run. For GNN and PDAF, a single (full weight) target will be declared on every time step whereas for GM-PHD the weight threshold will be set at 0.2 (see Table 4.2 for more details).

100 Monte Carlo runs were performed for each of a range of clutter levels (with λ between 10 and 1000) and with $p_D = 0.75$ and 1. The results are given in tabular form Table 6.2 and in visual form in fig. 6.30.

The results suggest that performance of the GM-PHD filter is equivalent to the GNN in 10-20 times the clutter level. This is not as great an improvement as the 10-100 times clutter level reported in [15] but it is still significant. More work on a direct performance comparison between MHT and the PHD filter would be useful. In one existing small-scale comparison [89] in which the PHD filter outperforms MHT (albeit only in 1 Monte Carlo run) the authors emphasize the comparative ease of implementation of the PHD filter as well as its lower computational load. What is clear is that GM-PHD offers a substantial improvement in track maintenance in high clutter levels compared with GNN and PDA and that taking all factors into account, it should be competitive with MHT.

	P_D	$= 1$		P_D	$= 0.75$	
clutter	NN	PDA	PHD	NN	PDA	PHD
10	1	1	1	0.96	0.97	1
20	0.99	1	1	0.94	0.97	1
50	0.97	0.99	1	0.92	0.96	1
100	0.93	0.99	1	0.84	0.95	0.99
200	0.88	0.99	1	0.73	0.93	0.99
500	0.65	0.91	1	0.47	0.47	0.92
1000	0.38	0.37	0.90	0.23	0.02	0.71
2000	0.14	0.09	0.41	0.07	0	0.14

Table 6.2: Proportion of tracks maintained from 1000 Monte Carlo Tests.

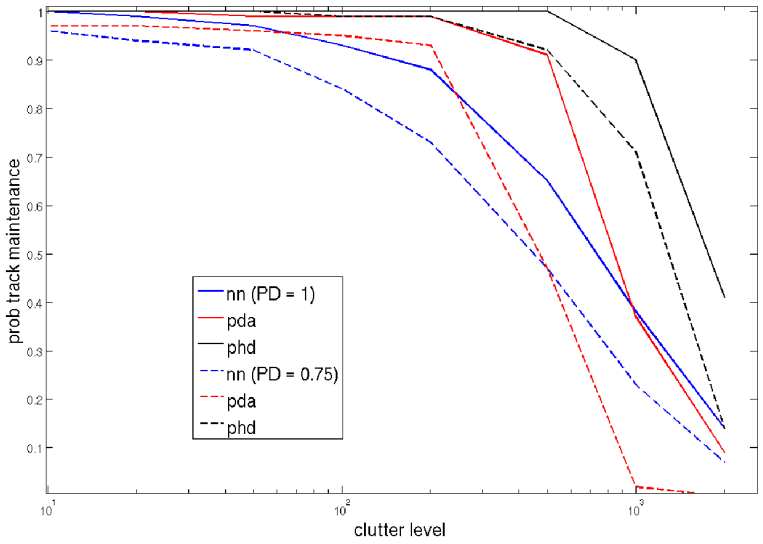


Figure 6.30: Proportion of tracks maintained from 1000 Monte Carlo Tests.

6.4.2 Track Initialisation and False Alarms

In this test, there is a single target in clutter. The ability of the filters to initialise a track (within 1 unit distance from the target) while minimizing the number of false tracks is measured. 100 Monte Carlo tests were run for a range of levels of clutter with $p_D = 1$ for all tests. Two track initialisation schemes were used with the conventional filters, 'M out of N' and SPRT as described in Section 3.1. Rather than exhaustively performing the test for every parameter regime, a smaller set of tests were carried out to determine reasonable parameter regimes. The parameters chosen for 'M out of N' were $M = 3$ and $N = 5$ while for SPRT the track confirmation threshold was set so that the false track confirmation probability was 10^{-3}, the weight threshold for declaring tracks in GM-PHD was set to 0.2.

The results are given in Table 6.3. Note that results are only given in the case where the target was tracked for more than 75% of the Monte Carlo tests and there were less than 50 false tracks per time step on average. Any filters not fulfilling both of these criteria would likely not be considered sufficiently reliable trackers in any context.

clutter	NN	M/N	NN	SPRT	PDA	M/N	PDA	SPRT	GM	PHD
	FT	Decl	FT	Decl	FT	Decl	FT	Decl	FT	Decl
10	0.5	3.1	0	3.2	1.9	3	0.2	3.1	0.2	2
20	2.7	3.1	0	3.5	7.2	3	0.3	3.5	0.4	2.2
50	23.1	3.1	0	4.6			0.9	4.4	0.3	3.1
100							1.3	5.2	0.3	4
200							1.2	6.6	0.3	5.4
500									0.1	10.4

Table 6.3: Number of false tracks (FT) declared per time step and time taken to declare real target in full tracking scenario (Decl.) averaged over 100 Monte Carlo tests.

The results show GM-PHD consistently declaring targets more than one time step before the next best filter while simultaneously declaring fewer false tracks. This combination of quicker track declaration and fewer false alarms demonstrates that there is an improvement in performance which goes beyond parameter adjustment.

6.4.3 Performance Comparison Using OSPA Metric

A more complex scenario was simulated containing eight targets appearing at various times during a 50 time step run. The trajectories for this scenario are shown in fig. 6.35(a). The OSPA metric, as discussed in Section 5.4 was used to evaluate performance by giving a distance of each tracker's multitarget estimate from the known ground truth. The OSPA metric depends on two parameters: p and c. For this test, $p = 2$ which may be interpreted as all single target distances corresponding to standard Euclidean distances. The parameter c determines the relative importances assigned to localisation and cardinality errors. For all tests, the OSPA metric was computed for a range of c values to give a better idea of the nature of each tracker's errors.

Due to the computational intractability of implementing the PDA filter in a multiple target scenario, as discussed in Section 3.1, the GM-PHD filter will be tested against the multiple target NN filter (GNN) for both 'M out of N' and SPRT track

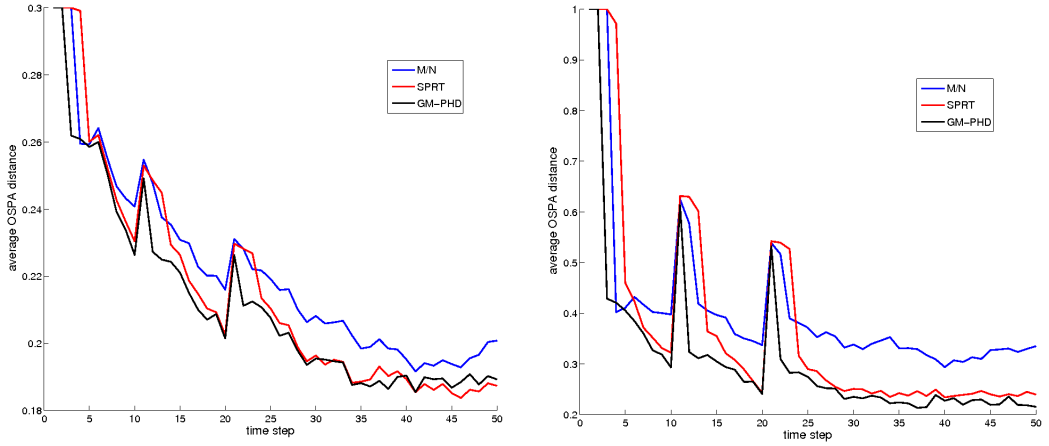
maintenance schemes only. The parameters from the tests in Section 6.4.2 are used here. 100 Monte Carlo runs are computed for a low clutter level, moderate clutter level and high clutter level ($\lambda = 5, 50, 200$ respectively).

The results are shown for each of the clutter levels in figs. 6.31-6.33 for both $c = 0.3$ and $c = 1$. The former case emphasizes localisation errors, while the latter case emphasizes cardinality errors. Note that the jumps in the OSPA distance at time steps 10 and 20 are caused by the appearance of two of the targets at each of those time steps. Figure 6.34 shows the ratio of the average OSPA distances for GM-PHD and GNN with SPRT track maintenance over a range of clutter levels and c.

Some conclusions emerge from a study of the results given in figs. 6.31-6.34. Firstly, GM-PHD performs better than GNN for all clutter levels, and the improvement in performance increases as the clutter level increases. This is in agreement with the results from Sections 6.4.1 and 6.4.2 but is based on a metric rather than a measure of effectiveness. Secondly, the performance advantage of the GM-PHD filter increases as the metric parameter c increases. This shows that the cardinality estimation in GM-PHD has a greater performance advantage than localisation estimation. This might have been expected; the localisation estimation in the GM-PHD is due to the same Kalman filter equations as in GNN. The real advantage of the random finite set methods over the conventional methods is the incorporation of explicit statistical models for the full set of relevant phenomena to the tracking scenario. In particular, explicit models are incorporated for those phenomena regarding cardinality issues such as target appearance and disappearance which are handled in an ad hoc manner in the conventional methods.

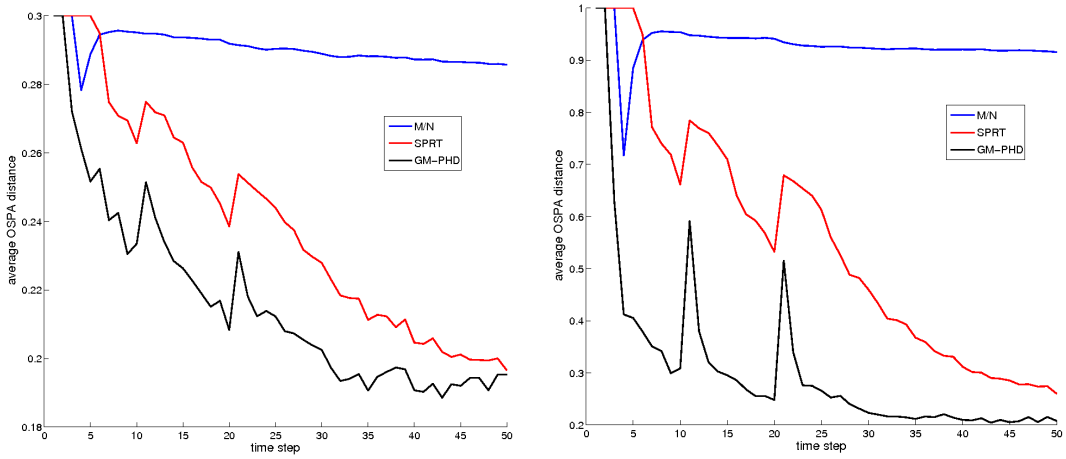
As a final demonstration of the ability of the GM-PHD filter to extract targets from dense clutter, an example is presented in fig. 6.35 which shows the noisy data set alongside the tracks declared by the GM-PHD filter.

