

**Faculty of Science and Engineering
School of Electrical Engineering, Computing and Mathematical Sciences**

**Robust Multi-Object Tracking:
A Labeled Random Finite Set Approach**

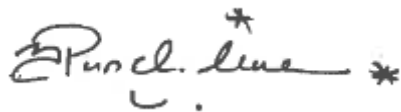
Yuthika Samanmali Gardiyawasam Punchihewa

**This thesis is presented for the Degree of
Doctor of Philosophy
of
Curtin University**

September 2018

DECLARATION

To the best of my knowledge and belief, this thesis contains no material previously published by any other person, except where due acknowledgement has been made. This thesis contains no material which has been accepted for the award of any other degree or diploma in any university.

A handwritten signature in black ink, appearing to read 'Yuthika Samanmali Gardiyawasam Punchihewa', with a small asterisk above the 'i' and another asterisk at the end of the signature.

Yuthika Samanmali Gardiyawasam Punchihewa

6 September 2018

ACKNOWLEDGMENTS

I would like to express my deep appreciation to my supervisors Professor Ba-Ngu Vo and Professor Ba-Tuong Vo. Your enthusiasm and patience kept me motivated when things were slow and tough and your guidance set me in the right directions when I strayed. It was a privilege and pleasure to have you for my mentors.

I would also like to deeply thank Dr Du-Yong Kim, Professor Reza Hoseinnezhad and Dr Francesco Papi for the collaborations and the intellectual discussions.

I also thank Dr Claudio Fantacci, Dr Angel Garcia, Dr Hung Gia Hoang, Dr Michael Beard and Jeff Wong for helping me during the initial stages of my study to clear up doubts and misconceptions.

My special thanks go to Professor Dilusha Silva at the University of Western Australia for the advice and the kind introduction to my supervisors. Without him, I would not have met such great supervisors and this PhD would not be a reality.

I gratefully acknowledge the financial support received from the Department of Electrical and Computer Engineering at Curtin University during this period. I also thank Robyn and the administrative staff from the department for their professional and friendly support.

My parents for raising me, inspiring me and believing in me when I didn't believe in my self. You went above and beyond to be there for me always. You are my pillars of strength.

My special thanks go to Wimal mama, Rupa nanda and Purnima akka for making my initial days in Australia comfortable and to Prasantha ayya and Buddhika akka for helping me with the babysitting. My sister Lochana for making adjustments in her life to make mine convenient. I would also like to remember my buddies from Curtin university Quang, Linh, Lara, Megha, Rajah, Suchi, Nimali and Nimsiri. Our lunchtime conversations were very entertaining and made me forget my troubles for a brief moment.

To Ruchitha, for being a wonderful partner in life, for helping in whichever way possible and for steadfastly cheering me towards the finish line. For the sacrifices you made for me to find success !!!

To Arundee and Apurva, for being my sunshine !!!

ABSTRACT

Multi-object tracking is the problem of tracking and estimation of an unknown and time-varying number of objects and their individual states from a sequence of observations. The nature of the standard multi-object observation, which encompasses measurement noise, missed detections, false detections and association uncertainty, makes this problem quite challenging. The *random finite set* approach for multi-object filtering founded by Mahler is an elegant formulation of the multi-object filtering problem in which the collection of object states is represented as a finite set. In general, the multi-object Bayes filter is intractable due to the inherent combinatorial nature of multi-target densities and the multiple integrations on the infinite dimensional multi-object state and observation spaces.

The recent establishment of the *labeled random finite set* framework for tracking paved the path for a tractable analytic solution to the multi-object Bayes filter popularly known as the *generalized labeled multi-Bernoulli* filter, which is also a *tracker* due to its use of distinct labels in tagging the object states. In its standard form, the generalized labeled multi-Bernoulli filter is capable of estimating the trajectories of an unknown and time-varying number of objects under the standard multi-object dynamic and observation models. As with most multi-object filters, the robustness of the tracking performance of the generalized labeled multi-Bernoulli filter is heavily dependent in certain knowledge and assumptions regarding the multi-object system. Knowledge regarding object dynamics, detection and clutter profile are a few critical examples. The central objective of this dissertation is to present new techniques constructed upon the labeled random finite set framework, for robust tracking, where complete information regarding the multi-object system is not available, thus broadening its applicability in the real world.

The uncertainty of object dynamics, clutter and detection are three phenomena that need to be contended with in robust multi-object tracking. The first contribution of this thesis is a labeled RFS algorithm that operates in a jump Markov setting where the object dynamics are time variant (maneuvering objects). The second contribution of this

dissertation is a labeled RFS based algorithm that operates with no knowledge of the clutter and detection profile. This can be useful since assigning significantly disparate values for clutter and detection model parameters in situations where these parameters are not known and time-varying, could lead to biased estimates. The third contribution is a labeled RFS algorithm that performs tracking on visual data without using detections and thus could also be treated as an alternative technique to the second contribution with regards to visual data. All algorithms produce trajectories in addition to object state estimates.

LIST OF PUBLICATIONS

The following papers have been published based on certain works in this dissertation.

1. **Y. Punchihewa**, B. T. Vo, B. N. Vo, and D. Y. Kim, Multiple Object Tracking in Unknown Backgrounds with Labeled Random Finite Sets, published in IEEE Transactions on Signal Processing, vol.66, no.11, pp. 3040-3055, June 2018.

The author's contribution includes the theoretical development of the algorithm, implementation (MATLAB), evaluation and drafting the paper. The co-authors contributed by way of documenting the theoretical development, editing the paper and surveying for suitable video data sets and object detection algorithms.

2. **Y. Punchihewa**, B.-N. Vo, and B.-T. Vo, A Generalized Labeled Multi-Bernoulli Filter for Maneuvering Targets, in 19th International Conference on Information Fusion, Heidelberg, Germany, July 2016.

The author's contribution includes the theoretical development of the algorithm, implementation (MATLAB), evaluation and drafting the paper. The co-authors contributed by way of editing the paper.

3. **Y. Punchihewa**, Efficient Generalized Labeled Multi-Bernoulli Filter for Jump Markov system, in International Conference on Control, Automation and Information Sciences (ICCAIS), Chiang Mai, Thailand, 2017.

4. **Y. Punchihewa**, F. Papi, and R. Hoseinnezhad, Multiple Target Tracking in Video Data using Labeled Random Finite Set, in International Conference on Control, Automation and Information Sciences(ICCAIS), Gwangju, South Korea, December 2014.

The author's contribution includes the theoretical development of the algorithm, implementation (MATLAB), evaluation and drafting the paper. The co-authors contributed by providing insight into image processing techniques, the development of the raw measurement likelihood function and support in editing the paper.

To Whom It May Concern,

I Yuthika Samanmali Gardiyawasam Punchedhewa, contributed to the above listed publications as indicated therein.

Yuthika Samanmali Gardiyawasam Punchedhewa

Supervisor: Prof. Ba-Ngu Vo

CONTENTS

Acknowledgements	v
Abstract	vii
List of abbreviations	xvii
List of notations	xix
1 INTRODUCTION	1
1.1 Motivation and Scope	2
1.2 Key Contributions	5
2 BACKGROUND	7
2.1 Bayesian estimation and tracking	7
2.2 Analytic solutions for the single object Bayes filter	9
2.2.1 Kalman Filter	9
2.2.2 Extended Kalman Filter	10
2.2.3 Unscented Kalman Filter	11
2.2.4 Particle Filter	12
2.2.5 Nearest Neighbour	13
2.2.6 Probabilistic Data Association Filter	14
2.3 Classical approaches to multiple object tracking	14
2.3.1 Global nearest Neighbour	14
2.3.2 Joint Probabilistic Data Association Filter	15
2.3.3 Multiple Hypothesis Tracking	15
2.4 Multiple object tracking with random finite sets	16
2.4.1 Probability density of a Random Finite Set	16
2.4.2 Types of Random Finite Sets	18
2.4.3 Multi-object Dynamic model	19
2.4.4 Multi-object Observation model	20

2.4.5	Multi-object Bayes filter	21
2.5	Multi-object tracking with labeled random finite sets	25
2.5.1	Labeled random finite sets	27
2.5.2	Generalized labeled multi-Bernoulli filter	31
3	TRACKING MANEUVERING OBJECTS USING JMS	43
3.1	Introduction	43
3.2	GLMB filter with separate prediction and update for a JMS	45
3.3	GLMB filter with joint prediction and update for a JMS	49
3.4	Implementation Issues	52
3.5	Simulation Studies	53
4	TRACKING WITH UNKNOWN CLUTTER AND DETECTION PROFILE	65
4.1	Introduction	65
4.2	Multi-Class GLMB	69
4.2.1	Standard multi-object likelihood with zero clutter	69
4.2.2	Multi class GLMB with zero clutter	70
4.3	GLMB Filtering with Unknown Background	74
4.3.1	GLMB Joint Object-Clutter Model	74
4.3.2	Implementation	76
4.3.3	Extension to Unknown Detection Probability	83
4.3.4	Numerical Studies	85
5	TRACKING WITH NO DETECTIONS	103
5.1	Introduction	103
5.2	Enhancing the Raw Measurement Via Background Subtraction	105
5.3	The Multi-object likelihood function	108
5.4	Track-before-detect GLMB filter	110
5.4.1	Implementation Issues	111
5.4.2	Experiment results	112
6	CONCLUSIONS AND FUTURE WORK	117
6.1	Conclusions	117

6.2 Future directions	119
Appendix	121
Bibliography	129

LIST OF FIGURES

2.1	The directed graph with nodes $\ell_1, \ell_2, \dots, \ell_{ I } \in I$ and corresponding costs $c^{(I, \xi)}(\ell_1), \dots, c^{(I, \xi)}(\ell_{ I })$ for solving the K shortest path problem for surviving objects from parent hypothesis (I, ξ) . Start and end nodes are denoted by S, E respectively [86].	33
2.2	The cost matrix for the joint prediction and update optimal assignment problem of hypothesis (I, ξ) . The 3 partitions correspond to survived and detected objects, survived and misdetected objects, and objects that died or were not born. [86]	38
3.1	True trajectories of the tracked objects in scenario 1	54
3.2	OSPA error (parameters $c = 500, p = 2$) reported for different implementations from 100 MC runs for scenario 1.	56
3.3	OSPA(2) error (parameters $c = 500, p = 2$) reported for different implementations from 100 MC runs for scenario 1.	57
3.4	Object trajectories from a single joint/Gibbs run for scenario 1. State estimates are color coded by object labels.	58
3.5	Object state estimates (color coded by estimated mode) generated by the Joint/Gibbs implementation for a single run of scenario 1.	59
3.6	True trajectories of the tracked objects in scenario 2	60
3.7	OSPA error (parameters $c = 500, p = 2$) reported for different implementations from 100 MC runs for scenario 2.	61
3.8	OSPA(2) error (parameters $c = 500, p = 2$) reported for different implementations from 100 MC runs for scenario 2.	61
3.9	Object trajectories from a single joint/Gibbs run for scenario 2. State estimates are color coded by object labels.	62
3.10	Object state estimates generated by the joint/Gibbs implementation for a single run of scenario 2	63
4.1	Frames 16, 48 of the image sequence from [108] and object detections obtained using the detector in [109]. The number of objects varies with time due to objects coming in and out of the scene. Object estimates (marked by blue boxes) using the standard GLMB filter for guessed clutter rate of 60 (top 2 frames) and 'true' clutter rate (bottom 2 frames). Tracking using 'true' clutter rate accurately estimated several objects that were missed in the frames on the top.	67
4.2	'True' clutter rate for the first 60 frames of the dataset [108]. Note that it is not possible to know the true clutter rate for real video data. For illustration, we assume that the clutter rate varies slowly and use the average clutter count over a moving 10-frame window as the 'true' clutter rate.	68

4.3	Tracking results for scenario 1.	89
4.4	Tracking results for scenario 2.	92
4.5	Tracking results for scenario 3.	94
4.6	Tracking results for scenario 4.	96
4.7	Tracking results for frames 20, 40, 100 in dataset 1.	98
4.8	Clutter statistics for dataset 1.	99
4.9	Tracking results for dataset 2.	102
4.10	Clutter statistics for dataset 2.	102
5.1	Stages of enhancing the raw measurement	108
5.2	Gaussian birth locations (in yellow and orange) and an object moving with his Gaussian neighbourhood (in blue). Hypotheses containing new births in the blue region are penalized.	113
5.3	Background subtracted image frames	114
5.4	Tracking results for frames 230,245,260,300,350,400. : Tracked objects are marked by blue rectangles with tracks in different colours and labels in yellow tags.	115
5.5	OSPA (parameters $c = 100, p = 1$) results for frames 100 to 600. Filtering starts after the initial 100 frames which are used to build the background model.	116
5.6	OSPA(2) (parameters $c = 100, p = 1$) results for frames 100 to 600.	116

LIST OF ABBREVIATIONS

CPHD	cardinalized probability hypothesis density filter
CBMemBer	cardinality balanced multi-Bernoulli
EAP	expected a posteriori
EKF	extended Kalman filter
GLMB	generalized labeled multi-Bernoulli
GNN	global nearest neighbour
i.i.d.	independent and identically distributed
IMM	interacting multiple models
JIPDA	joint integrated probabilistic data association
JMS	jump Markov system
JPDA	joint probabilistic data association
KDE	kernel density estimation
LMB	labeled multi-Bernoulli
MAP	maximum a posteriori
MCMC	Markov chain Monte Carlo
MC	Monte Carlo
MHT	multiple hypothesis tracking
NN	nearest neighbour
OSPA	optimal sub-pattern assignment
PDA	probabilistic data association
PHD	probability hypothesis density
RFS	random finite set
SMC	sequential Monte Carlo
SNR	signal-to-noise ratio
TBD	track-before-detect
UKF	unscented Kalman filter

LIST OF NOTATIONS

x	Unlabeled object state.
\boldsymbol{x}	Labeled object state.
X, Z, K, W	Unlabeled random finite sets.
$X^{(m)}$	Unlabeled random finite set of objects of class m .
\mathbf{X}	Labeled random finite set.
$\mathbf{X}^{(m)}$	Labeled random finite set of objects of class m in \mathbf{X} .
$\mathcal{L}(\cdot)$	Label set extraction function.
$\Delta(\cdot)$	Distinct label indicator function.
$\delta_X(\cdot)$	Generalised Kronecker delta function.
$1_{\mathbb{X}}(\cdot)$	Set inclusion function.
$\langle \cdot, \cdot \rangle$	Inner product of two functions.
$[h(\cdot)]^X$	Multi-object exponential.
$\mathcal{N}(\cdot; m, P)$	Gaussian probability density function with mean m and covariance P .
$\beta(\cdot; s, t)$	Beta distribution characterized by parameters s, t .
μ_β	Beta distribution mean.
σ_β^2	Beta distribution covariance.
$ X $	Cardinality of the set X .
$\mathbb{E}(\cdot)$	Expected value function.
\mathbb{R}	Space of real numbers.
\mathbb{R}^n	Space of n -dimensional real vectors.
\mathbb{X}	State space.
\mathbb{M}	Space of modes in a JMS.
\mathbb{L}	Label space.
$\mathbb{L}^{(m)}$	Label space for objects of class m .
\mathbb{B}	Label space for newly born objects.
$\mathbb{B}^{(m)}$	Label space for newly born objects of class m .
\mathbb{Z}	Measurement space.
\mathbb{C}	Discrete index space for components in a GLMB density.

ℓ	object label.
m	object class.
$\kappa(\cdot)$	Poisson clutter intensity function.
$\hat{\kappa}$	Predicted clutter intensity.
$\mathcal{F}_n(\cdot)$	Set of n -element subsets.
$\mathcal{F}(\cdot)$	Set of subsets with any number of elements.
$\phi(\cdot)$	State transition kernel.
$f(\cdot \cdot)$	State transition density.
$f^{(m)}(\cdot \cdot)$	State transition density for objects of class m .
$h(\cdot)$	Observation kernel.
$g(\cdot \cdot)$	Measurement likelihood function.
$g^{(m)}(\cdot \cdot)$	Measurement likelihood function for objects of class m .
$v(\cdot)$	Probability hypothesis density.
$\vartheta(\cdot \cdot)$	Markovian mode switching probability in a JMS.
ζ	Kinematic state of an object
$p(\cdot)$	probability density function.
$\pi(\cdot)$	Unlabeled RFS density.
$\boldsymbol{\pi}(\cdot)$	Labeled RFS density.
$\rho(\cdot)$	Cardinality probability density of a RFS
$P_D(\cdot)$	Detection probability.
$P_D^{(m)}(\cdot)$	Detection probability for objects of class m .
$P_S(\cdot)$	Survival probability.
$P_S^{(m)}(\cdot)$	Survival probability for objects of class m .
I	Random finite set of labels.
$I^{(m)}$	Random finite set of labels for objects of class m .
θ	Positive 1-1 mapping $\mathbb{L} \rightarrow Z$
$\theta^{(m)}$	Positive 1-1 mapping $\mathbb{L}^{(m)} \rightarrow Z$.
Θ	Space of association maps.
ξ	Association map history.
$\xi^{(m)}$	Map ξ restricted to $\mathbb{L}_0^{(m)} \times \dots \times \mathbb{L}_k^{(m)}$.
Ξ	Space of association map historys.
k	Time index.
k_n	Time index of n^{th} image frame.

CHAPTER 1

INTRODUCTION

The significance of object tracking was well recognized and motivated primarily by military defence applications in which precise detection and tracking of the enemy on ground, air and water were crucial. The Wiener filter [1] developed at the Massachusetts Institute of Technology and mounted on American anti-aircraft systems during the era of world war II, is one of the earliest examples. It detailed a method for reducing the amount of noise present in continuous time signals based on the minimum mean square error. A similar method for discrete time signals was proposed by Andrey Kolmogorov, contributing to the Russian war effort [2]. Both methods relied on frequency domain techniques for modelling the systems under concern. Postwar, scientists turned towards using time domain techniques for modelling such systems which gave rise to the famous Kalman filter that aptly demonstrated its usefulness in estimating the spacecraft trajectories during the first moon landing in 1969 [3]. Today object tracking is instrumental in many other applications such as bio-medicine [4–6], robotics [7, 8], autonomous vehicles [9, 10] and space science [11, 12] to name a few.

The Kalman filter is an exact solution to the Bayesian estimation problem under linear dynamic and observation models and Gaussian noise assumptions. The Kalman filter and its variations tackle the problem of estimating the true state from noisy observations, and is directly applicable to the single object tracking problem where the tracked object is known to be present throughout the course and the received measurements are affirmed to be produced by none other than the said object. Multi-object tracking is more complicated than solving several single object tracking problems supposing that information regarding the number of objects to be tracked and the origin of each measurement cannot be discerned, which is often the case. Consider a maritime surveillance system for example. Different marine objects enter (termed births in track-

ing terminology) the surveilled region and exit (termed deaths) at different times and there is no way of knowing exactly how many vessels are in the state space at a given time. Especially in instances where the state space is densely populated by objects of interest, it is too difficult to pinpoint which measurement should be attributed to which object. This is known as the data association problem. This problem is further exacerbated by the sensors/detection algorithms being unable to obtain a measurement from each object of interest (missed detections) and picking up incorrect measurements that cannot be attributed to a valid object (false measurements/clutter).

A plethora of techniques that attempt to solve the multi-object tracking problem has surfaced in the past few decades. The random finite set (RFS) approach to the multi-object estimation problem, which came to light circa at the beginning of the millennium, is a principled and mathematically consistent approach founded upon stochastic geometry and point process theory [13, 14]. Until 2013 the main criticism regarding RFS-based algorithms was that they were multi-object filters rather than trackers. i.e., these algorithms were able to produce state estimates for the objects in the system but they were unable to maintain a correlation between the estimates at different time steps innately without the aid of auxiliary heuristics/post-processing methods. The recent discovery of labeled random finite set based multi-object conjugate priors [15] laid these claims to rest.

The classical techniques and state-of-the-art RFS algorithms for the multi-object filtering/tracking problem is deliberated in [Chapter 2](#). Even though robust algorithms for multi-object *filtering* using RFS have been researched extensively, robust RFS-based multi-object *trackers* for diverse challenging scenarios and conditions remains a relatively unexplored territory. The overall objective of this thesis is to further strengthen the present day labeled RFS based multi-object trackers for such scenarios by investigating robust techniques to increase relevancy in real-world applications.

1.1 Motivation and Scope

Present day multiple object tracking algorithms operate with the aid of a significant amount of prior knowledge and assumptions regarding the multi-object system. This mainly includes knowledge and assumptions regarding the behaviour of the objects (dynamics and process noise), statistics of sensors (measurement noise, the rate of

false positive/negative measurements) and regions of interest in state space regarding object births/deaths and their respective probabilities. While some knowledge can be readily obtained by observing the state/observation space (e.g. in a radar-based tracking application an airport is a region of interest for object births/deaths), some knowledge has to be extracted by analysing training data over long periods of time (e.g. missed detection rate of a sensor). In the event of a change in the multi-object system conditions from the training data conditions (eg: weather), some prior knowledge withheld regarding the multi-object system would no longer be valid. Therefore in application to different real-world problems, algorithms that presume to acquire complete and accurate prior knowledge regarding the system fail to deliver a robust performance.

This thesis aims to approach the problem of robust tracking in the absence of the required prior knowledge and challenging background conditions from a labeled RFS perspective [13, 14]. The earlier work on multi-object filtering based on unlabeled RFS (discussed in [Section 2.4](#)) were unable to generate trajectories. The labeled RFS approach to multi-object tracking (discussed in [Section 2.5](#)) addresses this shortcoming in the unlabeled RFS-based algorithms by maintaining unique labels for the estimated objects. Even though research on robust unlabeled RFS algorithms that operate in the absence of prior knowledge has been carried out by others, labeled RFS algorithms in addressing this problem is sparse and this dissertation aims to bridge that gap. A few specific cases in multi-object tracking that are focused upon in this dissertation, where the robustness of a tracking algorithm is challenged are discussed in the following.

A typical expectation of generic multi-object tracking algorithms is that all objects in the system follow a common dynamic model. This is not the case where maneuvering objects are concerned, here dynamics could change drastically when different maneuvers are applied at different times. Therefore the motion of such an object cannot be encapsulated by a single dynamic model, and a combination of motion models that characterize different maneuvers may be needed. Maneuvering object tracking is the subject of numerous works as it is more challenging than standard tracking due to the uncertainty in object maneuvers on top of the uncertainties in noise, clutter, data association and detection. The interacting multiple models (IMM) and variable-structure IMM (VS-IMM) estimators [16–21] are two well known single-object filtering

algorithms for maneuvering objects. The number of modes in the IMM is kept fixed, whereas in the VS-IMM the number of modes is adaptively selected from a fixed set of modes for improved estimation accuracy and computational efficiency. The jump Markov system (JMS) or multiple models approach has proven to be an effective tool for maneuvering object tracking. In this approach, the object is assumed to be switching between a set of system models in a Markovian fashion. One objective of this dissertation is to address this problem of tracking objects with variable dynamics using a JMS labeled RFS algorithm.

In multi-object filtering, aside from assumptions regarding the dynamics, the various sensors and detection algorithms used invariably miss objects in the scene as well as generate false measurements. The detection of individual objects is characterized by the detection probability. False measurement generation is usually modeled as a homogeneous Poisson process characterized by a clutter rate. Knowledge of parameters such as clutter rate and detection probability is of critical importance in detection based tracking methods and their values vary with each detection method. While these parameters are assumed to be known in most multi-object tracking techniques, this is generally not the case in practice [22, 23]. In application areas such as radar, exact knowledge of these model parameters is given through the radar model. In some applications, these parameters are computed offline from training data trusting that the training data resembles the actual system. In the real world, the time-varying nature of these background parameters further aggravates this problem. For example, the clutter rate and detection probability parameters which are subject to the weather and lighting conditions undeniably vary with time in outdoor scenarios. Consequently, there is no guarantee that the parameters chosen from a particular training data set will be valid throughout. Thus algorithms that adapt to an unknown and varying clutter and detection profile remain an important topic in robust tracking. The second objective of this dissertation is to devise a labeled RFS based algorithm that performs joint tracking and estimation of unknown clutter and detection profile parameters.

Most multi-object tracking algorithms found in the literature are designed to operate on an observation set that consists of point measurements (detections). In practice, the sensor typically provides a raw data image. For example, in a radar application the antenna provides a raw image where each pixel corresponds to the received power in

a particular spatial location. This raw data is subsequently processed into point measurements by a detection algorithm. The popularity of such detection based tracking is understandable due to the efficiency in terms of memory and computation it provides by compressing raw data into a finite set of points [24]. However, for applications with a low signal to noise ratio (SNR), preprocessing data to extract detections can result in significant information loss [25]. In low SNR conditions, the threshold needs to be low enough to allow a decent detection probability, which also means a large number of false detections. In visual tracking, numerous techniques are employed to extract detections from raw data such as background/foreground matching via kernel density estimation [26], colour histograms [27], human shape models [28], multi-modal representations [29] and lazy background subtraction [30]. Where visual tracking applications are concerned, measurements obtained using such techniques typically output blobs/regions as opposed to point detections. Further, in some instances, the detection algorithms in visual applications are prone to detecting a single large object as multiple small blobs, to detecting multiple small objects as a single large blob and to miss detecting overlapping objects (occlusion). Track-before-detect (TBD) techniques attempt to avoid such issues encountered in detection based tracking and consequently avoid difficulties with unknown clutter and detection profile. The third objective of this dissertation is to produce a track-before-detect labeled RFS solution for visual data.

The structure of this dissertation, in summary, is as follows. Classical approaches to multi-object filtering/tracking, fundamental theories of the unlabeled and labeled RFS frameworks and algorithms constructed upon them are discussed in [Chapter 2](#). [Chapter 3](#) presents a labeled RFS algorithm for tracking maneuvering objects using jump Markov systems. [Chapter 4](#) proposes a labeled RFS algorithm for tracking under unknown clutter and detection profile. [Chapter 5](#) presents a labeled RFS method for visual tracking without using detections. [Chapter 6](#) concludes the dissertation with an analysis of the key findings and future directions.

1.2 Key Contributions

This thesis adopts the labeled RFS approach to the multi-object tracking problem. The key premise of this work is that the labeled random finite set framework is capable of producing tracking algorithms that deliver a robust performance under various challenging conditions that are encountered in the real world.

- The first contribution is a labeled RFS algorithm for tracking maneuvering objects. A JMS setting is utilized to model the object dynamics. This algorithm propagates the multi-object posterior as a generalized labeled multi-Bernoulli [15] density taking into account the JMS dynamic model. In addition to estimating the object states, the algorithm is capable of constructing trajectories and discerning the dynamic model which best represents the behaviour of each object at each time step. This contribution has appeared in the author's conference papers [31, 32] and journal article [33]. In this dissertation, it appears in [Chapter 3](#).
- The second contribution is a labeled RFS based algorithm that operates with no knowledge of the clutter and detection profile. By modeling clutter as originating from a special class of non-interacting objects which are not of interest, a JMS solution can be adapted to produce a tractable algorithm for tracking with unknown and varying clutter rate. The proposed algorithm is further extended to estimate the unknown detection probability by augmenting the object state vector with the detection probability. In addition to the object state estimates the algorithm produces trajectories and estimates for the clutter and detection profile parameters. This contribution has appeared in the author's journal article [33]. In this dissertation, it appears in [Chapter 4](#).
- The third contribution is a labeled RFS algorithm that performs tracking on visual data without using detections. The raw image is provided as a single observation to the algorithm without extracting point measurements on objects of interest. Utilizing a likelihood function for the raw measurement given the multi-object state, the algorithm produces trajectories in addition to object state estimates. This contribution has appeared in the author's conference paper [34]. In this dissertation, it appears in [Chapter 5](#).

CHAPTER 2

BACKGROUND

This chapter provides a brief introduction to the techniques that are prevalently used in present-day multiple object tracking and the theories upon which these techniques are constructed.

2.1 Bayesian estimation and tracking

Estimation theory can be broadly construed as studies on the estimation of underlying unknown parameters of a system utilizing empirical data of a stochastic nature. Object tracking is primarily an estimation problem as it aims to sequentially estimate the state of a dynamic system via a sequence of noisy measurements obtained from the system.

Object tracking with Bayesian inference can be roughly described as inferring the posterior probability density for object states using priors on the object states and the likelihood of the observation. All available information for the object state is obtainable from the posterior¹ probability density. The expected a posteriori (EAP) and the maximum a posteriori (MAP) are two popularly used estimators [16]. In situations where observations regarding the system state are received consecutively, an algorithm that recursively infers the posterior density with fresh observations is much preferred over an algorithm that processes the entire batch of observations at the end of the day. Such an iterative algorithm, also styled as a recursive algorithm, upon receiving fresh measurements, updates and feeds the posterior density back into the filter to be used in the next iteration. Customarily, in order to construct such a recursive algorithm, two probabilistic models are needed: one describing the prior knowledge regarding the kinematics/features of object states, referred to as the dynamic model, and a model de-

¹The term posterior density technically refers to $p(x_k, x_{k-1}, \dots, x_0 | z_k, z_{k-1}, \dots, z_1)$ and the term filtering density is used for $p(x_k | z_k, z_{k-1}, \dots, z_1)$. In this work we follow the standard nomenclature of using posterior and filtering density interchangeably.

scribing the noisy measurement generation referred to as the observation model need to be drafted. Consider the single object tracking problem. Let $x_k \in \mathbb{R}^{n_x}$ denote the state of a single object at time k (n_x being the dimension of the state vector) and let $x_{0:k}$ denote all states of that object from beginning up to time k . Let $z_k \in \mathbb{R}^{n_z}$ denote an observation received regarding the said object at time k (n_z being the dimension of the measurement) and let $z_{1:k}$ denote all such measurements received from time 1 up to k . In general, the dynamic model which describes the evolution of a single object state $x_{k-1} \in \mathbb{R}^{n_x}$ at time $k-1$ to $x_k \in \mathbb{R}^{n_x}$ at time k can be expressed as,

$$x_k = \phi_{k-1}(x_{k-1}) + v_{k-1}, \quad (2.1)$$

where ϕ_{k-1} is a state transition function and v_{k-1} is the process noise.

In general, the model for an observation $z_k \in \mathbb{R}^{n_z}$ for an object with state x_k at time k can be expressed as,

$$z_k = h_k(x_k) + w_k, \quad (2.2)$$

where h_k is a measurement generation function and w_k is the measurement noise.

A recursive filtering algorithm typically comprises of two stages, namely prediction and update. Let the state transition density that arises from the dynamic model [Eq. \(2.1\)](#) be denoted by $f_{k|k-1}(\cdot|\cdot)$ and hence $f_{k|k-1}(x'|x)$ the probability of an object with state x transitioning to new state x' . Let the filtering density at time $k-1$ be denoted by $\pi_{k-1}(\cdot|z_{1:k-1})$. By way of the Chapman-Kolmogorov equation, the predicted density $\pi_{k|k-1}(\cdot|z_{1:k-1})$ is,

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

Let $g_k(\cdot|\cdot)$ be the measurement likelihood arising from the observation model [Eq. \(2.2\)](#) and hence $g_k(z_k|x_k)$ the likelihood of receiving measurement z_k for an object with state x_k . In the update stage, using the new observation with Bayes rule, the posterior density $\pi_k(\cdot|z_{1:k})$ for the object state at time k is,

$$\pi_k(x_k|z_{1:k}) = \frac{g_k(z_k|x_k)\pi_{k|k-1}(x_k|z_{1:k-1})}{\int g_k(z_k|x)\pi_{k|k-1}(x|z_{1:k-1}) dx}. \quad (2.4)$$

The recursive single object Bayes filter is encapsulated by [Eq. \(2.3\)](#),[Eq. \(2.4\)](#). In this general form, it has no analytic solution.

2.2 Analytic solutions for the single object Bayes filter

2.2.1 Kalman Filter

An analytic solution for the recursive single object Bayes filter for the special case of linear state transition and observation functions with additive temporally uncorrelated Gaussian noise, coined the Kalman Filter, was first introduced by Rudolf Kalman in 1960 [35]. For such a special case, the Kalman filter dictates that the posterior density remains Gaussian at every time step [36]. In this special case, [Eq. \(2.1\)](#),[Eq. \(2.2\)](#) have the special form,

$$x_k = F_{k-1} x_{k-1} + v_{k-1}, \quad (2.5)$$

$$z_k = H_k x_k + w_k, \quad (2.6)$$

where $v_{k-1} \sim \mathcal{N}(\cdot; 0, Q_{k-1})$, is a sequence of uncorrelated noise, $w_k \sim \mathcal{N}(\cdot; 0, R_k)$, is a sequence of uncorrelated noise, F_{k-1} is a $n_x \times n_x$ matrix, H_k is a $n_z \times n_x$ matrix, Q_{k-1} is a $n_x \times n_x$ matrix and R_k is a $n_z \times n_z$ matrix. This means,

$$f_{k|k-1}(x_k|x_{k-1}) = \mathcal{N}(x_k; F_{k-1}x_{k-1}, Q_{k-1}), \quad (2.7)$$

$$g_k(z_k|x_k) = \mathcal{N}(z_k; H_k x_k, R_k). \quad (2.8)$$

Assuming that the posterior density $\pi_{k-1}(\cdot|z_{1:k-1})$ at time $k-1$ is the Gaussian density $\mathcal{N}(\cdot; m_{k-1}, P_{k-1})$, substituting $\pi_{k-1}(\cdot|z_{1:k-1})$ and [Eq. \(2.7\)](#) in [Eq. \(2.3\)](#), the following Kalman filter expression for predicted density $\pi_{k|k-1}(\cdot|z_{1:k-1})$ is obtained.

$$\pi_{k|k-1}(x_k|z_{1:k-1}) = \mathcal{N}(x_k; m_{k|k-1}, P_{k|k-1}), \quad (2.9)$$

$$m_{k|k-1} = F_{k-1}m_{k-1}, \quad (2.10)$$

$$P_{k|k-1} = F_{k-1}P_{k-1}F_{k-1}^T + Q_{k-1}. \quad (2.11)$$

Substituting $\pi_{k|k-1}(\cdot|z_{1:k-1})$ and Eq. (2.8) in Eq. (2.4) the following Kalman filter expression for posterior density $\pi_k(\cdot|z_{1:k})$ is obtained.

$$\pi_k(x_k|z_{1:k}) = \mathcal{N}(x_k; m_k, P_k), \quad (2.12)$$

$$m_k = m_{k|k-1} + K_k(z_k - H_k m_{k|k-1}), \quad (2.13)$$

$$P_k = P_{k|k-1} - K_k S_k K_k^T, \quad (2.14)$$

$$K_k = P_{k|k-1} H_k^T S_k^{-1}, \quad (2.15)$$

$$S_k = H_k P_{k|k-1} H_k^T + R_k. \quad (2.16)$$

As both the prior and posterior densities in the Kalman filter are Gaussian, the Kalman filter is a computationally tractable solution for the single object Bayes filter.

2.2.2 Extended Kalman Filter

The extended Kalman filter (EKF) [37, 38] has proved to be effective for scenarios with Gaussian noise where the dynamic/observation models are mildly non-linear. The first terms of the Taylor series expansions of the non-linear functions $\phi_{k-1}(\cdot)$, $h_k(\cdot)$, denoted by \hat{F}_{k-1} , \hat{H}_k as given below are used as linear approximations in computing the covariances of the predicted and updated densities. In case only one of the models is non-linear, it is sufficient to perform the EKF only at the corresponding stage and perform the standard Kalman filter at the other stage.

$$\hat{F}_{k-1} = \left. \frac{\partial \phi_{k-1}(x)}{\partial x} \right|_{x=m_{k-1}} \quad (2.17)$$

$$\hat{H}_k = \left. \frac{\partial h_k(x)}{\partial x} \right|_{x=m_{k|k-1}} \quad (2.18)$$

The extended Kalman filter equations for computing the predicted and posterior densities at time k given the posterior $\pi_{k-1}(\cdot|z_{1:k-1})$ at time $k-1$ as the Gaussian density $\mathcal{N}(\cdot; m_{k-1}, P_{k-1})$, is as follows.

$$\pi_{k|k-1}(x_k|z_{1:k-1}) = \mathcal{N}(x_k; m_{k|k-1}, P_{k|k-1}), \quad (2.19)$$

$$m_{k|k-1} = \phi_{k-1}(m_{k-1}), \quad (2.20)$$

$$P_{k|k-1} = \hat{F}_{k-1} P_{k-1} \hat{F}_{k-1}^T + Q_{k-1}, \quad (2.21)$$

$$\pi_k(x_k | z_{1:k}) = \mathcal{N}(x_k; m_k, P_k), \quad (2.22)$$

$$m_k = m_{k|k-1} + K_k (z_k - h_k(m_{k|k-1})), \quad (2.23)$$

$$P_k = P_{k|k-1} - K_k S_k K_k^T \quad (2.24)$$

$$K_k = P_{k|k-1} \hat{H}_k^T S_k^{-1}, \quad (2.25)$$

$$S_k = \hat{H}_k P_{k|k-1} \hat{H}_k^T + R_k. \quad (2.26)$$

2.2.3 Unscented Kalman Filter

In contrast to the EKF which performs analytic linearization, the unscented Kalman filter (UKF) [39–41] performs statistical linearization. Rather than propagating the mean and covariance of the Gaussian posterior density forward, the UKF propagates a set of $N = 2n_x + 1$ weighted sample points representing the density forward where n_x is the dimension of the state vector. Let us denote the set of weighted sample points at time $k - 1$ by $\left\{ \left(x_{k-1}^{(i)}, w^{(i)} \right) \right\}_{i=1:N}$. The selection of these sample points is such that they encompass the mean and the covariance of the density from which they are sampled.

Sample Point	Value	Weight
$i = 1$	$x_{k-1}^{(i)} = m_{k-1},$	$w^{(i)} = \frac{K}{(n_x + K)}$
$i = 2, \dots, n_x + 1$	$x_{k-1}^{(i)} = m_{k-1} + [A]_{i-1},$	$w^{(i)} = \frac{K}{2(n_x + K)}$
$i = n_x + 2, \dots, 2n_x + 1$	$x_{k-1}^{(i)} = m_{k-1} - [A]_{i-n-1}$	$w^{(i)} = \frac{K}{2(n_x + K)}$

Table 2.1: Computing sample points : $A = \sqrt{(n_x + K) P_{k-1}}$, the subscripted square brackets $[A]_i$ denotes the i^{th} row of matrix A and K is a scaling parameter such that $(n_x + K) \neq 0$.

The sample points are propagated forward through the non-linear transform for the object dynamics to compute the predicted mean $m_{k|k-1}$ and covariance $P_{k|k-1}$:

$$m_{k|k-1} = \sum_{i=1}^N w^{(i)} \phi_{k-1} \left(x_{k-1}^{(i)} \right), \quad (2.27)$$

$$P_{k|k-1} = Q_{k-1} + \sum_{i=1}^N w^{(i)} \left[\phi_{k-1} \left(x_{k-1}^{(i)} \right) - m_{k|k-1} \right] \left[\phi_{k-1} \left(x_{k-1}^{(i)} \right) - m_{k|k-1} \right]^T. \quad (2.28)$$

The predicted measurement $\hat{z}_{k|k-1}$ is calculated using the propagated sample points:

$$\hat{z}_{k|k-1} = \sum_{i=1}^N w^{(i)} h_k \left(\phi_{k-1} \left(x_{k-1}^{(i)} \right) \right). \quad (2.29)$$

The parameters of the posterior density of the UKF are calculated using the predicted measurement $\hat{z}_{k|k-1}$.

$$m_k = m_{k|k-1} + K_k \left(z_k - \hat{z}_{k|k-1} \right), \quad (2.30)$$

$$P_k = P_{k|k-1} - K_k S_k K_k^T, \quad (2.31)$$

$$K_k = P_{xz} S_k^{-1}, \quad (2.32)$$

$$S_k = P_{zz} + R_k, \quad (2.33)$$

$$P_{xz} = \sum_{i=1}^N w^{(i)} \left[\phi_{k-1} \left(x_{k-1}^{(i)} \right) - m_{k|k-1} \right] \left[h_k \left(\phi_{k-1} \left(x_{k-1}^{(i)} \right) \right) - \hat{z}_{k|k-1} \right]^T, \quad (2.34)$$

$$P_{zz} = \sum_{i=1}^N w^{(i)} \left[h_k \left(\phi_{k-1} \left(x_{k-1}^{(i)} \right) \right) - \hat{z}_{k|k-1} \right] \left[h_k \left(\phi_{k-1} \left(x_{k-1}^{(i)} \right) \right) - \hat{z}_{k|k-1} \right]^T. \quad (2.35)$$

The predicted/posterior density parameters obtained through the UKF are accurate up to the second order of the Taylor series expansion [40] whereas the EKF is accurate only up to the first-order. Further the UKF can be performed where the non-linear functions are discontinuous.

2.2.4 Particle Filter

The variations of the Kalman filter described in the previous sections operate under the assumption that the filtering density and the process/measurement noise models are Gaussian. The particle filter or sequential Monte Carlo (SMC) was proposed as a solution in situations where these assumptions do not hold. Even though the idea behind SMC estimation was available from the 1950s [42–44] it did not gain prominence in that era due to the fact that the available hardware struggled to cater to the computational power demanded by these methods.

In the particle filter, a density is approximated by a large set of weighted samples

(particles) denoted by, $\left\{ \left(x_{k-1}^{(i)}, w_{k-1}^{(i)} \right) \right\}_{i=1:N}$. i.e., the filtering density at time $k-1$,

$$\pi_{k-1} \left(x_{k-1} | z_{1:k-1} \right) \approx \sum_{i=1}^N w_{k-1}^{(i)} \delta_{x_{k-1}^{(i)}} \left(x_{k-1} \right), \quad \text{where} \quad \sum_{i=1}^N w_{k-1}^{(i)} = 1. \quad (2.36)$$

In principle, upon receiving a new measurement, a new set of weighted particles need to be drawn directly from the posterior density. Since that is infeasible in most circumstances, samples are drawn from a density that closely resembles the posterior. This is referred to as the importance or proposal density. A simple choice for the importance density is the predicted density. This results in the sequential importance sampling particle filter equation [45, 46].

$$\pi_k \left(x_k | z_{1:k} \right) \approx \sum_{i=1}^N w_k^{(i)} \delta_{x_k^{(i)}} \left(x_k \right), \quad (2.37)$$

$$x_k^{(i)} = \phi_{k-1} \left(x_{k-1}^{(i)} \right), \quad (2.38)$$

$$w_k^{(i)} \propto w_{k-1}^{(i)} g_k \left(z_k | x_k^{(i)} \right). \quad (2.39)$$

Propagating the filtering density forward in this manner leads to a problem known as particle depletion, where after a number of iterations, save for a few ones nearly all of the particles in the sample consist of negligible weights. To alleviate this problem a procedure termed resampling [47] is carried out when a significant particle depletion is noticed. The basic concept of resampling is the replacement of particles with negligible weights by replication of particles with higher weights. This can be performed in a number of different ways [48–50]. Another method called particle Markov chains Monte Carlo that combines Markov chain Monte Carlo and sequential Monte Carlo methods to design efficient high dimensional proposal distributions is discussed in [51]. Since resampling is a process with a high computation cost, methods to tackle this by way of using graphics processing units are discussed in [52, 53].