Whereas in Section 6.4.1 it was possible to consider performance comparisons of GNN and MHT in order to tentatively extrapolate that GM-PHD should be competitive in performance with MHT, this is not possible here. This is because no comparison of GNN and MHT using an OSPA metric is available. Such a comparison of GM-PHD with MHT using the OSPA metric would be valuable future work. Regardless, the full advantage of using the random finite sets formulation of multi-target tracking is the ability of the framework to incorporate rigorously factors such as amplitude information [29], extended targets [115], group target motion [28] and multiple sensors taking into account sensor field of views and transmission drop-outs [75].



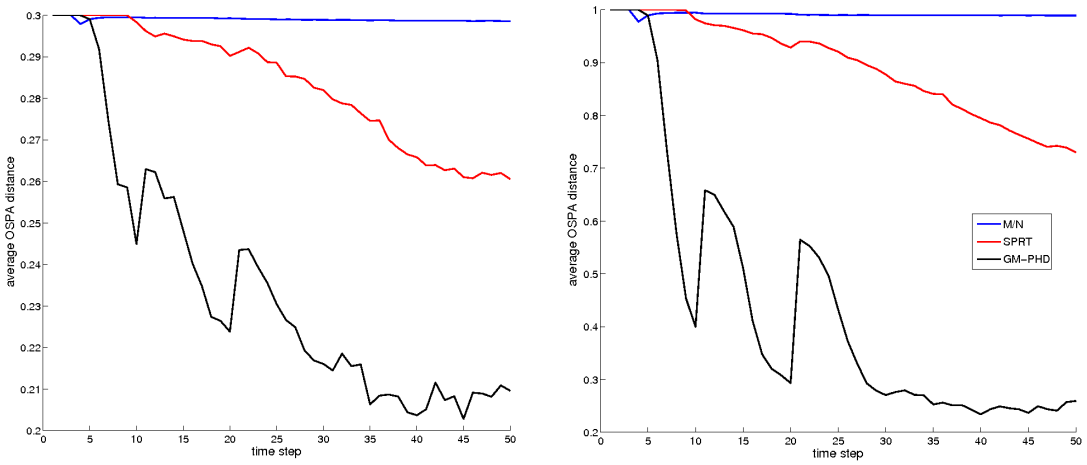
(a) (b)

Figure 6.31: Average OSPA distance for low clutter with (a) $c = 0.3$, (b) $c = 1$



(a) (b)

Figure 6.32: Average OSPA distance for moderate clutter with (a) $c = 0.3$, (b) $c = 1$



(a) (b)

Figure 6.33: Average OSPA distance for high clutter with (a) $c = 0.3$, (b) $c = 1$

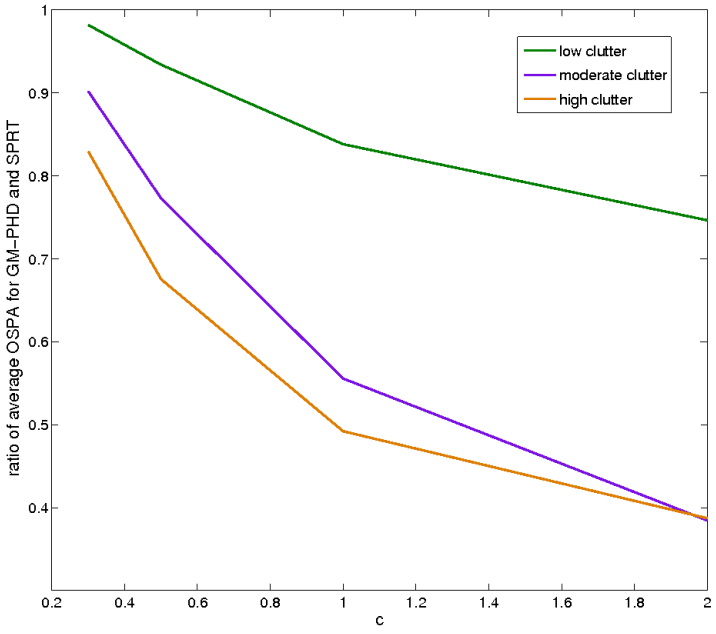


Figure 6.34: Ratio of average OSPA distance of GM-PHD and GNN with SPRT for various clutter levels and c.

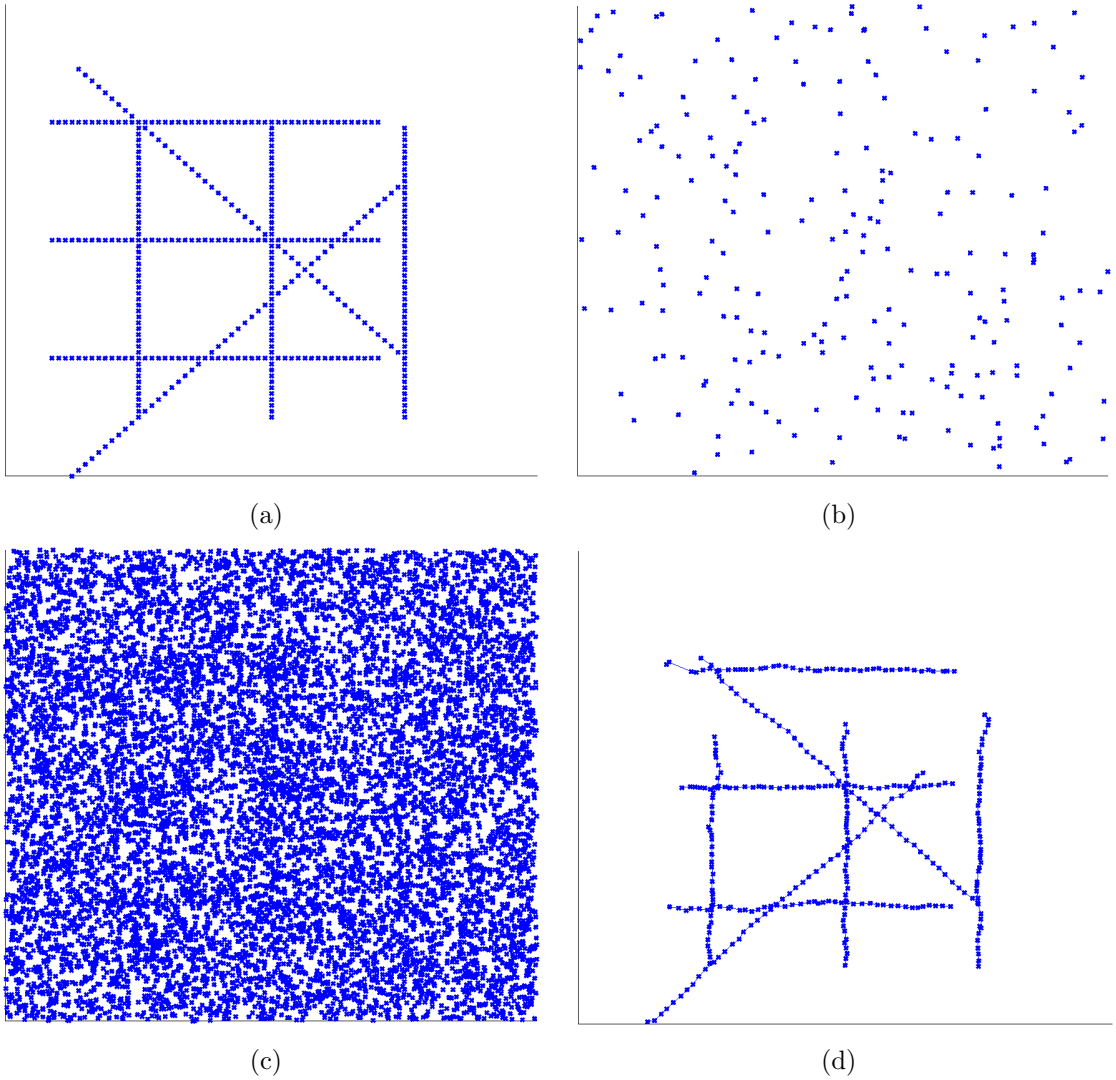


Figure 6.35: Illustrative results to demonstrate the ability of the GM-PHD filter to extract targets from dense clutter. (a) The target locations for all time steps. (b) The measurements from one time step. (c) The aggregated measurements from all time steps. (d) The tracks output by the GM-PHD filter.

Chapter 7

Summary, Conclusions and Future Work

The main aim of this thesis was to build on the existing work in the field of random finite set theory, extending the range of techniques and applications as well as improving understanding of the links with established Bayesian tracking theory. This was accomplished by:

1. Presenting an in-depth background to random finite set theory with a focus on connections with conventional probability theory (Chapters 3 and 4).

2. Developing new techniques to aid practical implementations of the probability hypothesis density filter and performance analysis thereof (Sections 5.1 and 5.4).

3. Deriving new links between extensions of the PHD filter and existing Bayesian methodologies (Sections 5.2 and 5.3).

4. Presenting new applications of the PHD filter, with particular focus on applications relating to sonar. Demonstrating the good performance of the PHD filter for these applications using a mixture of real and simulated data. (Chapter 6).

7.1 Summary and Conclusions

Chapter 2 focused on the more conventional single target tracking problem where the objective is to infer the state of a single target conditioned on noisy measurements received. A conceptual solution to this problem was given in the form of the Bayes filter in Section 2.1 and it was shown that in a special case this reduces to the well-known Kalman filter. As computation of the Bayes filter is intractable in general, a range of approximate solutions to the Bayes filter recursion has been developed. A selection

of the most popular broad approaches was presented. These approaches were: Taylor series approximation, sigma point approximation, Gaussian sum approximation, sequential Monte Carlo and grid approximation. The performance of these approximate methods was evaluated using the method of comparing the posteriors computed with the 'correct' Bayesian posterior in Section 2.2. A comparison was also made of the error standard deviation of each filter with its theoretical lower bound as given by the Cramer-Rao lower bound (CRLB). A test was carried out for a tracking scenario with measurements taken of range and bearing with bearing uncertainty large relative to range uncertainty, which is relevant to sonar and radar tracking. Results presented in Section 2.2.2 showed that, for larger values of bearing uncertainty, SMC methods gained a performance advantage over UKF, which in turn gained a performance advantage over EKF. These differences in performance became more significant as bearing uncertainty increased. Further results suggested that methods based on the assumption that the posterior density was Gaussian were very poor at approximating the posteriors even for smaller values of bearing uncertainty. This suggests that EKF and UKF methods should be implemented with caution. A new method for the selection of filter initialisation based on Bayesian principles was presented in Section 2.3 and it was demonstrated in a simple test that this method gives a significant performance increase for no additional computational cost. The background to handling target manoeuvres using jump Markov linear systems was presented.