2.2.5 Nearest Neighbour

In single object scenarios where the sensor picks up false measurements or is subject to missed detections along with true measurements, the above described filters are not directly applicable. The nearest neighbour (NN) method [22–24, 54] is a simple

tracking algorithm for such scenarios. The prediction step is performed similarly to the Kalman filter. For the update step, the measurement that is closest to the predicted measurement is used to compute the Kalman update or none is performed if no near measurement is available. Albeit being simple and efficient to implement, the NN filter has proven to be vulnerable in scenarios with dense clutter or low detection (inability to pick up the true measurement).

2.2.6 Probabilistic Data Association Filter

The prediction step of the probabilistic data association (PDA) [22, 55] filter is similar to the NN. Nevertheless, in the update step, rather than using a single measurement, several candidate measurements are selected using gating and the likelihood of each measurement being generated by the predicted object state (association probability) is calculated. Subsequently, the Kalman update is performed using an average of the candidate measurements weighted by their association probabilities. In [56] a variation of the PDA which allows the filtering density to be a Gaussian mixture rather than a single Gaussian as in the standard PDA is discussed. The PDA filter is less prone to the vulnerabilities of the NN [22–24, 54].

2.3 Classical approaches to multiple object tracking

The multiple object tracking problem is much more challenging than the single object tracking problem for several reasons. The number of objects in the tracking scenario is unknown, time-varying and could be zero. The mapping of the received measurements to objects, known as the data association problem, is combinatorially expensive. The received measurements could contain false measurements and be subject to mis-detection for every object in the scenario. Various multi-object filtering algorithms that stem from the single object algorithms mentioned in the preceding section have been constructed in order to address this problem. These are discussed in the following section.

2.3.1 Global nearest Neighbour

In scenarios where a known fixed number of multiple objects are present in the system, the global nearest neighbour (GNN) [22–24, 54] performs the standard Kalman prediction individually on each object, and the standard Kalman update individually

on each object, using an object-measurement mapping that minimizes a certain cost, on the condition that one measurement is mapped to one object at most. The cost function could be a simple function such as the total summed distance or association probability. Similar to NN the GNN too suffers in dense clutter environments and environments with low detection.

2.3.2 Joint Probabilistic Data Association Filter

The joint probabilistic data association (JPDA) filter [22] is an extension of the PDA filter. Similar to GNN it can only be applied in scenarios in which the multiple object count in the system is fixed and known. The Kalman prediction is carried out for each object individually, and the Kalman update for each object is performed separately, using an averaged measurement computed from selected neighbor measurements weighted according to their association probabilities. Care has to be taken to avoid conflicts when including the same measurement in different groups linked to different objects. Due to this, the complexity of this data association increases exponentially with the object count and measurement count. Sub-optimal strategies to reduce the complexity have been proposed in [57–59]. Further, in using JPDA on closely spaced objects, tracks tend to come together (clump) [60]. A variation of the JPDA named the joint integrated probabilistic data association filter (JIPDA) that accommodates an unknown and time-varying number of objects was proposed in [61].

2.3.3 Multiple Hypothesis Tracking

The technique of multiple hypothesis tracking (MHT) [23, 54, 62] functions by propagating multiple measurement-to-object associations, known as hypotheses. By propagating several alternative data association hypotheses rather than only the best hypothesis, it is able to defer difficult data association decisions with the expectation that future data will resolve the uncertainty. As it is impractical to carry all possible hypotheses, only the ones with the highest weights calculated using the Kalman filter are selected. Since a new set of hypotheses is generated with freshly received measurements which could be assigned to an existing track, new track or clutter, the MHT permits an unknown time-varying number of tracked objects.

The main limitation of the MHT is the exponentially growing number of hypotheses with time. Gating of measurements along with heuristic pruning/merging of hy-

potheses is a basic strategy to sidestep this problem. Stronger deterministic techniques to select the best hypotheses are given in [23, 63, 64]. A combinatorially optimum technique that uses reversible jump Markov chain Monte Carlo to generate samples from the posterior to find the hypotheses with highest posterior probabilities is discussed in [65].

2.4 Multiple object tracking with random finite sets

A random variable is a measurable mapping from sample space to state space. An RFS is a finite set-valued random variable. The number of elements is random and the values of the elements themselves are also random. The key difference between an RFS and a random vector is the following: the number of elements in a random vector is fixed and known and the elements appear in a certain order, but the number of elements in an RFS is random and there is no particular order in which these elements appear. An RFS is characterized by a discrete probability distribution on the cardinality of the set, and for a given cardinality a joint probability distribution on the elements.

The representation of multi-object states and the multi-object observations in set form has given rise to the RFS approach to multi-object tracking. This approach essentially focuses on jointly estimating both the number of objects in the state space and their individual states. Even though Point process based filters for multi-object tracking were introduced as early as the 1980s [66, 67] a systematic treatment for multi-object tracking with RFS was first introduced by Mahler in 1994 [68]. Detailed treatments are available in [13, 14, 69, 70].

The following sections briefly describe fundamental theories on RFS and brief descriptions on popular RFS filters that have attracted considerable interest in the recent years. Notation-wise, lower case letters e.g. x, z are used to denote single elements and upper case letters e.g. X, Z are used to denote sets from here onwards.

2.4.1 Probability density of a Random Finite Set

The probability density is a common descriptor of an RFS. For an RFS X on space $\mathbb{X} = \mathbb{R}^{n_x}$, the space $\mathcal{F}(\mathbb{X})$ of all finite subsets of \mathbb{X} does not inherit the usual Euclidean notion of density from \mathbb{R}^d . A mathematically consistent notion of a probability density on $\mathcal{F}(\mathbb{X})$ is given in point process theory [71]. The notion of a density is tied to measure

and integration. The conventional choice of reference measure in point process theory is the dimensionless measure given by,

$$\mu(\mathcal{T}) = \sum_{r=0}^{\infty} \frac{\lambda^r (\mathbb{X}^{-1}(\mathcal{T}) \cap \mathbb{X}^r)}{r!}, \quad (2.40)$$

for any subset $\mathcal{T} \subseteq \mathcal{F}(\mathbb{X})$ where \mathbb{X}^r is the r^{th} cartesian product of \mathbb{X} with the convention $\mathbb{X}^0 = \{\emptyset\}$, λ^r is the r^{th} product dimensionless Lebesgue measure on \mathbb{X}^r and \mathbb{X} is a mapping of vectors to sets defined by $\mathbb{X}(x_1, x_2, \dots, x_r) = \{x_i : i = 1, 2, \dots, r\}$.

The integral of a function $h : \mathcal{F}(\mathbb{X}) \rightarrow \mathbb{R}$ over a subset $\mathcal{T} \subseteq \mathcal{F}(\mathbb{X})$ with respect to μ is given as,

$$\int_{\mathcal{T}} h(X) \mu(dX) = \sum_{r=0}^{\infty} \frac{1}{r!} \int 1_{\mathcal{T}}(\mathbb{X}(x_1, \dots, x_r)) h(\{x_1, \dots, x_r\}) \lambda^r(dx_1 \dots dx_r), \quad (2.41)$$

where $1_{\mathcal{T}}(\cdot)$ is the indicator function for \mathcal{T} .

For a RFS X on \mathbb{X} , the probability density π with respect to measure μ satisfies,

$$P(X \in \mathcal{T}) = \int_{\mathcal{T}} \pi(Y) \mu(dY), \quad (2.42)$$

for any subset $\mathcal{T} \subseteq \mathcal{F}(\mathbb{X})$. A probability density on Euclidean space has the physical dimension of probability per unit hyper-volume. The probability density of the RFS π is dimensionless since the reference measure is dimensionless. The probability density of the RFS can be equivalently represented by the finite set statistics density and integration given by [13, 72],

$$\int h(X) \delta X = \sum_{i=0}^{\infty} \frac{1}{i!} \int_{\mathbb{X}^i} h(\{x_1, \dots, x_i\}) dx_1, \dots, dx_i, \quad (2.43)$$

for a function $h : \mathcal{F}(\mathbb{X}) \rightarrow \mathbb{R}$.

The probability hypothesis density (PHD) or intensity of a RFS is a 1^{st} order statistical moment of a RFS. For RFS X on \mathbb{X} , the PHD is a non-negative function v on \mathbb{X} such

that for each region $S \subseteq \mathbb{X}$,

$$\mathbb{E}(|X \cap S|) = \int_S v(x) dx. \quad (2.44)$$

Hence integrating v over the entire space gives the expected number of elements in X . The local maxima of function v give the points in \mathbb{X} with the highest concentration of elements.

2.4.2 Types of Random Finite Sets

Poisson RFS

An RFS X on \mathbb{X} is said to be Poisson with intensity $v(\cdot)$ if the cardinality distribution of that RFS is Poisson with mean N where $N = \int v(x) dx$, and for any finite cardinality the elements of X are independent and identically distributed (i.i.d) according to probability density $v(\cdot)/N$. Therefore, the probability density of a Poisson RFS with intensity v is given by [71],

$$\pi(\{x_1, \dots, x_n\}) = \frac{1}{e^N} \prod_{i=1}^n v(x_i). \quad (2.45)$$

with $\prod_{i=1}^0 v(x_i) = 1$ by convention.

Independent and identically distributed cluster RFS

An i.i.d cluster RFS X on \mathbb{X} is characterised by its cardinality distribution $\rho(\cdot)$ and intensity or PHD $v(\cdot)$. The cardinality distribution can be arbitrary on condition that the mean of the cardinality distribution matches $N = \int v(x) dx$. This means that an i.i.d cluster point process encapsulates the randomness of a Poisson RFS without being constrained to Poisson cardinality. The probability density of a i.i.d cluster RFS is given by,

$$\pi(\{x_1, \dots, x_n\}) = n! \rho(n) \prod_{i=1}^n \frac{v(x_i)}{N}. \quad (2.46)$$

Bernoulli RFS

A Bernoulli RFS X on \mathbb{X} has a probability $1 - r$ of being an empty set and a probability r of being singleton whose value is distributed according to a probability density $p(\cdot)$

defined on \mathbb{X} .

$$\pi(X) = \begin{cases} 1 - r, & X = \emptyset \\ r p(x), & X = \{x\} \\ 0, & |X| > 1 \end{cases} . \quad (2.47)$$

Multi-Bernoulli RFS

A multi-Bernoulli RFS is a collection of a finite and fixed number of independent Bernoulli RFSs. Therefore it is characterised by the parameter set $\{(r^{(i)}, p^{(i)}(\cdot))\}_{i=1}^M$ where M is the number of fixed Bernoulli RFSs and the pair $(r^{(i)}, p^{(i)}(\cdot))$ denote the existence probability and the spatial probability density of the i^{th} Bernoulli RFS. The probability density of such a multi-Bernoulli RFS is given by,

$$\pi(\{x_1, \dots, x_n\}) = \prod_{j=1}^M (1 - r^{(j)}) \sum_{1 \leq i_1 \neq \dots \neq i_n \leq M} \prod_{j=1}^n \frac{r^{(i_j)} p^{(i_j)}(x_j)}{1 - r^{(i_j)}}. \quad (2.48)$$

2.4.3 Multi-object Dynamic model

Consider a multi-object system where new objects spontaneously appear in the state space, some existing objects survive to the next time step with a new state, and some objects cease to exist. In this system let X_{k-1} denote the entire set of objects existing at time $k-1$. A object in this set with state x_{k-1} can either survive to the next time step with probability $P_{S,k}(x_{k-1})$ and change to a new state under transition density $f_{k|k-1}(\cdot | x_{k-1})$ or terminate with probability $1 - P_{S,k}(x_{k-1})$. In such a system the random finite set of the surviving objects could be modeled by,

$$S_k(X_{k-1}) = \bigcup_{x_{k-1} \in X_{k-1}} F_k(x_{k-1}), \quad (2.49)$$

where $F_k(x_{k-1})$ is a Bernoulli RFS with params $(P_{S,k}(x_{k-1}), f_{k|k-1}(\cdot | x_{k-1}))$ for the state transition of the object x_{k-1} .

In this scheme, appearance of a set of newly born objects in the state space at a particular time k , denoted by Λ_k can be modeled as either a Poisson RFS, an i.i.d cluster RFS or a multi-Bernoulli RFS. Taking these newborn objects into account, the entire set

of objects X_k at time k is given by,

$$X_k = S_k(X_{k-1}) \cup \Lambda_k. \quad (2.50)$$

Using this formulation, the multi-object transition density for the multi-object set X_k at time k given the multi-object set X_{k-1} at time $k-1$ can be derived as,

$$f_{k|k-1}(X_k|X_{k-1}) = \sum_{W \subseteq X_k} \pi_S(W|X_{k-1}) \pi_\Lambda(X_k - W) \quad (2.51)$$

where $\pi_S(\cdot|X_{k-1})$ is the transition density of the surviving object set $S_k(X_{k-1})$ and $\pi_\Lambda(\cdot)$ is the probability density of the spontaneously born object set Λ_k and $X_k - W$ denotes the set difference between X_k and W .

2.4.4 Multi-object Observation model

Consider a multi-object state X_k at a particular time, where certain objects are detected and measurements are obtained while some objects go undetected. If an object in state x_k at time k is detected with the probability $P_{D,k}(x_k)$ and generates a noisy measurement z_k with likelihood $g_k(z_k|x_k)$, the random finite set $D_k(\cdot)$ of all measurements generated at time k due to the true object set X_k is modeled by,

$$D_k(X_k) = \bigcup_{x_k \in X_k} G_k(x_k), \quad (2.52)$$

where $G_k(x_k)$ is a Benoulli RFS with params $(P_{D,k}(x_k), g_k(\cdot|x_k))$ for the detection of the object x_k .

Apart from the true object related measurements, sensors generate incorrect measurements (clutter). These clutter measurements are generally modeled as Poisson processes. Let the clutter measurement RFS be denoted by K_k with intensity $\kappa_k(\cdot)$. Consequently the entire set of measurements Z_k generated at time k is given by,

$$Z_k(X_k) = D_k(X_k) \cup K_k. \quad (2.53)$$

Using this formulation, the probability density for the multi-object observation set Z_k

at time k given the multi-object set X_k at time k can be derived as,

$$g_k(Z_k | X_k) = \sum_{W \subseteq Z_k} \pi_D(W | X_k) \pi_K(Z_k - W) \quad (2.54)$$

where $\pi_D(\cdot | X_k)$ is the probability density of the detected observations $D_k(X_k)$ and $\pi_K(\cdot)$ is the probability density of the false measurements K_k .

2.4.5 Multi-object Bayes filter

The multi-object Bayes filter is a generalization of the single object Bayes filter discussed in [Section 2.1](#). Let $\pi_{k-1}(\cdot | Z_{1:k-1})$ denote the posterior density for the multi-object state at time $k-1$ where $Z_{1:k-1}$ denotes all measurements received by the filter from time 1 up to time k . In the multi-object Bayes filter, this density $\pi_{k-1}(\cdot | Z_{1:k-1})$ at time $k-1$, is predicted forward to obtain the predicted density $\pi_{k|k-1}(\cdot | Z_{1:k-1})$ using the Chapman-Kolmogorov equation as follows.

$$\pi_{k|k-1}(X_k | Z_{1:k-1}) = \int f_{k|k-1}(X_k | X) \pi_{k-1}(X | Z_{1:k-1}) \delta X. \quad (2.55)$$

Here, $f_{k|k-1}(\cdot | \cdot)$ denotes the multi-object transition density given in [Eq. \(2.51\)](#), which is derived from the multi-object transition model that includes new object births, existing object survivals and deaths.

The predicted density for the multi-object state is updated with the measurement set Z_k received at time k using Bayes rule to obtain the new posterior density $\pi_k(\cdot | Z_{1:k})$ at time k .

$$\pi_k(X_k | Z_{1:k}) = \frac{g_k(Z_k | X_k) \pi_{k|k-1}(X_k | Z_{1:k-1})}{\int g_k(Z_k | X) \pi_{k|k-1}(X | Z_{1:k-1}) \delta X}. \quad (2.56)$$

Here $g_k(\cdot | \cdot)$ denotes the multi-object likelihood function given in [Eq. \(2.54\)](#), which is derived from the multi-object observation model that includes missed detections, true measurements and false alarms. Note that the integral with respect to δX is the set integral from finite set statistics [13].

Thus, equations [Eq. \(2.55\)](#) and [Eq. \(2.56\)](#) together establish the recursive multi-object Bayes filter. In general, the numerical integrations and the combinatorial complexities in the multi-object densities in these equations render this filter intractable

[73]. The computational costs of implementations based on SMC methods discussed in [74–77] are too expensive for scenarios with a high number of tracked objects. More solutions to the multi-object Bayes filter by way of moment and density approximations were proposed later on in [70, 78–81]. These are discussed in length in the following subsections.

Probability Hypothesis Density filter

The PHD filter operates by recursively propagating the first moment of the multi-object posterior density, (i.e., PHD/intensity) forward in time. Since the PHD is a function defined on single object state space, the PHD filter circumvents the data association complexity of the multi-object Bayes filter. The ground work for the PHD filter was laid in [70] followed by an SMC implementation in [74] and a Gaussian mixture implementation in [78]. Convergence analyses for the PHD filter implementation in [74] are given in [74, 82] and a convergence analysis for the implementation in [78] is given in [83]. Estimates for individual object states can be extracted by selecting the points in the single-object state space with the highest intensities.

Let $v_{k-1}(\cdot)$ denote the posterior intensity at time $k - 1$, and $v_{B,k}$ the intensity of Poisson distributed new object births. The predicted intensity $v_{k|k-1}(\cdot)$ at time k is computed using the following equation.

$$v_{k|k-1}(x_k) = v_{B,k}(x_k) + \int P_{S,k}(x_{k-1}) f_{k|k-1}(x_k|x_{k-1}) v_{k-1}(x_{k-1}) dx_{k-1} \quad (2.57)$$

The updated intensity v_k is computed using the following equation.

$$v_k(x_k) = [1 - P_{D,k}(x_k)] v_{k|k-1}(x_k) + \sum_{z \in Z_k} \frac{P_{D,k}(x_k) g_k(z|x_k) v_{k|k-1}(x_k)}{\kappa_k(z) + \int P_{D,k}(x) g_k(z|x) v_{k|k-1}(x) dx} \quad (2.58)$$

The PHD filter has a complexity that is linear in the number of filtered objects and measurements.

Cardinalized Probability Hypothesis Density filter

Similar to the PHD filter, the CPHD filter also propagates the intensity of the multi-object density. The key difference is that while the PHD filter cardinality distribution

is Poisson, the CPHD filter cardinality distribution is not constrained to be Poisson, and could be arbitrary. Therefore, apart from the intensity, the cardinality distribution also needs to be propagated. Even though this makes the CPHD filter computationally more expensive than the PHD filter, that is offset by better accuracy in object state estimations and lower variance in the cardinality distribution. The foundation for the CPHD filter was laid in [79] and analytic implementations were given in [80]. Different implementations presented in [80] are based on the Kalman filter, EKF, UKF and the particle filter.

Let $\rho_{k-1}(\cdot)$ denote the posterior cardinality distribution at time $k-1$, $\rho_{B,k}(\cdot)$ denote the cardinality distribution of the spontaneously newborn objects at time k , $C_j^l = \frac{l!}{j!(l-j)!}$ stand for the Binomial coefficient and $e_j(\cdot)$ denote the elementary symmetric function of order j . The standard inner product is denoted by $\langle \cdot, \cdot \rangle$. The rest of the symbols are defined as in the PHD filter equations. The CPHD recursion is given below.

$$v_{k|k-1}(x_k) = v_{B,k}(x_k) + \int P_{S,k}(x_{k-1}) f_{k|k-1}(x_k|x_{k-1}) v_{k-1}(x_{k-1}) dx_{k-1}, \quad (2.59)$$

$$\rho_{k|k-1}(n) = \sum_{j=0}^n \rho_{B,k}(n-j) \Pi_{k|k-1}[v_{k-1}, \rho_{k-1}](j), \quad (2.60)$$

where,

$$\Pi_{k|k-1}[v, \rho](j) = \sum_{l=j}^{\infty} C_j^l \frac{\langle P_{S,k}, v \rangle^j \langle 1 - P_{S,k}, v \rangle^{l-j}}{\langle 1, v \rangle^l}.$$

$$v_k(x_k) = (1 - P_{D,k}(x_k)) \frac{\langle \Upsilon_k^1[v_{k|k-1}, Z], \rho_{k|k-1} \rangle}{\langle \Upsilon_k^0[v_{k|k-1}, Z_k], \rho_{k|k-1} \rangle} v_{k|k-1}(x_k) + \quad (2.61)$$

$$\sum_{z \in Z_k} \psi_{k,z}(x_k) \frac{\langle \Upsilon_k^1[v_{k|k-1}, Z_k - \{z\}], \rho_{k|k-1} \rangle}{\langle \Upsilon_k^0[v_{k|k-1}, Z_k], \rho_{k|k-1} \rangle} v_{k|k-1}(x_k),$$

$$\rho_k(n) = \frac{\Upsilon_k^0[v_{k|k-1}, Z_k](n) \rho_{k|k-1}(n)}{\langle \Upsilon_k^0[v_{k|k-1}, Z_k], \rho_{k|k-1} \rangle}, \quad (2.62)$$

where,

$$\Upsilon_k^u [v, Z] = \sum_{j=0}^{\min(|Z|, n)} (|Z| - j)! \rho_{K,k}(|Z| - j) P_{j+u}^n \frac{\langle 1 - P_{D,k}, v \rangle^{n-(j+u)}}{\langle 1, v \rangle^n} e_j(\Xi_k(v, Z)),$$

$$\psi_{k,z}(x) = \frac{\langle 1, \kappa_k \rangle}{\kappa_k(z)} P_{D,k}(x) g_k(z|x),$$

$$\Xi_k(v, Z) = \{ \langle v, \psi_{k,z} \rangle : z \in Z \},$$

$\rho_{K,k}(\cdot)$ = cardinality distribution of clutter at time k ,

$$e_j(Z) = \sum_{S \subseteq Z, |S|=j} \left(\prod_{\zeta \in S} \zeta \right), \quad \text{with } e_0(Z) = 1 \text{ by convention.}$$

The CPHD filter has a complexity that is linear in the number of filtered objects and cubic in the number of measurements.

Cardinality balanced multi-Bernoulli filter

The cardinality balanced multi-Bernoulli (CBMeMber) filter [81] operates by propagating the parameters of a multi-Bernoulli RFS that approximates the posterior multi-object RFS. A Gaussian mixture implementation for linear Gaussian models (with EKF, UKF extensions for mildly non-linear Gaussian scenarios) as well as an SMC implementation to accommodate nonlinear models was proposed for the CBMeMber recursion in [81]. The SMC-PHD filters require particle clustering to extract state estimates which is computationally expensive, while the CBMeMber filter object state extraction is more accurate and less expensive. Formulas for the CBMeMber recursion is given as follows.

Let the posterior density at time $k - 1$ be represented as the multi-Bernoulli RFS with parameter set $\pi_{k-1} = \left\{ \left(r_{k-1}^{(i)}, p_{k-1}^{(i)} \right) \right\}_{i=1}^{M_{k-1}}$. If the density of the new born objects is distributed as the multi-Bernoulli RFS with parameter set $\pi_{B,k} = \left\{ r_{B,k}^{(i)}, p_{B,k}^{(i)} \right\}_{i=1}^{M_{B,k}}$, the predicted multi-Bernoulli density at time k is given by,

$$\pi_{k|k-1} = \left\{ \left(r_{P,k|k-1}^{(i)}, p_{P,k|k-1}^{(i)} \right) \right\}_{i=1}^{M_{k-1}} \cup \left\{ \left(r_{B,k}^{(i)}, p_{B,k}^{(i)} \right) \right\}_{i=1}^{M_{B,k}} \quad (2.63)$$

where

$$r_{P,k|k-1}^{(i)} = r_{k-1}^{(i)} \left\langle p_{k-1}^{(i)}, P_{S,k} \right\rangle,$$

$$p_{P,k|k-1}^{(i)}(x) = \frac{\left\langle f_{k|k-1}(x|\cdot), p_{k-1}^{(i)} P_{S,k} \right\rangle}{\left\langle p_{k-1}^{(i)}, P_{S,k} \right\rangle}.$$

Let $\left\{ \left(r_{k|k-1}^{(i)}, p_{k|k-1}^{(i)} \right) \right\}_{i=1}^{M_{k|k-1}}$ represent the (re-indexed) predicted multi-Bernoulli parameter set in Eq. (2.63). The updated multi-Bernoulli density has parameter set,

$$\pi_k \approx \left\{ \left(r_{L,k}^{(i)}, p_{L,k}^{(i)} \right) \right\}_{i=1}^{M_{k|k-1}} \cup \left\{ \left(r_{U,k}^*(z), p_{U,k}^*(\cdot; z) \right) \right\}_{z \in Z_k} \quad (2.64)$$

where

$$r_{L,k}^{(i)} = r_{k|k-1}^{(i)} \frac{1 - \left\langle p_{k|k-1}^{(i)}, P_{D,k} \right\rangle}{1 - r_{k|k-1}^{(i)} \left\langle p_{k|k-1}^{(i)}, P_{D,k} \right\rangle},$$

$$p_{L,k}^{(i)}(x) = p_{k|k-1}^{(i)}(x) \frac{1 - P_{D,k}(x)}{1 - \left\langle p_{k|k-1}^{(i)}, P_{D,k} \right\rangle},$$

$$r_{U,k}^*(z) = \frac{\sum_{i=1}^{M_{k|k-1}} \frac{r_{k|k-1}^{(i)} (1 - r_{k|k-1}^{(i)}) \left\langle p_{k|k-1}^{(i)}, \psi_{k,z} \right\rangle}{\left(1 - r_{k|k-1}^{(i)} \left\langle p_{k|k-1}^{(i)}, P_{D,k} \right\rangle \right)^2}}{\kappa_k(z) + \sum_{i=1}^{M_{k|k-1}} \frac{r_{k|k-1}^{(i)} \left\langle p_{k|k-1}^{(i)}, \psi_{k,z} \right\rangle}{1 - r_{k|k-1}^{(i)} \left\langle p_{k|k-1}^{(i)}, P_{D,k} \right\rangle}}$$

$$p_{U,k}^*(x; z) = \frac{\sum_{i=1}^{M_{k|k-1}} \frac{r_{k|k-1}^{(i)}}{1 - r_{k|k-1}^{(i)}} p_{k|k-1}^{(i)}(x) \psi_{k,z}(x)}{\sum_{i=1}^{M_{k|k-1}} \frac{r_{k|k-1}^{(i)}}{1 - r_{k|k-1}^{(i)}} \left\langle p_{k|k-1}^{(i)}, \psi_{k,z} \right\rangle},$$

$$\psi_{k,z}(x) = g_k(z|x) p_{D,k}(x).$$

Similar to the PHD filter the CBMeMber filter has a complexity that is linear in the number of hypothesized objects and received measurements. A convergence analysis of the SMC-CBMeMber implementation was given in [84].

2.5 Multi-object tracking with labeled random finite sets

Labeled random finite sets are RFSs where each element in the RFS is assigned a unique label/tag. Due to the uniqueness of the labels, the cardinality of the label set of a

labeled RFS is equal to the cardinality of the labeled RFS itself. The formal definition is given below.

Definition 1. A labeled RFS with state space \mathbb{X} and discrete label space \mathbb{L} , is an RFS on $\mathbb{X} \times \mathbb{L}$, such that the labels within each realization are always distinct.

By convention, a labeled element is represented by bold lower case e.g. $\mathbf{x} = (x, \ell)$ where x denotes the kinematic/feature object state and ℓ denotes the object label. A labeled set is denoted by a bold uppercase. e.g. $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$. Spaces are represented by blackboard bold. e.g. $\mathbb{X}, \mathbb{L}, \mathbb{Z}$. The cardinality of a set is denoted by $|\cdot|$. The rest of the notation and operators related to labeled RFSs is now introduced.

- Distinct label indicator

$$\Delta(\mathbf{X}) \triangleq \delta_{|\mathbf{X}|}(|\mathcal{L}(\mathbf{X})|). \quad (2.65)$$

- Standard inner product

$$\langle f, g \rangle \triangleq \int f(x) g(x) dx. \quad (2.66)$$

- Multi-object exponential

$$[h]^{\mathbf{X}} \triangleq \prod_{\mathbf{x} \in \mathbf{X}} h(\mathbf{x}). \quad (2.67)$$

- Kronecker delta function

$$\delta_Y(X) \triangleq \begin{cases} 1, & \text{if } X = Y \\ 0, & \text{otherwise} \end{cases}. \quad (2.68)$$

- Inclusion function

$$1_Y(X) \triangleq \begin{cases} 1, & \text{if } X \subseteq Y \\ 0, & \text{otherwise} \end{cases}. \quad (2.69)$$

- Label extractor

$$\mathcal{L} : \mathbb{X} \times \mathbb{L} \rightarrow \mathbb{L} \text{ is the projection } \mathcal{L}(\mathbf{X}) = \{\mathcal{L}(\mathbf{x}) : \mathbf{x} \in \mathbf{X}\} \text{ where } \mathcal{L}((x, \ell)) = \ell. \quad (2.70)$$

Note that a realisation \mathbf{X} of a labeled RFS always satisfies $\Delta(\mathbf{X}) = 1$ and $|\mathcal{L}(\mathbf{X})| = |\mathbf{X}|$.

The density of a labeled RFS is not the same as its unlabeled counterpart. For a labeled RFS distributed according to π , its unlabeled version π is distributed according to,

$$\pi(\{x_1, x_2, \dots, x_n\}) = \sum_{(\ell_1, \ell_2, \dots, \ell_n) \in \mathbb{L}^n} \pi(\{(x_1, \ell_1), (x_2, \ell_2), \dots, (x_n, \ell_n)\}). \quad (2.71)$$

Due to the discrete label space \mathbb{L} , the set integral for a function $h : \mathcal{F}(\mathbb{X} \times \mathbb{L}) \rightarrow \mathbb{R}$ is given as,

$$\int h(\mathbf{X}) \delta \mathbf{X} = \sum_{i=0}^{\infty} \frac{1}{i!} \sum_{(\ell_1, \dots, \ell_i) \in \mathbb{L}^i} \int_{\mathbb{X}^i} h(\{(x_1, \ell_1), \dots, (x_i, \ell_i)\}) dx_1, \dots, dx_i \quad (2.72)$$

By treating the multi-object state as a labeled RFS, the multi-object Bayes filter is able to output object IDs (trajectories) by extracting the label information from the label augmented state. The rest of this section gives details on such labeled RFS based multi-object tracking filters and the probability distribution families upon which they are constructed.

2.5.1 Labeled random finite sets

Two versatile classes of labeled RFS, named the labeled multi-Bernoulli (LMB) RFS [15] and the generalized labeled multi-Bernoulli (GLMB) RFS [15] are described in the following.

Definition 2. A *labeled multi-Bernoulli RFS* is a labeled RFS on state space \mathbb{X} and (discrete) label space \mathbb{L} parameterised by $\{r^{(\ell)}, p^{(\ell)}(\cdot)\}_{\ell \in \mathbb{L}}$ is distributed according to,

$$\pi(\mathbf{X}) = \Delta(\mathbf{X}) \omega(\mathcal{L}(\mathbf{X}) [p]^{\mathbf{X}}) \quad (2.73)$$

where

$$\omega(L) = \prod_{i \in \mathbb{L}} (1 - r^{(i)}) \prod_{\ell \in L} 1_L(\ell) \left(\frac{r^{(\ell)}}{1 - r^{(\ell)}} \right),$$

$$p(x, \ell) = p^{(\ell)}(x),$$

$$r^{(\ell)} = \text{existence probability of the element associated with label } \ell,$$

$$p^{(\ell)}(\cdot) = \text{probability density of the kinematic state associated with label } \ell.$$

Definition 3. A *generalized labeled multi-Bernoulli RFS* is a labeled RFS with state space \mathbb{X} and (discrete) label space \mathbb{L} distributed according to,

$$\pi(\mathbf{X}) = \Delta(\mathbf{X}) \sum_{c \in \mathbb{C}} \omega^{(c)}(\mathcal{L}(\mathbf{X})) [p^{(c)}]^{\mathbf{X}} \quad (2.74)$$

where \mathbb{C} is a discrete index set with $\omega^{(c)}(L)$ and $p^{(c)}$ satisfying

$$\begin{aligned} \sum_{L \subseteq \mathbb{L}} \sum_{c \in \mathbb{C}} \omega^{(c)}(L) &= 1, \\ \int_{x \in \mathbb{X}} p^{(c)}(x, \ell) dx &= 1. \end{aligned}$$

A GLMB is a mixture of multi-object exponentials. Each mixture component consists of a weight $\omega^{(c)}(\mathcal{L}(\mathbf{X}))$ and a multi-object exponential $[p^{(c)}]^{\mathbf{X}}$. The weight $\omega^{(c)}(\mathcal{L}(\mathbf{X}))$ only depends on the labels of the multi-object state, and $[p^{(c)}]^{\mathbf{X}}$ depends on the entire multi-object state.

Note that the LMB RFS is a special case of the GLMB RFS comprising of a single component.

It was also proven in [15] that if the posterior multi-object density at a particular time is distributed as a GLMB density, under the standard multi-object dynamic model which includes object survivals, death and new object births, the multi-object density predicted in the next time step can also be represented as a GLMB (i.e., closure under the Chapman Kolmogorov equation). Further, if the predicted GLMB density was updated under the standard multi-object observation model that includes object detections, missed detections and clutter, the updated density can also be written in the form of a GLMB density. In other words, the GLMB distribution is said to be a conjugate prior under Bayes rule. Since the prior, predicted and updated densities can all be written as GLMB densities this provides with a recursion mechanism to propagate the multi-object density forward in time. Further, in [15], an alternative representation of the GLMB distribution that facilitates implementation, referred to as the δ -GLMB distribution was introduced. It is also closed under the Chapman-Kolmogorov equation and Bayes rule. Representing the multi-object state as a δ -GLMB allows for a tractable solution for multi-object tracking that can be implemented via Gaussian mixture or SMC with lower computational cost and memory requirements.

Definition 4. A δ -generalized labeled multi-Bernoulli RFS with state space \mathbb{X} , (discrete) label space \mathbb{L} , and discrete space Ξ is a special case of the generalized labeled multi-Bernoulli with

$$\begin{aligned}\mathbb{C} &= \mathcal{F}(\mathbb{L}) \times \Xi, \\ \omega^{(c)}(L) &= \omega^{(I,\xi)}(L) = \omega^{(I,\xi)} \delta_I(L), \quad \text{where } L \in \mathcal{F}(\mathbb{L}), \\ p^{(c)} &= p^{(I,\xi)} = p^{(\xi)}.\end{aligned}$$

i.e., δ -generalized labeled multi-Bernoulli RFS is distributed according to,

$$\pi(\mathbf{X}) = \Delta(\mathbf{X}) \sum_{(I,\xi) \in \mathcal{F}(\mathbb{L}) \times \Xi} \omega^{(I,\xi)} \delta_I(\mathcal{L}(\mathbf{X})) \left[p^{(\xi)} \right]^{\mathbf{X}}. \quad (2.75)$$

In multi-object tracking, the δ -GLMB density can be used to represent the multi-object density which is viewed as a set of weighted components generated over the space of $\mathcal{F}(\mathbb{L}) \times \Xi$. Each component is represented by a pair (label set, association map history) denoted by (I, ξ) from space $\mathcal{F}(\mathbb{L}) \times \Xi$. The association map history ξ is a collection of object label-to-measurement mappings up to the current time which resulted in the current set of labels I . Therefore the component (I, ξ) represents the hypothesis that the label set I has the association map history ξ . The probability (also referred to as the weight) of the component (I, ξ) is given by $\omega^{(I,\xi)}$ and $p^{(\xi)}(\cdot, \ell)$ represents the density for the kinematic state of each object in label set I . The $\Delta(\cdot)$ operator ensures that the probability of a multi-object state with repeated labels is zero. The δ -GLMB filter for multi-object tracking operates by recursively propagating prior, predicted and posterior multi-object densities in δ -GLMB form forward in time.

GLMB representation of the multi-object state

Remark. For the remainder of this chapter, for succinctness of notations, we will suppress the time index k and use '+' to indicate the sets, functions, parameters and densities relating to the next time step. e.g., P_S used in place of $P_{S,k}$ and $P_{S,+}$ used in place of $P_{S,k+1}$.

As described in [Section 2.4.3](#), an object existing in the current time step may survive to the next time step with a certain probability and change state according to a particu-

lar state transition function or may not survive. Note that the label remains unchanged during the transition and only the kinematic state of the object changes. The surviving set of objects \mathbf{S} out of \mathbf{X} is distributed according to,

$$\pi_{\mathbf{S}}(\mathbf{S}|\mathbf{X}) = \Delta(\mathbf{S})\Delta(\mathbf{X}) 1_{\mathcal{L}(\mathbf{X})}(\mathcal{L}(\mathbf{S})) [\Phi(\mathbf{S}; \cdot)]^{\mathbf{X}} \quad (2.76)$$

where,

$$\Phi(\mathbf{S}; x, \ell) = \sum_{(x_+, \ell_+) \in \mathbf{S}} \delta_{\ell}(\ell_+) P_S(x, \ell) f_+(x_+|x, \ell) + (1 - 1_{\mathcal{L}(\mathbf{S})}(\ell)) (1 - P_S(x, \ell)).$$

Let \mathbb{B}_+ denote the entire label space for objects newly born in the next time step and \mathbf{B}_+ denote the set of the newborn objects. Let \mathbb{L} denote the label space for objects at the current time step (includes the labels of all objects born up to that time). i.e., $\mathbb{L}_+ = \mathbb{L} \cup \mathbb{B}_+$. A conventional scheme for labeling objects is to formulate the label as a pair consisting of the birth time step and a unique index for objects born during that time step $\ell = (k, i)$. This scheme of labeling ensures that $\mathbb{L} \cap \mathbb{B}_+ = \emptyset$. For a labeled multi-Bernoulli birth model with parameter set $\left\{ \left(r_{B,+}^{(\ell)}, p_{B,+}^{(\ell)}(\cdot) \right) \right\}_{\ell \in \mathbb{B}_+}$, the distribution of \mathbf{B}_+ can be given as the multi-Bernoulli RFS,

$$\pi_{\mathbf{B}}(\mathbf{B}_+) = \Delta(\mathbf{B}_+) \omega_{\mathbf{B}}(\mathcal{L}(\mathbf{B}_+)) [p_{B,+}(\cdot)]^{\mathbf{B}_+}. \quad (2.77)$$

$$\omega_{\mathbf{B}}(\mathcal{L}(\mathbf{B}_+)) = \prod_{i \in \mathbb{B}_+} \left(1 - r_{B,+}^{(i)} \right) \prod_{\ell \in \mathcal{L}(\mathbf{B}_+)} \frac{1_{\mathbb{B}_+}(\ell) r_{B,+}^{(\ell)}}{1 - r_{B,+}^{(\ell)}},$$

$$p_{B,+}(x, \ell) = p_{B,+}^{(\ell)}(x).$$

Hence, using Eq. (2.76) and Eq. (2.77) for the complete set of objects in the next time step denoted by $\mathbf{X}_+ = \mathbf{S} \uplus \mathbf{B}_+$, the multi-object transition kernel $f_+(\cdot|\cdot) : \mathcal{F}(\mathbb{X} \times \mathbb{L}_+) \times \mathcal{F}(\mathbb{X} \times \mathbb{L}) \rightarrow [0, \infty)$ is given by,

$$f_+(\mathbf{X}_+|\mathbf{X}) = \pi_{\mathbf{S}}(\mathbf{X}_+ \cap (\mathbb{X} \times \mathbb{L}) | \mathbf{X}) \pi_{\mathbf{B}}(\mathbf{X}_+ \cap (\mathbb{X} \times \mathbb{B}_+)) \quad (2.78)$$

Let \mathbb{Z} denote the observation space. An object $\mathbf{x} \in \mathbf{X}$ is either detected with probability $P_D(\mathbf{x})$ and generates a measurement z with likelihood $g(z|\mathbf{x})$ or misdetected

with probability $1 - P_D(\mathbf{x})$. Then the set $W \subset \mathbb{Z}$ of detections is distributed according to,

$$\pi_D(W|\mathbf{X}) = \{P_D(\mathbf{x}), g(\cdot|\mathbf{x}) : \mathbf{x} \in \mathbf{X}\}(W). \quad (2.79)$$

The set $K \in \mathcal{F}(\mathbb{Z})$ of false measurements generally modeled as a Poisson RFS with independent and identically distributed points is distributed according to,

$$\pi_K(K) = e^{-(\kappa,1)} [\kappa]^K \quad (2.80)$$

where $\kappa(\cdot)$ is the intensity function of the Poisson process.

Thus, the multi-object observation $Z = W \cup K$ is distributed according to the multi-object likelihood,

$$g(Z|\mathbf{X}) = \sum_{W \subseteq Z} \pi_D(W|\mathbf{X}) \pi_K(Z - W). \quad (2.81)$$

2.5.2 Generalized labeled multi-Bernoulli filter

The GLMB filter is constructed using the conjugacy of the GLMB density under the standard multi-object dynamic and observation models described in the previous section. The explicit formulas for computing the predicted and updated GLMB densities from the posterior density of the previous time step are given in the following.