A key aim of this thesis has been to give the reader an insight into the theory of random finite sets. To this end, Chapters 3 and 4 included a background of conventional multitarget tracking alongside a big picture view which gives an idea of the motivation behind the development of random finite set theory. A more detailed summary of the background to random finite set theory was presented in Section 3.3 emphasizing the links between multitarget and conventional statistics as well as the relationship to point process theory. The derivation of the multitarget Bayes filter was presented in detail in order to demonstrate how the theory may be used in calculating multitarget posterior densities by constructing belief mass functions and appealing to the multitarget analogue to the Radon-Nikodým theorem. The practicalities of a multitarget Bayes filter implementation were also discussed, and problems in defining the conventional estimators were elaborated. The first-order moment of these multitarget probability densities, the probability hypothesis density (PHD) was introduced in Section 4.1 and the PHD filter, which propagates only this moment, was derived in detail using probability generating functionals. This provides a more computationally tractable alternative to a full multitarget Bayes filter

implementation, but is based on the familiar and systematic approximation strategy of moment matching. Worked examples of both the multitarget Bayes filter and the PHD filter were given in a simple scenario with the aim of elucidating aspects of calculations of these filters. In order to give the reader the tools necessary to implement the PHD filter, the full details of the sequential Monte Carlo (SMC) and Gaussian mixture (GM) implementation of the PHD filter were given in Section 4.2.

Chapter 5 focused on extending the application of the PHD filter beyond the 'vanilla' tracking scenario analysed in Chapter 4. Also considered was analysis of the PHD filter equations in such scenarios by comparison with existing techniques based on Bayesian methods. Some problems arise in sequential Monte Carlo implementations of the PHD filter relating to the clustering step and target birth. Techniques were developed in Section 5.1 which alleviate these problems culminating in the improved SMC-PHD filter with track continuity. Numerical simulations were presented which demonstrate the good performance of these techniques. Handling manoeuvring targets within the jump Markov linear system framework was considered in the multitarget case in Section 5.2. A scheme for incorporating the jump Markov linear system model within the PHD filter was proposed which is based on interacting models. Although the proposed JMLS-PHD filter does not depend on the form of the forward model, it was shown that in the linear Gaussian case, it is equivalent to the JMLS-GM-PHD filter derived by Pasha et al. [90] using linear Gaussian assumptions. Furthermore, it was shown that the JMLS-PHD filter reduces to the conventional Bayes filter in the single target case. Incorporation of target amplitude information can assist tracking low SNR targets. In Section 5.3 the target-amplitude PHD filter due to Clark et al. [29] was presented and some relationships between this approach and the track-before-detect methodology were demonstrated. In particular, in the case where the target state space is discrete, the track-before-detect recursion is almost identical to a special case of the target-amplitude PHD filter and in the case where it is known that there are 0 or 1 targets present, the track-before-detect filter is mathematically equivalent to a multitarget Bayes filter. Finally, the issue of metrics on the multitarget state was considered in Section 5.4. The OSPA metric due to Schuhmacher et al. [107] was described and it was demonstrated that inclusion of target confidence information could assist the effectiveness of this metric. An extension to the OSPA metric including target confidence was proposed and it was shown that the distance derived is indeed a metric.

Chapter 6 presented a range of applications of random finite set theory, with a focus on problems relating to sonar. An implementation of the GM-PHD filter on real

active sonar data with an injected simulated target was presented in Section 6.1 showing good results, particularly when using the SNR-PHD filter for low SNR targets. In Section 6.2 the passive sonar problem of detecting low SNR tones in audio signal data was reformulated as a multitarget tracking problem. A novel denoising step was presented for passive sonar data and the GM-PHD was implemented permitting detection and monitoring of a time-varying number of tones in both real and simulated passive sonar data. The GM-PHD filter was also implemented to aid the extraction of bacterial trajectories from microscopic visual data in Section 6.3. The technique permitted the extraction of a much larger number of trajectories than was previously possible which allowed more detailed probability distributions to be inferred for key parameters of motion. These results were used to verify a fluid dynamical prediction about the behaviour of bacteria near to surfaces. Finally, in Section 6.4 a comparison between the GM-PHD filter and conventional multitarget tracking techniques was presented showing that the GM-PHD filter has a significant performance advantage compared with several existing techniques, particularly in high clutter levels.

7.2 Possible Directions for Future Work

- *Performance limits for multitarget tracking* - As a result of the recent proposal of the OSPA metric [107] and its extensions to include target labelling information [97] and target confidence information there now exists a consistent metric on the space of finite sets which may be used to assess the performance of a multitarget filter. A possible next step would be to consider the existence of limits of the performance of multitarget filters under this metric. In the single target case, the CRLB is a well known lower bound on the variance of an unbiased estimator. Knowledge of the CRLB was useful in assessing the performance of single target filters in Section 2.2 and an equivalent bound in the multitarget case would be valuable. As the CRLB is a bound on the second moment of the single target estimator this suggests that an understanding of an equivalent bound for the multitarget case might benefit from a better understanding of the second order moment of the multitarget posterior.

- *Investigation of higher order multitarget moments and filters* - The PHD filter propagates only the first-order multitarget moment, which is equivalent to the first order factorial moment densities of a finite point process. Higher order factorial moments densities as discussed in [32, p. 130] provide the higher order multitarget densities. The assumption, inherent to the PHD filter, that

the higher order moments are negligible may be too restrictive in some tracking scenarios. A desirable goal would be to propagate more multitarget moments. In [72], Mahler discusses the idea of a second-order multitarget filter but concludes that the computation currently eludes solution and may be too complex for practical use anyway. A higher order multitarget filter remains an open problem.

- *Understanding bacterial chemotaxis using the trajectories captured by the PHD filter* - There is ongoing work in analysing the bacterial trajectories captured using the PHD filter to understand bacterial chemotaxis and a paper is currently in preparation [101]. A topic of particular interest to current research is the stop-start motion of certain bacteria including *Rhodobacter sphaeroides* and the jump Markov linear system model discussed in Sections 2.4 and 5.2 is being considered. Applications for monitoring collective animal motion are also under investigation.

- *Performance comparison of GM-PHD and MHT* - MHT is generally considered to be the best performing competitor to the PHD filter for standard multitarget tracking problems such as the scenario in Section 6.4. A detailed comparison of the two would be valuable.

- *Improved SMC-PHD filter with track continuity and the multi-target multi-Bernoulli filter* - A similarity between the improved SMC-PHD filter and the multi-target multi-Bernoulli filter [127] was first observed in [96]. With the incorporation of track continuity, as in [136], the two filters are closely linked. Further investigation would be useful. Also of value would be a performance comparison between these two filters and the auxiliary particle implementation of the PHD filter proposed by Whiteley et al. [130].

Appendix A

Relationship to Measure Theoretic Probability Theory

This appendix will summarise the technical details of the relationship between finite set statistics (FISST) and conventional measure theoretic probability. Further discussion on this topic may be found in [75] and [125].

A.1 Definitions

Let $\mathcal{X}$ be an arbitrary space. A σ-algebra on $\mathcal{X}$, $\sigma(\mathcal{X})$, is a collection of subsets of $\mathcal{X}$ satisfying the following properties:

1. $\sigma(\mathcal{X})$ is non-empty.
2. $\sigma(\mathcal{X})$ is closed under complementation.
3. $\sigma(\mathcal{X})$ is closed under countable unions.

A *topology* on $\mathcal{X}$, $\tau(\mathcal{X})$, is a collection of subsets of $\mathcal{X}$ satisfying:

1. $\tau(\mathcal{X})$ contains the empty set and $\mathcal{X}$.
2. $\tau(\mathcal{X})$ is closed under arbitrary unions.
3. $\tau(\mathcal{X})$ is closed under finite intersections.

For a topological space $\tau(\mathcal{X})$, the *Borel algebra* on $\tau(\mathcal{X})$ is the smallest σ-algebra containing all open sets. A measure, defined on a σ-algebra of $\mathcal{X}$, is a function $\mu : \sigma(\mathcal{X}) \rightarrow \mathbb{R}$ satisfying:

1. $\mu(Y)$ is positive $\forall\ Y \in \sigma(\mathcal{X})$.

2. $\mu(\emptyset) = 0$.

3. $\mu(\bigcup_i Y_i) = \sum_i \mu(Y_i) \forall$ countable unions of pairwise disjoint sets Y_i.

A.2 Measure Theory for Random Finite Sets

Random finite sets are random variables which take their values in the space of finite sets, $\mathcal{F}(\mathcal{X})$ of some underlying space $\mathcal{X}$. It is typically assumed that $\mathcal{X} = \mathbb{R}^d$. In order to define a random variable, it is typically necessary to define a topology on the space of random elements and then define probability in terms of Borel subsets.

The space $\mathcal{F}(\mathcal{X})$ is topologised using a Mathéron "hit-or-miss" topology (for more detail see [75, App. F]). Using this topology, an RFS is a measurable map $X : \Omega \to \mathcal{F}(\mathcal{X})$ with probability measure

$$P_X(O) = \Pr(X \in O), \tag{A.1}$$

defined on Borel measurable subsets O of $\mathcal{F}(\mathcal{X})$ where Ω is the sample space.

As in conventional probability theory, it will be useful to be able to work with probability densities. This requires defining a measure on $\mathcal{F}(\mathcal{X})$ in order to permit integration. The conventional choice of measure, μ is given by [125]:

$$\mu(O) = \sum_{r=0}^{\infty} \frac{\lambda^r \chi^{-1}(\sigma_r \cap O)}{r!} \tag{A.2}$$

for all $O \subseteq \mathcal{F}(\mathcal{X})$. Here, λ^r is the rth order dimensionless Lebesgue measure, χ is the mapping from vectors to sets so that $\chi(\mathbf{x}_1, ..., \mathbf{x}_r) = \{\mathbf{x}_1, ..., \mathbf{x}_r\}$ and σ_r is the set of sets with exactly r members.