GLMB filter with separate prediction and update

GLMB Prediction

If the posterior multi-object density at the current time is given by Eq. (2.75), under the standard multi-object dynamic model the predicted multi-object density at the next time step is a GLMB of the form,

$$\pi_+(\mathbf{X}_+) = \Delta(\mathbf{X}_+) \sum_{(I_+, \xi) \in \mathcal{F}(\mathbb{L}_+) \times \Xi} \omega_+^{(I_+, \xi)} \delta_{I_+}(\mathcal{L}(\mathbf{X}_+)) \left[p_+^{(\xi)} \right]^{\mathbf{X}_+} \quad (2.82)$$

where,

$$\begin{aligned}
\omega_+^{(I_+, \xi)} &= \omega_B(I_+ \cap \mathbb{B}_+) \omega_S^{(\xi)}(I_+ \cap \mathbb{L}), \\
\omega_B(B) &= \prod_{i \in \mathbb{B}_+} (1 - r_{B,+}^{(i)}) \prod_{\ell \in B} \frac{1_{\mathbb{B}_+}(\ell) r_{B,+}^{(\ell)}}{1 - r_{B,+}^{(\ell)}}, \\
\omega_S^{(\xi)}(L) &= [\bar{P}_S^{(\xi)}]^L \sum_{I \ni L} [1 - \bar{P}_S^{(\xi)}]^{I-L} \omega^{(I, \xi)}, \\
p_+^{(\xi)}(x_+, \ell_+) &= 1_{\mathbb{L}}(\ell_+) p_S^{(\xi)}(x_+, \ell_+) + 1_{\mathbb{B}_+}(\ell_+) p_{B,+}(x_+, \ell_+), \\
p_S^{(\xi)}(x_+, \ell_+) &= \frac{\langle P_S(\cdot, \ell_+) f_+(x_+ | \cdot, \ell_+), p^{(\xi)}(\cdot, \ell_+) \rangle}{\bar{P}_S^{(\xi)}(\ell_+)}, \\
\bar{P}_S^{(\xi)}(\ell_+) &= \langle P_S(\cdot, \ell_+), p^{(\xi)}(\cdot, \ell_+) \rangle, \\
p_{B,+}(x_+, \ell_+) &= p_{B,+}^{(\ell_+)}(x_+).
\end{aligned}$$

For implementation purposes, it is not feasible to generate hypotheses on the entire space of mappings $\mathcal{F}(\mathbb{L}) \times \Xi$. Therefore the predicted density is approximated by a truncated version which consists of a selected number of hypotheses with the highest weights. The k-shortest path algorithm is used in [15] to generate the best hypotheses with highest weights for each parent hypothesis (I, ξ) . The K-shortest paths algorithm is a popular algorithm to find the K number of paths with minimum costs between two given nodes in a weighted graph [85]. In the GLMB filter implementation, best hypotheses for newborn objects are generated first, followed by best hypotheses for surviving objects. These two groups of hypotheses which were independently generated by solving two separate K-shortest path problems are then combined to create the overall best predicted hypotheses.

The usage of K-shortest path algorithm in generating the best surviving hypotheses from parent hypothesis (I, ξ) is illustrated in [Figure 2.1](#) (reproduced from [86]). The cost $C^{(I, \xi)}(\ell_j)$ of node ℓ_j is calculated as,

$$C^{(I, \xi)}(\ell_j) = -\ln \left[\frac{\bar{P}_S^{(\xi)}(\ell_j)}{1 - \bar{P}_S^{(\xi)}(\ell_j)} \right]. \quad (2.83)$$

The direction between two nodes is from the node with a lower cost to the one with a

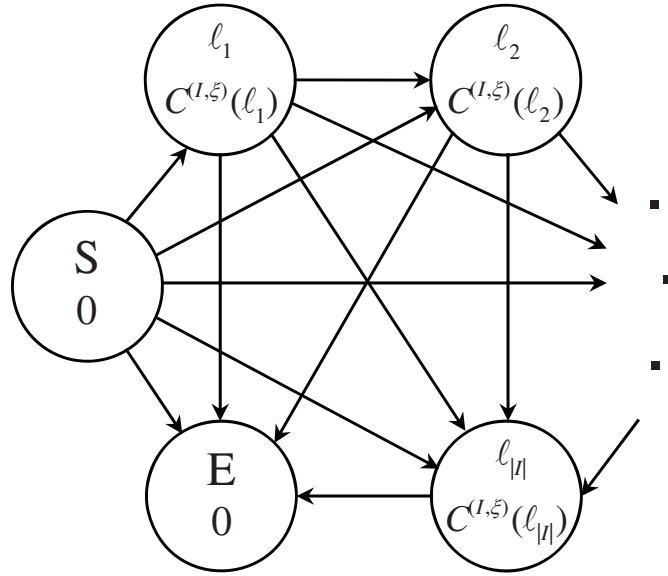


Figure 2.1: The directed graph with nodes $l_1, l_2, \dots, l_{|I|} \in I$ and corresponding costs $c^{(I, \xi)}(l_1), \dots, c^{(I, \xi)}(l_{|I|})$ for solving the K shortest path problem for surviving objects from parent hypothesis (I, ξ) . Start and end nodes are denoted by S, E respectively [86].

higher or equal cost. The distance $\mathcal{D}(l_j, l_{j'})$ from a node l_j to $l_{j'}$ is defined as,

$$\mathcal{D}(l_j, l_{j'}) = \begin{cases} C^{(I, \xi)}(l_{j'}) & \text{if } j < j' \\ \infty, & \text{otherwise} \end{cases}. \quad (2.84)$$

Consequently, a path from S to E through the set of nodes $J \in I$ has a total cost of,

$$\sum_{\ell \in J} C^{(I, \xi)}(\ell) = - \sum_{\ell \in J} \ln \left[\frac{\bar{P}_S^{(\xi)}(\ell)}{1 - \bar{P}_S^{(\xi)}(\ell)} \right] = -\ln \left(\prod_{\ell \in J} \left[\frac{\bar{P}_S^{(\xi)}(\ell)}{1 - \bar{P}_S^{(\xi)}(\ell)} \right] \right). \quad (2.85)$$

Thus solving the K-shortest path problem results in a K number of subsets of I in the order of increasing total cost.

In order to solve the K-shortest path problem for newborn objects, a second weighted graph is constructed in which the cost $C_{B,+}(l_j)$ of a node l_j is calculated as,

$$C_{B,+}(l_j) = -\ln \left[\frac{r_{B,+}^{(\ell)}(l_j)}{1 - r_{B,+}^{(\ell)}(l_j)} \right] \quad \text{where } \ell \in \mathbb{B}_+. \quad (2.86)$$

GLMB Update

If the multi-object prior at the current time step is given by Eq. (2.75), upon receiving a set of measurements Z , under the standard multi-object observation model the posterior density is δ -GLMB of the form,

$$\pi_Z(\mathbf{X}) = \Delta(\mathbf{X}) \sum_{(I,\xi) \in \mathcal{F}(\mathbb{L}) \times \Xi} \sum_{\theta \in \Theta} \omega^{(I,\xi,\theta)}(Z) \delta_I(\mathcal{L}(\mathbf{X})) \left[p^{(\xi,\theta)}(\cdot|Z) \right]^{\mathbf{X}} \quad (2.87)$$

where Θ is the space of mappings $\theta : \mathbb{L} \rightarrow \{0, 1, \dots, |Z|\}$, such that $\theta(i) = \theta(i') > 0$ implies $i = i'$ and

$$\begin{aligned} \omega^{(I,\xi,\theta)}(Z) &= \frac{\delta_{\theta^{-1}\{0;|Z|\}}(I) \omega^{(I,\xi)} \left[\bar{\psi}_Z^{(\xi,\theta)} \right]^I}{\sum_{(I,\xi) \in \mathcal{F}(\mathbb{L}) \times \Xi} \sum_{\theta \in \Theta} \delta_{\theta^{-1}\{0;|Z|\}}(I) \omega^{(I,\xi)} \left[\bar{\psi}_Z^{(\xi,\theta)} \right]^I}, \quad (2.88) \\ p^{(\xi,\theta)}(x, \ell|Z) &= \frac{p^{(\xi)}(x, \ell) \psi_Z^{\theta(\ell)}(x, \ell)}{\bar{\psi}_Z^{(\xi,\theta)}(\ell)}, \\ \bar{\psi}_Z^{(\xi,\theta)}(\ell) &= \left\langle p^{(\xi)}(\cdot, \ell) \psi_Z^{\theta(\ell)}(\cdot, \ell) \right\rangle, \\ \psi_Z^j(x, \ell) &= \delta_0(j) [1 - P_D(x, \ell)] + [1 - \delta_0(j)] \frac{[P_D(x, \ell)] g(z_j|x, \ell)}{\kappa(z_j)}. \end{aligned}$$

As in the prediction step, in the update, it is not feasible to generate hypotheses on the entire space of mappings $\mathcal{F}(\mathbb{L}) \times \Xi \times \Theta$. Therefore the updated density is approximated by a truncated version which consists of hypotheses with the highest weights. In order to do this, for each parent hypothesis (I, ξ) , mappings are generated on the smaller subset $\Theta^{(T^{(I,\xi)})}$, which is the set of $T^{(I,\xi)}$ elements of Θ with the highest weights. A different $T^{(I,\xi)}$ value could be chosen for each hypothesis (I, ξ) based on some heuristic. A good method for determining $T^{(I,\xi)}$ as mentioned in [15] is to compute $T^{(I,\xi)}$ proportional to $\omega^{(I,\xi,\theta)}(Z)$. Murty's algorithm or Gibbs sampling has been used in the literature [86, 87] to select the best $T^{(I,\xi)}$ mappings without exhaustively iterating through the entire mappings space.

The Hungarian method [88] is a traditional method that has been in use to solve the optimal assignment problem. Murty's algorithm [89] is an extension of the hungarian method to obtain the T best assignments. In order to utilize Murty's algorithm in tracking, a cost matrix with the costs of each object-to-measurement mapping needs

to be provided to the algorithm. To generate $T^{(I,\xi)}$ best assignments out of parent hypothesis (I, ξ) for measurement set Z , the cost matrix $C_Z^{(I,\xi)}$ with a row count equal to $|I|$ and a column count equal to $|Z|$ is generated where $C_{i,j}$ denotes the i^{th} row j^{th} column element (see Eq. (2.89)). From Eq. (2.88) it can be seen that the weight of the hypothesis $\omega^{(I,\xi,\theta)}(Z)$ is proportional to $\omega^{(I,\xi)} \left[\bar{\psi}_Z^{(\xi,\theta)} \right]^I$. Therefore $\left[\bar{\psi}_Z^{(\xi,\theta)} \right]^I$ can be treated as the overall cost of assigning objects with label set I to measurement set Z under mapping θ . The i^{th} row j^{th} column cost matrix element $C_{i,j}$ is populated with the cost of assigning the i^{th} object label ℓ_i to j^{th} measurement z_j . Any unassigned object ℓ_i giving $\theta(\ell_i) = 0$ symbolizes a missed detection and $\theta^{-1}(z_j) = 0$ indicates a false measurement. i.e.,

$$C_Z^{(I,\xi)} = \begin{bmatrix} C_{1,1} & C_{1,2} \dots C_{1,|Z|} \\ C_{2,1} & C_{2,2} \dots C_{2,|Z|} \\ \dots & \dots \\ C_{|I|,1} & C_{|I|,2} \dots C_{|I|,|Z|} \end{bmatrix}, \quad (2.89)$$

where

$$C_{i,j} = -\ln \left(\frac{\langle p^{(\xi)}(\cdot, \ell_i), (P_D(\cdot, \ell_i))g(j|\cdot, \ell_i) \rangle}{\langle p^{(\xi)}(\cdot, \ell_i), (1 - P_D(\cdot, \ell_i)) \kappa(z_j) \rangle} \right).$$

In addition to generating a predetermined number of best hypotheses in both the prediction and update stages, a further truncation of the multi-object density is performed after each stage due to the exponentially growing number of hypotheses in the density. An intuitive strategy to carry out this is simply to eliminate hypotheses with weights below a pre-determined threshold. It is established in [15] that retaining the hypotheses with highest weights and discarding those with lowest weights using previously described methods minimizes the L_1 error between the GLMB density and its truncated version. Extensive implementation details regarding the GLMB filter are given in [86]. The complexity of the implementation depends on the complexity of the Murty's algorithm which is of quartic in the number of measurements.

GLMB filter with joint prediction and update

More efficient implementations for the GLMB filter by combining the prediction and update into a single step are presented in [87, 90]. While the standard GLMB filter described in Section 2.5.2 requires separate truncations in the prediction and update

steps, the implementations with a joint prediction and update step require only one truncation step in each iteration.

Separating the truncation of the prediction from the update results in a significant portion of the predicted hypotheses generating updated hypotheses with insignificant weights. Hence, expensive ranked assignment problems are solved on these hypotheses that will later be discarded. The design proposed in [87] aims to mitigate this wastage of computations and has a complexity that is linear in the number of measurements and quadratic in the number of hypothesized objects.

If the GLMB posterior density at the current time is of the form [Eq. \(2.75\)](#), under the joint prediction and update scheme, upon receiving measurement set Z_+ in the next time step, the posterior density is given by,

$$\pi_{Z_+}(\mathbf{X}_+) = \Delta(\mathbf{X}_+) \sum_{I, \xi, I_+, \theta_+} \omega^{(I, \xi)} \omega_{Z_+}^{(I, \xi, I_+, \theta_+)} \delta_{I_+}[\mathcal{L}(\mathbf{X}_+)] \left[p_{Z_+}^{(\xi, \theta_+)} \right]^{\mathbf{X}_+} \quad (2.90)$$

where $I \in \mathcal{F}(\mathbb{L})$, $\xi \in \Xi$, $I_+ \in \mathcal{F}(\mathbb{L}_+)$, $\theta_+ \in \Theta_+$, and

$$\begin{aligned} \omega_{Z_+}^{(I, \xi, I_+, \theta_+)} &= 1_{\Theta_+(I_+)}(\theta_+) \left[1 - \bar{P}_S^{(\xi)} \right]^{I - I_+} \left[\bar{P}_S^{(\xi)} \right]^{I \cap I_+} \left[1 - r_{B,+} \right]^{\mathbb{B}_+ - I_+} \left[r_{B,+} \right]^{\mathbb{B}_+ \cap I_+} \left[\bar{\psi}_{Z_+}^{(\xi, \theta_+)} \right]^{I_+}, \\ \bar{P}_S^{(\xi)}(\ell_+) &= \left\langle p^{(\xi)}(\cdot, \ell_+), P_S(\cdot, \ell_+) \right\rangle, \\ \bar{\psi}_{Z_+}^{(\xi, \theta_+)}(\ell_+) &= \left\langle \bar{p}_+^{(\xi)}(\cdot, \ell_+), \psi_{Z_+}^{\theta_+(\ell_+)}(\cdot, \ell_+) \right\rangle, \\ \psi_{Z_+}^j(x_+, \ell_+) &= \delta_0(j) \left[1 - P_{D,+}(x_+, \ell_+) \right] + \\ &\quad \left[1 - \delta_0(j) \right] \frac{\left[P_{D,+}(x_+, \ell_+) \right] g_+(z_j | x_+, \ell_+)}{\kappa_+(z_j)}, \\ \bar{p}_+^{(\xi)}(x_+, \ell_+) &= 1_{\mathbb{L}}(\ell_+) \frac{\left\langle P_S(\cdot, \ell_+) f_+(x_+ | \cdot, \ell_+) p^{(\xi)}(\cdot, \ell_+) \right\rangle}{\bar{P}_S^{(\xi)}(\ell_+)} + \\ &\quad 1_{\mathbb{B}_+}(\ell_+) p_{B,+}(x_+, \ell_+), \\ p_{Z_+}^{(\xi, \theta_+)}(x_+, \ell_+) &= \frac{\bar{p}_+^{(\xi)}(x_+, \ell_+) \psi_{Z_+}^{\theta_+(\ell_+)}(x_+, \ell_+)}{\bar{\psi}_{Z_+}^{(\xi, \theta_+)}(\ell_+)}. \end{aligned}$$

It can be seen that in [equation Eq. \(2.90\)](#) the summation on the right is an enumeration of all possible combinations of newborn, dead and surviving object labels in association with the new measurements. To propagate the posterior, for each hypothesis (I, ξ) in the posterior density, a set of pairs $(I_+, \theta_+) \in \mathcal{F}(\mathbb{L}_+) \times \Theta_+(I_+)$ with

significant weights $\omega_{Z_+}^{(I,\xi,I_+,\theta_+)}$ need to be generated without exhaustively searching the space $\mathcal{F}(\mathbb{L}_+) \times \Theta_+(I_+)$. In [90], this is achieved by solving ranked assignment problems (Murty's algorithm) and in [87] Gibbs sampling is used to generate hypotheses with significant weights. Further, pruning of hypotheses with negligible weights at the end of each iteration is required to keep the exponentially growing hypotheses count in the density manageable.

Ranked assignment using Murty's algorithm is a technique that can be employed to obtain a finite number of best association mappings without exhaustively generating all possible mappings. It has a quartic complexity in the number of measurements. In using Murty's algorithm to extract the $T^{(I,\xi)}$ hypotheses, the cost matrix C with a row count equal to $P = |I| + |\mathbb{B}_+|$ and a column count equal to $|Z_+| + 2P$ is constructed. The extra $2P$ columns in the cost matrix correspond to misdetections and unsurviving/unborn objects and $j \in \{-1 : |Z_+|\}$ is the measurement index. This is illustrated in Figure 2.2 which has been reproduced from [87]. The i^{th} row j^{th} column element $\eta_i(j)$ of the cost matrix in Figure 2.2 is calculated as,

$$\eta_i(j) = \begin{cases} 1 - \bar{P}_S^{(\xi)}(\ell_i), & 1 \leq i \leq R, \quad j < 0, \\ \bar{P}_S^{(\xi)}(\ell_i) \bar{\psi}_{Z_+}^{(\xi,j)}(\ell_i), & 1 \leq i \leq R, \quad j \geq 0, \\ 1 - r_{B,+}(\ell_i), & R+1 \leq i \leq P, \quad j < 0, \\ r_{B,+}(\ell_i) \bar{\psi}_{Z_+}^{(\xi,j)}(\ell_i), & R+1 \leq i \leq P, \quad j \geq 0. \end{cases} \quad (2.91)$$

where, $Z_+ = \{z_{1:M}\}$, $I = \{\ell_{1:R}\}$, and $\mathbb{B}_+ = \{\ell_{R+1:P}\}$. Note that $j = 0$ indicates that ℓ_i was misdetection and $j = -1$ indicates that ℓ_i no longer exists.

Ranked assignment algorithms have a high computational cost in generating a sequence of positive 1-1 vectors ordered according to their weights, which is not necessary for the GLMB approximation. Exploiting the Gibbs sampler allows this problem to be broken down into simple, low-dimensional problems to achieve greater efficiency. The Gibbs sampler first introduced by the Geman brothers [91] is a more efficient special case of the Metropolis-Hasting MCMC algorithm. It is useful in generating a Markov chain where the state of the chain is correlated with adjacent ones

$$\begin{array}{c}
 \text{survived and detected} \\
 \left. \begin{array}{c} z_1 \quad \dots \quad z_j \quad \dots \quad z_M \\ \dots \quad \dots \quad \dots \quad \dots \quad \dots \\ \dots \quad \dots \quad \dots \quad \dots \quad \dots \\ \dots \quad \dots \quad \dots \quad \dots \quad \dots \end{array} \right\} \\
 \text{survived and misdetected} \\
 \left. \begin{array}{c} s_1 \quad \dots \quad s_P \\ \dots \quad \dots \quad \dots \quad \dots \quad \dots \\ \dots \quad \dots \quad \dots \quad \dots \quad \dots \\ \dots \quad \dots \quad \dots \quad \dots \quad \dots \end{array} \right\} \\
 \text{died or not born} \\
 \left. \begin{array}{c} \emptyset_1 \quad \dots \quad \emptyset_P \\ \dots \quad \dots \quad \dots \quad \dots \quad \dots \\ \dots \quad \dots \quad \dots \quad \dots \quad \dots \\ \dots \quad \dots \quad \dots \quad \dots \quad \dots \end{array} \right\} \\
 \emptyset_P
 \end{array}
 \begin{bmatrix}
 -\ln \eta_1(1) & \dots & -\ln \eta_1(M) \\
 \vdots & \ddots & \vdots \\
 -\ln \eta_i(j) & \dots & \dots \\
 \vdots & \ddots & \vdots \\
 -\ln \eta_P(1) & \dots & -\ln \eta_P(M) \\
 \vdots & \vdots & \vdots \\
 -\ln \eta_1(-1) & \dots & \dots \\
 \vdots & \vdots & \vdots \\
 \vdots & \vdots & \vdots \\
 \vdots & \vdots & \vdots \\
 -\ln \eta_P(-1) & \dots & \dots
 \end{bmatrix}$$

Figure 2.2: The cost matrix for the joint prediction and update optimal assignment problem of hypothesis (I, ξ) . The 3 partitions correspond to survived and detected objects, survived and misdetected objects, and objects that died or were not born. [86]

[91, 92].

In using the Gibbs algorithm for the multi-object density approximation, we first introduce a new notation $\gamma \in \Gamma$ which is a positive 1-1 P length vector such that $\gamma = [\gamma_1, \dots, \gamma_P] \in \{-1 : M\}^P$ where $M = |Z_+|$. Hence γ can be viewed as a realization of a random variable distributed according to a probability distribution π on $\{-1 : M\}$ and Γ is the set of positive 1-1 vectors in $\{-1 : M\}^P$.

In applying the Gibbs sampler to the GLMB filter label-to-measurement mapping generation, the relationship between a pair $(I_+, \theta_+) \in \mathcal{F}(\mathbb{L}_+) \times \Theta_+(I_+)$ and γ is given by,

$$\gamma_i = \begin{cases} \theta_+(\ell_i), & \text{if } \ell_i \in I_+ \\ -1, & \text{otherwise} \end{cases}. \quad (2.92)$$

Due to the positive 1-1 property of θ_+ , for each $i, i' \in \{1 : P\}$, $i \neq i' \implies \gamma_i \neq \gamma_{i'} > 0$.

Therefore, if the positive 1-1 P tuple γ is given, information (I_+, θ_+) regarding the associated label-to-measurement can be recovered by,

$$I_+ = \{\ell_i \in I \cup \mathbb{B}_+ : \gamma_i \geq 0\}, \quad (2.93)$$

$$\theta_+(\ell_i) = \gamma_i. \quad (2.94)$$

The usage of the Gibbs sampler in δ -GLMB filter implementation is described below. For each parent hypothesis (I, ξ) the number of child hypotheses to be generated $T^{(I, \xi)}$ is determined. A reasonable method is to allocate $T^{(I, \xi)}$ proportionate to the parent hypothesis weight. The task of the Gibbs sampler is to output $\{\gamma^{(t)}\}_{t=1:T^{(I, \xi)}}$ with significant weights when supplied with the initial tuple $\gamma^{(1)}$ and matrix $\eta = [\eta_i(j)]$ illustrated in Figure 2.2. The density π in the Gibbs sampler transition kernel where the γ tuples are sampled is given by,

$$\pi(\gamma) \propto 1_\Gamma(\gamma) \prod_{i=1}^P \eta_i(\gamma_i) \quad (2.95)$$

Recall that $\eta_i(j)$ given in Eq. (2.91) holds the cost of assigning label i to measurement j .

Inside the GLMB filter Gibbs sampler, the generation of γ tuples is performed as

follows. Element n of the t^{th} tuple $\gamma^{(t)}$ is sampled from the proposal distribution conditioned on the $1 : n-1$ elements sampled so far for that sample (i.e., $\gamma_{1:n-1}^{(t)}$) and $n+1 : P$ elements (i.e., $\gamma_{n+1:P}^{(t-1)}$) from previous sample number $t-1$.

$$\gamma_n^{(t)} \sim \pi_n \left(\cdot \mid \gamma_{1:n-1}^{(t)}, \gamma_{n+1:P}^{(t-1)} \right) \quad (2.96)$$

$$\therefore \pi(\gamma' \mid \gamma) = \prod_{i=1}^P \pi_n \left(\gamma'_i \mid \gamma'_{1:n-1}, \gamma_{n+1:P} \right). \quad (2.97)$$

Let $\bar{n} = \{1 : P\} - \{n\}$ and $\gamma_{\bar{n}} = (\gamma_{1:n-1}, \gamma_{n+1:P})$. In [87] it states that,

$$\pi_n(\gamma_n \mid \gamma_{\bar{n}}) \propto \eta_n(\gamma_n) \prod_{i \in \bar{n}} \left(1 - 1_{\{1:M\}}(\gamma_n) \delta_{\gamma_n}[\gamma_i] \right). \quad (2.98)$$

giving,

$$\pi_n(j \mid \gamma_{\bar{n}}) \propto \begin{cases} \eta_n(j) & , j \in \{-1, 0\} \\ \eta_n(j) \left(1 - 1_{\{\gamma_{1:n-1}, \gamma_{n+1:P}\}}(j) \right) & , j \in \{1 : M\} \end{cases}. \quad (2.99)$$

This result (used in algorithm 1a) allows for easy computation of the conditionals $\pi_n(\cdot \mid \cdot)$.

The pseudocode for the Gibbs sampler (re-produced from [87]) is given in algorithm 2.1 and algorithm 2.1a.

Algorithm 2.1: Gibbs

Input : $\gamma^{(1)}, T, \eta = [\eta_i(j)]$
Output : $\gamma^{(1)}, \dots, \gamma^{(T)}$

- 1 $P = \text{size}(\eta, 1)$;
- 2 $M = \text{size}(\eta, 2) - 2$;
- 3 **for** $t = 2 : T$ **do**
- 4 **for** $n = 1 : P$ **do**
- 5 $\gamma_n^{(t)} \sim \pi_n(\cdot \mid \gamma_{1:n-1}^{(t)}, \gamma_{n+1:P}^{(t-1)})$;
- 6 **end**
- 7 $\gamma^{(t)} = [\gamma_1^{(t)}, \dots, \gamma_P^{(t)}]$;
- 8 **end**

Both the Murty's based and Gibbs sampler based implementations are highly parallelizable. However, the Gibbs sampler based solution is more attractive since it has a complexity that is linear in the number of measurements and quadratic in the number of hypothesized objects while the Murty's based implementation has a complexity that

Algorithm 2.1a: $\gamma_n^{(t)} \sim \pi_n \left(\cdot \mid \gamma_{1:n-1}^{(t)}, \gamma_{n+1:P}^{(t-1)} \right)$

- 1 $C = [-1 : M];$
- 2 $\eta_n = [\eta_n(-1), \dots, \eta_n(M)];$
- 3 **for** $j = 1 : M$ **do**
- 4 $\eta_n(j) = \eta_n(j) \left(1 - 1_{\{\gamma_{1:n-1}^{(t)}, \gamma_{n+1:P}^{(t-1)}\}}(j) \right);$
- 5 **end**
- 6 $\gamma_n^{(t)} \sim \text{Categorical}(c, \eta_n)$

is quartic in the number of measurements.

An efficient approximation to the GLMB filter named the labeled multi-Bernoulli (LMB) filter is presented in [93]. The LMB filter seeks to approximate the true posterior density such that the first moment of the approximated version matches the first moment of the true posterior. Compared to the CBMeMber filter which exhibits a cardinality bias, the LMB approximation is more accurate with no bias and also produces labels for the object estimates. It operates via propagating an LMB RFS. Even though the LMB family is closed under prediction, it is not closed under the update operation. Therefore, in order to facilitate a recursion, the filter carries out a three-step LMB-to-GLMB-to-LMB conversion. Initially, clusters containing closely spaced objects and their associated measurements are formed using standard gating. Next, the LMB density corresponding to each cluster is expanded to GLMB form. Secondly, the standard GLMB update is performed on the GLMB density of each cluster. Finally, the updated GLMB density is converted back to LMB form. Note that since the three-step update for each cluster can be carried out independently, this allows for parallel execution. Further details regarding the LMB recursion and the LMB-to-GLMB-to-LMB conversion can be found in [93].

TRACKING MANEUVERING OBJECTS USING JUMP MARKOV SYSTEMS

In this chapter, a novel labeled RFS algorithm that uses jump Markov systems to address the specific problem of tracking maneuvering objects is presented. The δ -GLMB filter from the RFS paradigm [13, 69] is adopted as the base for the proposed algorithm since it outputs tracks, is provably Bayes optimal [14] and admits efficient implementation [87]. Further, the GLMB is a versatile model that offers good trade-offs between tractability and fidelity [94]. The proposed algorithm is verified via numerical examples. The results of this chapter have appeared in the author's conference papers [31, 32] and journal article [33].

3.1 Introduction

The Bayes optimal approach to the multi-object tracking problem is the Bayes multi-object filter that recursively propagates the multi-object posterior density forward in time incorporating both the uncertainty in the number of objects as well as their states. As described in [Chapter 2](#), under the standard multi-object system model which takes into account object births, deaths, survivals, object detections, missed detections and clutter, the multi-object posterior densities at each time could be represented as GLMB densities. Hence the GLMB filter is an analytic solution to the multi-object Bayes filter.

Jump Markov systems (JMS) are prevalently used in a variety of areas spanning from control systems, communication networks to economics and is applicable to tracking. The JMS approach for tracking, also termed the multiple models approach has proven to be an effective tool for single maneuvering object tracking. In this approach,

the object can switch between a set of models. A JMS provides a mechanism to model such a behavior in the system in a Markovian fashion.

A JMS consists of a parameterized state space model, whose parameters evolve with time according to a finite state Markov chain. An example of a maneuvering object scenario which can be successfully represented using a JMS model is the dynamics of an aircraft, which can fly with a nearly constant velocity motion, accelerated/decelerated motion, and coordinated turn. In a JMS formulation for such a system an object that is moving according to a certain motion model at any time step is assumed to follow the same motion model with a certain probability or switch to a different motion model (out of a pre-selected set of motion models) with a certain probability in the next time step.

A Markovian transition probability matrix describes the probabilities with which a particular object changes/retains the motion model in the next time step given the motion model at the current time step. Let $\vartheta(m_+|m)$ denote the probability of switching to motion model m_+ from m as given by this Markovian transition matrix. In the Markovian transition matrix, the sum of the conditional probabilities of all possible motion models in the next time step given the current model adds up to 1, i.e.,

$$\sum_{m_+ \in \mathbb{M}} \vartheta(m_+|m) = 1. \quad (3.1)$$

where \mathbb{M} is the discrete set of motion models in the system.

A PHD filter based on JMS for maneuvering target tracking was derived in [95, 96] with a Gaussian mixture implementation and particle implementation. More work on PHD and CPHD filters proposed for JMS can be found in [97, 98]. A discussion on these works can be found in [99]. Recently, multi-Bernoulli and labeled multi-Bernoulli filters were also derived for JMS in [100–102]. These filters, however, are only approximate solutions to the Bayes multi-object filter for maneuvering objects, and at present there is no exact solution in the literature. This work aims to address that shortfall. In [Section 3.2](#), we extend the standard GLMB filter with separate prediction and update (detailed in [Section 2.5.2](#)) to operate in a JMS setting by deriving the equations for the predicted GLMB density under a multiple motion model transition that can be imple-

mented via Gaussian mixture or sequential Monte Carlo methods. In [Section 3.3](#) of this chapter, we extend the GLMB filter with the joint prediction and update (detailed in [Section 2.5.2](#)) to operate in a JMS setting. In addition to being an analytic solution, the proposed solutions output object trajectories, whereas the PHD and CBMemM-Ber filters do not. Implementation issues are discussed in [Section 3.4](#). The proposed algorithms are verified via two numerical examples and comparison on tracking performance and computational efficiency are presented in [Section 3.5](#).

Remark. The notations previously defined in [Chapter 2](#) will apply throughout this chapter, and the convention of suppressing the time index k is will also be continued.

3.2 GLMB filter with separate prediction and update for a JMS

The Bayes multi-object prediction is given by equation [Eq. \(3.2\)](#) which propagates the multi-target density $\pi(\cdot|Z)$ at the current time step to density $\pi_+(\cdot|Z_+)$ at the next time step.

$$\pi_+(\mathbf{X}_+|Z) = \int f_+(\mathbf{X}_+|\mathbf{X})\pi(\mathbf{X}|Z)\delta\mathbf{X}. \quad (3.2)$$

In the JMS model discussed in this chapter, the posterior is predicted forward under the different modes (motion models) according to the mode switching probabilities. Since we are interested in inferring the mode for each object at each time step in addition to the kinematics, the mode is included in the object state and a modification is made to the notation to accommodate that. Let the mode augmented unlabeled object state be denoted by $x = (\zeta, m) \in \mathbb{X} \times \mathbb{M}$ where ζ denotes the object kinematics and m denotes the mode. Therefore the model augmented labeled object state be denoted by $\mathbf{x} = (x, \ell) = (\zeta, m, \ell)$ where ℓ denotes the label.

By augmenting the object state with the mode, the mode augmented state space model can be expressed with the following transition density and measurement likelihood functions.

$$f_+(\zeta_+, m_+|\zeta, m, \ell) \triangleq f_+^{(m)}(\zeta_+|\zeta, \ell)\vartheta(m_+|m) \quad (3.3)$$

$$g_+(z|\zeta, m, \ell) \triangleq g_+^{(m)}(z|\zeta, \ell) \quad (3.4)$$

where $f_+^{(m)}(\cdot|\cdot, \cdot)$ is the kinematic state transition density under mode m and $g_+^{(m)}(\cdot|\cdot, \cdot)$ is the measurement likelihood function under mode m .

Additionally, to emphasize the dependence on the mode, the probability of survival, probability of birth and probability of detection parameters are denoted as,

$$p_{B,+}^{(m_+)}(\zeta_+, \ell_+) \triangleq p_{B,+}(\zeta_+, m_+, \ell_+), \quad (3.5)$$

$$P_S^{(m)}(\zeta, \ell) \triangleq P_S(\zeta, m, \ell), \quad (3.6)$$

$$P_D^{(m)}(\zeta, \ell) \triangleq P_D(\zeta, m, \ell). \quad (3.7)$$

As described in [Section 2.5.1](#), in a multi-object system with state space \mathbb{X} and discrete label space \mathbb{L} , the multi-target filtering density at the current time can be represented as a δ -GLMB of the form:

$$\pi(\mathbf{X}) = \Delta(\mathbf{X}) \sum_{(I, \xi) \in \mathcal{F}(\mathbb{L}) \times \Xi} \omega^{(I, \xi)} \delta_I(\mathcal{L}(\mathbf{X})) \left[p^{(\xi)} \right]^{\mathbf{X}}, \quad (3.8)$$

where each hypothesis in the multi-object density is represented by a pair (label set, association map history) (I, ξ) from space $\mathcal{F}(\mathbb{L}) \times \Xi$. The weight of the hypothesis represented by (I, ξ) is given by $\omega^{(I, \xi)}$ and the kinematic state density of its tracks is represented by $p^{(\xi)}(\cdot)$.

It is assumed that object births follow a labeled multi-Bernoulli birth model parameterised by $\left\{ \left(r_{B,+}^{(\ell)}, p_{B,+}^{(\ell)}(\cdot) \right) \right\}_{\ell \in \mathbb{B}_+}$.

Proposition 5. *In a multi-object system where each object is moving according to a JMS, if the posterior at the current time is a δ -GLMB of the form [Eq. \(3.8\)](#), then the predicted density at the following time is a δ -GLMB of the form,*

$$\pi_+(\mathbf{X}_+) = \Delta(\mathbf{X}_+) \sum_{(I_+, \xi) \in \mathcal{F}(\mathbb{L}_+) \times \Xi} \omega_+^{(I_+, \xi)} \delta_{I_+}(\mathcal{L}(\mathbf{X}_+)) \left[p_+^{(\xi)} \right]^{\mathbf{X}_+}, \quad (3.9)$$

where,

$$\begin{aligned} \omega_+^{(I_+, \xi)} &= \omega_S^{(I, \xi)}(I_+ \cap \mathbb{L}) \omega_B(I_+ \cap \mathbb{B}_+), \\ \omega_S^{(I, \xi)}(L) &= [\bar{P}_s^{(\xi)}]^L \sum_{I \supseteq L} 1_I(L) [1 - \bar{P}_s^{(\xi)}]^{I-L} \omega^{(I, \xi)}, \end{aligned}$$

$$\begin{aligned}
\omega_B(B) &= \prod_{i \in \mathbb{B}_+} (1 - r_{B,+}^{(i)}) \prod_{\ell \in B} \frac{1_{\mathbb{B}_+}(\ell) r_{B,+}^{(\ell)}}{1 - r_{B,+}^{(\ell)}}, \\
p_+^{(\xi)}(\zeta_+, m_+, \ell_+) &= 1_{\mathbb{L}}(\ell_+) p_S^{(\xi)}(\zeta_+, m_+, \ell_+) + 1_{\mathbb{B}_+}(\ell_+) p_{B,+}(\zeta_+, m_+, \ell_+), \\
p_S^{(\xi)}(\zeta_+, m_+, \ell_+) &= \frac{\sum_{m \in \mathbb{M}} \left\{ P_S^{(m)}(\cdot, \ell_+) f_+^{(m)}(\zeta_+ | \cdot, \ell_+) \times \vartheta(m_+ | m), p^{(\xi)}(\cdot, m, \ell_+) \right\}}{\bar{P}_S^{(\xi)}(\ell_+)}, \\
\bar{P}_S^{(\xi)}(\ell_+) &= \sum_{m_+ \in \mathbb{M}} \bar{P}_S^{(\xi)}(\ell_+, m_+), \\
\bar{P}_S^{(\xi)}(\ell_+, m_+) &= \sum_{m \in \mathbb{M}} \left\{ P_S^{(m)}(\cdot, \ell) \times \vartheta(m_+ | m), p^{(\xi)}(\cdot, m, \ell) \right\}, \\
p_{B,+}(\zeta_+, m_+, \ell_+) &= p_{B,+}^{(\ell_+)}(\zeta_+, m_+).
\end{aligned}$$

Proposition 6. *In a multi-object system where each object is moving according to a JMS, if the predicted density is a δ -GLMB of the form Eq. (3.9), then upon receiving measurement set Z_+ the posterior density is a δ -GLMB of the form,*

$$\pi_{Z_+}(\mathbf{X}_+) = \Delta(\mathbf{X}_+) \sum_{(I_+, \xi) \in \mathcal{F}(\mathbb{L}_+) \times \Xi} \sum_{\theta_+ \in \Theta_+} \omega^{(I_+, \xi, \theta_+)}(Z_+) \delta_{I_+}(\mathcal{L}(\mathbf{X}_+)) \left[p^{(\xi, \theta_+)}(\cdot | Z_+) \right]^{\mathbf{X}_+}, \quad (3.10)$$

where,

Θ_+ is the space of mappings $\theta_+ : \mathbb{L}_+ \rightarrow \{0 : |Z_+|\}$ such that $\theta_+(i) = \theta_+(i') > 0$ implies $i = i'$,

$$\begin{aligned}
\omega^{(I_+, \xi, \theta_+)}(Z_+) &= \frac{\delta_{\theta_+^{-1}\{0:|Z_+|\}}(I_+) \omega_+^{(I_+, \xi)} \left[\bar{\psi}_{Z_+}^{(\xi, \theta_+)} \right]^{I_+}}{\sum_{(I_+, \xi) \in \mathcal{F}(\mathbb{L}_+) \times \Xi} \sum_{\theta_+ \in \Theta_+} \delta_{\theta_+^{-1}\{0:|Z_+|\}}(I_+) \omega^{(I_+, \xi)} \left[\bar{\psi}_{Z_+}^{(\xi, \theta_+)} \right]^{I_+}}, \\
p^{(\xi, \theta_+)}(\zeta_+, m_+, \ell_+ | Z_+) &= \frac{p_+^{(\xi)}(\zeta_+, m_+, \ell_+) \psi_{Z_+}^{\theta_+(\ell_+)}(\zeta_+, m_+, \ell_+)}{\bar{\psi}_{Z_+}^{(\xi, \theta_+)}(\ell_+)}, \\
\bar{\psi}_{Z_+}^{(\xi, \theta_+)}(\ell_+) &= \sum_{m_+ \in \mathbb{M}} \bar{\psi}_{Z_+}^{(\xi, \theta_+)}(m_+, \ell_+), \\
\bar{\psi}_{Z_+}^{(\xi, \theta_+)}(m_+, \ell_+) &= \left\{ p_+^{(\xi)}(\cdot, m_+, \ell_+) \psi_{Z_+}^{\theta_+(\ell_+)}(\cdot, m_+, \ell_+) \right\}, \\
\psi_{Z_+}^j(\zeta_+, m_+, \ell_+) &= \delta_0(j) \left[1 - P_{D,+}^{(m_+)}(\zeta_+, \ell_+) \right] + [1 - \delta_0(j)] \frac{\left[P_{D,+}^{(m_+)}(\zeta_+, \ell_+) \right] g_+^{(m_+)}(z_j | \zeta_+, \ell_+)}{\kappa_+(z_j)},
\end{aligned}$$

$\kappa(\cdot)$ = intensity function for Poisson clutter.

Extracting State Estimates

The multi-object state extraction is performed akin to the single model system, by first determining the MAP cardinality estimate from the cardinality distribution and extracting the hypothesis of the highest weight with the MAP cardinality. The mean of the posterior density for each label in the selected hypothesis is computed to extract the kinematics. To estimate the mode for each label, the mode that maximizes the marginal probability of that mode over the entire density for that label is selected. i.e., for label ℓ of hypothesis (I, ξ) , the estimated motion model \hat{m} is selected as,

$$\hat{m} = \operatorname{argmax}_m \int p^{(\xi)}(\zeta, m, \ell) d\zeta. \quad (3.11)$$

Anlytic Solution

Consider the special case where the object birth model, dynamic models and observation models in the JMS are all linear with Gaussian noise. i.e.,

$$p_{B,+}^{(\ell_+)}(\zeta_+, m_+) = \mathcal{N}\left(\zeta_+; b_+^{(\ell_+)}, Q_{b,+}^{(\ell_+)}\right) \times \vartheta_b^{(\ell_+)}(m_+), \quad (3.12)$$

$$f_+^{(m)}(\zeta_+|\zeta, \ell_+) = \mathcal{N}\left(\zeta_+; F^{(m)}\zeta, Q_f^{(m)}\right), \quad (3.13)$$

$$g_+^{(m_+)}(z|\zeta_+, m_+, \ell_+) = \mathcal{N}\left(z; H^{(m_+)}\zeta_+, Q_h^{(m_+)}\right), \quad (3.14)$$

where,

$b_+^{(\ell_+)}$ = mean of the Gaussian birth density of birth component ℓ_+ ,

$Q_{b,+}^{(\ell_+)}$ = covariance of the Gaussian birth density of birth component ℓ_+ ,

$\vartheta_b^{(\ell_+)}(m_+)$ = probability of an object birth from component ℓ_+ having initial mode m_+ ,

$F^{(m)}$ = state transition matrix for mode m ,

$Q_f^{(m)}$ = process noise covariance matrix for mode m ,

$H^{(m_+)}$ = observation matrix for mode m_+ ,

$Q_h^{(m_+)}$ = measurement noise covariance matrix for mode m_+ .

Note: $\sum_{m \in \mathbb{M}} \vartheta_b^{(\ell)}(m) = 1$.

Using the following identities,

$$\int \mathcal{N}(\zeta; \bar{\zeta}, P) \mathcal{N}(\zeta_+; F^{(m)}\zeta, Q_f^{(m+)}) d\zeta = \mathcal{N}\left(\zeta_+; F^{(m)}\bar{\zeta}, F^{(m)}P[F^{(m)}]^T + Q_f^{(m)}\right), \quad (3.15)$$

$$\mathcal{N}(\zeta; \bar{\zeta}, P) \mathcal{N}(z; H^{(m+)}\zeta, Q_h^{(m+)}) = q(z) \mathcal{N}\left(\zeta; \bar{\zeta} + K(z - H^{(m+)}\bar{\zeta}), [I - KH^{(m+)}]P\right), \quad (3.16)$$

$$q(z) = \mathcal{N}\left(z; H^{(m+)}\bar{\zeta}, H^{(m+)}P[H^{(m+)}]^T + Q_h^{(m+)}\right), \quad (3.17)$$

$$K = P[H^{(m+)}]^T \left[H^{(m+)}P[H^{(m+)}]^T + Q_h^{(m+)} \right]^{-1}, \quad (3.18)$$

in Eq. (3.9), Eq. (3.10) we obtain the analytic solution.

3.3 GLMB filter with joint prediction and update for a JMS

The recursion described in the previous section is implemented via separate truncations of the multi-object density during both the prediction and update stages. This is carried out by eliminating hypotheses with insignificant weights and allowing existing hypotheses to generate child hypotheses in the next stage proportionate to their weights. In the prediction step, the generation of best predicted hypotheses is carried out using two independent K-shortest path algorithms. In the update step, generation of best predicted hypotheses is performed by solving a ranked assignment problem for each predicted GLMB hypothesis. As a result of the disjoint nature of these truncations, a significant number of the parent hypotheses with high weights that are allocated a large quota for generating child hypotheses, generate a significant number of child hypotheses with insignificant weights. Hence considerable computational resources are wasted in solving ranked assignment problems which have cubic complexity in the number of measurements. In [87] a more efficient design which consists of a combined prediction and update step and resultantly a single density truncation is presented. Further details and explicit formulas regarding this implementation was given in [Section 2.5.2](#).

Substituting the JMS state equations [Eq. \(3.3\)](#), [Eq. \(3.4\)](#) and the parameters given

in Eq. (3.5), Eq. (3.6), Eq. (3.7) in the GLMB filter equation for the joint prediction and update given by Eq. (2.90) produces the JMS-GLMB recursion with a joint prediction and update step given in proposition 7.

Proposition 7. *If the filtering density at current time is a δ -GLMB density of the form Eq. (3.8), then the filtering density at next time is a δ -GLMB given by,*

$$\pi_{Z_+}(\mathbf{X}_+) = \Delta(\mathbf{X}_+) \sum_{I, \xi, I_+, \theta_+} \omega^{(I, \xi)} \omega_{Z_+}^{(I, \xi, I_+, \theta_+)} \delta_{I_+}[\mathcal{L}(\mathbf{X}_+)] \left[p_{Z_+}^{(\xi, \theta_+)} \right]^{\mathbf{X}_+} \quad (3.19)$$

where

$$\begin{aligned} I &\in \mathcal{F}(\mathbb{L}), \quad \xi \in \Xi, \quad I_+ \in \mathcal{F}(\mathbb{L}_+), \quad \theta_+ \in \Theta_+(I_+), \\ \omega_{Z_+}^{(I, \xi, I_+, \theta_+)} &= 1_{\Theta_+(I_+)}(\theta_+) \left[1 - \bar{P}_S^{(\xi)} \right]^{I-I_+} \left[\bar{P}_S^{(\xi)} \right]^{I \cap I_+} \left[1 - r_B \right]^{\mathbb{B}-I_+} \left[r_B \right]^{\mathbb{B} \cap I_+} \left[\bar{\psi}_{Z_+}^{(\xi, \theta_+)} \right]^{I_+}, \\ \bar{P}_S^{(\xi)}(\ell_+) &= \sum_{m_+ \in \mathbb{M}} \bar{P}_S^{(\xi)}(\ell_+, m_+), \\ \bar{P}_S^{(\xi)}(\ell_+, m_+) &= \sum_{m \in \mathbb{M}} \left\langle P_S^{(m)}(\cdot, \ell) \vartheta(m_+ | m), p^{(\xi)}(\cdot, m, \ell) \right\rangle, \\ \bar{\psi}_{Z_+}^{(\xi, \theta_+)}(\ell_+) &= \sum_{m_+ \in \mathbb{M}} \bar{\psi}_{Z_+}^{(\xi, \theta_+)}(m_+, \ell_+), \\ \bar{\psi}_{Z_+}^{(\xi, \theta_+)}(m_+, \ell_+) &= \left\langle \bar{p}_+^{(\xi)}(\cdot, m_+, \ell_+), \psi_{Z_+}^{\theta_+(\ell_+)}(\cdot, m_+, \ell_+) \right\rangle, \\ \bar{p}_+^{(\xi)}(\zeta_+, m_+, \ell_+) &= 1_{\mathbb{B}_+}(\ell_+) p_B(\zeta_+, m_+, \ell_+) + 1_{\mathbb{L}}(\ell) p_S^{(\xi)}(\zeta_+, m_+, \ell_+), \\ p_S^{(\xi)}(\zeta_+, m_+, \ell_+) &= \frac{\sum_{m \in \mathbb{M}} \left\langle P_S^{(m)}(\cdot, \ell) f_+^{(m)}(\zeta_+ | \cdot, \ell) \vartheta(m_+ | m), p^{(\xi)}(\cdot, m, \ell) \right\rangle}{\bar{P}_S^{(\xi)}(\ell_+)}, \\ p_{Z_+}^{(\xi, \theta_+)}(\zeta_+, m_+, \ell_+) &= \frac{\bar{p}_+^{(\xi)}(\zeta_+, m_+, \ell_+) \psi_{Z_+}^{\theta_+(\ell_+)}(\zeta_+, m_+, \ell_+)}{\bar{\psi}_{Z_+}^{(\xi, \theta_+)}(m_+, \ell_+)}, \\ \psi_{Z_+}^j(\zeta_+, m_+, \ell_+) &= \delta_0(j) \left[1 - P_{D,+}^{(m_+)}(\zeta_+, \ell_+) \right] + [1 - \delta_0(j)] \frac{P_{D,+}^{(m_+)}(\zeta_+, \ell_+) \times g_+^{(m_+)}(z_j | \zeta_+, \ell_+)}{\kappa_+(z_j)}. \end{aligned}$$

Analytic solution

Consider the special case where the object birth model, dynamic models and observation model are all linear with Gaussian noise for which Eq. (3.12), Eq. (3.13), Eq. (3.14) apply. For such a system if the posterior density at the current time is of the form

Eq. (3.8) then the filtering density at the next time step is the δ -GLMB of the form,

$$\pi_{Z_+}(\mathbf{X}_+) = \Delta(\mathbf{X}_+) \sum_{I, \xi, I_+, \theta_+} \omega^{(I, \xi)} \omega_{Z_+}^{(I, \xi, I_+, \theta_+)} \delta_{I_+}[\mathcal{L}(\mathbf{X}_+)] \left[p_{Z_+}^{(\xi, \theta_+)} \right]^{\mathbf{X}_+} \quad (3.20)$$

where

$$\begin{aligned} I &\in \mathcal{F}(\mathbb{L}), \xi \in \Xi, I_+ \in \mathcal{F}(\mathbb{L}_+), \theta_+ \in \Theta_+(I_+), \\ \omega_{Z_+}^{(I, \xi, I_+, \theta_+)} &= \left[1 - \bar{P}_S^{(\xi)} \right]^{I - I_+} \left[\bar{P}_S^{(\xi)} \right]^{I \cap I_+} \left[1 - r_{B, +} \right]^{\mathbb{B}_+ - I_+} \left[r_{B, +} \right]^{\mathbb{B}_+ \cap I_+} \left[\bar{\psi}_{Z_+}^{(\xi, \theta_+)} \right]^{I_+}, \\ \bar{P}_S^{(\xi)}(\ell_+) &= \sum_{m_+ \in \mathbb{M}} \bar{P}_S^{(\xi)}(\ell_+, m_+), \\ \bar{P}_S^{(\xi)}(\ell_+, m_+) &= \sum_{m \in \mathbb{M}} \left\langle p^{(\xi)}(\cdot, m_+, \ell_+) \times \vartheta(m_+ | m), P_S^{(m)}(\cdot, \ell_+) \right\rangle, \\ \bar{\psi}_{Z_+}^{(\xi, \theta_+)}(\ell_+) &= \sum_{m_+ \in \mathbb{M}} \bar{\psi}_{Z_+}^{(\xi, \theta_+)}(m_+, \ell_+), \\ \bar{\psi}_{Z_+}^{(\xi, \theta_+)}(m_+, \ell_+) &= \left\langle \bar{p}_+^{(\xi)}(\cdot, m_+, \ell_+), \psi_{Z_+}^{\theta_+(\ell_+)}(\cdot, m_+, \ell_+) \right\rangle, \\ \bar{p}_+^{(\xi)}(\zeta_+, m_+, \ell_+) &= 1_{\mathbb{B}_+}(\ell_+) p_B(\zeta_+, m_+, \ell_+) + \\ &\quad 1_{\mathbb{L}}(\ell) p_S^{(\xi)}(\zeta_+, m_+, \ell_+), \\ p_S^{(\xi)}(\zeta_+, m_+, \ell_+) &= \frac{\sum_{m \in \mathbb{M}} \left\langle P_S^{(m)}(\cdot, \ell) f_+^{(m)}(\zeta_+ | \cdot, \ell) \vartheta(m_+ | m), p^{(\xi)}(\cdot, m, \ell) \right\rangle}{\bar{P}_S^{(\xi)}(\ell_+)}, \\ p_{Z_+}^{(\xi, \theta_+)}(\zeta_+, m_+, \ell_+) &= \frac{\bar{p}_+^{(\xi)}(\zeta_+, m_+, \ell_+) \psi_{Z_+}^{\theta_+(\ell_+)}(\zeta_+, m_+, \ell_+)}{\bar{\psi}_{Z_+}^{(\xi, \theta_+)}(m_+, \ell_+)}, \\ \psi_{Z_+}^j(\zeta_+, m_+, \ell_+) &= \delta_0(j) \left[1 - P_{D, +}^{(m_+)}(\zeta_+, \ell_+) \right] + [1 - \delta_0(j)] \frac{P_{D, +}^{(m_+)}(\zeta_+, \ell_+) \times g_+^{(m_+)}(z_j | \zeta_+, \ell_+)}{\kappa_+(z_j)}, \\ f_+^{(m)}(\zeta_+ | \zeta, \ell) &= \mathcal{N} \left(\zeta_+; F^{(m)} \zeta, Q_f^{(m)} \right), \\ F^{(m)} &= \text{state transition matrix for mode } m, \\ Q_f^{(m)} &= \text{process noise covariance matrix for mode } m, \\ p_{B, +}^{(\ell_+)}(\zeta_+, m_+) &= \mathcal{N} \left(\zeta_+; b^{(\ell_+)}, Q_b^{(\ell_+)} \right) \times \vartheta_b^{(\ell_+)}(m_+), \\ b^{(\ell_+)} &= \text{mean of the Gaussian birth density of birth component } \ell_+, \\ Q_b^{(\ell_+)} &= \text{covariance of the Gaussian birth density of birth component } \ell_+, \\ \vartheta_b^{(\ell_+)}(m_+) &= \text{probability of an object birth from component } \ell_+ \text{ having initial mode } m_+, \end{aligned}$$

$$g_+^{(m_+)}(z|\zeta_+, m_+, \ell_+) = \mathcal{N}\left(z; H^{(m_+)}\zeta_+, Q_g^{(m_+)}\right),$$

$\kappa(\cdot)$ = intensity function for Poisson clutter,

$H^{(m_+)}$ = observation matrix for mode m_+ ,

$Q_g^{(m_+)}$ = measurement noise covariance for mode m_+ .

3.4 Implementation Issues

When the analytic solutions for the above described filters are considered, it is evident that the posterior density for each track is a Gaussian mixture, with each mixture component relating to one of the available modes in the system. For a particular track, at each new time step, the posterior is predicted forward taking modes in the system into consideration with adjusted weights, thereby generating a new Gaussian mixture with more components. As a result, the number of mixture components escalates exponentially. Hence extensive pruning and merging of mixture components must be carried out for each track density in each GLMB hypothesis after the update step to keep the computation manageable. Pruning is carried out by dropping the Gaussian mixture components with weights below a pre-determined threshold. Gaussian mixture components that are quite close to each other are merged into a single Gaussian. The pseudo code of the merging procedure used in this work is given in Algorithm 3.1. Additionally, hypothesis pruning is carried out at the end of each time step by dropping hypotheses below a certain threshold and retaining a (pre-determined) maximum number of hypotheses with the highest weights.

For mildly non-linear motion models and measurement models, the UKF can be utilized for predicting and updating each Gaussian component in the mixture forward. Alternatively, instead of a making use of a Gaussian mixture to represent the posterior density of each track in a hypothesis, a particle-based representation can be employed. In such case, the density is represented using a large set of particles which are propagated forward under the different modes with adjusted weights for each particle. As in the case of the Gaussian mixture, the number of particles in the density increase by a fold of the number of motion models considered during each prediction forward. Thus resampling needs to be carried out to discard particles with negligible weights to keep the total count of particles manageable.

Algorithm 3.1: Merging procedure for Gaussian mixture components

Input : $\{\omega^{(i)}, m^{(i)}, P^{(i)}\}_{i=1}^J$,
Input : T_{merge}, J_{max} .
Output : $\{\tilde{\omega}^{(i)}, \tilde{m}^{(i)}, \tilde{P}^{(i)}\}_{i=1}^J$
1 $S = \{1, \dots, J\}$;
2 for $n = 1 : J$ **do**
4 $j = \underset{i \in S}{\operatorname{argmax}} \omega^{(i)}$;
5 $S' = \left\{ i \in S \mid \frac{(m^{(i)} - m^{(j)})^T (m^{(i)} - m^{(j)})}{(P^{(i)})} < T_{merge} \right\}$;
6 $\tilde{\omega}^{(n)} = \sum_{i \in S'} \omega^{(i)}$;
7 $\tilde{m}^{(n)} = \frac{1}{\tilde{\omega}^{(n)}} \sum_{i \in S'} \omega^{(i)} m^{(i)}$;
8 $\tilde{P}^{(n)} = \frac{1}{\tilde{\omega}^{(n)}} \sum_{i \in S'} \omega^{(i)} \left[P^{(i)} + (\tilde{m}^{(n)} - m^{(i)})^T (\tilde{m}^{(n)} - m^{(i)}) \right]$;
9 $S = S \setminus S'$;
10 end
5 if $n > J_{max}$ **then**
11 replace $\{\tilde{\omega}^{(n)}, \tilde{m}^{(n)}, \tilde{P}^{(n)}\}_{i=1:n}$ by the J_{max} Gaussians with largest weights ;
12 end

3.5 Simulation Studies

In this section, using two simulated scenarios we demonstrate the performance of the proposed JMS-GLMB filter with joint prediction and update (for both cases of Murty's algorithm and Gibbs sampling) in comparison to the JMS-GLMB filter with separate prediction and update. The second scenario is simulated to be more challenging than the first scenario with a higher number of objects and a higher intensity of clutter.

Example 1

The first scenario consists of 3 objects and Poisson clutter with an average of 60 per scan. The true tracks of the three objects present in this scenario are presented in [Figure 3.1](#).

The object state vector consists of Cartesian x, y coordinates and the velocities in those directions. The sampling interval $T = 5s$. Therefore the gap between consecutive time steps in the filter is taken to be $5s$. The dynamics of the objects are modelled using three types of motion models; constant velocity, right turn (coordinated turn with a 3° angle), and left turn (coordinated turn with a -3° angle).

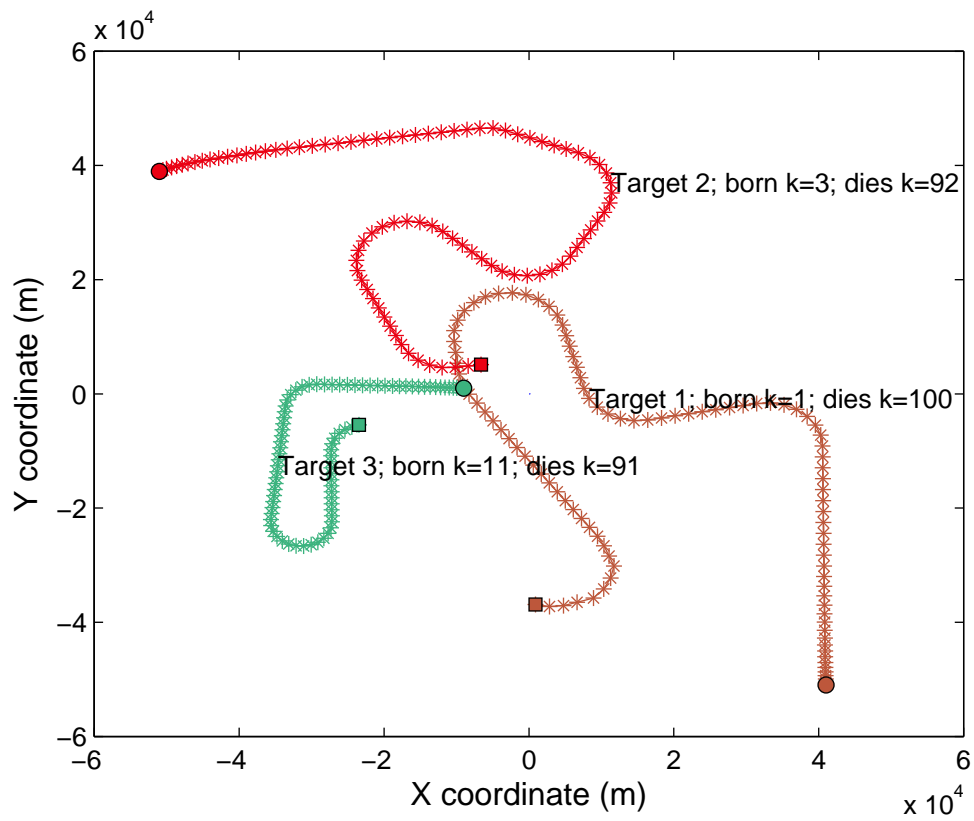


Figure 3.1: True trajectories of the tracked objects in scenario 1

The state transition matrix for the first mode is given by,

$$F^{(1)} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

and the state transition matrices for the second and third modes can be obtained by substituting $\omega = 0.05$ and $\omega = -0.05$, respectively, in

$$F^{(2,3)} = \begin{bmatrix} 1 & \sin(T\omega)/\omega & 0 & (\cos(T\omega) - 1)/\omega \\ 0 & \cos(T\omega) & 0 & -\sin(T\omega) \\ 0 & (1 - \cos(T\omega))/\omega & 1 & \sin(T\omega)/\omega \\ 0 & \sin(T\omega) & 0 & \cos(T\omega) \end{bmatrix}$$

The objects are assumed to transition between these different modes according to the mode switching probability matrix A where the probability of switching from mode m to m_+ denoted by $\vartheta(m_+|m) = [A_{m,m_+}]$.

$$A = \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.2 & 0.8 & 0 \\ 0.2 & 0 & 0.8 \end{bmatrix}$$

The process noise is distributed zero-mean Gaussian with covariance

$$Q_f = \sigma_f^2 \begin{bmatrix} T^4/4 & T^3/2 & 0 & 0 \\ T^3/2 & T^2 & 0 & 0 \\ 0 & 0 & T^4/4 & T^3/2 \\ 0 & 0 & T^3/2 & T^2 \end{bmatrix},$$

where $\sigma_f = 5$ for mode 1 and $\sigma_f = 20$ for modes 2 and 3. The probability of survival is set at 0.97. The objects are observed in a $[-60, 60] \times [-60, 60] km^2$ region and are assumed to be born from a labeled multi-Bernoulli distribution with three components each with 0.05 birth probability and birth densities $\mathcal{N}(m_1, P_L)$, $\mathcal{N}(m_2, P_L)$, $\mathcal{N}(m_3, P_L)$ where $m_2 = [-50000, 0, 40000, 0]$, $m_3 = [-10000, 0, 0, 0]$ and $P_L = \text{diag}([300, 30, 300, 30])$.

A sensor located at $(0, 0)$ extracts the target x and y coordinates. The probability of detection is set at $P_D = 0.99$. Measurement noise is assumed to be distributed zero-mean Gaussian with covariance $\sigma_h^2 I_2$ where $\sigma_h = 40m$ and I_2 is the identity matrix of dimension 2.

The Optimal Subpattern Assignment Metric (OSPA)[103] values calculated for 100 Monte Carlo (MC) runs for scenario 1 using the three different implementations are illustrated in Figure 3.2. The OSPA metric is capable of computing the multi-object miss-distance between a set of true object states and a set of estimated object states in a mathematically consistent and physically meaningful manner and accommodates cardinality differences eliminating weaknesses found in other methods [104–106].

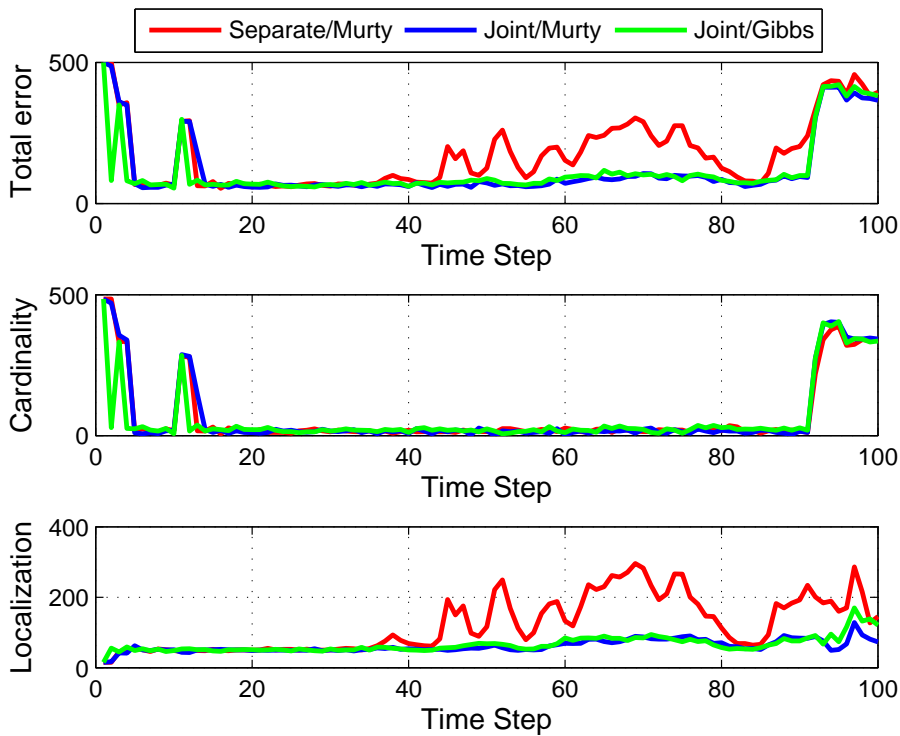


Figure 3.2: OSPA error (parameters $c = 500$, $p = 2$) reported for different implementations from 100 MC runs for scenario 1.

Even though popularly used for providing a good indication of multi-object *filtering* performance, the OSPA metric does not account for errors between the estimated and true sets of *tracks*. Therefore it does not penalize track switching and fragmentation in a consistent manner. A new metric based on the existing OSPA metric named OSPA(2) or OSPA-on-OSPA was recently proposed in [107] for evaluating

tracking performance. This is achieved by defining a base distance between two tracks as a weighted time-averaged OSPA distance between the object states of each track and substituting this base distance into the original OSPA calculation. The OSPA(2) error calculated for 100 MC runs is displayed in [Figure 3.3](#).

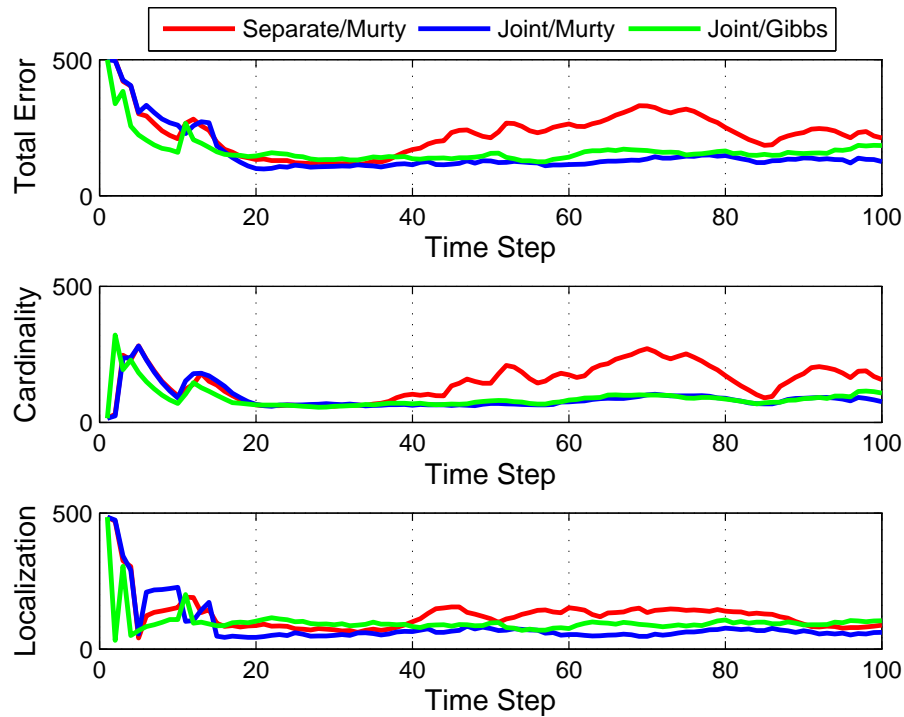


Figure 3.3: OSPA(2) error (parameters $c = 500$, $p = 2$) reported for different implementations from 100 MC runs for scenario 1.

State estimates obtained for a single run by the single stage implementation using Gibbs is shown in [Figure 3.4](#). Since the state estimates are colour coded by the object labels implicit trajectory formation by the filter can be observed.

Object state estimates for the same single run displayed in [Figure 3.4](#) albeit colour coded by the estimated mode is shown in [Figure 3.5](#). Observe how the mode change has been identified when the objects undergo the different maneuvers constant velocity, right turn and left turn.

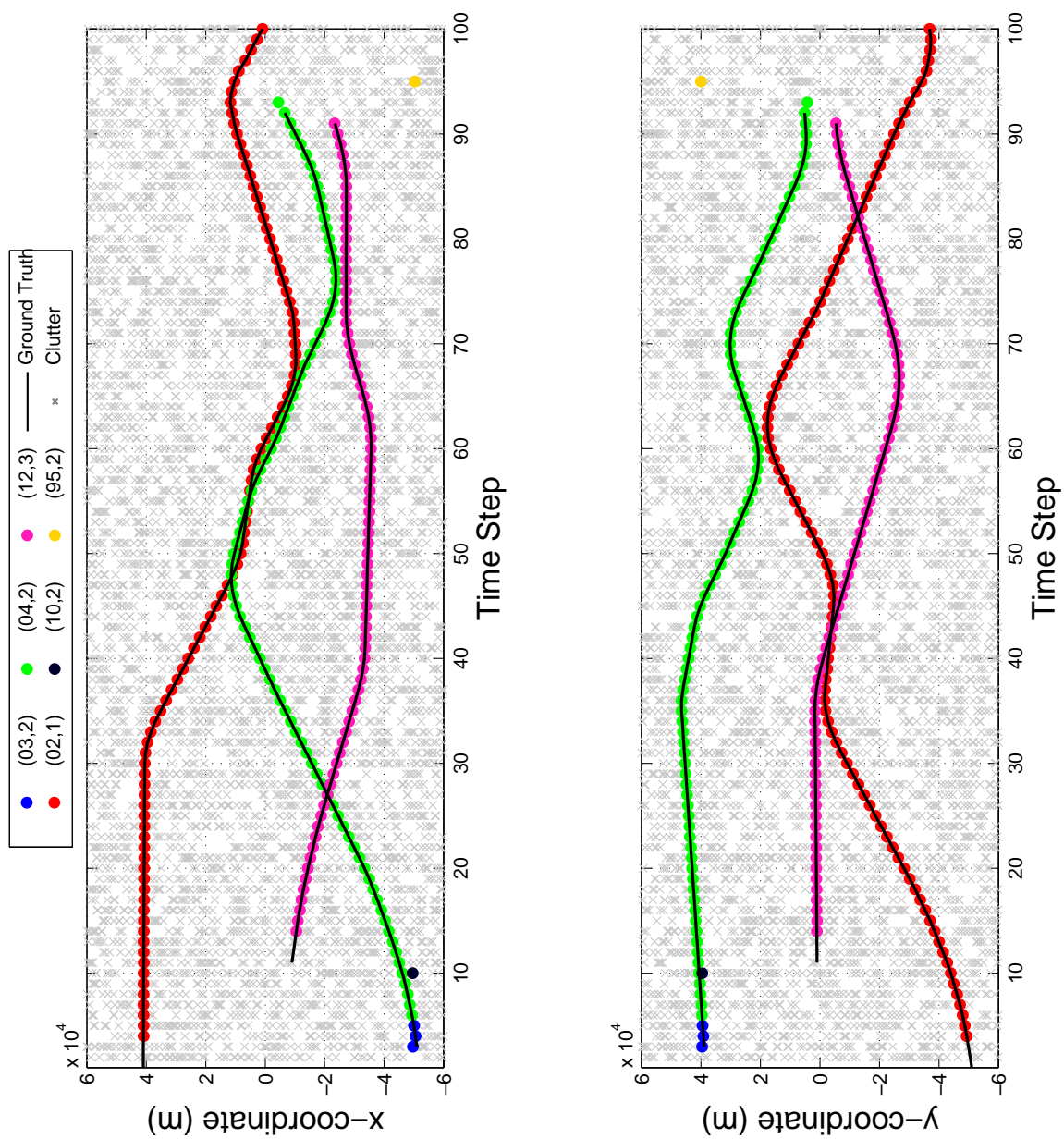


Figure 3.4: Object trajectories from a single joint/Gibbs run for scenario 1. State estimates are color coded by object labels.

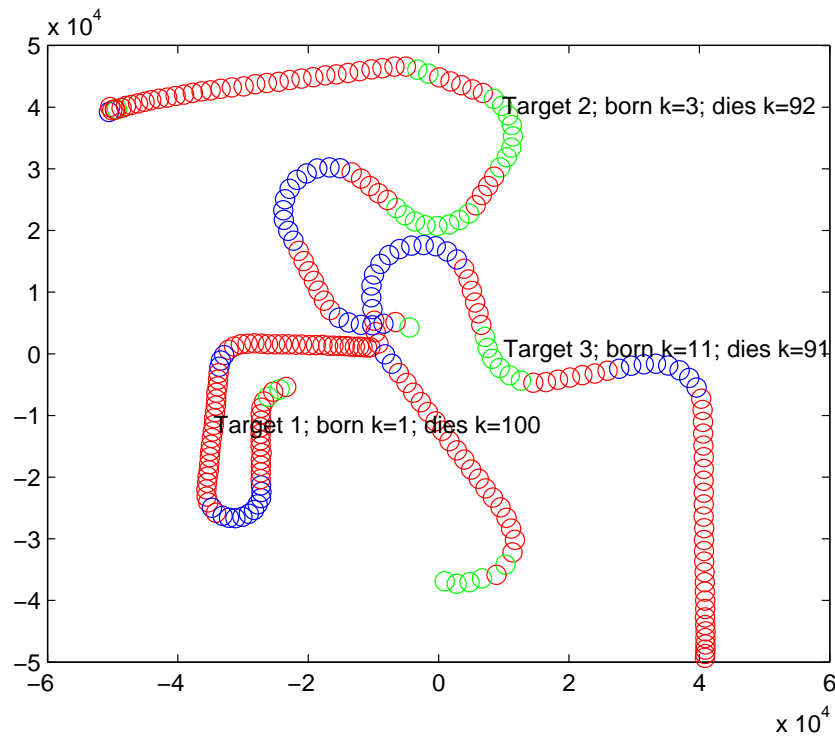


Figure 3.5: Object state estimates (color coded by estimated mode) generated by the Joint/Gibbs implementation for a single run of scenario 1.

Example 2

The second scenario consists of 8 objects observed in a $[-80, 80] \times [-80, 80] km^2$ region and Poisson clutter with an average of 120 per scan. The true tracks of the eight objects present in the scenario are shown in [Figure 3.6](#). It is modeled identically to scenario 1 in terms of state dynamics, birth model and measurement model.

The OSPA [103] values calculated for 100 MC runs for scenario 2 using the three different implementations are illustrated in [Figure 3.7](#) and the OSPA(2) statistics for the same 100 MC runs are illustrated in [Figure 3.8](#).

The x, y coordinate estimates obtained for a single run by the single stage (joint prediction and update) implementation using Gibbs for this scenario is shown in [Figure 3.9](#).

Object state estimates for the same single run displayed in [Figure 3.9](#) colour coded by the estimated mode is shown in [Figure 3.10](#).

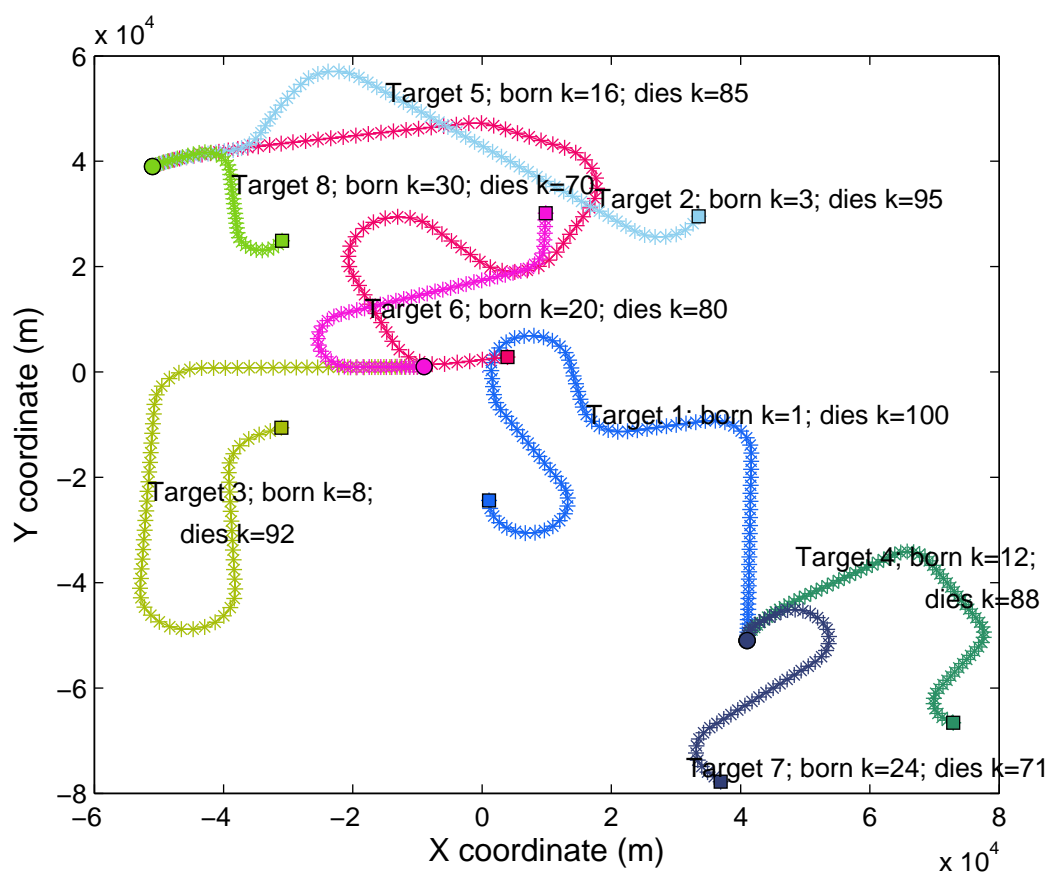


Figure 3.6: True trajectories of the tracked objects in scenario 2

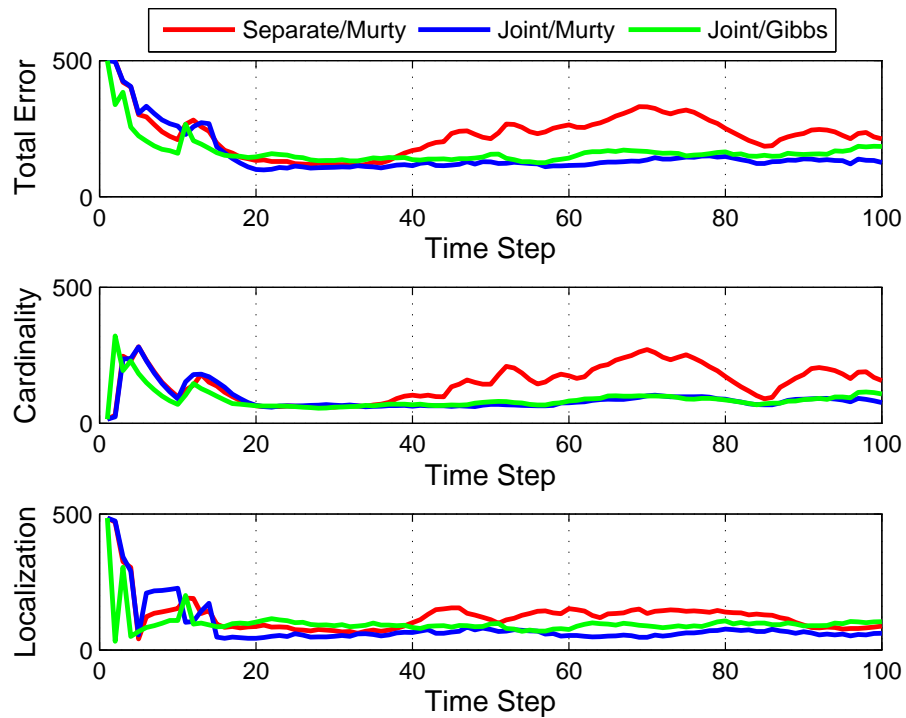


Figure 3.7: OSPA error (parameters $c = 500$, $p = 2$) reported for different implementations from 100 MC runs for scenario 2.

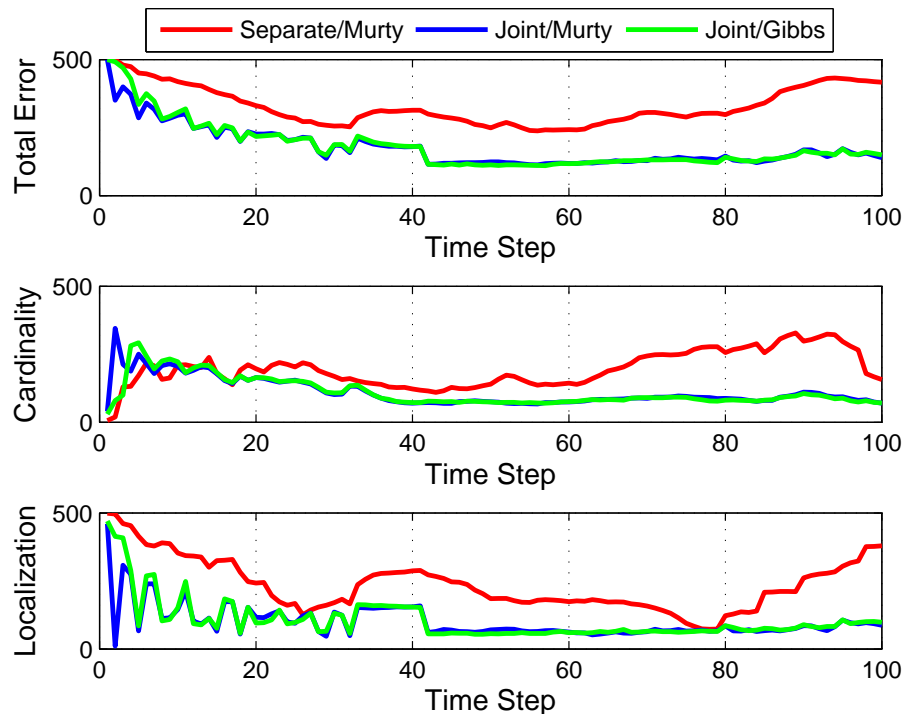


Figure 3.8: OSPA(2) error (parameters $c = 500$, $p = 2$) reported for different implementations from 100 MC runs for scenario 2.