Using this measure, the integral of a function $f : \mathcal{F}(\mathcal{X}) \to \mathbb{R}$ is given by:

$$\int_O f(X)\mu(dX) = \sum_{r=0}^{\infty} \frac{1}{r!} \int_{\chi^{-1}(\sigma_r \cap O)} \mathbf{1}_O(\chi(\mathbf{x}_1, ..., \mathbf{x}_r)) f(\{\mathbf{x}_1, ..., \mathbf{x}_r\}) \lambda^r(d\mathbf{x}_1, ..., d\mathbf{x}_r). \tag{A.3}$$

A.3 Radon-Nikodým Theorem and Random Finite Set Densities

The Radon-Nikodým theorem states that if μ and μ' are two measures on $\mathcal{F}(\mathcal{X})$ and μ' is absolutely continuous with respect to μ then there exists a function, unique almost everywhere such that

$$\mu'(O) = \int_O f(X)\mu(dX), \tag{A.4}$$

and hence there exists a probability density f_X such that

$$f_X = \frac{d\mu'_X}{d\mu}. \tag{A.5}$$

A.4 Belief Mass Functions and Probability Measures

Using the theory above, probability measures exist only as defined on the abstract hyperspace $\mathcal{F}(\mathcal{X})$. It would be more intuitive to work on the underlying space $\mathcal{X}$. The belief mass function

$$\beta_X(S) = \Pr(X \subseteq S) \tag{A.6}$$

is a function somewhat analogous to a probability measure which is defined on $S \subseteq \mathcal{X}$.

The Choquet-Mathéron theorem has a consequence that P_X is equivalent to β_X so this allows substitution of the latter for the former without information loss [75, App. F]. Furthermore, it is shown in [125] that the set derivative as defined in (3.26) applied to the belief mass function is essentially the probability density function.

This permits working in $\mathcal{X}$ rather than $\mathcal{F}(\mathcal{X})$ which is intuitively appealing but does require defining substitutes for the set integral and Radon-Nikodým derivative. These are given in (3.19) and (3.31) respectively.

Appendix B

Additional Proofs and Results

B.1 Intermediate Results From PHD Corrector

Let $F[g,h]$ be a two variable functional given by:

$$F[g,h] = \exp(\lambda\kappa[g] - \lambda + \mu s[h(1-p_D)] + \mu s[hp_D p_g] - \mu). \tag{B.1}$$

where the notation $a[b]$ is shorthand for $\int a(\mathbf{y})b(\mathbf{y})\mathrm{d}\mathbf{y}$ and $p_g = \int g(\mathbf{z})L_{\mathbf{z}}(\mathbf{x})\mathrm{d}\mathbf{z}$. Then,

$$\text{(i)} \quad \left.\frac{\delta F}{\delta Z}\right|_{[g,h]=[0,1]} = e^{-\lambda-\mu s[p_D]} \cdot \prod_{\mathbf{z}\in Z}(\lambda\kappa(\mathbf{z}) + \mu s[p_D L_{\mathbf{z}}]) \tag{B.2}$$

$$\begin{aligned}\text{(ii)} \quad \left.\frac{\delta}{\delta \mathbf{x}}\frac{\delta F}{\delta Z}\right|_{[g,h]=[0,1]} &= e^{-\lambda-\mu s[p_D]} \cdot \mu(1-p_D(\mathbf{x}))s(\mathbf{x}) \prod_{z\in Z}(\lambda\kappa(\mathbf{z}) + \mu s[p_D L_{\mathbf{z}}]) \\ &+ e^{-\lambda-\mu s[p_D]} \prod_{\mathbf{z}\in Z}(\lambda\kappa(\mathbf{z}) + \mu s[p_D L_{\mathbf{z}}]) \cdot \sum_{\mathbf{z}\in Z} \frac{\mu p_D(\mathbf{x}) L_{\mathbf{z}}(\mathbf{x}) s(\mathbf{x})}{\lambda\kappa(\mathbf{z}) + \mu s[p_D L_{\mathbf{z}}]}\end{aligned} \tag{B.3}$$

Proof - i)

$$F[g,1] = \exp(\lambda\kappa[g] - \lambda - \mu s[p_D] + \mu s[p_D p_g]), \tag{B.4}$$

therefore, using (3.33) and (4.9)

$$\begin{aligned}\frac{\delta F}{\delta \mathbf{z}}[g,1] &= F[g,1] \cdot \frac{\delta}{\delta \mathbf{z}}(\lambda\kappa[g] - \lambda - \mu s[p_D] + \mu s[p_D p_g]) &\text{(B.5)}\\ &= F[g,1](\lambda\kappa(z)\mu s[p_D L_{\mathbf{z}}]). &\text{(B.6)}\end{aligned}$$

Since

$$\frac{\delta}{\delta \mathbf{z}'}(\lambda\kappa(z)\mu s[p_D L_{\mathbf{z}}]) = 0 \tag{B.7}$$

for $\mathbf{z} \neq \mathbf{z}'$. The result follows immediately by iterating the derivative.
ii) Letting $1 - p_D = q_D$,

$$\begin{aligned} \frac{\delta F}{\delta \mathbf{x}}[g,h] &= F[g,h] \cdot \frac{\delta}{\delta \mathbf{x}}(\lambda\kappa[g] - \lambda + \mu s[hq_D] + \mu s[hp_D p_g] - \mu) && \text{(B.8)} \\ &= F[g,h](\mu q_D(\mathbf{x})s(\mathbf{x}) + \mu p_D(\mathbf{x})p_g(\mathbf{x})s(\mathbf{x}) && \text{(B.9)} \\ \Rightarrow &= \frac{\delta F}{\delta \mathbf{x}}[g,1] = F[g,1]E[g] && \text{(B.10)} \end{aligned}$$

where $E[g] = \mu q_D(\mathbf{x})s(\mathbf{x}) + \mu p_D(\mathbf{x})p_g(\mathbf{x})s(\mathbf{x})$. The product rule for set derivatives (3.37) therefore gives:

$$\frac{\delta}{\delta Z}\left(\frac{\delta F}{\delta \mathbf{x}}\right) = \sum_{W \subseteq Z} \frac{\delta E}{\delta W} \cdot \frac{\delta F}{\delta Z - W}. \tag{B.11}$$

Noting that,

$$\frac{\delta E}{\delta \mathbf{z}} = \mu p_D L_{\mathbf{z}}(\mathbf{x})s(\mathbf{x}) \tag{B.12}$$

using (4.9) from which it is also clear that $\delta E/\delta W = 0$ for $|W| > 1$. Hence, remembering the form of $\delta F/\delta Z$ from (B.6),

$$\begin{aligned} \frac{\delta}{\delta Z}\Big(\frac{\delta F}{\delta \mathbf{x}}\Big) &= \sum_{W \subseteq Z : |W| \leq 1} \frac{\delta E}{\delta W} \cdot \frac{\delta F}{\delta Z - W} && \text{(B.13)} \\ &= E \cdot \frac{\delta F}{\delta Z} + \sum_{\mathbf{z} \in Z} \frac{\delta E}{\delta \mathbf{z}} \frac{\delta F}{\delta (Z - \{\mathbf{z}\})} && \text{(B.14)} \\ &= (\mu q_D(\mathbf{x})s(\mathbf{x}) + \mu p_D(\mathbf{x})p_g(\mathbf{x})s(\mathbf{x})) \cdot F[g,1] \prod_{z \in Z} (\lambda\kappa(\mathbf{z}) + \mu s[p_D L_{\mathbf{z}}]) \\ &\quad + \sum_{z \in Z} \mu p_D(\mathbf{x}) L_{\mathbf{z}}(\mathbf{x}) s(\mathbf{x}) \cdot F[g,1] \prod_{w \in Z - \{\mathbf{z}\}} (\lambda\kappa(\mathbf{w}) + \mu s[p_D L_{\mathbf{w}}]) && \text{(B.15)} \end{aligned}$$

from which the result follows. □

B.2 Proofs for GM-PHD filter

Predictor - Assume that:

- $f_{k+1|k}(\mathbf{x}|\mathbf{x}_k) = \mathcal{N}(\mathbf{x}; F_k\mathbf{x}_k, Q_k)$,
- P_S is constant
- $b_{k+1|k}(\mathbf{x}) = \sum_{i=1}^{J_{\gamma,k}} w_{\gamma,k}^i \mathcal{N}(\mathbf{x}; m_{\gamma,k}^i, P_{\gamma,k}^i)$.

If the PHD at time k is a Gaussian mixture of the form $D_{k|k}(\mathbf{x}) = \sum_{i=1}^{J_k} w_k^i \mathcal{N}(\mathbf{x}; m_k^i, P_k^i)$ then the PHD predictor

$$D_{k+1|k}(\mathbf{x}) = b_{k+1|k}(\mathbf{x}) + \int p_S f_{k+1|k}(\mathbf{x}|\mathbf{w}) D_k(\mathbf{w}) \mathrm{d}\mathbf{w} \tag{B.16}$$

can be written as

$$D_{k+1|k}(\mathbf{x}) = D_{S,k+1|k}(\mathbf{x}) + b_{k+1|k}(\mathbf{x}), \tag{B.17}$$

where

$$\begin{aligned} D_{S,k+1|k}(\mathbf{x}) &= p_S \sum_{i=1}^{J_k} w_k^i \mathcal{N}(\mathbf{x}; m_{S|k+1|k}, P_{S,k+1|k}), & \text{(B.18)} \\ m_{S,k+1|k}^i &= F_k m_k^i, & \text{(B.19)} \\ P_{S,k+1|k}^i &= Q_k + F_k P_k^i F_k^T. & \text{(B.20)} \end{aligned}$$

Proof - Due to the assumption on $b_{k+1|k}$ it is only necessary to show that $\int p_S f_{k+1|k}(\mathbf{x}|\mathbf{w}) D_k(\mathbf{w}) = D_{S,k+1|k}$.

$$\begin{aligned} \int p_S f_{k+1|k}(\mathbf{x}|\mathbf{w}) D_k(\mathbf{w}) &= p_S \int \mathcal{N}(\mathbf{x}; F_k\mathbf{w}, Q_k) \sum_{i=1}^{J_k} w_k^i \mathcal{N}(\mathbf{w}; m_k^i, P_k^i) \mathrm{d}\mathbf{w} & \text{(B.21)} \\ &= p_S \sum_{i=1}^{J_k} w_k^i \int \mathcal{N}(\mathbf{x}; F_k\mathbf{w}, Q_k) \cdot \mathcal{N}(\mathbf{w}; m_k^i, P_k^i) \mathrm{d}\mathbf{w} & \text{(B.22)} \end{aligned}$$

Using the form of $D_{k|k}(\mathbf{x})$ and the first two assumptions. Thus, using (2.19)

$$\begin{aligned} \int p_S f_{k+1|k}(\mathbf{x}|\mathbf{w}) D_k(\mathbf{w}) \mathrm{d}\mathbf{w} &= p_S \sum_{i=1}^{J_k} \mathcal{N}(\mathbf{x}, F_k m_k^i, Q_k + F_k P_k^i F_k^T) & \text{(B.23)} \\ &= D_{S,k+1|k} \quad \square & \text{(B.24)} \end{aligned}$$

Update - Assume that:

- $L_{\mathbf{z}}(\mathbf{x}) = \mathcal{N}(\mathbf{z}; H_k\mathbf{x}, R_k)$
- P_D is constant

If the predicted PHD at time $k+1$ is a Gaussian mixture of the form $D_{k+1|k}(\mathbf{x}) = \sum_{i=1} J_{k+1|k} w_{k+1|k}^i \mathcal{N}(\mathbf{x}; m_{k+1|k}^i, P_{k+1|k}^i)$, then the updated PHD,

$$D_{k+1|k+1}(\mathbf{x}) = (1 - p_D(\mathbf{x})) D_{k+1|k}(\mathbf{x}) + \sum_{\mathbf{z} \in Z_{k+1}} \frac{p_D(\mathbf{x}) L_{\mathbf{z}} D_{k+1|k}(\mathbf{x})}{\lambda\kappa(\mathbf{z}) + D_{k+1|k}[p_D L_{\mathbf{z}}]}, \tag{B.25}$$

can be written as

$$D_{k+1|k+1}(\mathbf{x}) = (1-p_D)D_{k+1|k}(\mathbf{x}) + \sum_{\mathbf{z}\in Z_{k+1}} D_{D,k}(\mathbf{x};\mathbf{z}), \tag{B.26}$$

where

$$D_{D,k}(\mathbf{x};\mathbf{z}) = \sum_{i=1}^{J_{k+1|k}} w_k^i(\mathbf{z})\mathcal{N}(\mathbf{x}; m_{k+1|k+1}^i(\mathbf{z}), P_{k+1|k+1}^i), \tag{B.27}$$

$$w_k^i(\mathbf{z}) = \frac{p_D w_{k+1|k}^i q_k^i(\mathbf{z})}{\kappa(\mathbf{z}) + p_D \sum_{j=1}^{J_{k+1|k}} w_{k+1|k}^j q_k^j(\mathbf{z})}, \tag{B.28}$$

$$q_k^i(\mathbf{z}) = \mathcal{N}(\mathbf{z}; H_k m_{k+1|k}^i, R_k + H_k P_{k+1|k}^i H_k^T), \tag{B.29}$$

$$m_{k+1|k+1}^i(\mathbf{z}) = m_{k+1|k}^i + K_k^i(z - H_k m_{k+1|k}^i), \tag{B.30}$$

$$P_{k+1|k+1}^i = [\mathbb{I} - K_k H_k] P_{k+1|k}^i, \tag{B.31}$$

$$K_k^i = P_{k+1|k}^i H_k^T (H_k P_{k+1|k}^i H_k^T + R_k)^{-1}. \tag{B.32}$$

Proof - It only needs to be shown that

$$D_{D,k}(\mathbf{x},\mathbf{z}) = \frac{P_D L_{\mathbf{z}}(\mathbf{x}) D_{k+1|k}(\mathbf{x})}{\lambda\kappa(z) + D_{k+1|k}[P_D L_{\mathbf{z}}(\mathbf{x})]}. \tag{B.33}$$

Using the form of $D_{k+1|k}(\mathbf{x})$ and the assumptions:

$$\frac{P_D L_{\mathbf{z}}(\mathbf{x}) D_{k+1|k}(\mathbf{x})}{\lambda\kappa(\mathbf{z}) + D_{k+1|k}[P_D L_{\mathbf{z}}(\mathbf{x})]} \tag{B.34}$$

$$= \frac{p_D \mathcal{N}(\mathbf{z}; H_k\mathbf{x}, R_k) \sum_{i=1}^{J_{k+1|k}} w_{k+1|k}^i \mathcal{N}(\mathbf{x}; m_{k+1|k}^i, P_{k+1|k}^i)}{\lambda\kappa(\mathbf{z}) + \int p_D \mathcal{N}(\mathbf{z}; H_k\mathbf{x}, R_k) \cdot \sum_{i=1}^{J_{k+1|k}} w_{k+1|k}^i \mathcal{N}(\mathbf{x}; m_{k+1|k}^i, P_{k+1|k}^i) \mathrm{d}\mathbf{x}} \tag{B.35}$$

$$= \frac{p_D \sum_{i=1}^{J_{k+1|k}} w_{k+1|k}^i \mathcal{N}(\mathbf{z}; H_k\mathbf{x}, R_k) \mathcal{N}(\mathbf{x}; m_{k+1|k}^i, P_{k+1|k}^i)}{\lambda\kappa(\mathbf{z}) + p_D \int w_{k+1|k}^i \mathcal{N}(\mathbf{z}; H_k\mathbf{x}, R_k) \mathcal{N}(\mathbf{x}; m_{k+1|k}^i, P_{k+1|k}^i) \mathrm{d}\mathbf{x}}. \tag{B.36}$$

Therefore, using (2.19)

$$\frac{P_D L_{\mathbf{z}}(\mathbf{x}) D_{k+1|k}(\mathbf{x})}{\lambda\kappa(\mathbf{z}) + D_{k+1|k}[P_D L_{\mathbf{z}}(\mathbf{x})]} \tag{B.37}$$

$$= \frac{P_D \sum_{i=1}^{J_{k+1|k}} w_{k+1|k}^i q_k^i(\mathbf{z}) \mathcal{N}(\mathbf{x}; \tilde{m}_{k+1|k}^i, \tilde{P}_{k+1|k}^i)}{\lambda\kappa(\mathbf{z}) + p_D \sum_{i=1}^{J_{k+1|k}} w_{k+1|k}^i \int q_k^i(\mathbf{z}) \mathcal{N}(\mathbf{x}; \tilde{m}_{k+1|k}^i, \tilde{P}_{k+1|k}^i) d\mathbf{x}} \tag{B.38}$$

$$= \frac{P_D \sum_{i=1}^{J_{k+1|k}} w_{k+1|k}^i q_k^i(\mathbf{z}) \mathcal{N}(\mathbf{x}; \tilde{m}_{k+1|k}^i, \tilde{P}_{k+1|k}^i)}{\lambda\kappa(\mathbf{z}) + p_D \sum_{i=1}^{J_{k+1|k}} w_{k+1|k}^i q_k^i(\mathbf{z})} \tag{B.39}$$

where $\tilde{m}_{k+1|k}^i$ and $\tilde{P}_{k+1|k}^i$ have the required form $\square$.

B.3 Derivation of the particle TBD filter as a multitarget Bayes filter

Prediction Step - The following assumptions are made:

- If no target is present, one appears with probability p_B and a distributed according to the target birth density γ.
- If a target is present with state $\mathbf{x}'$, it survives to the next time step with probability p_S in which case its state is distributed according to $p_{k+1|k}(\mathbf{x}|\mathbf{x}')$. Otherwise it disappears.

The aim is to calculate the multitarget Markov density $f(X|X')$. First, consider the belief mass function for the set of birth targets, denoted B, noting that birth targets are only possible when there is no target already present:

$$\begin{aligned} \beta_B(T|X' = \emptyset) &= p(B \subseteq T|\emptyset) && \text{(B.40)} \\ &= p_B p(\{\mathbf{x}\} \subseteq T) + 1 - p_B && \text{(B.41)} \end{aligned}$$

and hence,

$$\frac{\delta\beta_B}{\delta B}(T|\emptyset) = \begin{cases} p_B p(\{\mathbf{x}\} \subseteq T) + 1 - p_B & \text{if } B = \emptyset \\ p_B \gamma(\mathbf{x}) & \text{if } B = \{\mathbf{x}\} \end{cases} \tag{B.42}$$

using (3.36). Therefore, by (3.31):

$$f(X|\emptyset) = \frac{\delta\beta_B}{\delta B}(\emptyset|\emptyset) = \begin{cases} 1 - p_B & \text{if } X = \emptyset \\ p_B \gamma(\mathbf{x}) & \text{if } X = \{\mathbf{x}\} \end{cases} \tag{B.43}$$

Similarly, starting with the probability mass function for the set of surviving targets, denoted Y:

$$\begin{aligned} \beta_Y(T|X' = \{\mathbf{x}'\}) &= p(Y \subseteq T|\{\mathbf{x}'\}) && \text{(B.44)} \\ &= p_s p(\{\mathbf{y}\} \subseteq T) + 1 - p_s && \text{(B.45)} \end{aligned}$$

and hence,

$$\frac{\delta\beta_Y}{\delta Y}(T|\emptyset) = \begin{cases} p_S p(\{\mathbf{y}\} \subseteq T) + 1 - p_S & \text{if } Y = \emptyset \\ p_S p_{k+1|k}(\mathbf{y}|\mathbf{x}') & \text{if } Y = \{\mathbf{y}\} \end{cases} \tag{B.46}$$

using (3.36). Therefore, by (3.31):

$$f(X|\mathbf{x}') = \frac{\delta\beta_Y}{\delta Y}(\emptyset|\emptyset) = \begin{cases} 1 - p_S & \text{if } X = \emptyset \\ p_S p_{k+1|k}(\mathbf{y}|\mathbf{x}') & \text{if } X = \{\mathbf{x}\} \end{cases} \tag{B.47}$$

Finally, it is clear that:

$$f(X|X') = p_T f(X|\{x'\}) + (1 - p_T) f(X|\emptyset) \tag{B.48}$$

where p_T is the probability that there is a target present. Putting all of this together gives a prediction step equivalent to the one described in [105].