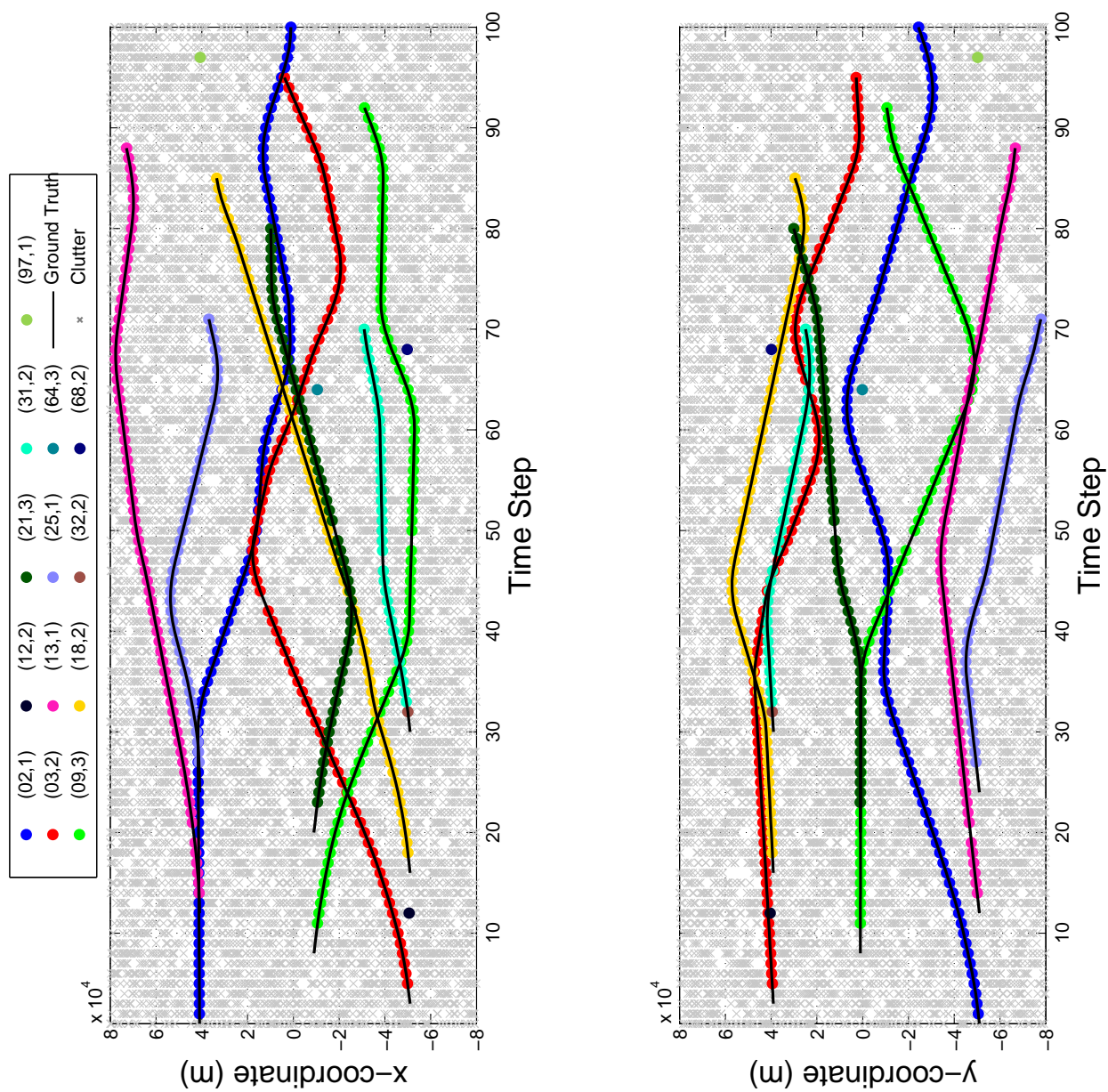


Figure 3.9: Object trajectories from a single joint/Gibbs run for scenario 2. State estimates are color coded by object labels.

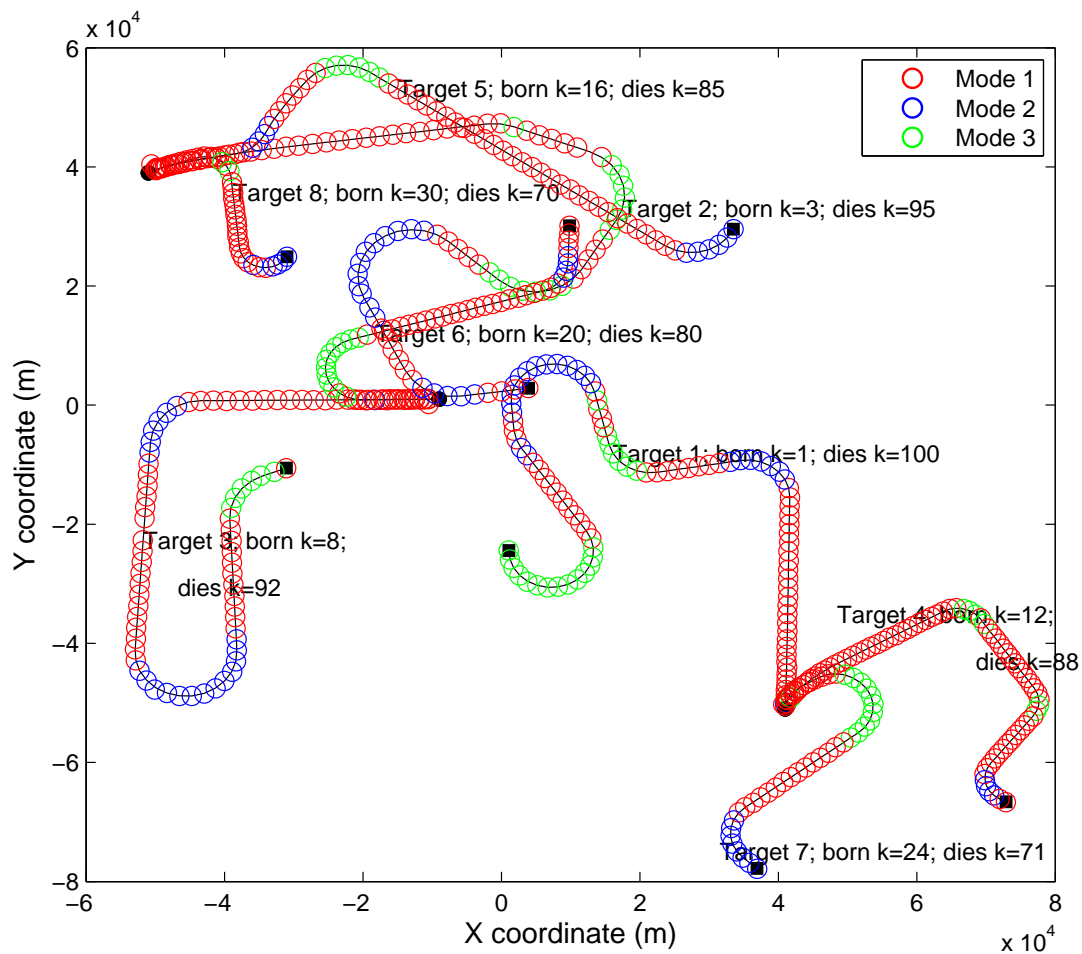


Figure 3.10: Object state estimates generated by the joint/Gibbs implementation for a single run of scenario 2

TRACKING WITH UNKNOWN CLUTTER AND DETECTION PROFILE

In this chapter, we first present how the GLMB filter for JMS proposed in [Chapter 3](#) is applied for joint tracking and classification of multiple objects. Further, we extend that multi-object tracker/classifier to one that can adaptively learn clutter rate and detection profile while tracking, provided that the detection profile and clutter background do not change too rapidly compared to the measurement-update rate. This is a reasonable assumption particularly in applications such as visual tracking where the data rate is around 30 frames per second, and the background parameters vary slowly within this period. An efficient implementation with a complexity that is linear in the number of measurements and quadratic in the number of hypothesized tracks is presented with experiments confirming markedly improved performance over existing multi-object filters for unknown background. The results of this chapter have appeared in the author's journal article [33].

4.1 Introduction

In a multi-object scenario, the number of objects and their individual states evolve in time, compounded by false detections, misdetections and measurement origin uncertainty. For example, in the KITTI-17 sequence from KITTI video datasets [108], see [Figure 4.1](#), the number of objects varies with time due to objects coming in and out of the scene, and the detector used to convert each image into point measurements (e.g. background subtraction, foreground modeling [26]), invariably misses objects in the scene as well as generates false measurements or clutter.

Knowledge of parameters for uncertainty sources such as clutter and detection profile is of critical importance in Bayesian multi-object filtering, arguably, more so than the measurement noise model. The detection profile is characterized by the probability that an object with a particular state generates an observation. Clutter, often modeled as a Poisson point process characterized by the clutter rate, are false measurements not originated from any object. While these parameters are assumed to be known in most multi-object tracking techniques, this is generally not the case in practice [13, 14, 22, 23]. Significant mismatches in clutter and detection model parameters inevitably result in erroneous estimates. For the video tracking example in [Figure 4.1](#) the clutter rate and detection profile are not known and have to be guessed before a multi-object tracker can be applied. The tracking performance of the Bayes optimal multi-object tracking filter [15, 86], for the ‘guessed’ clutter rate and ‘true’ clutter rate (that varies with time as shown in [Figure 4.2](#)), demonstrates significant performance degradation.

Except for a few applications such as radar, and where filtering is directly performed on pre-detection data [110–112] the required clutter rate and detection profile of the sensor are not available. Usually, these parameters are either estimated from training data or manually tuned. However, a major problem in many applications is the time-varying nature of the misdetection and clutter processes, see [Figure 4.2](#) for example. Consequently, there is no guarantee that the model parameters chosen from training data will be sufficient for the multi-object filter at subsequent frames. Thus, current multi-object tracking algorithms are far from being a ‘plug-and-play’ technology, since their application still requires cumbersome and error-prone user configuration.

We remark that robust Bayesian approaches to problems with model mismatch in the literature such as [113–118] are classical approaches that are too computationally intensive for an online multi-object tracker. A Sequential Monte Carlo technique for calibration of time-invariant multi-object model parameters was proposed in [119]. While this approach is quite general, it is not directly applicable to time-varying clutter rate and detection profile, and is also too computationally intensive for an online tracker.

A PHD filter that performs joint clutter background estimation and filtering is given in [120]. It uses nonhomogeneous Poisson point process to model the spatial



Figure 4.1: Frames 16, 48 of the image sequence from [108] and object detections obtained using the detector in [109]. The number of objects varies with time due to objects coming in and out of the scene. Object estimates (marked by blue boxes) using the standard GLMB filter for guessed clutter rate of 60 (top 2 frames) and ‘true’ clutter rate (bottom 2 frames). Tracking using ‘true’ clutter rate accurately estimated several objects that were missed in the frames on the top.

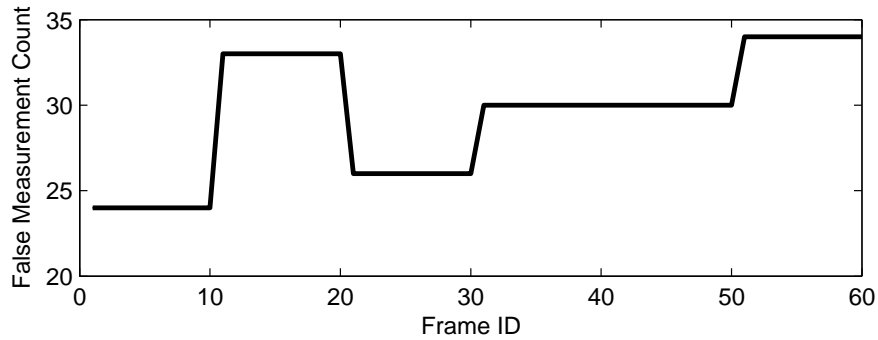


Figure 4.2: ‘True’ clutter rate for the first 60 frames of the dataset [108]. Note that it is not possible to know the true clutter rate for real video data. For illustration, we assume that the clutter rate varies slowly and use the average clutter count over a moving 10-frame window as the ‘true’ clutter rate.

distribution of clutter points and the PHD filter output is fed back into the clutter estimation algorithm for improved performance. The foundation for a CPHD filter in unknown detection probability is described in [121] and for a CPHD filter in unknown clutter rate is described in [122]. An analytic implementation for a CPHD filter built upon [121],[122] is presented in [123]. A multi-Bernoulli filter that operates amidst unknown clutter rate and detection probability is presented in [124]. A Kronecker Delta Mixture and Poisson filter for estimating clutter rate and detection probability is given in [125]. An inverse gamma Gaussian mixture model CPHD solution for filtering amidst unknown detection probability, which makes use of the inverse gamma component to propagate features such as signal amplitude and SNR to is given in [126]. Application of [124] to track cell microscopy data with unknown background parameters is presented in [127]. Nonetheless, none of these filters produces tracks. Further, the CPHD, PHD and multi-Bernoulli filters require more drastic approximations than the GLMB filter. The track-before-detect approach, (eg. the multi-Bernoulli filters in [110–112] does not require knowledge of clutter rate and detection probability. However, they are still expensive relative to detection based algorithms.

This chapter presents an online multi-object tracker that learns the clutter and detection model parameters while tracking built upon the δ -GLMB filter from the RFS paradigm because it outputs tracks, is provably Bayes optimal [14] and admits efficient implementation [87]. Further, the GLMB is a versatile model that offers good trade-offs between tractability and fidelity, see [94] and the references therein.

4.2 Multi-Class GLMB

4.2.1 Standard multi-object likelihood with zero clutter

For a given multi-object state \mathbf{X} , each $(x, \ell) \in \mathbf{X}$ is either detected with probability $P_D(x, \ell)$ and generates a detection $z \in Z$ with likelihood $g(z|x, \ell)$ or missed with probability $1 - P_D(x, \ell)$. The *multi-object observation* is the superposition of the observations from detected objects and Poisson clutter with (positive) intensity κ . Assuming that, conditional on \mathbf{X} , detections are independent of each other and clutter, the multi-object likelihood function is given by [15, 86],

$$g(Z|\mathbf{X}) \propto \sum_{\theta \in \Theta} 1_{\Theta(\mathcal{L}(\mathbf{X}))}(\theta) \prod_{(x, \ell) \in \mathbf{X}} \psi_Z^{(\theta(\ell))}(x, \ell) \quad (4.1)$$

where: Θ is the set of *positive 1-1* maps $\theta : \mathbb{L} \rightarrow \{0:|Z|\}$, i.e. maps such that *no two distinct arguments are mapped to the same positive value*, $\Theta(I)$ is the set of *positive 1-1* maps with domain I ; and

$$\psi_{\{z_{1:M}\}}^{(j)}(x, \ell) = \begin{cases} \frac{P_D(x, \ell)g(z_j|x, \ell)}{\kappa(z_j)}, & \text{if } j = 1:M \\ 1 - P_D(x, \ell), & \text{if } j = 0 \end{cases} \quad (4.2)$$

The map θ specifies which objects generated which detections, i.e. object ℓ generates detection $z_{\theta(\ell)} \in Z$, with undetected objects assigned to 0. The positive 1-1 property means that θ is 1-1 on $\{\ell : \theta(\ell) > 0\}$, the set of labels that are assigned positive values, and ensures that any detection in Z is assigned to at most one object.

For the special case with zero-clutter, i.e. κ is identically zero, the multi-object likelihood function still takes the same form, but with $P_D(x, \ell)g(z_j|x, \ell)/\kappa(z_j)$ replaced by $P_D(x, \ell)g(z_j|x, \ell)$, see [13, 14]. To cover both positive and identically-zero clutter intensities we write

$$\psi_{\{z_{1:M}\}}^{(j)}(x, \ell) = \begin{cases} \frac{P_D(x, \ell)g(z_j|x, \ell)}{\kappa(z_j) + \delta_0[\kappa(z_j)]}, & \text{if } j = 1:M \\ 1 - P_D(x, \ell), & \text{if } j = 0 \end{cases} \quad (4.3)$$

4.2.2 Multi class GLMB with zero clutter

The efficient JMS-GLMB recursion from Section 3.3 can be applied to the joint multi-object tracking and classification problem by using the mode (motion model) as the class label (not to be confused with the object label). What distinguishes this problem from generic JMS-GLMB filtering is that the modes do not interact with each other in the following sense:

1. All possible states of a new object with the same object label share a common mode (class label);
2. An object cannot switch between different modes from one time step to the next.

Let us introduce some new notation related to class labels first. Recall that \mathbb{X} denotes the space for object kinematic state, \mathbb{M} denotes the space of modes and \mathbb{B} denote the entire label space for new born objects at the current time. Let $\mathbb{B}^{(m)}$ denote the set of labels of all elements in the space $\mathbb{X} \times \mathbb{M} \times \mathbb{B}$ with mode m . Then condition 1 implies that the label sets $\mathbb{B}^{(m)}$ and $\mathbb{B}^{(m')}$ for different modes m and m' are disjoint (otherwise there exist a label ℓ in both $\mathbb{B}^{(m)}$ and $\mathbb{B}^{(m')}$, which means there are states in $\mathbb{X} \times \mathbb{M} \times \mathbb{B}$ with different modes m and m' but share a common label ℓ). Furthermore, the sets $\mathbb{B}^{(m)}$, $m \in \mathbb{M}$ cover \mathbb{B} , i.e. $\mathbb{B} = \biguplus_{m \in \mathbb{M}} \mathbb{B}^{(m)}$, and thus form a partition of the space \mathbb{B} . A new object is classified as class m (and has mode m) if and only if its label falls into $\mathbb{B}^{(m)}$. Thus for an LMB birth model, condition 1 means

$$r_{B,+}(\ell_+) = \sum_{m_+ \in \mathbb{M}} r_{B,+}^{(m_+)} 1_{\mathbb{B}_+^{(m_+)}}(\ell_+), \quad (4.4)$$

$$p_{B,+}^{(m_+)}(\zeta_+, \ell_+) = p_{B,+}^{(m_+)}(\zeta_+) 1_{\mathbb{B}_+^{(m_+)}}(\ell_+). \quad (4.5)$$

Note that $r_{B,+}^{(m_+)}$ and $p_{B,+}^{(m_+)}(\zeta_+)$ are respectively the existence probability and probability density of the kinematics ζ_+ of a new object given mode m_+ , while $1_{\mathbb{B}_+^{(m_+)}}(\ell_+)$ is the probability of mode m_+ given label ℓ_+ .

Condition 2 means that the mode transition probability

$$\vartheta(m_+|m) = \delta_m[m_+], \quad (4.6)$$

which implies that each object belongs to exactly one of the classes in \mathbb{M} for its entire

life. Consequently, the non-interacting mode condition means that at time k , the label space for all class m objects is $\mathbb{L}^{(m)} = \bigcup_{t=0:k} \mathbb{B}_t^{(m)}$, and the set of all possible labels is given by the disjoint union $\mathbb{L} = \bigcup_{m \in \mathbb{M}} \mathbb{L}^{(m)}$.

For a multi-object JMS system with non-interacting modes, the JMS-GLMB recursion reduces to a form where the weights and multi-object exponentials can be separated according to classes. We call this form the multi-class GLMB.

Proposition 8. Let $\mathbf{X}^{(m)}$ denote the subset of \mathbf{X} with mode m , and hence $\mathbf{X} = \bigcup_{m \in \mathbb{M}} \mathbf{X}^{(m)}$. Suppose that the hybrid multi-object density at the current time is a GLMB of the form

$$\pi(\mathbf{X}) = \sum_{\xi, I} 1_{\Theta(I)}(\xi \perp \Theta) \prod_{m \in \mathbb{M}} \pi^{(I^{(m)}, \xi^{(m)})}(\mathbf{X}^{(m)}) \quad (4.7)$$

where $\xi \in \Xi$, $I \subseteq \mathbb{L}$, $\xi \perp \Theta$ denotes the projection ξ into the space Θ (i.e., if $\xi = (\theta_1; \theta_k)$ then $\xi \perp \Theta_k = \theta_k$),

$$I^{(m)} \triangleq I \cap \mathbb{L}^{(m)}, \quad \xi^{(m)} = \xi|_{\mathbb{L}_0^{(m)} \times \dots \times \mathbb{L}_k^{(m)}} \quad (\text{i.e. the map } \xi \text{ restricted to } \mathbb{L}_0^{(m)} \times \dots \times \mathbb{L}_k^{(m)}), \quad \text{and} \quad (4.8)$$

$$\pi^{(I^{(m)}, \xi^{(m)})}(\mathbf{X}^{(m)}) \triangleq \Delta(\mathbf{X}^{(m)}) \omega^{(I^{(m)}, \xi^{(m)})} \delta_{I^{(m)}}[\mathcal{L}(\mathbf{X}^{(m)})] [p^{(\xi^{(m)})}]^{\mathbf{X}^{(m)}} \quad (4.9)$$

Then the hybrid multi-object filtering density at the next time step is the GLMB

$$\pi_{Z_+}(\mathbf{X}_+) \propto \sum_{\xi, I, \theta_+, I_+} 1_{\Theta_+(I_+)}(\theta_+) \prod_{m \in \mathbb{M}} \pi_{Z_+}^{(m, I^{(m)}, \xi^{(m)}, I_+^{(m)}, \theta_+^{(m)})}(\mathbf{X}_+^{(m)}) \quad (4.10)$$

where $I_+ \in \mathcal{F}(\mathbb{L}_+)$, $\theta_+ \in \Theta_+$, $I_+^{(m)} = I_+ \cap \mathbb{L}_+^{(m)}$, $\theta_+^{(m)} = \theta_+|_{\mathbb{L}_+^{(m)}}$.

$$\pi_{Z_+}^{(m, I, \xi, I_+, \theta_+)}(\mathbf{Y}_+) = \Delta(\mathbf{Y}_+) \omega_{Z_+}^{(m, I, \xi, I_+, \theta_+)} \omega^{(I, \xi)} \delta_{I_+}[\mathcal{L}(\mathbf{Y}_+)] [p_{Z_+}^{(\xi, \theta_+)}]^{\mathbf{Y}_+} \quad (4.11)$$

$$\begin{aligned} \omega_{Z_+}^{(m, I, \xi, I_+, \theta_+)} &= \left[\bar{\psi}_{Z_+}^{(\xi, \theta_+)}(m, \cdot) \right]^{I_+} [1 - r_{B,+}]^{\mathbb{B}_+^{(m)} - I_+} [r_{B,+}]^{\mathbb{B}_+^{(m)} \cap I_+} \times \\ &\quad \left[1 - \bar{P}_S^{(\xi)}(m, \cdot) \right]^{I - I_+} \left[\bar{P}_S^{(\xi)}(m, \cdot) \right]^{I \cap I_+}, \end{aligned} \quad (4.12)$$

$$\bar{P}_S^{(\xi)}(m, \ell) = \left\langle p^{(\xi)}(\cdot, m, \ell), P_S^{(m)}(\cdot, \ell) \right\rangle,$$

$$\bar{\psi}_{Z_+}^{(\xi, \theta_+)}(m, \ell) = \left\langle \bar{p}_+^{(\xi)}(\cdot, m, \ell), \psi_{Z_+}^{(\theta_+(\ell))}(\cdot, m, \ell) \right\rangle, \quad (4.13)$$

$$\begin{aligned} \bar{p}_+^{(\xi)}(\zeta, m, \ell) &= 1_{\mathbb{L}^{(m)}}(\ell) \frac{\left\langle P_S^{(m)}(\cdot, \ell) f_+^{(m)}(\zeta|\cdot, \ell), p^{(\xi)}(\cdot, m, \ell) \right\rangle}{\bar{P}_S^{(\xi)}(m, \ell)} \\ &\quad + 1_{\mathbb{B}_+^{(m)}}(\ell) p_B^{(m)}(\zeta, \ell) \end{aligned} \quad (4.14)$$

$$p_{Z_+}^{(\xi, \theta_+)}(\zeta, m, \ell) = \frac{\bar{p}_+^{(\xi)}(\zeta, m, \ell) \psi_{Z_+}^{(\theta_+, \ell)}(\zeta, m, \ell)}{\bar{\psi}_{Z_+}^{(\xi, \theta_+)}(m, \ell)} \quad (4.15)$$

$$\psi_{\{z_1, \dots, z_{|Z_+|}\}}^{(j)}(\zeta, m, \ell) = \begin{cases} \frac{P_{D,+}^{(m)}(\zeta, \ell) g_+^{(m)}(z_j|\zeta, \ell)}{\kappa_+(z_j) + \delta_0[\kappa_+(z_j)]}, & \text{if } j \in \{1, \dots, |Z_+|\} \\ 1 - P_{D,+}^{(m)}(\zeta, \ell), & \text{if } j = 0. \end{cases}$$

Proof: Note that the $\mathbb{L}_0^{(m)} \times \dots \times \mathbb{L}_k^{(m)}$, $m \in \mathbb{M}$ form a partition of $\mathbb{L}_0 \times \dots \times \mathbb{L}_k$, and since each $\xi^{(m)}$ was defined as a restrictions of ξ over $\mathbb{L}_0^{(m)} \times \dots \times \mathbb{L}_k^{(m)}$, ξ is completely characterized by the $\xi^{(m)}$, $m \in \mathbb{M}$.

Recall from Eq. (2.75) the definition of a δ -GLMB RFS with state space \mathbb{X} , discrete space Ξ and discrete label space \mathbb{L} as,

$$\pi(\mathbf{X}) = \Delta(\mathbf{X}) \sum_{(I, \xi) \in \mathcal{F}(\mathbb{L}) \times \Xi} \omega^{(I, \xi)} \delta_I(\mathcal{L}(\mathbf{X})) \left[p^{(\xi)} \right]^{\mathbf{X}}. \quad (4.16)$$

By defining

$$\omega^{(I, \xi)} = 1_{\Theta(I)}(\xi \perp \Theta) \prod_{m \in \mathbb{M}} \omega^{(I^{(m)}, \xi^{(m)})} \quad (4.17)$$

$$p^{(\xi)}(\zeta, m, \ell) = \left[p^{(\xi^{(m)})}(\zeta, m, \ell) \right]^{1_{\mathbb{L}^{(m)}}(\ell)}. \quad (4.18)$$

It can be seen that Eq. (4.7) is a GLMB of the form Eq. (4.16) since

$$\delta_I[\mathcal{L}(\mathbf{X})] = \prod_{m \in \mathbb{M}} \delta_{I^{(m)}}[\mathcal{L}(\mathbf{X}^{(m)})], \quad (4.19)$$

$$\left[p^{(\xi)} \right]^{\mathbf{X}} = \left[p^{(\xi)} \right]_{m \in \mathbb{M}}^{\uplus \mathbf{X}^{(m)}} = \prod_{m \in \mathbb{M}} \left[p^{(\xi^{(m)})} \right]^{\mathbf{X}^{(m)}} \quad (4.20)$$

Proposition 7 from Chapter 3, gives the hybrid multi-object filtering density at time $k + 1$. Substituting Eq. (4.21), Eq. (4.22), Eq. (4.23), Eq. (4.4) into Proposition 7 from Chapter 3 and decomposing

$$\mathbf{X}_+ = \bigsqcup_{m \in \mathbb{M}} \mathbf{X}_+^{(m)} \quad (4.21)$$

$$\omega_{Z_+}^{(I, \xi, I_+, \theta_+)} = 1_{\Theta_+(I_+)}(\theta_+) \prod_{m \in \mathbb{M}} \omega_{Z_+}^{(m, I^{(m)}, \xi^{(m)}, I_+^{(m)}, \theta_+^{(m)})} \quad (4.22)$$

$$p_{Z_+}^{(\xi, \theta_+)} = \left(p_{Z_+}^{(\xi^{(m)}, \theta_+^{(m)})} \right)_{\mathbb{L}_+^{(m)}(\ell)} \quad (4.23)$$

and rearranging yields Eq. (4.10). Note that Eq. (4.6) ensures $m_+ = m$.

Given a GLMB filtering density of the multi-class form Eq. (4.7), the GLMB filtering density for class $c \in \mathbb{M}$, can be obtained by marginalizing the other classes according to the following proposition.

Proposition 9. *For the multi-class GLMB Eq. (4.7), the marginal GLMB for class c is given by*

$$\pi(\mathbf{X}^{(c)}) = \Delta(\mathbf{X}^{(c)}) \sum_{\xi, I} \omega^{(I, \xi)} \delta_{I^{(c)}}[\mathcal{L}(\mathbf{X}^{(c)})] \left[p^{(\xi^{(c)})} \right]^{\mathbf{X}^{(c)}}. \quad (4.24)$$

Proof: Note that

$$\begin{aligned} \int \pi^{(I^{(m)}, \xi^{(m)})}(\mathbf{X}^{(m)}) \delta \mathbf{X}^{(m)} &= \int \Delta(\mathbf{X}^{(m)}) \omega^{(I^{(m)}, \xi^{(m)})} \delta_{I^{(m)}}[\mathcal{L}(\mathbf{X}^{(m)})] \left[p^{(\xi)} \right]^{\mathbf{X}^{(m)}} \delta \mathbf{X}^{(m)}, \\ &= \omega^{(I^{(m)}, \xi^{(m)})}. \end{aligned}$$

Since, the $\mathbf{X}^{(m)}$, $m \in \mathbb{M}$ are disjoint

$$\begin{aligned} \pi(\mathbf{X}^{(c)}) &= \int \pi \left(\bigsqcup_{m \in \mathbb{M}} \mathbf{X}^{(m)} \right) \delta \left(\bigsqcup_{m \in \mathbb{M} - \{c\}} \mathbf{X}^{(m)} \right), \\ &= \int \sum_{\xi, I} 1_{\Theta(I)}(\xi \perp \Theta) \times \prod_{m \in \mathbb{M}} \pi^{(I^{(m)}, \xi^{(m)})}(\mathbf{X}^{(m)}) \delta \left(\bigsqcup_{m \in \mathbb{M} - \{c\}} \mathbf{X}^{(m)} \right), \\ &= \sum_{\xi, I} 1_{\Theta(I)}(\xi \perp \Theta) \pi^{(I^{(c)}, \xi^{(c)})}(\mathbf{X}^{(c)}) \times \prod_{m \in \mathbb{M} - \{c\}} \int \pi^{(I^{(m)}, \xi^{(m)})}(\mathbf{X}^{(m)}) \delta \mathbf{X}^{(m)}, \\ &= \sum_{\xi, I} 1_{\Theta(I)}(\xi \perp \Theta) \pi^{(I^{(c)}, \xi^{(c)})}(\mathbf{X}^{(c)}) \times \prod_{m \in \mathbb{M} - \{c\}} \omega^{(I^{(m)}, \xi^{(m)})}, \\ &= \Delta(\mathbf{X}^{(c)}) \sum_{I, \xi} \omega^{(I, \xi)} \delta_{I^{(c)}}[\mathcal{L}(\mathbf{X}^{(c)})] \left[p^{(\xi^{(c)})} \right]^{\mathbf{X}^{(c)}}. \end{aligned}$$

4.3 GLMB Filtering with Unknown Background

Clutter or false detections are generally understood as detections that do not correspond to any object [13, 14, 22, 23]. Since the number of false detections and their values are random, clutter is usually modeled by RFSs in the literature [13, 14, 70]. The simplest and the most commonly used clutter model is the Poisson RFS [70], in most cases, with a uniform intensity over the surveillance region. Alternatively, clutter can be treated as detections originating from *clutter generators*-objects that are not of interest to the tracker [121–124].

In [123] a CPHD recursion was derived to propagate separate intensity functions for clutter generators and objects of interest, and their collective cardinality distribution of the hybrid multi-object state. Similarly, in [124] analogous multi-Bernoulli recursions were derived to propagate the disjoint union of objects of interest and clutter generators. In this work, we show that the multi-class GLMB filter is an effective multi-object object tracker that can operate under unknown background by learning the clutter and detection model on-the-fly. In particular, a GLMB clutter model is proposed in Section 4.3.1 by treating clutter as a special class of objects with completely uncertain dynamics, along with a dedicated GLMB recursion for propagating the joint filtering density of clutter generators and objects of interest. Implementation details are given in Section 4.3.2. Extension of the proposed algorithm to accommodate unknown detection profile is described in Section 4.3.3.

4.3.1 GLMB Joint Object-Clutter Model

We propose to model the finite set of *clutter generators* and *objects of interest* as two non-interacting classes of objects, and propagate this so-called *hybrid multi-object* filtering density forward in time via the multi-class GLMB recursion. The GLMB posterior density of the hybrid multi-object state captures all relevant statistical information on the objects of interest as well as the clutter generators. What distinguishes the objects of interest from clutter generators is that the former have relatively predictable dynamics whereas the latter have essentially no dynamics.

In the hybrid multi-object model, the Poisson clutter intensity κ is identically 0 and each detection is generated from either a clutter generator or an object of interest, which constitute, respectively, the two modes (or classes) 0 and 1 of the mode space

$\mathbb{M} = \{0, 1\}$. Since the classes are non-interacting, there is no switching between objects of interest and clutter generators. Moreover, the label space for newborn clutter generators $\mathbb{B}^{(0)}$ and the label space for newborn objects of interest $\mathbb{B}^{(1)}$ are disjoint and the LMB birth parameters are given by

$$\begin{aligned} r_{B,+}(\ell_+) &= r_{B,+}^{(0)} 1_{\mathbb{B}^{(0)}}(\ell_+) + r_{B,+}^{(1)} 1_{\mathbb{B}^{(1)}}(\ell_+), \\ p_{B,+}^{(m_+)}(\zeta_+, \ell_+) &= p_{B,+}^{(m_+)}(\zeta_+) 1_{\mathbb{B}^{(m_+)}}(\ell_+). \end{aligned}$$

Since clutter has no dynamics, each clutter generator has a transition density independent of the previous state and a uniform measurement likelihood in the observation region with volume V .

$$\begin{aligned} f_+^{(0)}(\zeta_+|\zeta, \ell) &= s(\zeta_+), \\ g^{(0)}(z|\zeta, \ell) &= u(z)V^{-1}. \end{aligned}$$

Note that the labels of clutter generators can effectively be ignored since it is implicit that their labels are distinct but are otherwise uninformative. Further, for Gaussian implementations, it is assumed that the survival and detection probabilities for clutter generators are state independent.

$$\begin{aligned} P_S^{(0)}(\zeta, \ell) &= P_S^{(0)}, \\ P_{D,+}^{(0)}(\zeta, \ell) &= P_{D,+}^{(0)}. \end{aligned}$$

Applying the multi-class GLMB recursion to this model, it can be easily seen that all clutter generators are functionally identical (from birth through prediction and update).

$$p_B^{(0)}(\zeta, \ell) = \bar{p}_+^{(\xi^{(0)})}(\zeta, 0, \ell) = p_{Z_+}^{(\xi^{(0)}, \theta_+^{(0)})}(\zeta, 0, \ell) = s(\zeta) \quad (4.25)$$

and that the weight update for clutter generators reduces to

$$\begin{aligned} \omega_{Z_+}^{(0, I^{(0)}, \xi^{(0)}, I_+^{(0)}, \theta_+^{(0)})} &= \left[1 - P_S^{(0)} \right]^{|I^{(0)} - I_+^{(0)}|} \left[P_S^{(0)} \right]^{|I^{(0)} \cap I_+^{(0)}|} \left[1 - r_{B,+}^{(0)} \right]^{|B_+^{(0)} - I_+^{(0)}|} \left[r_{B,+}^{(0)} \right]^{|B_+^{(0)} \cap I_+^{(0)}|} \times \\ &\quad \left[1 - P_{D,+}^{(0)} \right]^{|I \in I_+^{(0)}: \theta_+^{(0)}(\ell)=0|} \left[P_{D,+}^{(0)} V^{-1} \right]^{|I \in I_+^{(0)}: \theta_+^{(0)}(\ell)>0|}. \end{aligned} \quad (4.26)$$

Thus the propagation of clutter generators within each GLMB component reduces to

the propagation of their weights

$$\omega_{Z_+}^{(0, I_+^{(0)}, \xi^{(0)}, \theta_+^{(0)})} = \sum_{I^{(0)}} \omega^{(I^{(0)}, \xi^{(0)})} \omega_Z^{(0, I^{(0)}, \xi^{(0)}, I_+^{(0)}, \theta_+^{(0)})}. \quad (4.27)$$

4.3.2 Implementation

The key challenge in the implementation of the multi-class GLMB filter is the propagation of the GLMB components, which involves, for each parent GLMB component (I, ξ) , searching the space $\mathcal{F}(\mathbb{L}_+) \times \Theta_+$ to find a set of (I_+, θ_+) such that the children components (I, ξ, I_+, θ_+) have significant weights $\omega_{Z_+}^{(I, \xi, I_+, \theta_+)}$. In [87], the set of (I_+, θ_+) is generated from a Gibbs sampler, with stationary distribution constructed so that only valid children components have positive probabilities, and those with high weights are more likely to be sampled than those with low weights. A direct application of this approach to generate new children would, however, would be expensive, for the following reasons.

Let $P = |I|$, $P^{(0)} = |I^{(0)}|$, $P^{(1)} = |I^{(1)}|$ and $M = |Z_+|$. According to [87] the complexity of the joint prediction and update via Gibbs sampling with T iterations is $\mathcal{O}(TP^2M)$. Since the present formulation treats clutter as objects, the total number of hypothesized objects $P \geq P^{(0)} \geq M$, and hence the complexity is at least $\mathcal{O}(TM^3)$, which is cubic in the number of measurements and results in a relatively inefficient implementation. This occurs because the majority of the computational effort is spent on clutter generators even though they are not of interest. This problem is exacerbated as the clutter rate increases.

In the following we propose a more efficient implementation by focusing on the filtering density of the objects of interest instead of the hybrid multi-object filtering density. Observe that given any $(I_+^{(1)}, \theta_+^{(1)}) \in \mathcal{F}(\mathbb{L}_+^{(1)}) \times \Theta_+^{(1)}$, and $(I_+^{(0)}, \theta_+^{(0)}) \in \mathcal{F}(\mathbb{L}_+^{(0)}) \times \Theta_+^{(0)}$, where $\Theta_+^{(m)}$ denotes the space of positive 1-1 maps from $\mathbb{L}_+^{(m)}$ to $\{0, 1, \dots, M\}$, we can uniquely define

$$(I_+, \theta_+) \triangleq \left(I_+^{(1)} \uplus I_+^{(0)}, 1_{\mathbb{L}_+^{(1)}}(\cdot) \theta_+^{(1)}(\cdot) + 1_{\mathbb{L}_+^{(0)}}(\cdot) \theta_+^{(0)}(\cdot) \right) \quad (4.28)$$

Further, the weight of the resulting component (I, ξ, I_+, θ_+) is

$$\omega_{Z_+}^{(I, \xi, I_+, \theta_+)} = 1_{\Theta(I_+)(\theta_+)} \omega_{Z_+}^{(0, I^{(0)}, \xi^{(0)}, I_+^{(0)}, \theta_+^{(0)})} \omega_{Z_+}^{(1, I^{(1)}, \xi^{(1)}, I_+^{(1)}, \theta_+^{(1)})} \quad (4.29)$$

see Proposition 9, Eq. (4.22). Note that if θ_+ is not a valid association map then $1_{\Theta(I_+)}(\theta_+) = 0$, and hence the weight is zero.

For each parent GLMB component (I, ξ) , rather than searching for (I_+, θ_+) with significant $\omega_{Z_+}^{(I, \xi, I_+, \theta_+)}$ in the space $\mathcal{F}(\mathbb{L}_+) \times \Theta_+$, we:

1. seek $(I_+^{(1)}, \theta_+^{(1)})$ with significant $\omega_{Z_+}^{(1, I^{(1)}, \xi^{(1)}, I_+^{(1)}, \theta_+^{(1)})}$ from the smaller space $\mathcal{F}(\mathbb{L}_+^{(1)}) \times \Theta_+^{(1)}$;
2. for each such $(I_+^{(1)}, \theta_+^{(1)})$ find the $(I_+^{(0)}, \theta_+^{(0)})$ with the best $\omega_{Z_+}^{(0, I^{(0)}, \xi^{(0)}, I_+^{(0)}, \theta_+^{(0)})}$, subject to the constraint,

$$1_{\mathbb{L}_+^{(1)}}(\cdot) \theta_+^{(1)}(\cdot) + 1_{\mathbb{L}_+^{(0)}}(\cdot) \theta_+^{(0)}(\cdot) \in \Theta \left(I_+^{(1)} \uplus I_+^{(0)} \right); \quad (4.30)$$

3. construct (I_+, θ_+) from $(I_+^{(1)}, \theta_+^{(1)})$ and $(I_+^{(0)}, \theta_+^{(0)})$ via Eq. (4.28) and compute the corresponding weight via Eq. (4.29).

Due to the constraint Eq. (4.30), $1_{\Theta(I_+)}(\theta_+) = 1$, and hence, it follows from Eq. (4.29) that the resulting GLMB component (I, ξ, I_+, θ_+) also has significant weight. The goal here is to generate a set of mappings with significant weights, but not necessarily in the order of the weight, since the latter is highly computationally intensive.

The advantage of this strategy is two fold:

- searching over a much smaller space $\mathcal{F}(\mathbb{L}_+^{(1)}) \times \Theta_+^{(1)}$ results in a linear complexity in the measurements $\mathcal{O}(T(P^{(1)})^2 M)$ since typically $P^{(1)} \ll M$;
- finding $(I_+^{(0)}, \theta_+^{(0)})$ with the best weight subject to the constraint $\theta_+ \in \Theta(I_+)$ is straight forward and requires minimal computation.

Propagating Objects of Interest

One way to generate significant $(I_+^{(1)}, \theta_+^{(1)})$ is to design a Gibbs sampler with stationary distribution $\omega^{(1, I^{(1)}, \xi^{(1)}, I_+^{(1)}, \theta_+^{(1)})}$. However, this approach requires computing the hybrid multi-object density, which we try to avoid in the first place.

A much more efficient alternative is to treat the multi-Bernoulli clutter as Poisson with matching intensity, and apply the standard GLMB filter given in [Section 2.5.2](#) (i.e., the JMS-GLMB filter with a single-mode), where the Gibbs sampler [89] (or Murty's algorithm [92]) can be used to obtain significant $(I_+^{(1)}, \theta_+^{(1)})$ [87]. Since there are $|I^{(0)}|$ clutter generators from the previous time with survival probability $P_S^{(0)}$, and $|\mathbb{B}_+^{(0)}|$ clutter birth with probability $r_{B,+}^{(0)}$, the predicted clutter intensity is given by

$$\hat{\kappa}_+ = \left(P_S^{(0)} |I^{(0)}| + r_{B,+}^{(0)} |\mathbb{B}_+^{(0)}| \right) P_{D,+}^{(0)} V^{-1}. \quad (4.31)$$

Note that a Poisson RFS has larger variance on the number of clutter points than a multi-Bernoulli with matching intensity. Hence, in treating clutter as a Poisson RFS, we are effectively tempering with the clutter model to induce the Gibbs sampler (or Murty's algorithm) to generate more diverse components [87].

Following [87], let us enumerate $Z_+ = \{z_{1:M}\}$, $I^{(1)} = \{\ell_{1:R}\}$, and $\mathbb{B}_+^{(1)} = \{\ell_{R+1:P}\}$. The $(I_+^{(1)}, \theta_+^{(1)}) \in \mathcal{F}(\mathbb{L}_+^{(1)}) \times \Theta(I_+^{(1)})$ at time $k+1$ with significant weights are determined by solving a ranked assignment problem with cost matrix $[\eta_i^{(\xi^{(1)})}(j)]$, $i = 1:P$, $j = -1:M$, computed as,

$$\eta_i^{(\xi^{(1)})}(j) = \begin{cases} 1 - \bar{P}_S^{(\xi^{(1)})}(1, \ell_i) & \ell_i \in I^{(1)}, j < 0 \\ \bar{P}_S^{(\xi^{(1)})}(1, \ell_i) \bar{\psi}_{Z_+}^{(\xi^{(1)}, \theta_+^{(1)})}(1, \ell_i) & \ell_i \in I^{(1)}, j \geq 0 \\ 1 - r_{B,+}(\ell_i) & \ell_i \in \mathbb{B}_+^{(1)}, j < 0 \\ r_{B,+}(\ell_i) \bar{\psi}_{Z_+}^{(\xi^{(1)}, \theta_+^{(1)})}(1, \ell_i) & \ell_i \in \mathbb{B}_+^{(1)}, j \geq 0 \end{cases} \quad (4.32)$$

where,

$$\begin{aligned} \bar{\psi}_{Z_+}^{(\xi^{(1)}, \theta_+^{(1)})}(1, \ell) &= \left\langle \bar{p}_+^{(\xi^{(1)})}(\cdot, 1, \ell), \psi_{Z_+}^{(\theta_+^{(1)}, \ell)}(\cdot, 1, \ell) \right\rangle \\ \psi_{Z_+}^{(j)}(\zeta, 1, \ell) &= \begin{cases} \frac{P_{D,+}^{(1)}(\zeta, \ell) g_+^{(1)}(z_j | \zeta, \ell)}{\hat{\kappa}_+}, & \text{if } j \in \{1, \dots, M\} \\ 1 - P_{D,+}^{(1)}(\zeta, \ell), & \text{if } j = 0 \end{cases} \end{aligned}$$

Such a ranked assignment problem can be solved by Murty's algorithm or the Gibbs sampler.

Propagating Clutter Generators

Given $(I_+^{(1)}, \theta_+^{(1)})$ pertaining to the objects of interest, we proceed to determine $(I_+^{(0)}, \theta_+^{(0)})$ pertaining to clutter generators, which maximizes $w_{Z_+}^{(0, I^{(0)}, \xi^{(0)}, I_+^{(0)}, \theta_+^{(0)})}$ where $I_+^{(0)} \subseteq I^{(0)} \cup \mathbb{B}_+^{(0)}$ and $\theta_+^{(0)} : I_+^{(0)} \rightarrow \{0 : M\}$ subject to constraint Eq. (4.30).

Denote by $Z_+^{(1)} \subseteq Z_+$ the set of measurements assigned to $I_+^{(1)}$ by $\theta_+^{(1)}$ and the remaining set of measurements $Z_+ - Z_+^{(1)}$, due to clutter generators, by $Z_+^{(0)}$. Recall that clutter generators are functionally identical except in label and that their propagation reduces to calculating their corresponding weights Eq. (4.26). Let $N_S^{(0)} = |I^{(0)} \cap I_+^{(0)}|$ and $N_{B,+}^{(0)} = |\mathbb{B}_+^{(0)} \cap I_+^{(0)}|$ denote the counts of surviving and new born clutter generators respectively. Then $|I^{(0)} - I_+^{(0)}| = |I^{(0)}| - N_S^{(0)}$ and $|\mathbb{B}_+^{(0)} - I_+^{(0)}| = |\mathbb{B}_+^{(0)}| - N_{B,+}^{(0)}$. Observe that the count $|Z_+^{(0)}|$ of clutter must equal the number of detections of clutter generators according to $(I_+^{(0)}, \theta_+^{(0)})$, i.e. $|Z_+^{(0)}| = |\{\ell \in I_+^{(0)} : \theta_+^{(0)}(\ell) > 0\}|$ and hence the count of misdetections of clutter generators according to $(I_+^{(0)}, \theta_+^{(0)})$ is $N_S^{(0)} + N_{B,+}^{(0)} - |Z_+^{(0)}| = |\{\ell \in I_+^{(0)} : \theta_+^{(0)}(\ell) = 0\}|$. Consequently the weight Eq. (4.26) can be rewritten as

$$\begin{aligned} w_{Z_+}^{(0, I^{(0)}, \xi^{(0)}, I_+^{(0)}, \theta_+^{(0)})} &= \left[1 - P_S^{(0)}\right]^{|I^{(0)}| - N_S^{(0)}} \left[P_S^{(0)}\right]^{N_S^{(0)}} \left[1 - r_{B,+}^{(0)}\right]^{|\mathbb{B}_+^{(0)}| - N_{B,+}^{(0)}} \times \\ &\quad \left[r_{B,+}^{(0)}\right]^{N_{B,+}^{(0)}} \left[1 - P_{D,+}^{(0)}\right]^{N_S^{(0)} + N_{B,+}^{(0)} - |Z_+^{(0)}|} \left[P_{D,+}^{(0)} V^{-1}\right]^{|Z_+^{(0)}|} \\ &\propto \left[\frac{P_S^{(0)}(1 - P_{D,+}^{(0)})}{1 - P_S^{(0)}}\right]^{N_S^{(0)}} \left[\frac{r_{B,+}^{(0)}(1 - P_{D,+}^{(0)})}{1 - r_{B,+}^{(0)}}\right]^{N_{B,+}^{(0)}}. \end{aligned}$$

Thus seeking the best $(I_+^{(0)}, \theta_+^{(0)})$ subject to constraint Eq. (4.30) reduces to seeking the best $(N_S^{(0)}, N_{B,+}^{(0)})$ subject to the constraints $0 \leq N_S^{(0)} \leq |I^{(0)}|$, $0 \leq N_{B,+}^{(0)} \leq |\mathbb{B}_+^{(0)}|$ and $N_S^{(0)} + N_{B,+}^{(0)} \geq |Z_+^{(0)}|$.

Linear Gaussian Update Parameters

Let $\mathcal{N}(\cdot; \bar{\zeta}, P)$ denote a Gaussian density with mean $\bar{\zeta}$ and covariance P . Then for a linear Gaussian multi-object model of the objects of interest,

$$P_S^{(1)}(\zeta, \ell) = P_S^{(1)},$$

$$P_D^{(1)}(\zeta, \ell) = P_D^{(1)},$$

$$\begin{aligned}
f_+^{(1)}(\zeta_+|\zeta, \ell) &= \mathcal{N}(\zeta_+; F\zeta, Q), \\
g^{(1)}(z|\zeta, \ell) &= \mathcal{N}(z; H\zeta, R), \\
p_{B,+}^{(1)}(\zeta_+) &= \mathcal{N}(\zeta_+; \bar{\zeta}_+^{(1)}, P_+^{(1)}),
\end{aligned}$$

where F is the transition matrix, Q_f is the process noise covariance, H is the observation matrix, Q_h is the observation noise covariance, $\bar{\zeta}_+^{(1)}$ and $P_+^{(1)}$ are the mean and covariance of the kinematic state of a new object of interest. If each current density of an object of interest is a Gaussian of the form

$$p^{(\xi^{(1)})}(\zeta, 1, \ell) = \mathcal{N}\left(\zeta; \bar{\zeta}^{(\xi^{(1)})}(\ell), P^{(\xi^{(1)})}(\ell)\right) \quad (4.33)$$

then the terms Eq. (4.13), Eq. (4.14), Eq. (4.15) can be computed analytically using the following identities:

$$\begin{aligned}
\int \mathcal{N}(\zeta; \bar{\zeta}, P) \mathcal{N}(\zeta_+; F\zeta, Q_f) d\zeta &= \mathcal{N}(\zeta_+; F\bar{\zeta}, FPF^T + Q_f), \\
\mathcal{N}(\zeta; \bar{\zeta}, P) \mathcal{N}(z; H\zeta, Q_h) &= q(z) \mathcal{N}(\zeta; \bar{\zeta} + K(z - H\bar{\zeta}), [I - KH]P), \\
q(z) &= \mathcal{N}(z; H\bar{\zeta}, HPH^T + Q_h), \\
K &= PH^T [HPH^T + Q_h]^{-1}.
\end{aligned} \quad (4.34)$$

Pseudo-code

For the GLMB joint object-clutter model, propagation of a multi-class GLMB of the form Eq. (4.7) reduces to propagation of Eq. (4.9) for each mode $m \in \{0, 1\}$. The density in Eq. (4.7) is thus completely described by the parameters,

$$\left(w^{(I^{(m)}, \xi^{(m)})}, p^{(\xi^{(m)})} \right) \text{ for } (I^{(m)}, \xi^{(m)}) \in \mathcal{F}(\mathbb{L}^{(m)}) \times \Theta^{(m)} \text{ and } m \in \{0, 1\},$$

which can be enumerated as $\{I^{(m,h)}, \xi^{(m,h)}, w^{(m,h)}, p^{(m,h)}\}$ where $w^{(m,h)} \triangleq w^{(I^{(m,h)}, \xi^{(m,h)})}$ and $p^{(m,h)} \triangleq p^{(\xi^{(m,h)})}$ for $h \in \{1, \dots, H\}$ and $m \in \{0, 1\}$. Consequently the multi-class GLMB of the form Eq. (4.7) can be written as a sum over $h \in \{1, \dots, H\}$, thus obviating the need to store the history vector $\xi^{(m,h)}$ of association maps $\theta^{(m,h)}$. Implementing the filter for the GLMB joint object-clutter model reduces to propagation of the parameter set $\{I^{(m,h)}, w^{(m,h)}, p^{(m,h)}\}$ forward in time for each mode $m \in \{0, 1\}$.

For $m = 1$ (objects of interest), the propagation of the parameter set is given by the procedure in section [Section 4.3.2](#), which adapts the Gibbs sampler solution in [87]. Accordingly, we continue the convention of indexing by h for the cost $\eta_i^{(1,h)}(j) \triangleq \eta_i^{(\xi^{(1,h)})}(j)$ and similarly for the parameters in the expression for the cost in [Eq. \(4.32\)](#). Following the approach in [87], for any valid pairs $(I^{(1)}, \theta^{(1)})$, it is necessary to introduce an auxiliary variable $\gamma = (\gamma_i)_{i=1}^{|I^{(1)}|}$ which is positive 1-1 given by,

$$\gamma_i = \begin{cases} \theta^{(1)}(\ell_i), & \text{if } \ell_i \in I^{(1)} \\ -1 & \text{otherwise.} \end{cases} \quad (4.35)$$

For $m = 0$ (clutter generators), the propagation of the parameter set is given by the by the procedure in section [Section 4.3.2](#), for which the values can be pre-computed offline. Recall that clutter generators are by assumption identical except in label. The propagation of their parameter set further reduces to the weights $w^{(0,h)}$ and the counts $N^{(0,h)} \triangleq |I^{(0,h)}|$ of clutter generators. Thus minimal storage is required for propagating their parameters.

As shown by [Eq. \(4.17\)](#) the weight $\omega^{(h)}$ of any valid component $(I^{(h)}, \xi^{(h)})$ is simply the product $\omega^{(h)} = w^{(0,h)} w^{(1,h)}$, thus for the GLMB joint object-clutter model, the filter implementation propagates the parameter set,

$$\left\{ \omega^{(h)}, I^{(1,h)}, N^{(0,h)}, p^{(1,h)} \right\}_{h=1}^H$$

forward in time to obtain the next parameter set ,

$$\left\{ \omega_+^{(h_+)}, I_+^{(1,h_+)}, N_+^{(0,h_+)}, p_+^{(1,h_+)} \right\}_{h_+=1}^{H_+}$$

as detailed in the following pseudo-code.¹

¹The pseudo-code is presented in the style of [28]. The notation $\{ \}$ is used for a MATLAB cell array of possibly non-unique elements. The ‘‘Unique’’ function is similar to the MATLAB call which outputs non repeated elements of the input, as well as the indices $U_{h,u,v}$ matching each of the outputs to all occurrences in the input. Algorithm 2 for Gibbs sampling is reproduced from [?].

Algorithm 4.1: Propagation of hypotheses with objects of interest and clutter generators

Input : $\{\omega^{(h)}, I^{(1,h)}, N^{(0,h)}, p^{(1,h)}\}_{h=1}^H, H_+^{0,\max}, H_+^{1,\max},$

Input : $\left\{ \left(r_{B,+}^{(1)}(\ell_i), p_{B,+}^{(1)}(\cdot, \ell_i) \right) \right\}_{i=1}^{|\mathbb{B}_+^{(1)}|},$

Input : $P_S^{(1)}, f_+^{(1)}(\cdot|\cdot), P_{D_+}^{(1)}, g_+^{(1)}(\cdot|\cdot), Z_+$

Output: $\{\omega_+^{(h_+)}, I_+^{(1,h_+)}, N_+^{(0,h_+)}, p_+^{(1,h_+)}\}$

- 1 sample counts $[T_+^{(m,h)}]_{h=1}^H$ from a multinomial distribution with parameters $H_+^{m,\max}$ trials and weights $[\omega_+^{(h)}]_{h=1}^H$ for $m = 0, 1$;
- 2 **for** $h \leftarrow 1$ **to** H **do**
- 3 initialize $\gamma^{(1,h,1)}$ (with any positive 1-1 vector);
- 4 $\eta^{(1,h)} := \left[\eta_i^{(1,h)}(j) \right]_{(i,j)=(1,-1)}^{\left(|I^{(1,h)} \cup \mathbb{B}_+^{(1)}|, |Z_+| \right)}$;
- 5 $\{\gamma^{(1,h,u)}\}_{u=1}^{T_+^{(1,h)}} := \text{Unique} \left(\text{Gibbs} \left(\gamma^{(1,h,1)}, T_+^{(1,h)}, \eta^{(1,h)} \right) \right)$;
- 6 **for** $u \leftarrow 1$ **to** $T_+^{(1,h)}$ **do**
- 7 $I_+^{(1,h,u)} := \{\ell_i \in I^{(1,h)} \cup \mathbb{B}_+^{(1)} : \gamma_i^{(1,h,u)} \geq 0\}$;
- 8 $w_+^{(1,h,u)} := \prod_{i=1}^{|I^{(1,h)} \cup \mathbb{B}_+^{(1)}|} \eta_i^{(1,h)} \left(\gamma_i^{(1,h,u)} \right)$;
- 9 $\bar{p}_+^{(1,h,u)}(\cdot, \ell_i) := 1_{\mathbb{L}(1)}(\ell_i) \frac{\langle P_S^{(1)}(\cdot, \ell_i) f_+^{(1)}(\zeta[\cdot, \ell_i]), p^{(1,h)}(\cdot, \ell_i) \rangle}{\langle p^{(1,h)}(\cdot, \ell_i), P_S^{(1)}(\cdot, \ell_i) \rangle} + 1_{\mathbb{B}_+^{(1)}}(\ell_i) p_B^{(1)}(\cdot, \ell_i)$, for each $\ell_i \in I_+^{(1,h,u)}$;
- 10 $p_+^{(1,h,u)}(\cdot, \ell_i) \propto \bar{p}_+^{(1,h,u)}(\cdot, \ell_i) P_{D_+}^{(1)}(\cdot, \ell_i) g_+^{(1)}(z_{\gamma_i^{(1,h,u)}}|\cdot, \ell_i)$, for each $\ell_i \in I_+^{(1,h,u)}$ with $\gamma_i^{(1,h,u)} > 0$;
- 11 $p_+^{(1,h,u)}(\cdot, \ell_i) \propto \bar{p}_+^{(1,h,u)}(\cdot, \ell_i) (1 - P_{D_+}^{(1)}(\cdot, \ell_i))$, for each $\ell_i \in I_+^{(1,h,u)}$ with $\gamma_i^{(1,h,u)} = 0$;
- 12 $|Z_+^{(0)}| := |Z_+| - |\{\ell_i : \gamma_i^{(1,h,u)} > 0\}|$;
- 13 $\{N_+^{(0,h,v)}, w_+^{(0,h,v)}\}_{v=1}^{T_+^{(0,h)}} := \text{SolveClutter}(|Z_+^{(0)}|, N^{(0,h)}, T_+^{(0,h)}, Z_+)$;
- 14 **for** $v \leftarrow 1$ **to** $T_+^{(0,h)}$ **do**
- 15 $w_+^{(h,u,v)} := \omega_+^{(h)} w_+^{(1,h,u)} w_+^{(0,h,v)}$;
- 16 create solutions $S := \left\{ \left(w_+^{(h,u,v)}, I_+^{(1,h,u)}, N_+^{(0,h,v)}, p_+^{(1,h,u)} \right) \right\}_{(h,u,v)=(1,1,1)}^{(H, T_+^{(1,h)}, T_+^{(0,h)})}$;
- 17 $\left(\left\{ \left(w_+^{(h_+)}, I_+^{(1,h_+)}, N_+^{(0,h_+)}, p_+^{(1,h_+)} \right) \right\}_{h_+=1}^{H_+}, \sim, [U_{h,u,v}] \right) := \text{Unique}(S)$;
- 18 $\omega_+^{(h_+)} = \sum_{h,u,v: U_{h,u,v}=h_+} w_+^{(h,u,v)}$ for all h_+ ;
- 19 normalize weights $\left\{ \omega_+^{(h_+)} \right\}_{h_+=1}^{H_+}$;

Algorithm 4.2: Gibbs Sampler

Input : $\gamma^{(1)}, T, \eta = [\eta_i(j)]$
Output: $\gamma^{(1)}, \dots, \gamma^{(T)}$

- 1 $P = \text{size}(\eta, 1);$
- 2 $M = \text{size}(\eta, 2) - 2;$
- 3 **for** $t = 2 : T$ **do**
- 4 **for** $n = 1 : P$ **do**
- 5 $\gamma_n^{(t)} \sim \pi_n(\cdot | \gamma_{1:n-1}^{(t)}, \gamma_{n+1:P}^{(t-1)});$
- 6 $\gamma^{(t)} = [\gamma_1^{(t)}, \dots, \gamma_P^{(t)}];$

Algorithm 4.3: SolveClutter

Input : $|Z_+^{(0)}|, N^{(0)}, T, |Z_+|$
Input : $r_{B,+}^{(0)}, \mathbb{B}_+^{(0)}, P_S^{(0)}, P_{D,+}^{(0)}, V$
Output: $(N_+^{(0,t)}, w_+^{(0,t)})_{t=1}^T$

- 1 find pairs $\{(n_S^{(t)}, n_B^{(t)})\}_{t=1}^T$ giving T -highest costs $w(n_S, n_B)$ subject to
 $0 \leq n_S \leq N^{(0)}, 0 \leq n_B \leq |\mathbb{B}_+^{(0)}|, n_S + n_B \geq |Z_+^{(0)}|$ where,

$$w(n_S, n_B) = [1 - P_S^{(0)}]^{(N^{(0)} - n_S)} [P_S^{(0)}]^{n_S} \times [1 - r_{B,+}^{(0)}]^{|\mathbb{B}_+^{(0)}| - n_B} [r_{B,+}^{(0)}]^{n_B} \times$$

$$[1 - P_{D,+}^{(0)}]^{(n_B + n_S - |Z_+^{(0)}|)} [P_{D,+}^{(0)} V^{-1}]^{|Z_+^{(0)}|};$$
- 2 $N_+^{(0,t)} := n_S^{(t)} + n_B^{(t)}$ for all t ;
- 3 $w_+^{(0,t)} := w(n_S^{(t)}, n_B^{(t)})$ for all t ;

4.3.3 Extension to Unknown Detection Probability

Following the approach in [87], to jointly estimate an unknown detection probability, we augment a variable $a \in [0, 1]$ to the state, i.e. $\mathbf{x} = (\zeta, m, a, \ell)$, so that

$$P_D^{(m)}(\zeta, a, \ell) = a. \quad (4.36)$$

Additionally, in this model

$$\begin{aligned} g^{(m)}(z|\zeta, a, \ell) &= g^{(m)}(z|\zeta, \ell), \\ P_S^{(m)}(\zeta, a, \ell) &= P_S^{(m)}, \\ p_{B,+}^{(1)}(\zeta_+, a_+) &= p_{B,+}^{(1)}(\zeta_+) p_{B,+}^{(1)}(a_+), \end{aligned}$$

and the transition density is given by

$$f_+^{(m)}(\zeta_+, a_+|\zeta, a, \ell) = f_+^{(m)}(\zeta_+, |\zeta, \ell) f_+^{(\Delta)}(a_+|a). \quad (4.37)$$

The Beta distribution is a natural choice for modeling quantities that vary between $[0, 1]$. Following [87] the unknown detection probability is modeled as a Beta distribution $\beta(\cdot, s, t)$ where s and t are positive shape parameters. Consequently, the state density of an object of interest is modeled by a Beta-Gaussian density:

$$p^{(\xi^{(1)})}(\zeta, 1, a, \ell) = \beta(a; s^{(\xi^{(1)})}(\ell), t^{(\xi^{(1)})}(\ell)) \mathcal{N}(\zeta; m^{(\xi^{(1)})}(\ell), P^{(\xi^{(1)})}(\ell)) \quad (4.38)$$

Analytic computation of the terms ??, Eq. (4.14) and Eq. (4.15) can be performed separately for the Gaussian part (which has been given in the previous subsection) and the Beta part using [87]:

$$\beta(a_+; s_+, t_+) = \int \beta(a; s, t) f_+^{(\Delta)}(a_+|a) da \quad (4.39)$$

where,

$$\begin{aligned} s_+ &= \left(\frac{\mu_\beta(1 - \mu_\beta)}{\sigma_{\beta,+}^2} - 1 \right) \mu_\beta, \\ t_+ &= \left(\frac{\mu_\beta(1 - \mu_\beta)}{\sigma_{\beta,+}^2} - 1 \right) (1 - \mu_\beta), \\ \sigma_{\beta,+}^2 &= C_\beta \sigma_\beta^2, \quad C_\beta \geq 1. \\ (1 - a)\beta(a; s, t) &= \frac{B(s, t + 1)}{B(s, t)} \beta(a; s, t + 1), \\ a\beta(a; s, t) &= \frac{B(s + 1, t)}{B(s, t)} \beta(a; s + 1, t), \\ B(s, t) &= \int_0^1 a^{s-1} (1 - a)^{t-1} da. \end{aligned}$$

Note that $\beta(\cdot; s_+, t_+)$ has the same mean μ_β as $\beta(\cdot; s, t)$ but a larger variance than the original σ_β^2 due to the scaling by the factor C_β .

4.3.4 Numerical Studies

Simulations

The following simulation scenario is used to test the proposed robust multi-object filter. The object state vector $[x, y, \dot{x}, \dot{y}]^T$ consists of Cartesian coordinates and the velocities. Objects of interest move according to a constant velocity model, with zero-mean Gaussian process noise of covariance Q_f where $v_f = 5ms^{-1}$ and $T = 1s$.

$$Q_f = v_f^2 \begin{bmatrix} T^4/4 & T^3/2 & 0 & 0 \\ T^3/2 & T^2 & 0 & 0 \\ 0 & 0 & T^4/4 & T^3/2 \\ 0 & 0 & T^3/2 & T^2 \end{bmatrix}$$

The probability of survival for an object of interest is set at 0.99. Objects of interest are born from a labeled multi-Bernoulli distribution with four components of 0.03 birth probability, and birth densities

$$\begin{aligned} & \mathcal{N}(\cdot; [0, 0, 0, 0,] P_\gamma)^T, \mathcal{N}(\cdot; [400, -600, 0, 0] P_\gamma)^T, \\ & \mathcal{N}(\cdot; [-200, 800, 0, 0] P_\gamma)^T, \mathcal{N}(\cdot; [-800, -200, 0, 0] P_\gamma)^T, \\ & \text{where } P_\gamma = \text{diag}([50, 50, 50, 50]). \end{aligned}$$

Objects of interest enter and leave the $[-1000, 1000]m \times [-1000, 1000]m$ observation region at different times reaching a maximum of ten targets. The measurements are the object positions obtained through a sensor located at coordinate $(0, 0)$. Measurement noise is assumed to be distributed Gaussian with zero mean and covariance Q_h where $v_h = 3ms^{-1}$.

$$Q_h = v_h^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

The detection model parameters for all newborn objects of interest are set at $s = 9$ and $t = 1$ resulting in a mean of 0.9 for the detection probability.

At the initial time step, clutter generators are born from a (labeled) multi-Bernoulli distribution with 120 components, each with 0.5 birth probability and uniform birth

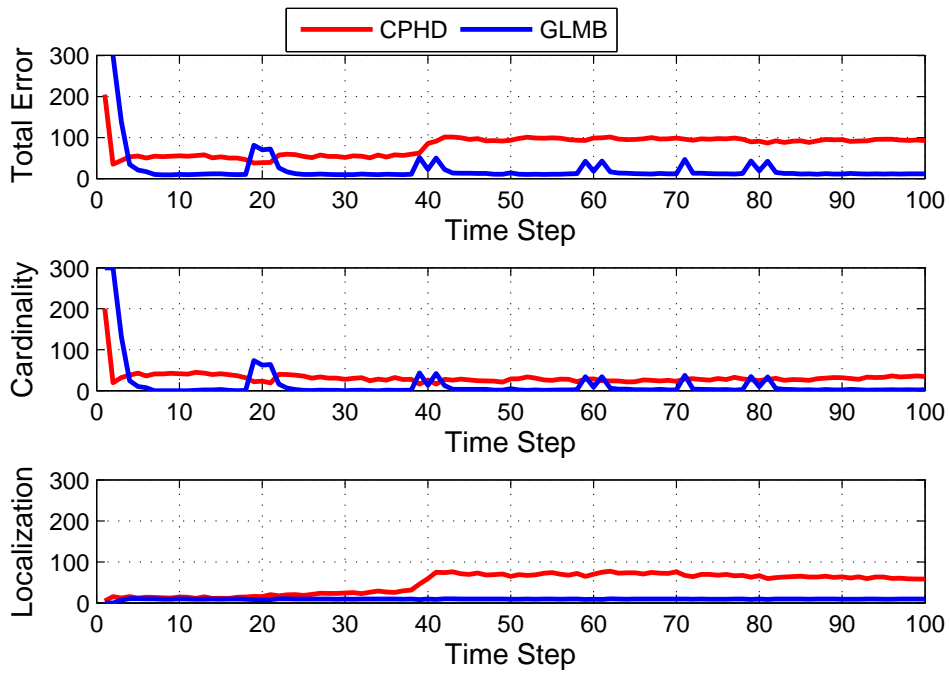
density. At subsequent time steps clutter generators are born from a (labeled) multi-Bernoulli distribution with 30 components, each with 0.5 birth probability and uniform birth density. The probability of survival and probability of detection of the clutter generators are both set at 0.9.

Four scenarios corresponding to different pairings of average (unknown) clutter rate and detection probability (see Table 1) are studied.

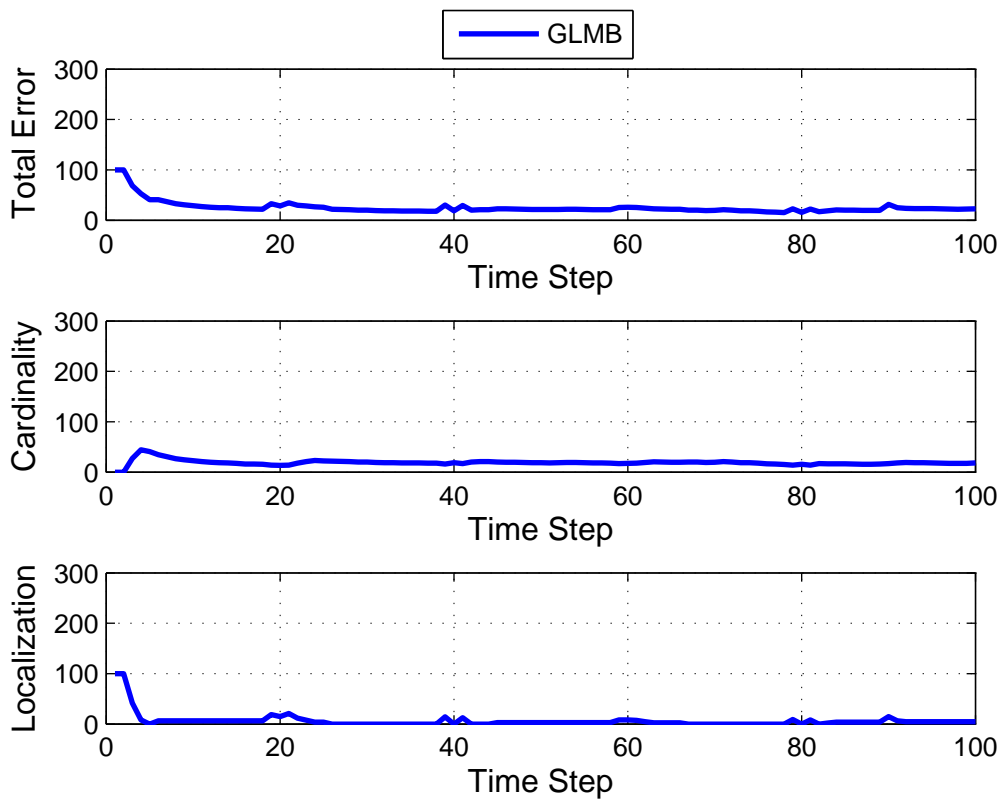
Scenario ID	Clutter Rate	Detection Probability
1	10	0.97
2	10	0.85
3	70	0.97
4	varying between 25-35	0.95

Table 4.1: Simulation Parameters unknown to the filter

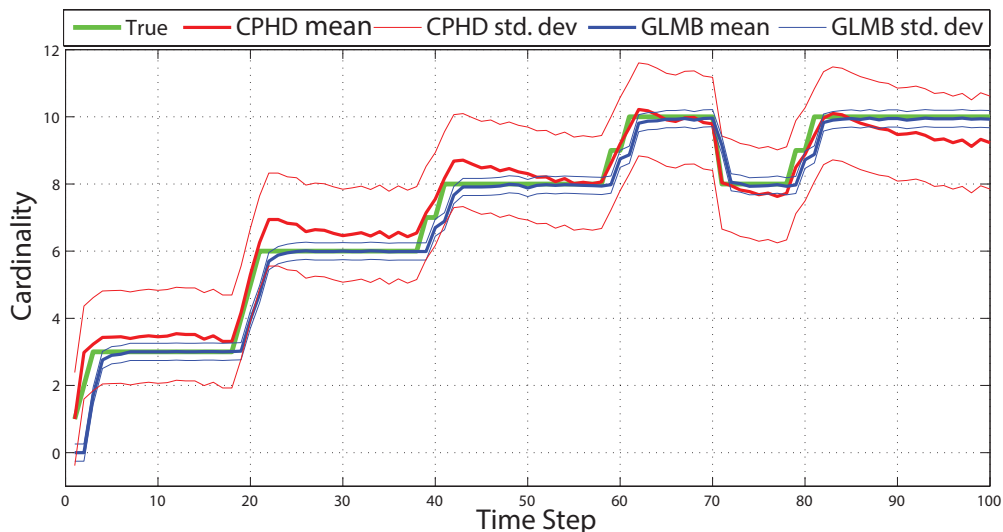
The [Figure 4.3a](#) shows the optimal subpattern assignment metric (OSPA) [103] errors obtained from 100 Monte Carlo runs (OSPA parameters $c = 300m$, $p = 1$) for the proposed GLMB filter in comparison with λ -CPHD [123] filter for scenario 1. Monte Carlo average results for the cardinality of objects of interest count are given in [Figure 4.3b](#). The [Figure 4.3b](#) shows the OSPA(2) [103] errors obtained from 100 Monte Carlo runs (OSPA parameters $c = 300m$, $p = 1$) for the proposed GLMB filter. Note that this metric cannot be calculated for the λ -CPHD as it does not have the concept of tracks. Cardinality statistics for both filters against true cardinality are given in [Figure 4.3c](#). Estimated clutter rates and detection probabilities by the two filters are shown in [Figure 4.3d](#), while estimated tracks for objects of interest taken from a single run is shown in [Figure 4.3e](#). It can be seen that for the given parameters, the GLMB filter performs far better than the λ -CPHD in terms of clutter rate, detection probability and state estimation for objects of interest.



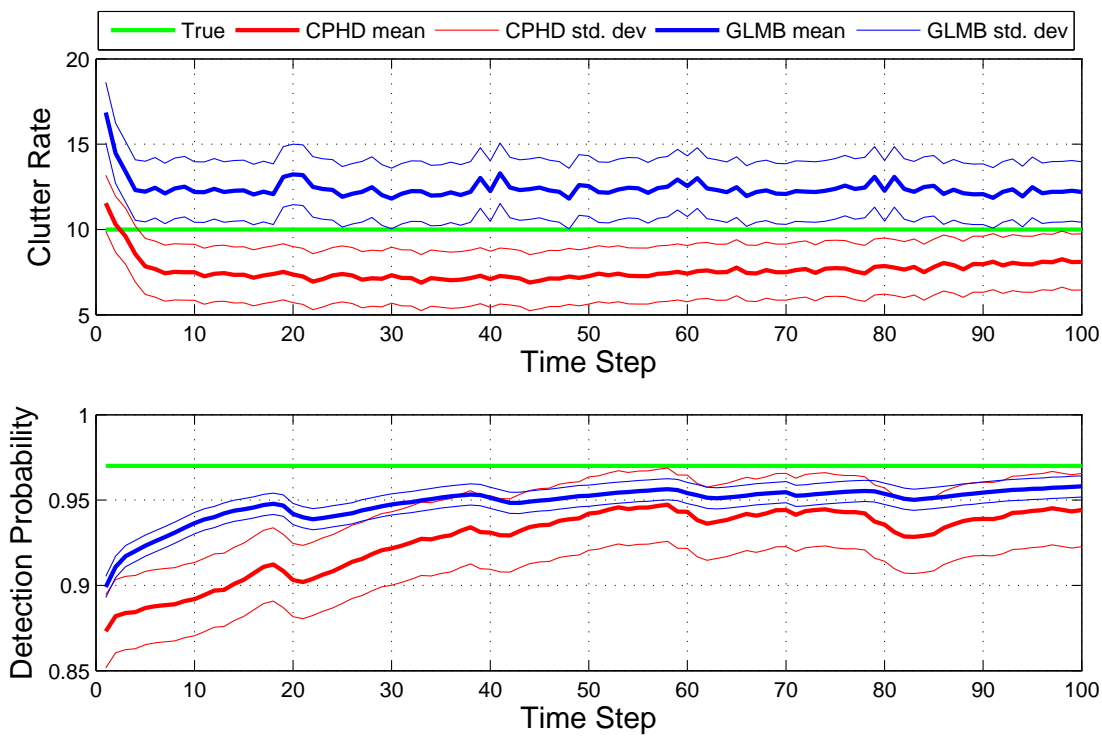
(a) Average OSPA error (parameters $c = 300m$, $p=1$) over 100 Monte Carlo runs. The bumps in the cardinality error for GLMB appear close to time steps where a new birth or a death of an object of interest occurs.



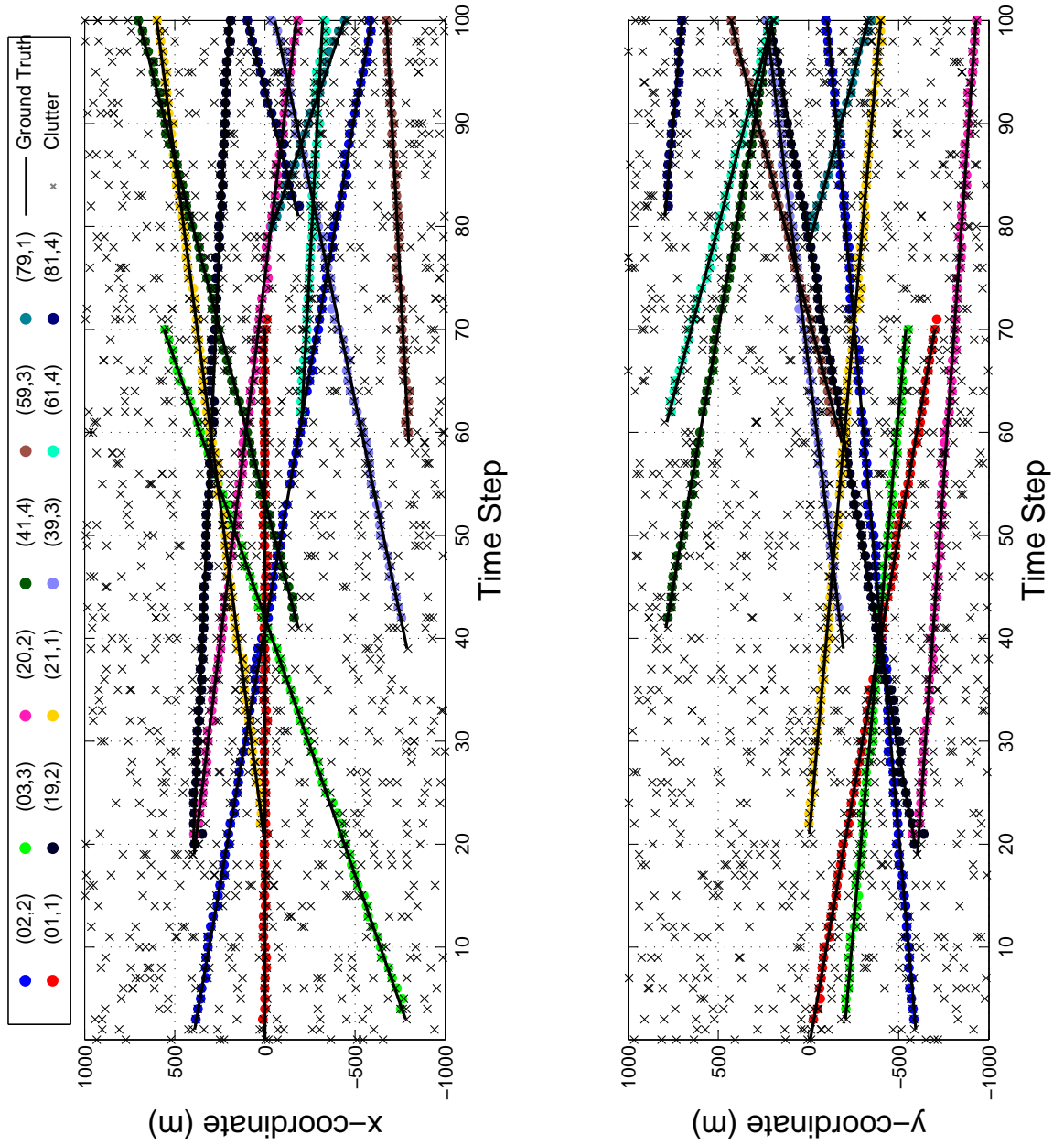
(b) Average OSPA(2) error (parameters $c = 300m$, $p=1$) over 100 Monte Carlo runs.



(c) Cardinality estimations for objects of interest.



(d) Estimated clutter and detection parameters.

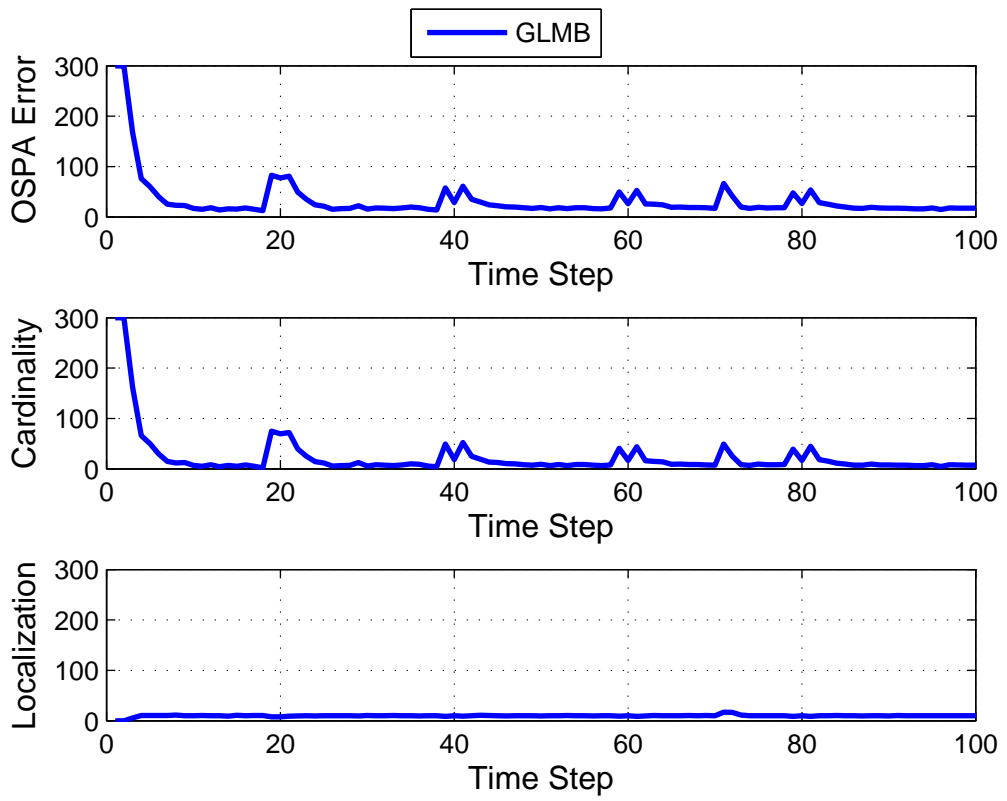


(e) Estimated tracks obtained from a single run.