Update Step: The update step is easier because the measurement set is not a random finite set. It is known that a measurement will be received for each pixel and that the only options are that any pixel corresponds to a target, or that none of them do. The relative probabilities of these options can then be simply calculated in the way described in [105].

B.4 Unscented Kalman Filter

Let $\mathbf{x}_k$ be an N_x dimensional Gaussian random variable with mean $\hat{\mathbf{x}}_{k|k}$ and variance $\mathbf{P}_{k|k}$. A set of $2N_x + 1$ sample points are chosen. These are called sigma points and are deterministically chosen so that they have properties that allow information about the distribution of the random variable to be captured by this small number of points. The $2N_x + 1$ sigma points, χ_k^i and their respective weights W^i are given by:

$$\chi_k^0 = \hat{\mathbf{x}}_{k|k} \tag{B.49}$$

$$W^0 = \frac{\kappa}{N_x + \kappa} \tag{B.50}$$

$$\chi_k^i = \hat{\mathbf{x}}_{k|k} + \left(\sqrt{(N_x + \kappa)\mathbf{P}_{k|k}}\right)_i \tag{B.51}$$

$$W^i = \frac{1}{2(N_x + \kappa)} \tag{B.52}$$

$$\chi_k^{i+N_x} = \hat{\mathbf{x}}_{k|k} - \left(\sqrt{(N_x + \kappa)\mathbf{P}_{k|k}}\right)_i \tag{B.53}$$

$$W^{i+N_x} = \frac{1}{2(N_x + \kappa)} \tag{B.54}$$

For $i = 1, ..., N_x$, where κ is a tunable parameter and $\left(\sqrt{(N_x + \kappa)\mathbf{P}_{k|k}}\right)_i$ is the ith row of the matrix square root of $(N_x + \kappa)\mathbf{P}_{k|k}$.

It will be shown that this weighted set of sigma point has the same mean and covariance, and all higher order odd-ordered central moments as the distribution of $\mathbf{x}_k$.

As the points are symmetrically distributed and chosen with equal weights about $\hat{\mathbf{x}}_{k|k}$, it is clear that the sample mean is $\hat{\mathbf{x}}_{k|k}$ and that the odd-ordered moments are all zero, so it only remains to be shown that the sample covariance is equal to $\mathbf{P}_{k|k}$.

The sample covariance, $\mathbf{P}$ is

$$\begin{aligned}
\mathbf{P} &= \sum_{i=0}^{2N_x} W^i [\chi^i_{k|k} - \hat{\mathbf{x}}_{k|k}][\chi^i_{k|k} - \hat{\mathbf{x}}_{k|k}]^T && \text{(B.55)} \\
&= \sum_{i=1}^{N_x} 2W^i(N_x + \kappa)\big(\sqrt{\mathbf{P}_{k|k}}\big)_i \big(\sqrt{\mathbf{P}_{k|k}}\big)_i^T && \text{(B.56)} \\
&= \mathbf{P}_{k|k} && \text{(B.57)}
\end{aligned}$$

which proves the claim.

The unscented Kalman filter (UKF) is given by:

$$\begin{aligned}
\hat{\mathbf{x}}_{k+1|k} &= \sum_{i=0}^{2N_x} W^i f_k(\chi^i_k) && \text{(B.58)} \\
\mathbf{P}_{k+1|k} &= \mathbf{Q}_{k+1} + \sum_{i=0}^{2N_x} W^i [\mathbf{f}_k(\chi^i_k) - \hat{\mathbf{x}}_{k|k}][\mathbf{f}_k(\chi^i_k) - \hat{\mathbf{x}}_{k|k}]^T && \text{(B.59)} \\
\chi^i_{k+1|k} &= \mathbf{f}_{k+1}(\chi^i_k) && \text{(B.60)} \\
\hat{\mathbf{z}}_k &= \sum_{i=0}^{2N_x} W^i \mathbf{h}_k(\chi^i_{k+1|k}) && \text{(B.61)} \\
\hat{\mathbf{x}}_{k+1} &= \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1}(\mathbf{z}_k - \hat{\mathbf{z}}_k) && \text{(B.62)} \\
\mathbf{P}_{k+1} &= \mathbf{P}_{k+1|k} - \mathbf{K}_{k+1}\mathbf{S}_{k+1}\mathbf{K}^T_{k+1} && \text{(B.63)}
\end{aligned}$$

where

$$\begin{aligned}
\mathbf{S}_{k+1} &= \mathbf{R}_{k+1} + \sum_{i=0}^{2N_x} W^i [\mathbf{h}_{k+1}(\chi^i_{k+1|k}) - \hat{\mathbf{z}}_k][h_{k+1}(\chi^i_{k+1|k}) - \hat{\mathbf{z}}_k]^T && \text{(B.64)} \\
\mathbf{K}_{k+1} &= \Big(\sum_{i=0}^{2N_x} W^i [\chi^i_{k+1|k} - \hat{\mathbf{x}}_{k+1|k}][h_{k+1}(\chi^i_{k+1|k}) - \hat{\mathbf{z}}_k]^T \Big) \mathbf{S}^{-1}_{k+1} && \text{(B.65)}
\end{aligned}$$

B.5 Gaussian Sum Filter

In order to compute the Gaussian sum approximation to the Bayes filter, assume that the Markov density is given by the Gaussian sum

$$f_{k+1|k}(\mathbf{x}|\mathbf{y}) = \sum_{i=1}^{T_k} \tau^i_k \mathcal{N}(\mathbf{x}; F_k\mathbf{y}, Q^i_k), \tag{B.66}$$

where $\tau^i_k > 0$, and $\sum_{i=1}^{T_k} \tau^i_k = 1$. Also assume that the posterior from time k is a Gaussian sum of the form

$$f_{k|k}(\mathbf{x}|Z^k) = \sum_{i=1}^{N_{k|k}} w^i_{k|k} \mathcal{N}(\mathbf{x}; \mathbf{x}^i_{k|k}, P^i_{k|k}). \tag{B.67}$$

Plugging these into (2.13) gives

$$f_{k+1|k}(\mathbf{x}|Z^k) = \sum_{i=1}^{T_k}\sum_{j=1}^{N_{k|k}} \tau_k^i w_{k|k}^j \int \mathcal{N}(\mathbf{x}; F_k y, Q_k^i)\mathcal{N}(\mathbf{x}; \mathbf{x}_{k|k}^j, P_{k|k}^j) \quad \text{(B.68)}$$

$$= \sum_{i=1}^{T_k}\sum_{j=1}^{N_{k|k}} \tau_k^i w_{k|k}^j \mathcal{N}(\mathbf{x}; F_k \mathbf{x}_{k|k}^j, Q_k^i + F_k P_{k|k}^j F_k^T. \quad \text{(B.69)}$$

This follows from (2.19) in the same manner as presented in Section 2.1.2.

The posterior density can be calculated in a similar way using (2.19) and following the algebra from Section 2.1.2 for each component in a double sum as above.

Appendix C

Indicative Computation times for GM-PHD and SMC-PHD

It can be seen from (4.75) and (4.68) that the computational complexity of the GM-PHD and SMC-PHD filters should both be linearly dependent on the number of targets and number of measurements.

In order to give some an indication of the computation time required for an implementation of the PHD filter, the computation time was measured for GM-PHD and (improved) SMC-PHD implementations over a range of clutter levels and target numbers. The trials were for a 10 time step run and were computed using a machine with an AMD Phenom(tm) II X3 720 2.8GHz Processor.

Let M be the average number of false alarms per time step, N be the number of targets and P the number of particles per target (for SMC-PHD). Denote the approximate computation times for GM-PHD and (improved) SMC-PHD by $C_{GM}(M, N)$ and $C_{SMC}(M, N, P)$ respectively. The linear dependence on M, N and P that was expected was found over the range of tests and the relationship was given by

$$C_{GM} = 6 \times 10^{-4} \times M \times N \tag{C.1}$$

$$C_{SMC} = 5 \times 10^{-3} \times M \times N \times (P/1000) \tag{C.2}$$

Bibliography

[1] D. Akselrod, A. Sinha, T. Kirubarajan, M. Farooq, and Z.J. Ding. A distributed multisensor-multitarget tracking testbed for maritime surveillance. *Proceedings of the Society of Photo-Optical Instrumentation Engineers (SPIE)*, 5809:111–122, 2005.

[2] B.D. Anderson and J.B. Moore. *Optimal Filtering*. Prentice-Hall, 1979.

[3] J.P. Armitage and H.L. Packer. *Bacterial Motility and Chemotaxis*. Wiley-Blackwell, 1998.

[4] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035, 2007.

[5] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions On Signal Processing*, 50:174–188, 2002.

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

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

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

[9] G. Battistelli, L. Chisci, S. Morrocchi, F. Papi, A. Benavoli, A. Di Lalla, A. Farina, and A. Graziano. Traffic intensity estimation via phd filtering. In *Proceedings of the 5th European Radar Conference*, 2008.

[10] H.C. Berg. How to track bacteria. *Review of Scientific Instruments*, 42:868871, 1971.

[11] H.C. Berg and D.A. Brown. Chemotaxis in escherichia coli analysed by three-dimensional tracking. *Nature*, 239:500–504, 1972.

[12] D.P. Bertsekas. *Linear Network Optimization*. MIT Press, 1991.

[13] D. Billon. Hmm automatic detection and tracking for passive sonar. In *UDT Europe*, 2008.

[14] S. Blackman. *Multiple Target Tracking with Radar Applications*. Artech House, 1986.

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

[16] S.S. Blackman, R.J. Dempster, and S.H. Roszkowski. Imm/mht applications to radar and ir multitarget tracking. *Proceedings of the Society of Photo-Optical Instrumentation Engineers (SPIE)*, 3163:429–439, 1997.

[17] H.A.P. Blom and Y. Bar-Shalom. The interacting multiple model algorithm for systems with markovian switching coefficients. *IEEE Transactions on Automatic Control*, 33(8):780–783, August 1988.

[18] Z.I. Botev, J.F. Grotowski, and D.P. Kroese. Kernel density estimation via diffusion. *Annals of Statistics*, 38:2916–2957, 2009.

[19] A.W. Bowman and A. Azzalini. *Applied Smoothing Techniques for Data Analysis*. Oxford University Press, 1997.