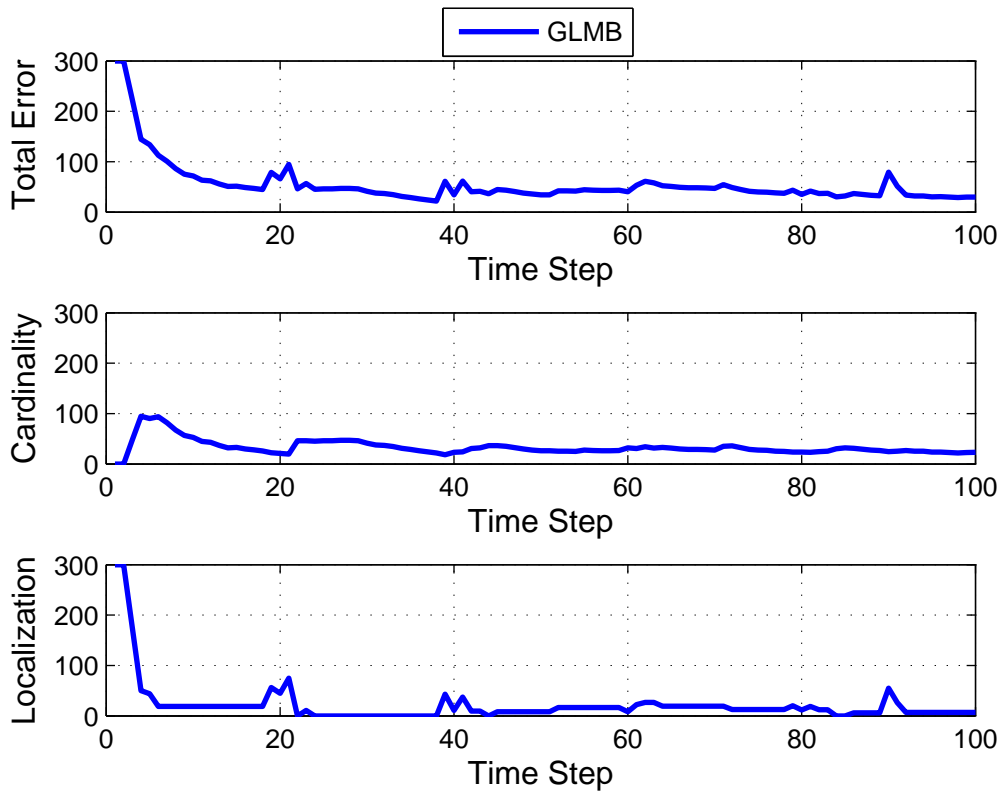
Figure 4.3: Tracking results for scenario 1.

We further investigate the performance of the proposed algorithm by varying the background parameters in scenarios 2 and 3. The average detection probability in scenario 2 is lower than that of scenario 1, while the average clutter rate in scenario 3 is higher than that of scenario 1. Note from [Figure 4.3](#) that the λ -CPHD filter begins to fail in scenario 1. For scenarios 2 and 3, the λ -CPHD completely breaks down. This is due to the low SNR in scenarios 2 and 3 which is too challenging for the λ -CPHD approximation that usually requires a high SNR [123]. On the other hand, the proposed GLMB filter is capable of accurately tracking the objects of interest, as well as estimating the unknown clutter and detection parameters. The OSPA errors and cardinality estimation results for objects of interest over 100 Monte Carlo runs, along with estimates of the clutter rate and detection probabilities, for the robust GLMB filter are given in [Figure 4.4](#) and [Figure 4.5](#). Scenario 4 introduces a time-varying clutter rate for which the results are shown in [Figure 4.6](#).

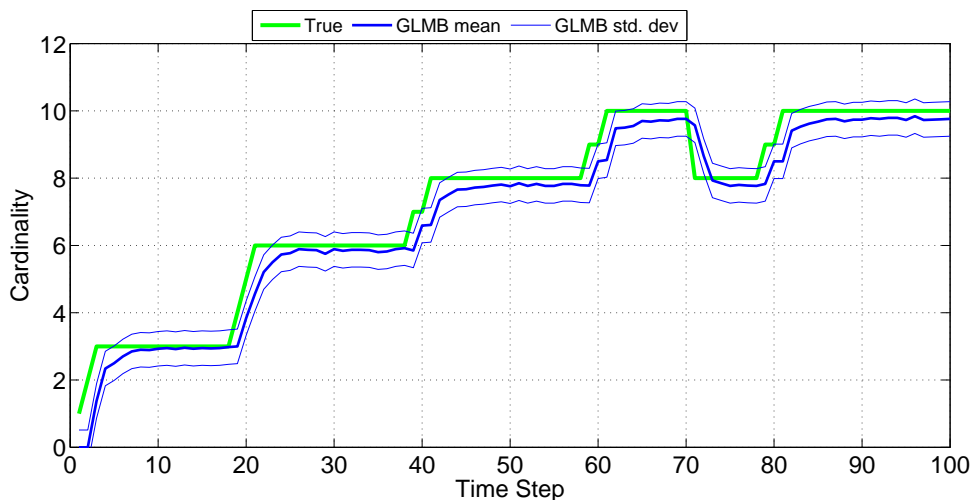
It is clear that the proposed filter outperforms the λ -CPHD and is faster to converge to the true clutter rate. The λ -CPHD has a complexity that is linear in the number of measurements and linear in the number of targets while the proposed GLMB tracker has a complexity that is linear in the number of measurements and quadratic in the number of hypothesised tracks. While the drastic approximation in the λ -CPHD results in a cheaper complexity, it also degrades tracking performance.



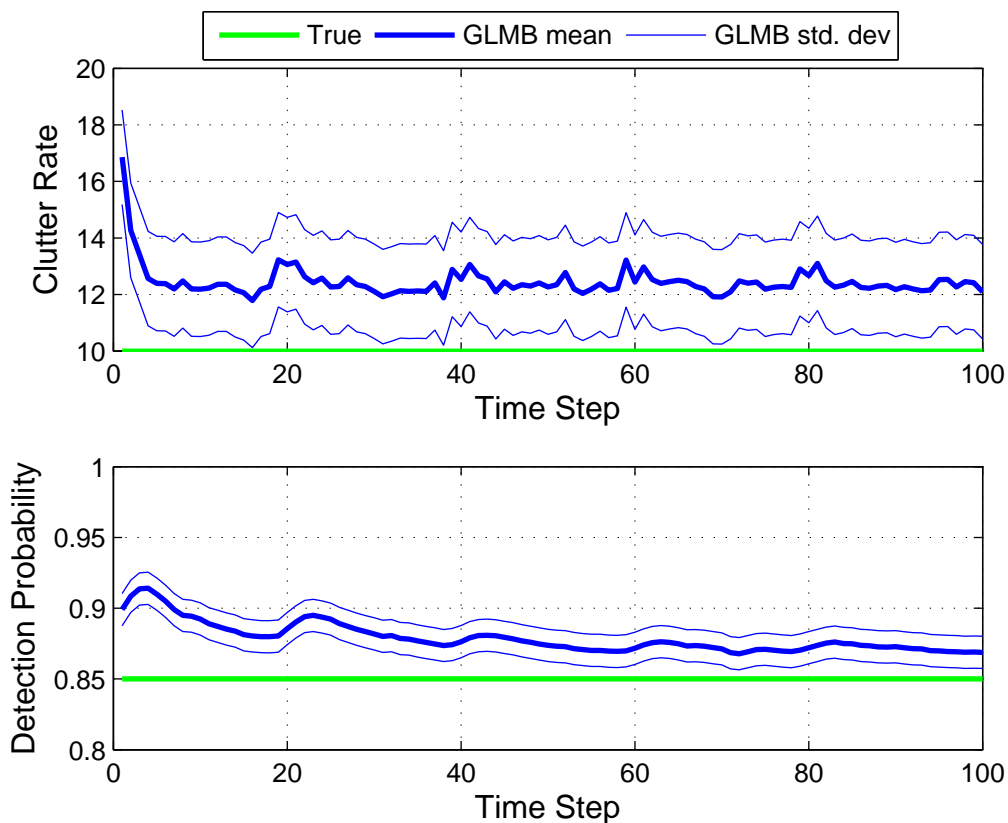
(a) Average OSPA error (parameters $c=300m$, $p=1$) over 100 Monte Carlo runs.



(b) Average OSPA(2) error (parameters $c=300m$, $p=1$) over 100 Monte Carlo runs.

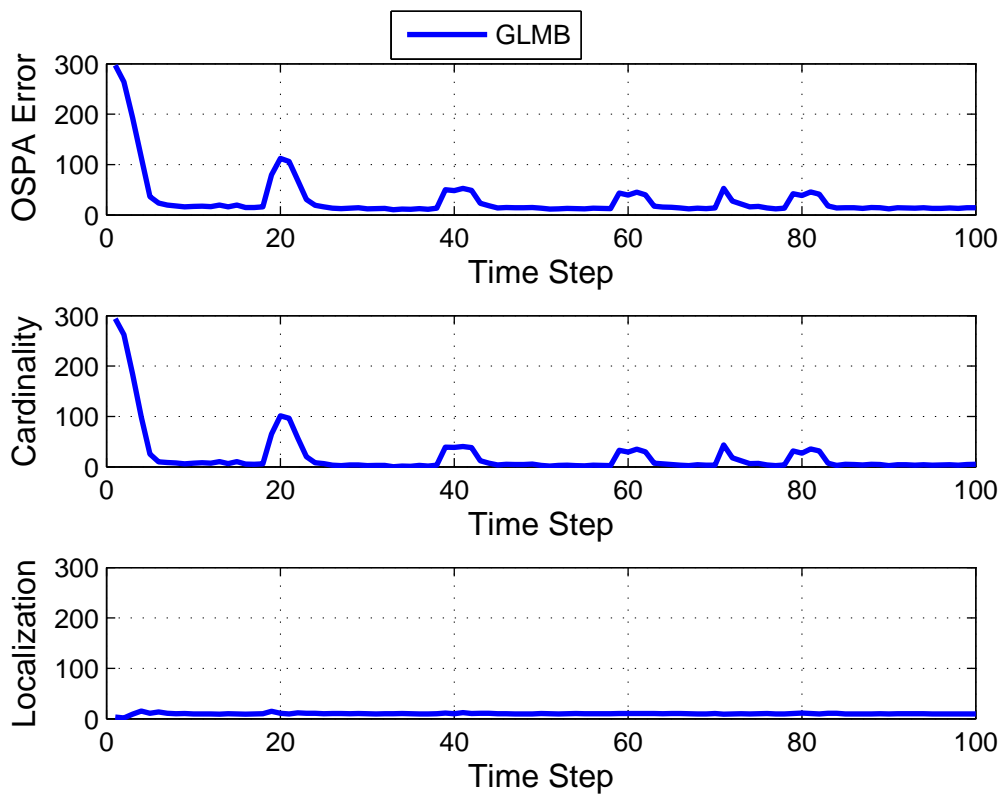


(c) Cardinality estimations for objects of interest.

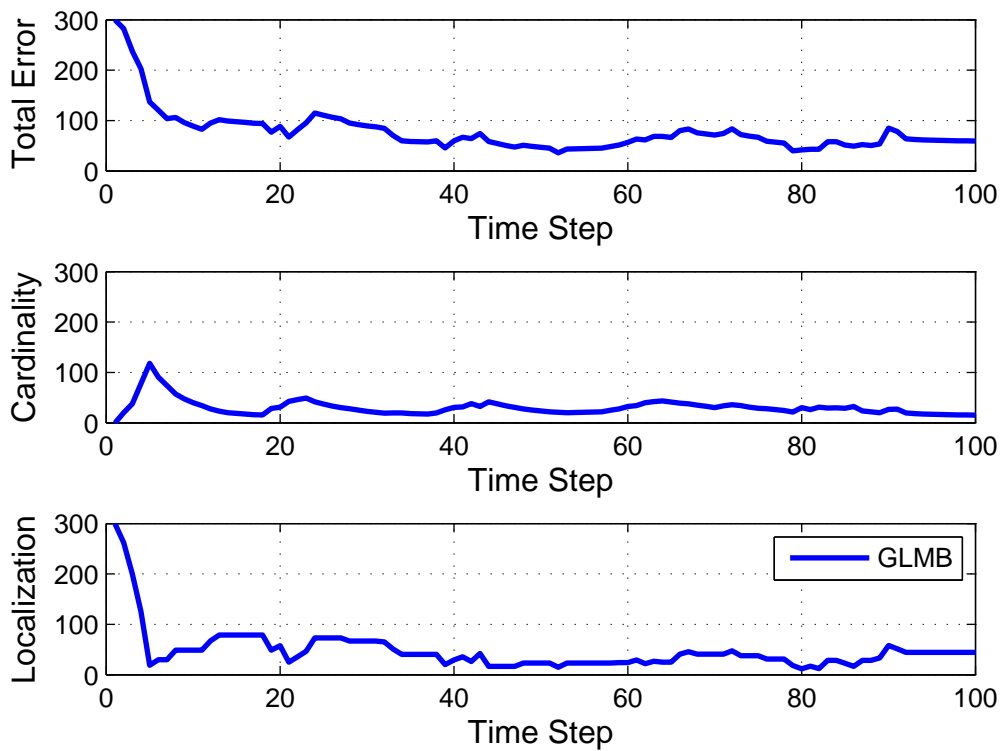


(d) Estimated clutter and detection parameters.

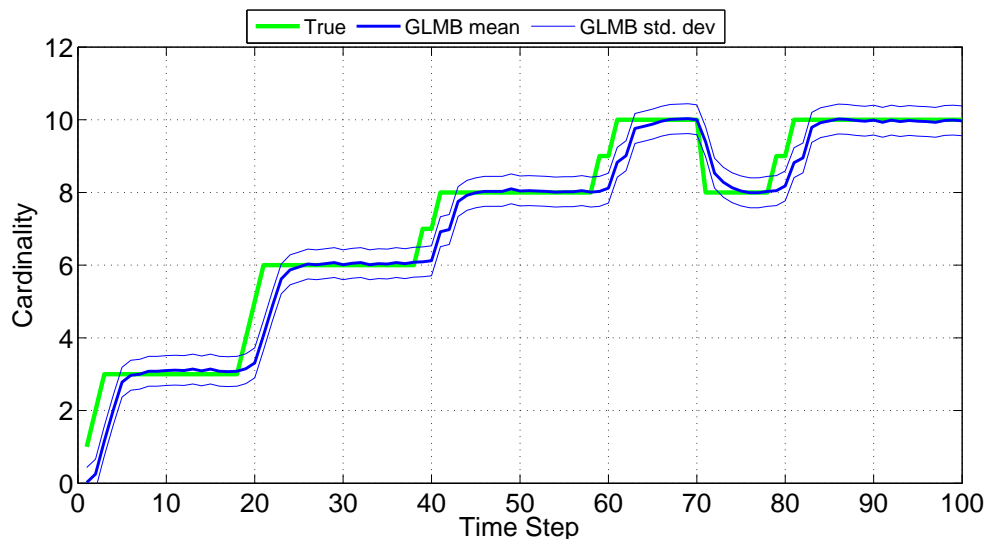
Figure 4.4: Tracking results for scenario 2.



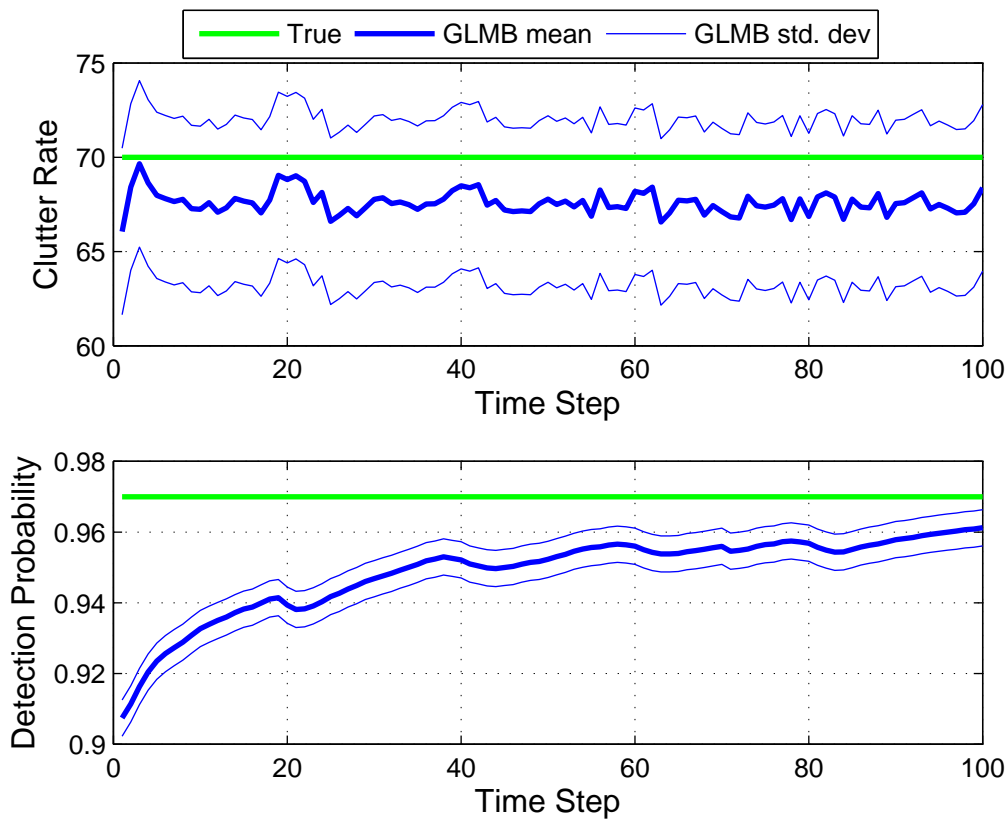
(a) Average OSPA error (parameters $c=300m$, $p=1$) over 100 Monte Carlo runs. The bumps in the cardinality error for GLMB appear close to time steps where a new birth or a death of an object of interest occurs.



(b) Average OSPA(2) error (parameters $c=300m$, $p=1$) over 100 Monte Carlo runs.

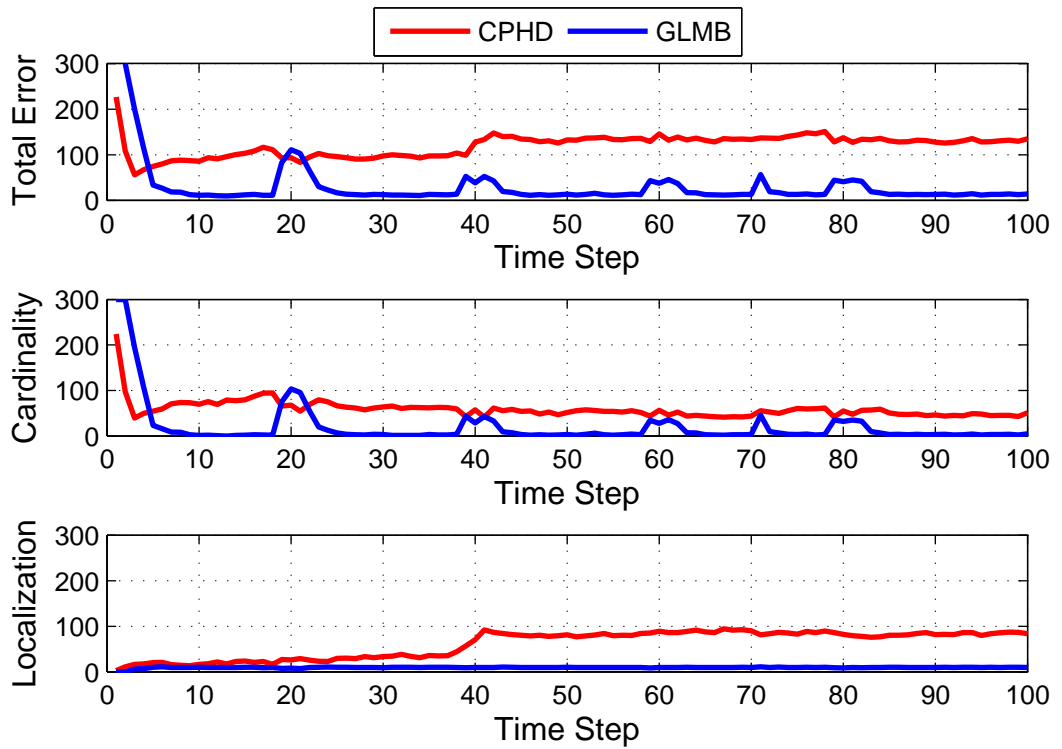


(c) Cardinality estimations for objects of interest.

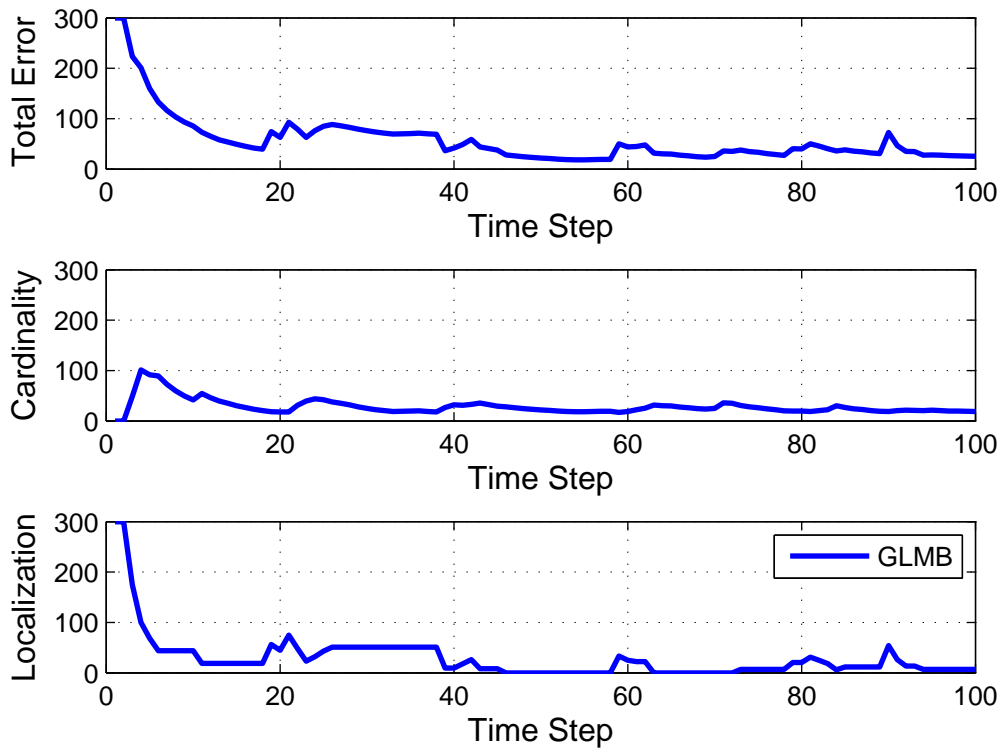


(d) Estimated clutter and detection parameters.

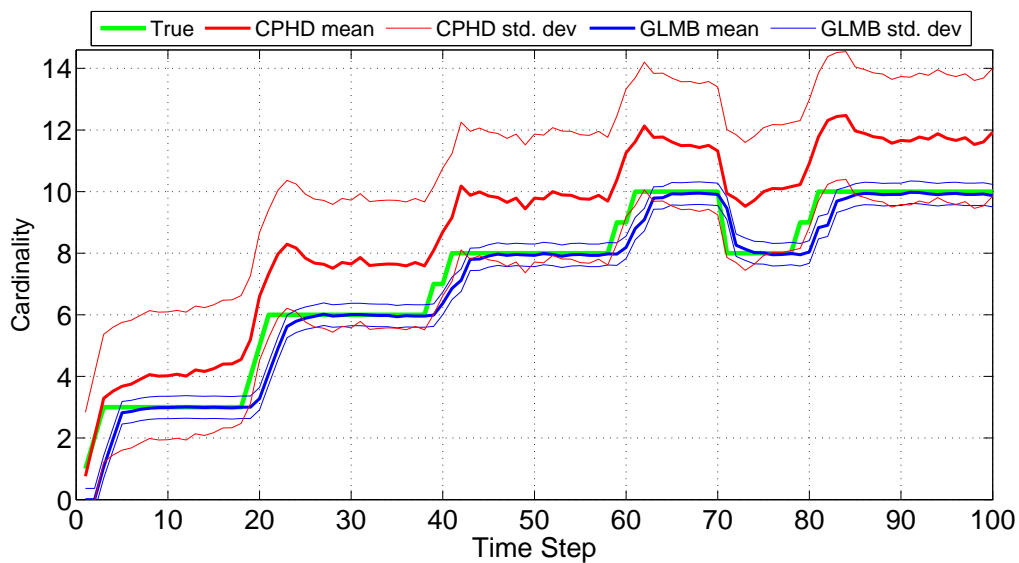
Figure 4.5: Tracking results for scenario 3.



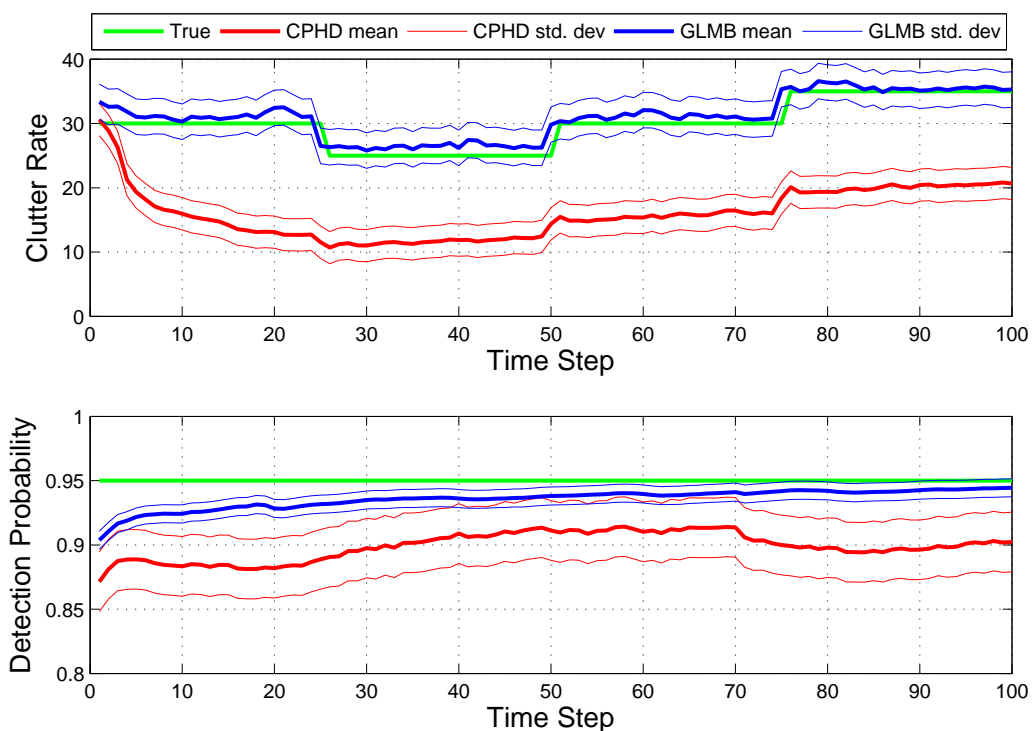
(a) Average OSPA error (parameters $c=300m$, $p=1$) over 100 Monte Carlo runs.



(b) Average OSPA(2) error (parameters $c=300m$, $p=1$) over 100 Monte Carlo runs.



(c) Cardinality estimations for objects of interest.



(d) Estimated clutter and detection parameters.

Figure 4.6: Tracking results for scenario 4.

Video Data

The proposed filter for jointly unknown clutter rate and detection probability is tested on two image sequences: S2.L1 from PETS2009 datasets [128] and KITTI-17 from KITTI datasets [108]. The detections are obtained using the detection algorithm in [109].

Dataset 1:

The state vector consists of the target x, y positions and the velocities in each direction. The process noise is assumed to be distributed from a zero-mean Gaussian with covariance Q_f where $v_f = 2$ pixels. Actual targets are assumed to be born from a labeled multi-Bernoulli distribution with seven components of 0.03 birth probability, and Gaussian birth densities,

$$\begin{aligned} & \mathcal{N}(\cdot, [260; 260; 0; 0]^T, P_\gamma), \mathcal{N}(\cdot, [740; 370; 0; 0]^T, P_\gamma), \\ & \mathcal{N}(\cdot, [10; 200; 0; 0]^T, P_\gamma), \mathcal{N}(\cdot, [280; 80; 0; 0]^T, P_\gamma), \\ & \mathcal{N}(\cdot, [750; 130; 0; 0]^T, P_\gamma), \mathcal{N}(\cdot, [650; 270; 0; 0]^T, P_\gamma), \\ & \mathcal{N}(\cdot, [500; 200; 0; 0]^T, P_\gamma), \\ & \text{where } P_\gamma = \text{diag}([10; 10; 3; 3]). \end{aligned}$$

The observation space is a 756×560 pixel image frame. Actual target measurements contain the x, y positions with measurement noise assumed to be distributed zero-mean Gaussian with covariance Q_h with $v_r = 3$ pixels. Clutter targets are born from a multi-Bernoulli distribution with 30 birth components in the first most time step and 12 components in subsequent time steps each with 0.5 birth probability and uniform birth density. The probability of survival and detection for clutter targets are both set at 0.9.

The [Figure 4.7](#) shows tracking results at frames 20, 40 and 100 respectively. True and estimated clutter cardinality statistics are given in [Figure 4.8](#). From these figures, it can be observed that the filter successfully outputs object tracks and that the estimated clutter rate nearly overlays the true clutter rate.



Figure 4.7: Tracking results for frames 20, 40, 100 in dataset 1.

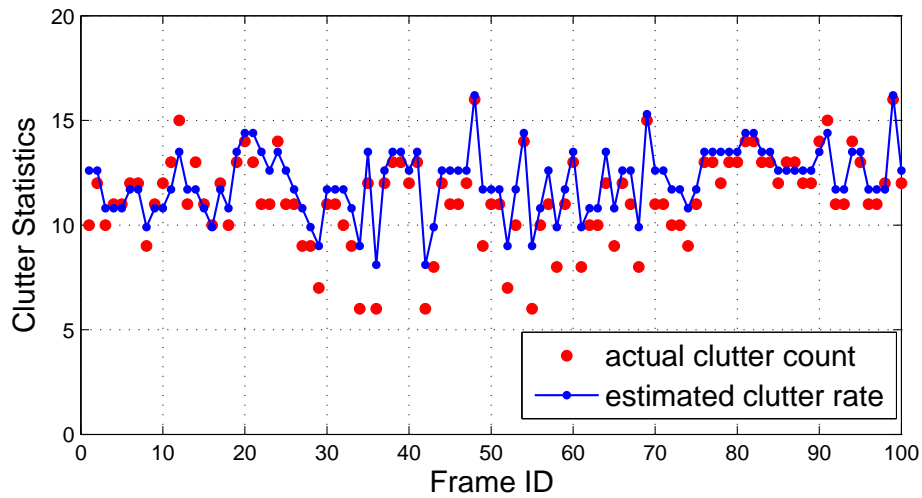


Figure 4.8: Clutter statistics for dataset 1.

Dataset 2: The detection results from this dataset (KITTI17) comprises of a higher number of false measurements than the PETS2009 S2.L1 dataset. The state vector consists of the target x, y positions and the velocities in each direction. The process noise is assumed to be distributed from a zero-mean Gaussian with covariance Q_f where $v_f = 2$ pixels. Actual targets are assumed to be born from a labeled multi-Bernoulli distribution with three components of 0.05 birth probability, and birth densities

$$\mathcal{N}(\cdot, [550; 200; 0; 0]^T, P_\gamma),$$

$$\mathcal{N}(\cdot, [1200; 250; 0; 0]^T, P_\gamma),$$

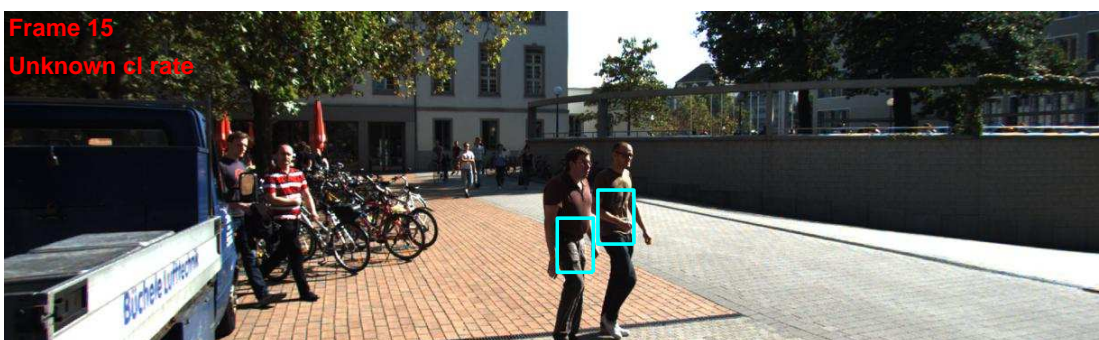
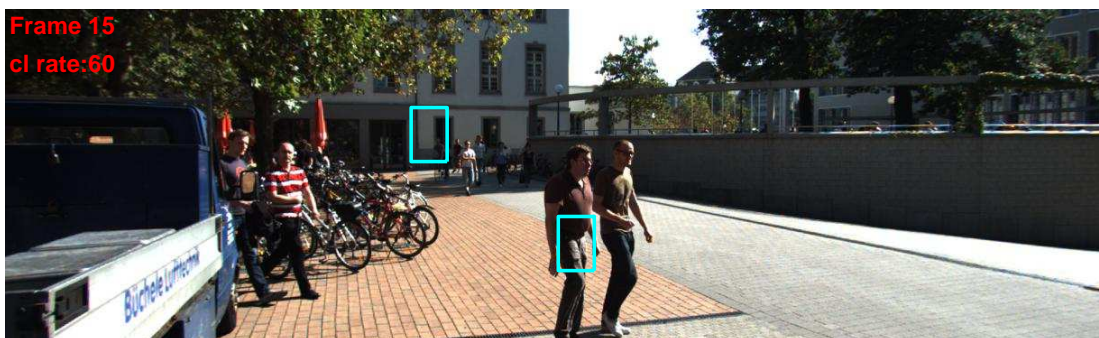
$$\mathcal{N}(\cdot, [500; 250; 0; 0]^T, P_\gamma),$$

$$\text{where } P_\gamma = \text{diag}([10; 10; 1; 1]).$$

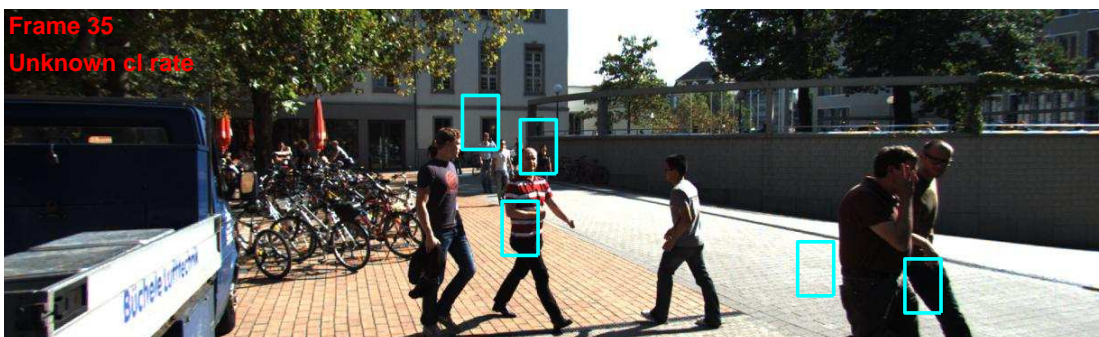
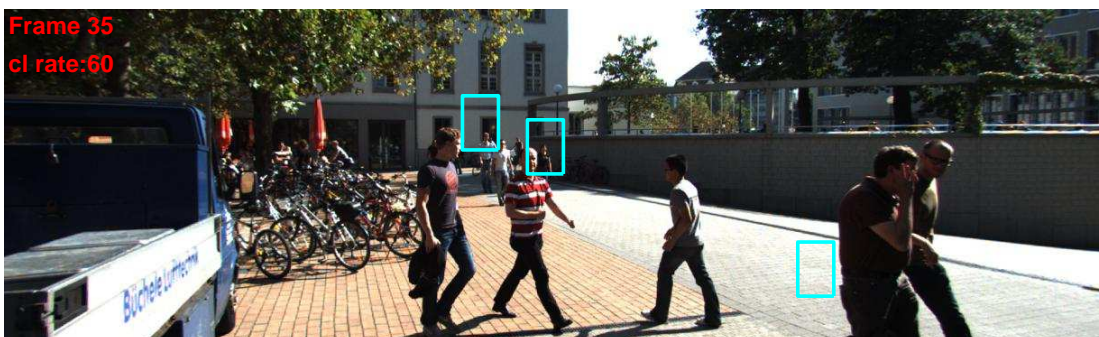
State transition function for actual targets is based on constant velocity model with a 0.99 probability of survival. Process noise is assumed to be distributed from a zero-mean Gaussian with covariance Q_f with $v_f = 2$ pixels per frame. The observation space is a 1220×350 pixel image frame. Actual target measurements contain the x, y positions with measurement noise assumed to be distributed zero-mean Gaussian with covariance Q_r with $v_r = 3$ pixels.

Clutter targets are born from 60 identical and uniformly distributed birth regions in the first most time step and 20 birth regions in the subsequent time steps each with a birth probability of 0.5. The probability of survival and detection for clutter targets are both set at 0.9.

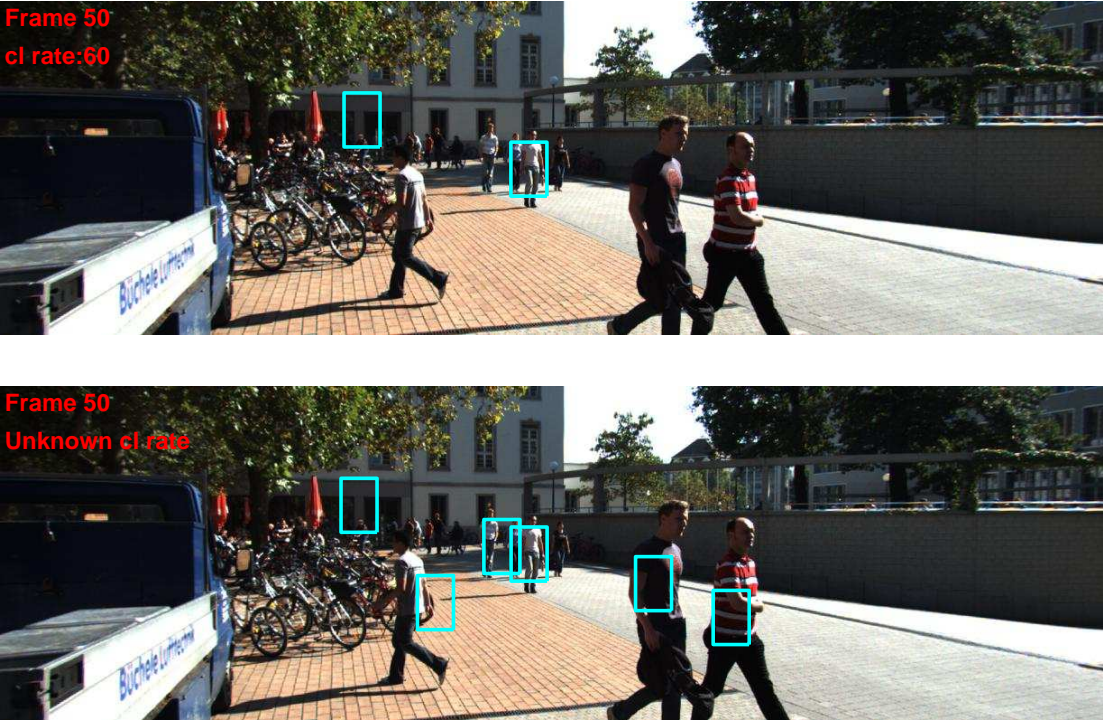
The top frames of [Figure 4.9a](#), [Figure 4.9b](#), [Figure 4.9a](#) shows tracking results for frames 15, 35 and 50 obtained from the standard GLMB filter for the guessed clutter rate of 60. The bottom frames of [Figure 4.9a](#), [Figure 4.9b](#), [Figure 4.9a](#) shows tracking results for the same frames using the proposed filter. When comparing each frame pair it can be noted that some objects that were missed by the standard algorithm with the guessed clutter rate has been picked up by the proposed algorithm. Comparison between true clutter cardinality and estimated clutter cardinality as given in [Figure 4.10](#) demonstrates that the estimated clutter rate is within close range of the true clutter rate. Feeding this estimated clutter rate back to the standard GLMB algorithm [86] could help achieve a performance similar to that of the standard algorithm with known clutter rate.



(a) Tracking results for frame 15 with guessed clutter rate 60 (top) and the proposed filter (bottom)



(b) Tracking results for frame 35 with guessed clutter rate 60 (top) and the proposed filter (bottom)



(c) Tracking results for frame 50 with guessed clutter rate 60 (top) and the proposed filter (bottom)

Figure 4.9: Tracking results for dataset 2.

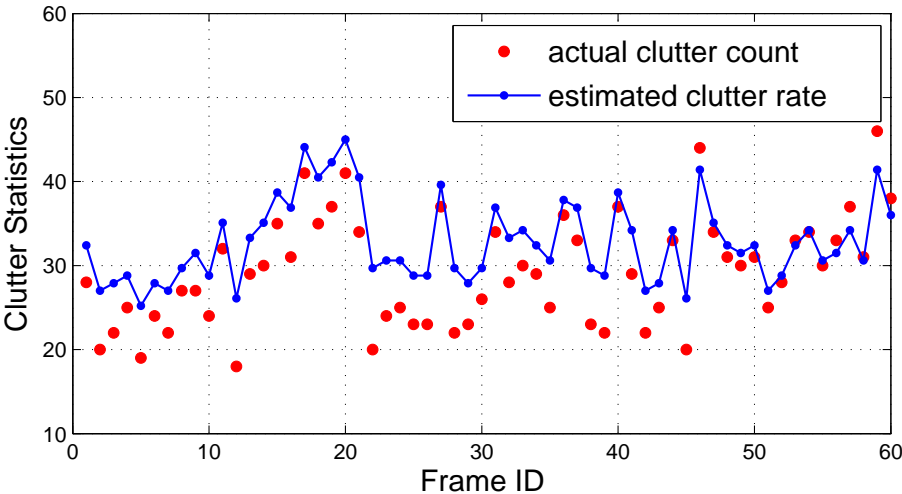


Figure 4.10: Clutter statistics for dataset 2.