[20] S. Chapman. On the brownian displacements and thermal diffusion of grains suspended in non-uniform fluid. *Proceedings of the Royal Society A*, 119:34–54, 1928.

[21] A.A. Chernova, J.P. Armitage, H.L. Packer, and P.K. Maini. Response kinetics of tethered bacteria to stepwise changes in nutrient concentration. *Biosystems*, 71:51–59, 2003.

[22] D. Clark, B. Ristic, and B.-N. Vo. Phd filtering with target amplitude feature. In *Proceedings of the 11th International Conference on Information Fusion*, 2008.

[23] D. Clark, I.T. Ruiz, Y. Petillot, and J. Bell. Particle phd filter multiple target tracking in sonar image. *IEEE Transactions on Aerospace and Electronic Systems*, 43(1):409–416, 2007.

[24] D.E Clark. *Multiple target tracking with the probability hypothesis density filter.* PhD thesis, Heriot-Watt University, 2006.

[25] D.E. Clark and J. Bell. Data association for the phd filter. *Proceedings of the 2005 Intelligent Sensors, Sensor Networks & Information Processing Conference,*, 1:217–222, 2005.

[26] D.E Clark and J. Bell. Convergence results for the particle phd filter. *IEEE Transaction on Signal Processing*, 54:2652–2661, 2006.

[27] D.E. Clark, J. Bell, Y. de Saint-Pern, and Y. Petillot. Phd filter multi-target tracking in 3d sonar. In *Oceans 2005 Europe International Conference*, pages 265–270, 2005.

[28] D.E. Clark and S. Godsill. Group target tracking with the gaussian mixture probability hypothesis density filter. *Proceedings of the International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, 1:149–154, 2007.

[29] D.E Clark, B. Ristic, B.-N. Vo, and B.T. Vo. Bayesian multi-object filtering wth amplitude feature likelihood for unknown object snr. *IEEE Transactions on Signal Processing*, 58:26–37, 2010.

[30] D.E. Clark and B. Vo. Convergence analysis of the gaussian mixture phd filter. *IEEE Transations on Signal Processing*, 55:1204–1212, 2007.

[31] D. Crisan and A. Doucet. A survey of convergeance results on particle filtering methods for practitioners. *IEEE Transactions on Signal Processing*, 50:736–746, 2002.

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

[33] M. de Feo, A. Graziano, R. Miglioli, and A. Farina. Immjpda versus mht and kalman filter with nn correlation: performance comparison. *IEE Proceedings on Radar, Sonar and Navigation*, 144:49–56, 1997.

[34] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistics Society*, 39:1–38, 1977.

[35] D.L. Donoho. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41:613–627, 1995.

[36] A. Doucet, S. Godsill, and C. Andrieu. On sequential monte carlo sampling for bayesian filtering. *Statistics and Computing*, 10:197–208, 2000.

[37] A. Doucet and A.M. Johansen. A tutorial on particle filtering and smoothing: fifteen years later. In *The Oxford Handbook of Nonlinear Filtering*, 2011.

[38] O. Erdinc, Willett P., and Y. Bar-Shalom. Probability hypothesis density filter for multitarget multisensor tracking. *Proceedings of the 7th International Conference on Information Fusion (FUSION)*, 1:146–153, 2005.

[39] L. Etgar, A. Nakhmani, A. Tannenbaum, E. Lifschitz, and R. Tannenbaum. Trajectory control of pbse$-\gamma - fe_2o_3$ nanoplatforms under viscous flow and an external magnetic field. *Nanotechnology*, 21:1–9, 2010.

[40] G. Evensen. *Data Assimilation: The Ensemble Kalman Filter*. Springer, 2007.

[41] F. Folster and H. Rohling. Data association and tracking for automotive radar networks. *IEEE Transactions on Intelligent Transportation Systems*, 6(4):370–377, 2005.

[42] A. Goobic, M. Welser, S. Acton, and K. Ley. Biomedical application of target tracking in clutter. In *35th Asilomar Conference on Signals, Systems & Computers*, pages 4–7, 2001.

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

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

[45] M.S. Grewal and P.A. Andrews. *Kalman Filtering: Theory and Practise Using MATLAB*. Wiley, 2001.

[46] B. Hammarberg, C. Forster, and E. Torebjork. Parameter estimation of human nerve cfibers using matched filtering and multiple hypothesis tracking. *IEEE Transactions on Biomedical Engineering*, 49:329–336, 2002.

[47] T. L. Hill. *Statistical Mechanics: Principles and Selected Applications*. Dover Publications, 1987.

[48] Y. C. Ho and R.C.K. Lee Lee. A bayesian approach to problems in stochastic estimation and control. *IEEE Transactions on Automatic Control*, 9:333–339, 1964.

[49] J.R. Hoffman and R.P.S. Mahler. Multitarget miss distance via optimal assignment. *IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans*, 34:3, 2004.

[50] K. Jaqaman, D. Loerke, M. Mettlen, H. Kuwata, S. Grinstein, S.L. Schmidt, and G. Danuser. Robust single-particle tracking in live-cell time-lapse sequences. *Nature Methods*, 5:695–702, 2008.

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

[52] B. Jida, R. Lherbier, M. Wahl, and J.-C. Noyer. Bayesian networks and probabilistic data association methods for multi-object tracking: Application to road safety. In *3rd International Conference on Information and Communication Technologies: From Theory to Applications*, 2008.

[53] A. Johansen, S. Singh, A. Doucet, and B.-N. Vo. Convergence of the smc-phd filter. *Methodology and Computing in Applied Probability*, 8:265–291, 2006.

[54] R. Juang, A. Levchenko, and P. Burlina. Tracking cell motion using gm-phd. *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 1:1154–1157, 2009.

[55] S.J. Julier and J.K. Uhlmann. A new extension of the kalman filter to nonlinear systems. *Proceedings of the Society of Photo-Optical Instrumentation Engineers*, 3068:182–193, 1997.

[56] S.J. Julier and J.K. Uhlmann. A new method for the nonlinear transformation of means and covariances in filters and estimation. *IEEE Transcations on Automatic Control*, 43:477–482, 2000.

[57] S.J. Julier and J.K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE Aerospace and Electronic Systems Magazine*, 92:401–422, 2004.

[58] R.E. Kalman. A new approach to linear filtering and prediction problems. *ASME Journal of Basic Engineering*, 82:34–35, 1960.

[59] T. Kanugo, D.M. Mount, N. Netanyahu, C. Piatko, and A.Y. Wu. A local search approximation algorithm for k-means clustering. *Proceedings of the 18th Annual ACM Symp. on Computational Geometry*, 1:1018, 2002.

[60] D.B. Kearns. A field guide to bacterial swarming motility. *Nature Reviews Microbiology*, 8:634–644, 2010.

[61] D. Kocak, N. Lobo, and E. Widder. Computer vision techniques for quantifying, tracking, and identifying bioluminescent plankton. *IEEE Journal. of Ocean. Engineering*, 24:81–95, 1999.

[62] W. Koch. On bayesian mht for well-seperated targets in densely cluttered environment. In *Proceedings of the IEEE International Radar Conference*, pages 323–328, 1995.

[63] A.N. Kolmogorov. On analytic methods in probability theory. *Uspeki Matematicheskih Nauk*, 5:5–41, 1938.

[64] J. Leonard and H. Durrant-Whyte. Application of multi-target tracking to sonar based mobile robot navigation. In *Proceedings of the 29th Conference on Decision & Control*, pages 3118–3123, 1990.

[65] D. Lerro and Y. Bar-Shalom. Tracking with debiased consistent converted measurements versus ekf. *IEEE Transactions on Aerospace and Electronic System*, 29:1015–1022, 1993.

[66] H. Lew and D.M. Drunheller. Estimation of non-rayleigh clutter and fluctuating-target models. *IEE Proceeding on Radar, Sonar and Navigation*, 149:231–241, 2002.

[67] X.R. Li and V.P. Jilkov. Survey of maneuvering target tracking. part i: Dynamic models. *IEEE Transactions on Aerospace and Electronic Systems*, 39:1333–1364, 2003.

[68] Jae S. Lim. *Two-Dimensional Signal and Image Processing*. Prentice-Hall, 1990.

[69] X. Lin, T. Kirubarajan, Y. Bar-Shalom, and S. Maskell. Comparison of ekf, pseudomeasurement and particle filters for a bearing-only target tracking problems. *Proceedings of the Society of Photographic Instrumentation Engineers*, 4728:240–250, 2002.

[70] W.K. Ma, B. Vo, S. Singh, and A. Baddeley. Tracking an unknown time-varying number of speakers using tdoa measurements. *IEEE Transactions on Signal Processing*, 54(9):3291–3304, 2006.

[71] E. Maggio, M. Taj, and A. Cavallaro. Efficient multitarget visual tracking using random finite sets. *IEEE Transactions on Circuits and Systems for Video Technology*, 18:1016–1027, 2008.

[72] R.P.S. Mahler. Multitarget moments and their application to multitarget tracking. In *Proceedings of the Workshop on Estimation, Tracking and Fusion: A Tribute to Yaakov Bar-Shalom*, 2001.

[73] R.P.S. Mahler. Multi-target bayes filtering via first-order multi-target moments. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1152–1178, 2003.

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

[75] R.P.S. Mahler. *Statistical Multisource-Multitarget Information Fusion*. Artech House, 2007.

[76] D. Maksarov and H. Durrant-Whyte. Mobile vehicle navigation in unknown environments - a multiple hypothesis approach. *IEE Proceedings: Control Theory and Applications*, 142(4):385–400, 1995.

[77] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 1999.

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

[79] J. Moyal. The general theory of stochastic population processes. *Acta Mathematica*, 108:1–31, 1962.

[80] D. Musicki, R. Evans, and S. Stankovic. Integrated probabilistic data association. *IEEE Transactions on Automatic Contol*, 39(6):1237–1241, 1994.

[81] D. Musicki, Wang X.Z., R. Ellem, and F. Fletcher. Efficient active sonar multitarget tracking. In *Oceans 2006-Asia Pacific Conference*, 2006.

[82] J.C. Naylor and A.F.M. Smith. Application of a method for the efficient computation of posterior distributions. *Applied Statistics*, 31(3):214–225, 1982.

[83] M. Nicoli, V. Rampa, and Spagnolini U. Hidden markov model for multidimensional wavefront tracking. *IEEE Transactions on Geoscience & Remote Sensing*, 40:651–662, 2002.

[84] A.V. Oppeneheim and R.W. Shafer. *Discrete-Time Signal Processing*. Prentice-Hall, 1989.

[85] E. Oron, Y. Bar-Shalom, and M. Lachish. Advanced ir imaging and tracking for hypersonic intercept. In *1997 IEEE Aerospace Conference Proceedings*, 1997.

[86] N.T. Ouellette, H. Xu, and E. Bodenschatz. A quantitative study of three-dimensional lagrangian particle tracking algorithms. *Experiments in Fluids*, 40:301–313, 2006.

[87] K. Panta, D.E Clark, and B. Vo. Data association and track management for the gaussian mixture probability hypothesis density filter. *IEEE Transactions on Aerospace and Electronic Systems*, 45(3):1003–1016, 2009.

[88] K Panta, B. Vo, and S. Singh. Improved probability hypothesis density (phd) filter for multitarget tracking. *Proceedings of the 3rd International Conference on Intelligent Sensing and Information Processing*, 1:213–218, 2005.

[89] K Panta, B. Vo, S. Singh, and A. Doucet. Probability hypothesis density versus multiple hypothesis tracking. *Signal Processing, Sensor Fusion and Target Recognition XIII*, 5429:284–295, 2004.

[90] S. A. Pasha, B.-N. Vo, H. D. Tuan, and W.-K. Ma. A gaussian mixture phd filter for jump markov system models. *IEEE Transactions on Aerospace and Electronic Systems*, 45(3):919–936, July 2009.

[91] S.L. Porter, G.H. Wadhams, and J.P. Armitage. Rhodobacter spheroides: complexity in chemotactic signalling. *Trends in Microbiology*, 16:677–687, 2008.

[92] J. G. Proakis and D.K. Manolakis. *Digital Signal Processing*. Prentice-Hall, 2006.

[93] K. Punithakumar, T. Kirubarajan, and A. Sinha. Multiple-model probability hypothesis density filter for tracking maneuvering targets. *IEEE Transactions on Aerospace and Electronic Systems*, 44(1):87–98, January 2008.

[94] B. Ristic. Notes from short course on tracking and data fusion. University College London, May 2008.

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

[96] B. Ristic, D.E. Clark, and B.-N. Vo. Improved smc implementation of the phd filter. In *Proceedings of the 13th International Conference on Information Fusion*, 2010.

[97] B. Ristic, B.-N. Vo, and Clark D. Performance evaluation of mulit-target tracking using the ospa metric. In *Proceedings of the 13th International Conference on Information Fusion*, 2010.

[98] C.P. Robert and G Casella. *Monte Carlo Statistical Methods*. Springer-Verlag, 2004.

[99] J. Roecker. Suboptimal joint probabilistic data association. *IEEE Transcations on Aerospace and Electronic Systems*, 29(2):510–517, 1993.

[100] J. Roecker. A class of near optimal jpda algorithms. *IEEE Transcations on Aerospace and Electronic Systems*, 30(2):504–510, 1994.

[101] G. Rosser, C.A. Yates, T.M. Wood, and D.A. Wilkinson. Analysis of free swimming bacteria trajectories in video data. In preparation.

[102] L.H. Ryder. *Quantum Field Theory*. Cambridge University Press, 1996.

[103] D. Salmond. *Tracking in Uncertain Environments*. PhD thesis, University of Sussex, 1989.

[104] D. Salmond. Mixture reduction algorithms for target tracking in clutter. *Signal & Data Processing of Small Targets, Proc. SPIE*, 1305:434–445, 1990.

[105] D.J. Salmond and H. Birch. A particle filter for track-before-detect. *Proceedings of the American Control Conference*, 1:3755–3760, 2001.

[106] M.J. Saxton. Single-particle tracking: connecting the dots. *Nature Methods*, 5:671–672, 2008.

[107] D. Schuhmacher, B.-T. Vo, and B.-N. Vo. A consistent metric from performance evaluation of multi-object filters. *IEEE Transactions on Singal Processing*, 56:3447–3457, 2008.

[108] K. Shafique and M. Shah. A noniterative greedy algorithm for multiframe point correspondance. *IEEE. Transactions on Pattern Analysis and Machine Intelligence*, 466:1725–1748, 2005.

[109] H. Shum, E.A. Gaffney, and D.J. Smith. Modelling bacterial behaviour close to a no-slip plane boundary: the influence of bacterial geometry. *Proceedings of the Royal Society A*, 466:1725–1748, 2010.

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

[111] H.W. Sorenson and D.L. Alspach. Recursive bayesian estimation using gaussian sums. *Automatica*, 7:465–479, 1971.

[112] U. Spagnolini and V. Rampa. Multitarget detection/tracking for monostatic ground penetrating radar: application to pavement profiling. *IEEE Transactions on Geoscience & Remote Sensing*, 37:383–394, 1999.

[113] L.A. Steen and J.A. Seebach. *Counterexamples in Topolgy*. Rhinehart and Winston, Inc., 1970.

[114] Y. Sun and P. Willett. Hough transform for long chirp detection. *IEEE Transactions on Aerospace and Electronic Systems*, 38:553–569, 2002.

[115] A. Swain and D. Clark. Extended object filtering using spatial independent cluster processes. In *Proceedings of the 13th International Conference on Information Fusion*, 2010.

[116] B. Taboada, S. Poggio, L. Camarena, and G. Cordiki. Automatic tracking and analysis system for free-swimming bacteria. In *Proceedings of the 25th Annual Conference of the IEEE EMBS*, 2003.

[117] M. Tobias and L.D. Lanterman. Probability hypothesis density-based multitarget tracking with bistatic range and doppler observations. *IEE Proceedings: Radar, Sonar and Navigation*, 152:195–205, 2005.

[118] S.M. Tonissen and Y. Bar-Shalom. Maximum likelihood track-before-detect with fluctuating target amplitude. *IEEE Transactions on Aerospace and Electronic System*, 34(3):796–809, 1998.

[119] H.L. Van Trees. *Optimum Array Processing*. Wiley, 2002.

[120] N. Verstraeten, K. Braeken, B. Debkumari, M. Fauvart, J. Fransaer, J. Vermant, and J. Michiels. Living on a surface: Swarming and biofilm formation. *Trends in Microbiology*, 16:496–506, 2008.

[121] M. Vihola. Rao-blackwellised particle filtering in random set multitarget tracking. *IEEE Transcations on Aerospace and Electronic Systems*, 43(2):689–705, 2007.

[122] B. Vo and W. Ma. The gaussian mixture probability hypothesis density. *IEEE Transactions on Signal Processing*, 54:4091–4104, 2006.

[123] B. Vo, A. Pasha, and H.D. Tuan. A gaussian mixture phd filter for nonlinear jump markov models. *Proceedings of the 45th IEEE Conference on Decision and Control*, 1-14:6063–6068, 2006.

[124] B. Vo, Singh S., and A. Doucet. Sequential monte carlo methods for multitarget filtering with random finite sets. *IEEE Transactions on Aerospace and Electronic Systems*, 41:1224–1245, 2005.

[125] B.-N. Vo and S. Singh. On the bayes filtering equations of finite set statistics. In *Proceedings of the 5th Asian Control Conference*, 2004.

[126] B.-N. Vo, B.-T. Vo, N.-T. Pham, and D. Suter. Joint detection and estimation of multiple object from image observations. *IEEE Transactions on Signal Processing*, 58:5129–5141, 2010.

[127] B.-T. Vo, B.-N. Vo, and A. Cantoni. The cardinality balanced multi-target multi-bernoulli filter and its implementation. *IEEE Transactions on Signal Processing*, 57:409–423, 2009.

[128] B.T. Vo. *Random Finite Sets in Multi-Object Filtering*. PhD thesis, The University of Western Australia, 2008.

[129] Y.-D. Wang, J.-K. Wu, A.A. Kassim, and W. Huang. Data-driven probability hypothesis density filter for visual tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 18:1085–1095, 2008.

[130] N. Whiteley, S. Singh, and S. Godsill. Auxiliary particle implementation of probability hypothesis density filter. *IEEE Transactions on Aerospace and Electronic Systems*, 46:1437–1454, 2010.

[131] T.M. Wood. Interacting methods for manoeuvre handling in the gm-phd filter. *IEEE Transactions on Aerospace and Electronic Systems*, 47(4):3021–3025.

[132] T.M. Wood. Mathematical modelling of single target sonar and radar tracking. DPhil Transfer of Status Thesis.

[133] T.M Wood. Tracking in dense clutter with the phd filter. In *Proceedings of the IMA Mathematics in Defence Conference 2009*, 2009.

[134] T.M. Wood. Detecting and tracking multiple stealthy targets: comparison of phd filter and track-before-detect approaches. In *To appear in Proceedings of the IMA Mathematics in Defence Conference*, 2011.

[135] T.M Wood, D. Allwright, P. Bond, S. Long, and I. Moroz. A new method for processing passive sonar. In *Proceedings of the 13th International Conference on Information Fusion*, 2010.

[136] T.M. Wood, D.E. Clark, and B. Ristic. Efficient resampling and basic track continuity for the smc-phd filter. In *Proceedings of the 2010 Conference on Cognitive Systems with Interactive Sensors (COGIS)*, 2010.

[137] T.M. Wood, C.A Yates, D.A. Wilkinson, and G. Rosser. Simplified multitarget tracking using the phd filter for microscopic video data. In Press, Submitted to IEEE Transactions on Circuits and Systems for Video Technology.

[138] J. Xie, S. Khan, and M. Shah. Automatic tracking of escherichia coli in phase contrast microscopy. *IEEE Transactions on Biomedical Engineering*, 56:390–399, 2009.

[139] Y. Xie. A range-dependent echo-association algorithm and its application in split-beam sonar tracking of migratory salmon in the fraser river watershed. *IEEE Journal. of Ocean. Engineering*, 25:387–398, 2000.

[140] R. Xu and D. Wunsch II. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678, 2005.

[141] Y. Zhang, H. Leung, M. Blanchette, and Litva J. Lo, T. An efficient decentralized multiradar multitarget tracker for air surveillance. *IEEE Transactions on Aerospace and Electronic Systems*, 33(4):1357–1363, 1997.

[142] Z. Zhao, X. Rong Li, and V.P. Jilkov. Best linear unbiased filtering with nonlinear measurements for target tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 40:1324–1336, 2004.