TRACKING WITH NO DETECTIONS

In this chapter, an alternative approach to address the problem of tracking with unknown clutter and detection profile, in visual tracking, in way of a GLMB filter that operates on raw video data without using detections (track-before-detect) is presented. An image sequence from the CAVIAR1¹ benchmark dataset is used to validate the utility of this method. The results of this chapter have appeared in the author's conference paper [34].

5.1 Introduction

In multi-object tracking, the raw data received by the sensor is often preprocessed into point measurements referred to as detections before being fed into the filtering algorithm. Even though preprocessing the image data and condensing it into point measurements is efficient in terms of bandwidth, memory and computational cost, significant information loss can occur under low SNR conditions. For example, in tracking via detection with radar, the most straightforward method to obtain measurements is to apply a threshold to the data and to treat those cells that exceed the threshold as point measurements. In low SNR conditions, the threshold needs to be low enough to allow a decent detection probability, which also means a large number of false detections. In instances where the required clutter rate and detection profile of the system are not available, these parameters need to be either estimated from training data or manually tuned. In [Chapter 4](#) it was shown that it's possible to successfully learn the parameters in the absence of the clutter rate and detection profile information.

The notion of track-before-detect was born out of the need to prevent the informa-

¹<http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/>

tion loss that occurs when data is pre-processed to obtain point measurements under low SNR. In the TBD method, the entire raw data set is treated as a single measurement and each object in the multi-object state is considered to have influenced the value of that raw measurement. Consequently, the issues of clutter and missed detections do not arise, since there is no one-to-one pairing of an object to measurement. Since there are no explicit data association problems to be solved, the computational burden due to the exponential explosion caused by data association in presence of a high number of objects and measurements is relaxed.

The TBD technique was first used in 2001 on by [16, 18–20, 22, 25, 56] and later in [129] all based on particle filters. An MHT implementation with a TBD approach named the histogram probabilistic multi-hypothesis tracker (H-PMHT) filter was introduced in [130]. A performance comparison of the classical algorithms using TBD methods in a radar-like simulation is found in [131]. A CBMeMBER filter based TBD algorithm is given in [132] which was subsequently applied to tracking ground targets constrained to move on a road network using raw image observations in [133]. The same was successfully applied to visual tracking in [110] for tracking sports players and in [111] on a surveillance dataset. A TBD particle filter applied to acoustic source tracking is presented in [134], and a TBD technique for sensor networks are presented in [135][136]. Various other TBD strategies for radar, infrared and optical sensors are presented in [137, 138]. From a RFS perspective, a TBD Bernoulli filter application on multiple input multiple output enhanced linear short sar (MELISSA) radar image data is mentioned in [139]. A TBD δ -GLMB filter for radar signals is presented in [140].

The main difficulty in the TBD problem is the formulation of the likelihood function due to its highly non-linear nature. The likelihood function expresses the probability density of the single common (raw) measurement given a multi-object state. In order to enable online filtering, the pre-detection likelihood function (for the raw image observation) should be constructed such that the multi-object distribution under consideration remains conjugate.

Object tracking in computer vision is used in a wide range of diverse application areas such as visual surveillance, medical imaging, augmented reality and traffic control. The omnipresence of surveillance cameras from security sensitive areas such as air-

ports, banks and government agencies to more ordinary venues such as shops, parking lots and private homes has resulted in the emergence of monitoring via analog closed-circuit televisions, digital video recorders and IP network camera systems [141, 142]. The cumbersome and monotonous task of visually monitoring a massive collection of archived or live video footage by human operators is impractical and intelligent systems are needed. Research on visual surveillance using RFS-based multi-object filtering algorithms includes [143–146] that perform filtering on detection based data and [110, 111, 133, 147] that directly process the pre-detection data. None of the aforementioned TBD multi-object filters are trackers for the reason that they are unable to identify the object trajectories (note that the RFS-based filters used in [110] cannot estimate tracks and a separate track management algorithm was required to post-process the filter output to produce tracks). To date, there is no RFS tracker that works directly on pre-detection raw visual data. Such a tracking algorithm based on the δ -GLMB filter [15, 86] that directly performs visual multi-object tracking without detecting the individual objects first is presented in the following.

5.2 Enhancing the Raw Measurement Via Background Subtraction

Background subtraction or foreground detection is the extraction of an image's foreground from the original image. In image processing, *foreground* is recognized as the regions of interest in the image, which in the scope of visual surveillance is the set of objects to be detected. The remaining regions are treated as the background. A moving object would create dissimilar regions in consecutive frames. When the regions of interest are moving objects, the dissimilar regions in the compared frames would belong to the foreground and the rest would belong to the background.

A colour image observation comprises of an array of pixel values. Each pixel value is a 3-dimensional vector which carries the intensity values for the three colour bands red, green and blue. The background model at each pixel location is based on the pixel's recent history. It could be based on the simple average of the recent n frames or a weighted average with more recent frames having higher weights. In this work, the RGB colour values of each pixel are first converted to chromaticity (RGI) space as it is more resilient to ambience light changes and shadows. Thus each pixel value would be a 3-dimensional vector carrying RGI values. RGB to RGI conversion is simply done by normalizing the RGB values to add up to 1 and recording the normalized R value,

G value and the normalizing constant divided by 256 (assuming 8-bit colour coding) as the I value. For the i^{th} pixel of the k^{th} frame this operation is given by the following equations.

$$r_k(i) = \frac{R_k(i)}{R_k(i) + G_k(i) + B_k(i)} \quad (5.1)$$

$$g_k(i) = \frac{G_k(i)}{R_k(i) + G_k(i) + B_k(i)} \quad (5.2)$$

$$I_k(i) = \frac{R_k(i) + G_k(i) + B_k(i)}{256} \quad (5.3)$$

where $R_k(i)$ is the value of the colour band red of the i^{th} pixel in k^{th} frame and $B_k(i)$, $G_k(i)$ are defined likewise for colour bands blue and green. The background model for each frame will be built using a window of the N_0 previous image frames collected in K_0 frame intervals (every K_0^{th} frame). The smaller the interval K_0 is, the more sensitive the background model would be to frame differences. If the frames are produced at a rate of 25 per second and if K_0 is chosen to be 125 (the values used in this experiment), we would not detect any movement that occurred during a period smaller than 5 seconds. Selecting a too small value N_0 will result in the background model being affected by the foreground pixel values (outliers). After the first N_0 frames are collected, each new addition at K_0 intervals will result in the removal of the frame at the bottom.

Let $d(c, i, k)$ denote the value of the colour channel c in RGI space of the i^{th} pixel of frame k

$$d(c, i, k) = \begin{cases} r_k(i), & c = r, \\ g_k(i), & c = g, \\ I_k(i), & c = I, \end{cases} \quad (5.4)$$

and k_s denote the frame number of the s^{th} image frame in the N_0 image frame stack which was collected in K_0 intervals where k is the frame currently being processed.

$$k_s = K_0 \left(\left\lfloor \frac{k}{K_0} \right\rfloor - (s - 1) \right) \quad (5.5)$$

The background model can be interpreted as the probability of each pixel in the current frame belonging to the background given the recent N_0 frames. The Kernel density estimation (KDE) [148] method has been used in this work to build the probability density of the background model. For each colour channel of the i^{th} pixel of a frame under consideration, a histogram is built out of the corresponding i^{th} pixel values from the current N_0 frame window. Next, the histogram is smoothed with a Gaussian kernel. The kernel bandwidth is a free parameter which strongly influences the result of the smoothing. Moving objects change the variation of pixel intensities and therefore different pixels would have differently shaped histograms. Therefore the bandwidth of the Gaussian kernel is separately calculated for each histogram (per pixel per colour channel) since it is not justifiable to use the same bandwidth to smooth differently shaped histograms. In this experiment, the *median absolute deviation* over the N_0 pixel samples as computed by Eq. (5.6) is used as the kernel bandwidth.

For colour channel c of the i^{th} pixel of frame k the kernel bandwidth $\sigma(c, i, k)$ is calculated by (k_s denotes the median frame of the N_0 frame stack),

$$\sigma(c, i, k) = \underset{s}{median} \quad |d(c, i, k) - d(c, i, k_s)| \quad (5.6)$$

Therefore the likelihood of i^{th} pixel of frame k belonging to the background (i.e., the probability distribution of the background model) is,

$$p_{bg}(i, k) = \frac{1}{N_0} \sum_{s=0}^{N_0-1} \prod_{c \in \{r, g, I\}} \mathcal{N} (d(c, i, k); d(c, i, k_s), \sigma(c, i, k)^2) \quad (5.7)$$

Equivalently Eq. (5.7) can be written as,

$$y(i, k) = \frac{1}{N_0} \sum_{s=0}^{N_0-1} \exp \left[- \sum_{c \in \{r, g, I\}} \frac{[d(c, i, k) - d(c, i, k_s)]^2}{2\sigma(c, i, k)^2} \right] \quad (5.8)$$

The 3-dimensional pixel array is now reduced to a 1- dimensional pixel array and it's values $p_{bg}(i, k)$ lie in the interval $[0, 1]$. This essentially represents a grayscale image. This image may be affected by minute background noise such as the salt and pepper noise, which can be reduced by morphological erosion and dilation on the $p_{bg}(\cdot, \cdot)$

values using a small structural element. Finally, the output image is processed by a low pass filter (0.5 selected as the threshold in this experiment). This sets lighter shades to all white and gives more prominence to the variations in darker shades.

$$p_{bg}(i, k) = (\min y(i, k), \text{threshold})$$

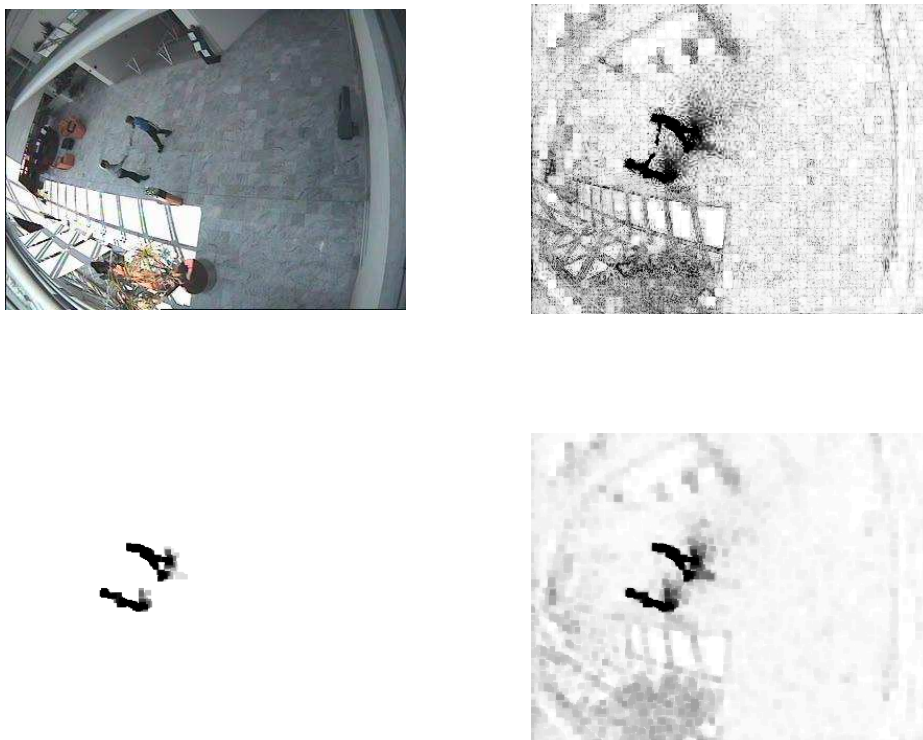


Figure 5.1: Stages of enhancing the raw measurement

5.3 The Multi-object likelihood function

Following [111], the following measurement model is adapted. An object appearing in state \mathbf{x} in the final background subtracted image, illuminates a set of pixels denoted by $T(\mathbf{x})$. It is clear that the intensity of the illumination is higher for pixels nearer to the object and less for pixels further away. Let us denote this intensity function by $g_F(\mathbf{x})$. Any pixel that is not illuminated by the object would have its intensity distributed according to a different distribution which we denote by $g_B(\mathbf{x})$. Therefore probability

density of the intensity of a particular pixel i is given by,

$$p(y(i, k) | \mathbf{x}) = \begin{cases} g_F(\mathbf{x}) & \text{if } i \in T(\mathbf{x}) \\ g_B(\mathbf{x}) & \text{otherwise} \end{cases} \quad (5.9)$$

We consider only non-overlapping objects considering scenarios similar to the case study that has been used, where a camera propped up from an elevation is monitoring the objects moving beneath. This assumption results in a likelihood function that leads to a numerically tractable algorithm.

Following [111] the densities of $g_F(\mathbf{x})$ and $g_B(\mathbf{x})$ are of the form:

$$g_F(\bar{\mathbf{y}}_F) = \zeta_F \exp\left(\frac{-\bar{\mathbf{y}}_F}{\delta_F}\right), \quad (5.10)$$

$$g_B(\bar{\mathbf{y}}_B) = \zeta_B \exp\left(\frac{\bar{\mathbf{y}}_B}{\delta_B}\right), \quad (5.11)$$

where ζ_F, ζ_B are normalizing constants and δ_F and δ_B determine the spread rates of the foreground and background intensities. In this experiment δ_B is chosen to be several times smaller than δ_F under the assumption that the background intensity would generally be unaffected by an object and will remain constant unless quite close to the object whereas the foreground intensity caused by an object would have a much varying and spreading intensity function. Let $\bar{y}_B(\mathbf{X})$ denote the average of pixel intensities of the image constructed by replacing all regions illuminated by the objects (i.e., $\sum_{\mathbf{x} \in \mathbf{X}} T(\mathbf{x})$) with background pixel value 1. The resulting image could be considered as entirely background. Let m be the total number of pixels. Then we have,

$$\bar{y}_B(\mathbf{X}) = \frac{1}{m} \left(\sum_{i=0}^m y(i, k) + \sum_{\mathbf{x} \in \mathbf{X}} \sum_{i \in T(\mathbf{x})} (1 - y(i, k)) \right). \quad (5.12)$$

Ideally the probability that this image is background should be 1. Substituting $\bar{y}_B(\mathbf{X})$ in Eq. (5.11) gives,

$$\begin{aligned}
g_B(\bar{y}_B(\mathbf{X})) &= \zeta_B \exp \frac{(\sum_{i=0}^m y(i, k) + \sum_{\mathbf{x} \in \mathbf{X}} \sum_{i \in T(\mathbf{x})} (1 - y(i, k)))}{m \delta_B} \\
&= \zeta_B \exp \frac{(\sum_{i=0}^m y(i, k))}{m \delta_B} \prod_{\mathbf{x} \in \mathbf{X}} \exp \left(\frac{|T(\mathbf{x}) - \sum_{i \in T(\mathbf{x})} (1 - y(i, k))|}{m \delta_B} \right) \\
&= \zeta_B \exp \frac{(\sum_{i=0}^m y(i, k))}{m \delta_B} \prod_{\mathbf{x} \in \mathbf{X}} \exp \left(\frac{|T(\mathbf{x}) - (1 - \bar{y}(\mathbf{x}))|}{m \delta_B} \right) \quad (5.13)
\end{aligned}$$

where $|T(\mathbf{x})|$ denotes the number of pixels in the object region specified by the state \mathbf{x} . If the above background likelihood value turns out to be very small, it is probably due to some object(s) not being included in the set \mathbf{X} . Such missing objects contribute to the sum of pixel values and thereby reduce the value of the average $\bar{y}_B(\mathbf{X})$.

By Eq. (5.10) the likelihood of a certain object \mathbf{x} illuminating the set of pixels $T(\mathbf{x})$ is given by,

$$g_F(\bar{y}(\mathbf{x})) = \zeta_F \exp \frac{(-\bar{y}(\mathbf{x}))}{\delta_F} \quad (5.14)$$

The likelihood of the background subtracted image measurement given the object set \mathbf{X} is the product of the foreground and background terms formulated as in Eq. (5.13) and Eq. (5.14).

$$g(y|\mathbf{X}) = g_B(\bar{y}_B(\mathbf{X})) \prod_{\mathbf{x} \in \mathbf{X}} g_F(\bar{y}(\mathbf{x})). \quad (5.15)$$

$$\begin{aligned}
&= \underbrace{\zeta_B \exp \frac{(\sum_{i=0}^m y(i, k))}{m \delta_B}}_{\text{independent of } \mathbf{X}} \underbrace{\prod_{\mathbf{x} \in \mathbf{X}} \exp \left(\frac{|T(\mathbf{x}) - (1 - \bar{y}(\mathbf{x}))|}{m \delta_B} \right) \zeta_F \exp \frac{(-\bar{y}(\mathbf{x}))}{\delta_F}}_{\text{dependent on } \mathbf{x}} \quad (5.16)
\end{aligned}$$

Note that this likelihood function is separable.

5.4 Track-before-detect GLMB filter

A GLMB filter which remains a conjugate prior under a separable likelihood function is proposed in [140] for TBD applications with emphasis on radar images. Since the likelihood function discussed in the previous section is separable, this GLMB filter can be directly applied. The prediction step is the same as the standard GLMB prediction. Since the observation consists of a single raw measurement, false measurements or missed detections does not come into play in the update. The update step is as follows.

Let the prior GLMB density be given by,

$$\pi(\mathbf{X}) = \Delta(\mathbf{X}) \sum_{(I,\xi) \in \mathcal{F}(\mathbb{L}) \times \Xi} \omega^{(I,\xi)} \delta_I(\mathcal{L}(\mathbf{X})) \left[p^{(\xi)} \right]^{\mathbf{X}}. \quad (5.17)$$

The posterior density under a separable likelihood with only a single observation z is given by the following equation.

$$\pi(\mathbf{X}|z) = \Delta(\mathbf{X}) \sum_{(I,\xi) \in \mathcal{F}(\mathbb{L}) \times \Xi} \omega_z^{(I,\xi)} \delta_I(\mathcal{L}(\mathbf{X})) \left[p^{(\xi)}(\cdot|z) \right]^{\mathbf{X}} \quad (5.18)$$

where,

$$\begin{aligned} \omega_z^{(I,\xi)} &\propto \omega^{(I,\xi)} [\bar{\psi}_z]^I, \\ p^{(\xi)}(x, \ell|z) &= \frac{p^{(\xi)}(x, \ell) \psi_z(x, \ell)}{\bar{\psi}_z(\ell)}, \\ \bar{\psi}_z(\ell) &= \left\langle p^{(\xi)}(\cdot, \ell), \psi_z(\cdot, \ell) \right\rangle, \\ \psi_z(x, \ell) &= \exp \left(\frac{|T(x, \ell) - (1 - \bar{y}(x, \ell))|}{m \delta_B} \right) \zeta_F \exp \frac{(-\bar{y}(x, \ell))}{\delta_F}. \end{aligned}$$

Note that term $\psi_z(x, \ell)$ is the term $g_F(\bar{y}(\mathbf{x}))$ that is dependent of \mathbf{x} in Eq. (5.16).

For state estimation at each time step, the maximum a posteriori cardinality estimate is computed and the hypothesis with the highest weight component of that cardinality is extracted as the multi-object state.

Note that the growth in the number of hypotheses occurs only during the prediction phase. This is due to the reason, that we have only a single measurement which contains contributions from all the objects in the scenario.

5.4.1 Implementation Issues

A tricky issue in this model is getting multiple estimates for the same object when it is near birth locations. In a particular hypothesis if there happened to be a surviving object that is promoted by the foreground image with a likelihood value over 1 and there is also a newborn object nearby which is similarly promoted by the same foreground image (since there is no data association), that hypothesis will gain a higher weight

than the hypotheses with just one of those objects and surely be selected as the best hypothesis. Now there are two estimates for the same object and this number of estimates will continue to multiply until the object gets sufficiently far from the birth Gaussian. Likewise, when the object is exiting through a doorway the new births at that doorway will get promoted by the likelihood of the approaching object and increase the number of estimates for that single object.

A simple solution for this is to merge the estimates that overlap in a hypothesis. Merging changes the cardinality of a hypothesis and care should be taken not to destroy the cardinality distribution as a result. Further, merging combined with recalculating cardinality distribution is computationally expensive. Thus a simpler scheme is employed to circumvent this issue. We mark a Gaussian space around each surviving object which is essentially its immediate neighbourhood. See [Figure 5.2](#). For each new birth in a particular hypothesis, the probability of the newborn being in the neighbourhood of each surviving object divided by a normalized constant is calculated. (The constant is selected such that it returns a value of 0.5 if the new birth lies at the center of the Gaussian neighbourhood). The closest surviving object gives the highest probability and this probability is deducted from the probability of that hypothesis as a penalty. (If the object neighbourhood did not overlap with any new birth the penalty would be zero). In this method we are adapting the birth model not by imposing restrictions on the birth regions, but by penalizing a hypothesis by reducing its weight if it contains a new birth that is too close to a survival in that set.

5.4.2 Experiment results

An image sequence showing people entering and exiting a lobby area through several doors has been used as the test dataset. The state vector consists of the x position, y position, width and height of the object. All objects are assumed to be moving according to the random walk model. The probability of survival is a state independent value of 0.98. Objects are assumed to be born at birth locations which are the five entryways to the lobby. The probability of an object being born is set to 0.005 for each location which is Gaussian distributed with small variances in x and y directions measured in pixels. For doorways nearer to the camera a much larger variance is set (e.g., $diag(100, 25)$) and for doorways at the far end of the picture smaller variances are set (e.g., $diag(25, 9)$). The rationale behind this is that doorways far from the camera ap-

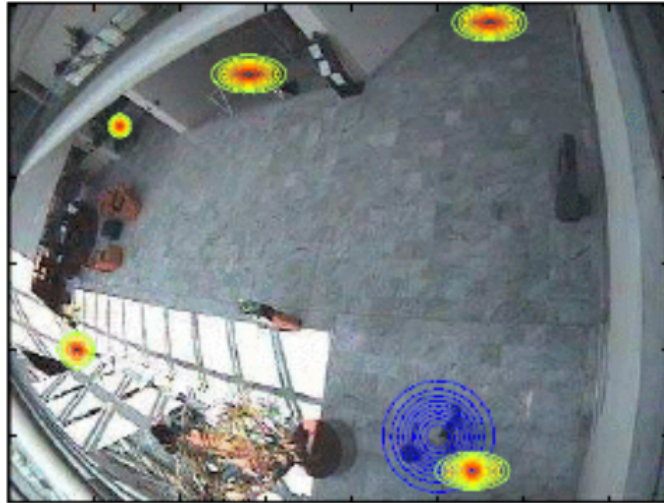


Figure 5.2: Gaussian birth locations (in yellow and orange) and an object moving with his Gaussian neighbourhood (in blue). Hypotheses containing new births in the blue region are penalized.

pear smaller in the video (perspective effect) and take a fewer number of pixels to cover. The variance in x direction is higher as it should cover the width of the entrance. The number of particles for each object in a hypothesis is set to 1000. Resampling is carried out after each update step. The [Figure 5.4](#) shows several frames from the test data set where two objects are tracked successfully along with their trajectories and labels. Approximate midpoint of objects were manually recorded as ground truth and the OSPA [103] results are given in [Figure 5.5](#). At the beginning of the sequence (upto frame 230) there are no objects recorded on the frames and the tracking algorithms accordingly does not indicate any false positives. The same applies to the frames post frames 473 where there are no actual objects recorded. There is a delay in confirming the track for the firstly appearing object since the information provided by the enhanced measurements during that period is insufficient to do so. Eg., See frame 191 of [Figure 5.3](#).

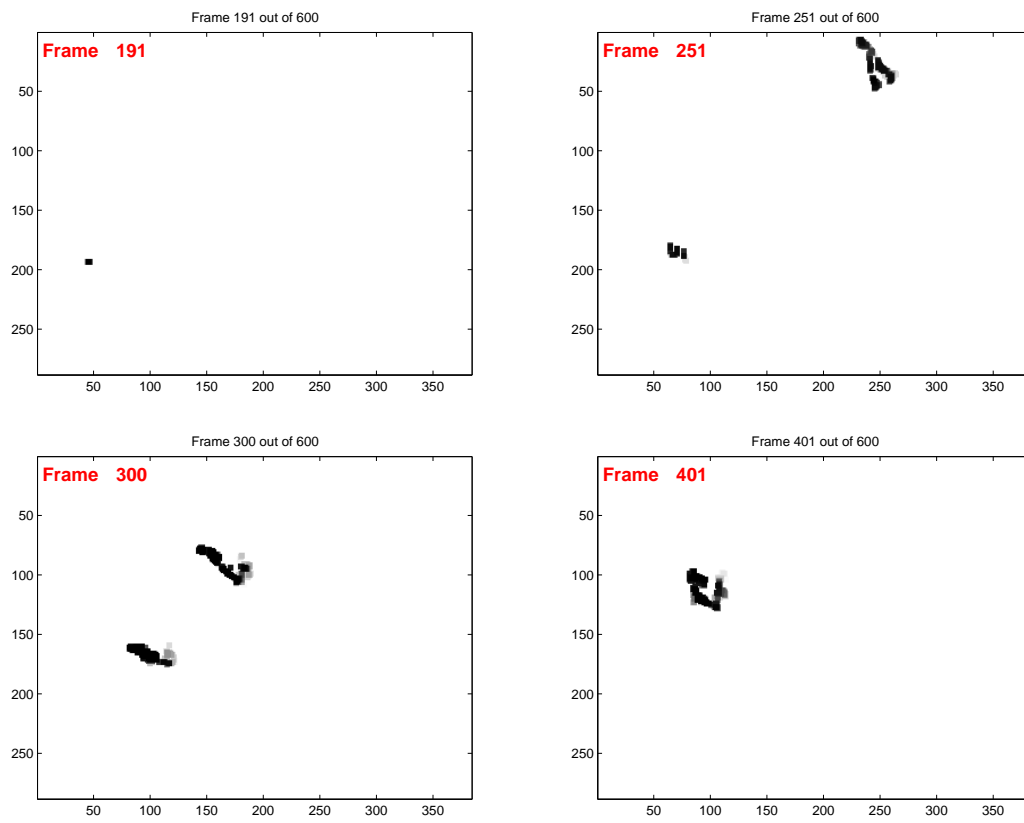


Figure 5.3: Background subtracted image frames

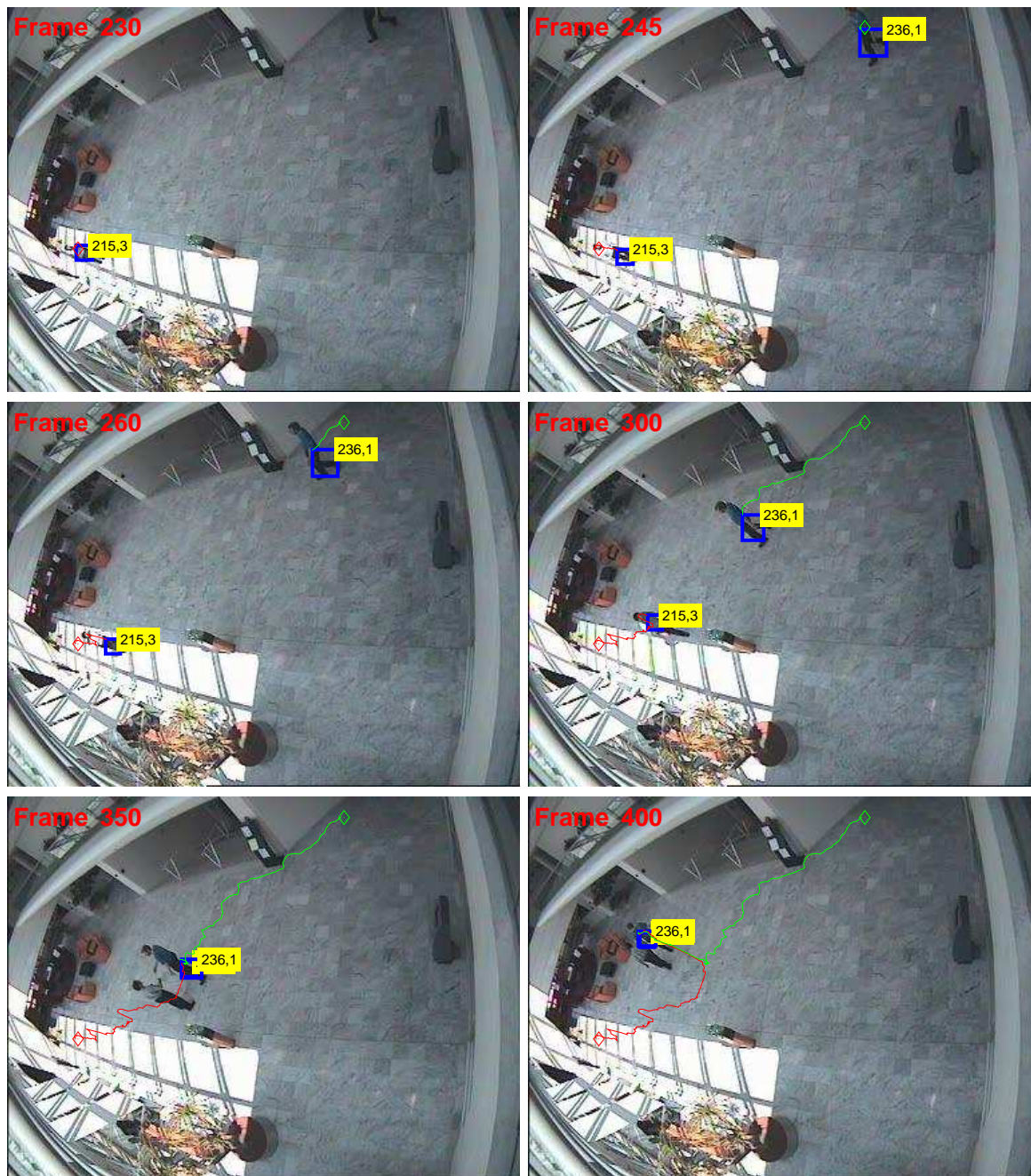


Figure 5.4: Tracking results for frames 230,245,260,300,350,400. : Tracked objects are marked by blue rectangles with tracks in different colours and labels in yellow tags.

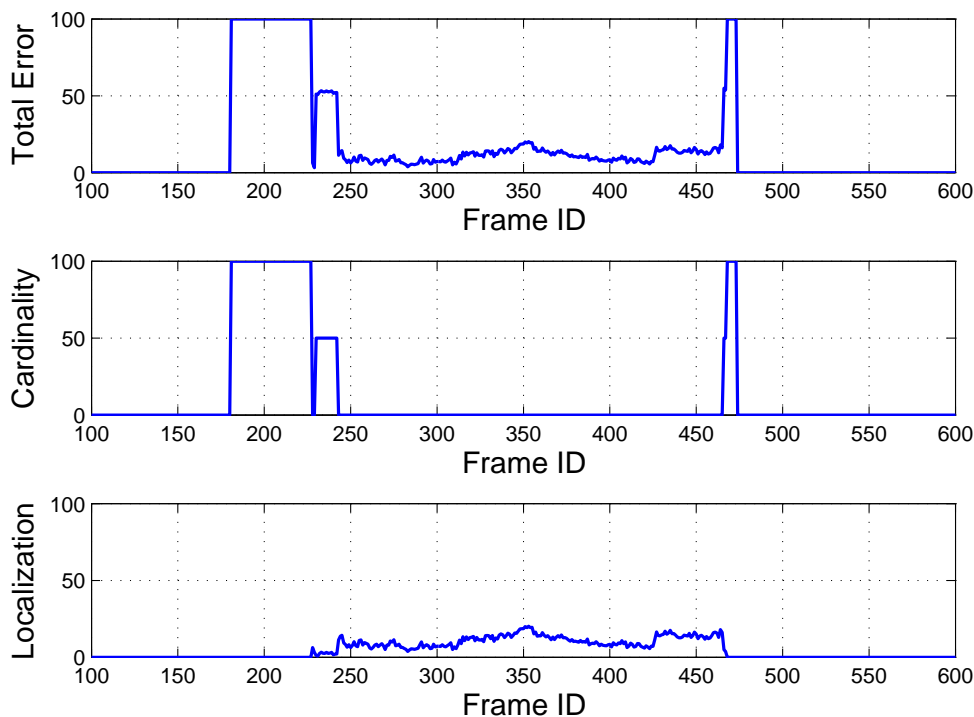


Figure 5.5: OSPA (parameters $c = 100$, $p = 1$) results for frames 100 to 600. Filtering starts after the initial 100 frames which are used to build the background model.

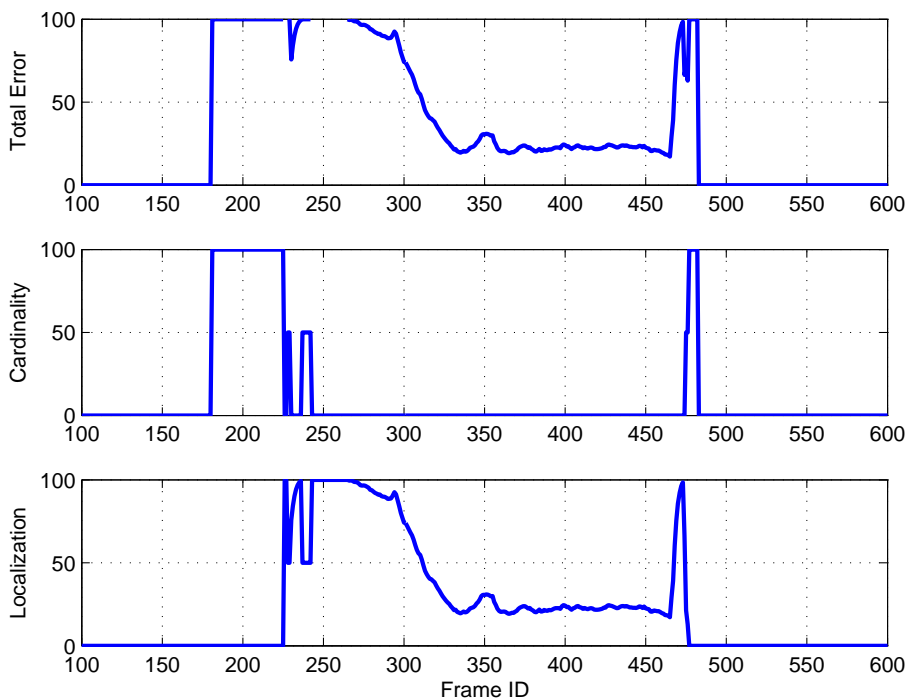


Figure 5.6: OSPA(2) (parameters $c = 100$, $p = 1$) results for frames 100 to 600.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

Multiple object tracking can be broadly defined as jointly estimating the time-varying number of objects and their states from a sequence of incomplete noisy observation sets which could also include false and dropped measurements. The RFS framework envisaged by Mahler is a Bayes optimal systematic approach to solving this problem. Earlier tractable solutions found in the literature for the multi-object Bayes filter from the RFS framework operated via approximation of moment or multi-object density parameters. The first tractable analytic solution for the multi-object Bayes filter is the GLMB filter which has proven to be superior to contemporary algorithms in terms of filtering accuracy and track generation. The standard GLMB filter jointly estimates the number of objects and their states using prior knowledge regarding the multi-object state space. Knowledge regarding the dynamic/measurement models, clutter and detection profile of the multi-object system which is assumed to be integral to such prior knowledge in the generic algorithms is not always readily available in practice. Incorrect estimations and ambiguities regarding such knowledge affect the robustness of the multi-object filtering/tracking algorithms and lead to erroneous estimations.

In this dissertation, a few techniques to overcome such hindrances that affect the robustness of filtering algorithms are presented. Specifically, in [Chapter 3](#), the case of tracking objects where the object dynamic cannot be fairly represented using a single dynamic model is investigated. Such an investigation would assist in tracking objects that are maneuvered by an operator/control system to change their dynamic behaviour. In this chapter, three different GLMB implementations for tracking objects

with varying dynamic models are presented. In these designs, multi-object dynamic model is represented as a JMS. The first algorithm operates in two stages at each iteration; prediction and update. It uses the K-shortest path algorithm in the prediction stage and uses the Murty's algorithm in the update stage to generate the hypotheses with the highest weights. The second algorithm uses a single (joint) prediction and update stage at each iteration and uses the Murty's algorithm to generate the hypotheses with the highest weights. The third algorithm also features a single (joint) prediction and update stage at each iteration but uses Gibbs sampler to generate the hypotheses with significant weights.

Two simulation scenarios are used to evaluate the filtering accuracy of the proposed algorithms. The second scenario consists of a higher number of objects and a higher clutter intensity than the first scenario. In both simulation scenarios, the single stage algorithms confirm to be significantly more accurate in terms of the OSPA and the OSPA(2) errors than the two-stage (separate prediction and update) algorithm. When the single stage implementations are compared, it can be concluded that while equal in terms of tracking accuracy, the Gibbs based implementation is much efficient than the Murty's based implementations. Recall that the Murty's based two-stage algorithm of the standard GLMB filter is quartic in the number of measurements, the single stage implementation of the Murty's based standard filter is quartic in the number of measurements and the single stage algorithm of the Gibbs based standard filter is linear in the number of measurements.

In [Chapter 4](#) the GLMB filter for JMS from the previous chapter is further extended to a non-interacting multi-class GLMB where an object does not to change the motion model initially established. Next, the multi-class GLMB is reduced to a two-class GLMB to represent actual objects of interest and objects that are of no interest which generate clutter measurements, and applied to tracking with unknown clutter rate and detection probability. Dynamic and observation generation models for the two classes of objects are developed and a GLMB recursion for propagating the two-class joint filtering density is derived. Details regarding an efficient implementation are given along with the pseudocode of the algorithm. Four simulation scenarios with a time-varying count of the objects of interest (reaching a maximum of ten) and different values for the unknown clutter rate and detection probability parameters are employed to showcase

the range of performance of the filter. Inspection of the OSPA and OSPA(2) results, as well as the cardinality estimates and estimates of the unknown background parameters, provide confirmation that the proposed filter has good tracking performance, surpasses the λ -CPHD filter and is the better filter for this problem of tracking with unknown background parameters. Experiments with benchmark visual datasets further confirm the findings from simulation studies.

In [Chapter 5](#) an alternative strategy for tracking in the presence of unknown background for visual data is presented. The proposed algorithm is a track-before-detect GLMB filter which eliminates the requirement to extract detections (hence no issue of missing detections and false detections). The measurement likelihood function is constructed to accommodate the raw visual data in its entirety as a single observation and to derive the updated GLMB density. Inspection of the OSPA and OSPA(2) results, as well as the tracking results from a visual dataset, confirm the suitability of the method.

6.2 Future directions

It is of interest to investigate how the work presented in this dissertation could be further extended in terms of theoretical developments, implementation and applications in the directions detailed in the following.

- Advances in sensor technology have resulted in cheap, small sized nodes that are easily deployable in a large-scale environment. Consequently, multi-object tracking through centralized and distributed sensor networks has become a trending topic and is highly motivated by its applicability in areas such as military surveillance, habitat monitoring, illegal hunting tracking and simultaneous localization and mapping. Further, in visual surveillance, occlusion remains a highly challenging topic to date. Multiple cameras observing the scenario from different vantage points provides vital information in such situations and is a good example of the benefit of using sensor networks in visual tracking. Work relating to RFS in sensor networks are discussed in [146, 149–156]. Research into deploying the proposed δ -GLMB filter algorithms in sensor networks with centralized/decentralized data fusion is an interesting future direction.
- The ability to localize and track multiple moving speakers from audio and visual data has many useful applications a few being boardroom video conferences,

traffic navigation for the vision impaired and environment navigation for robots. Work relating to speaker tracking using RFS are discussed in [157, 158]. The possibility of applying the proposed GLMB filter algorithms for JMS and unknown background parameters to visual data and audio data could be investigated. Audio/visual source localization using a track-before-detect GLMB filter is also an interesting research direction.

- In some applications, the tracked objects give rise to more than one point measurement, particularly when the objects are large compared to the size of the sensor resolution cell. These secondary measurements are different from clutter measurements which are obtained independently of the objects of interest. This can occur due to the detection of an extended object using high-resolution sensors. The opposite of this is the merged measurement problem, where multiple small objects result in a single measurement. Work relating to extended object tracking using RFS has been carried out by [159–162] and work relating to merged object tracking using RFS has been carried out by [163–165]. It is of interest to investigate the applicability of the GLMB filter for unknown clutter and detection profile in scenarios with such merged/extended objects.
- In filtering the current state is estimated using the measurements received up to the current time. In smoothing, the decision is delayed and uses a batch of measurements some of which may have been obtained at later times than the current time. This yields in more accurate estimates than filtering due to the reason that more information regarding the system is available. Several RFS based smoothing algorithms in different directions such as forward-backward smoothing [166, 167], batch smoothing [168] and smoothing-while-filtering [169] are found in the literature. Applying the techniques presented in this dissertation (which are for filtering) in smoothing algorithms could result in a significant theoretical development and more importantly improve performance.

APPENDIX

Statement of Contribution by Others

To Whom It May Concern, I, Yuthika Samanmali Gardiyawasam Punchihewa, contributed to the theoretical development of the algorithm, implementation (MATLAB), evaluation and drafting the paper entitled:

Y. Punchihewa, B.-N. Vo, and B.-T. Vo, A Generalized Labeled Multi-Bernoulli Filter for Maneuvering Targets, in 19th International Conference on Information Fusion, Heidelberg, Germany, July 2016.

The co-authors contributed by way of editing the paper.

Yuthika Samanmali Gardiyawasam Punchihewa

I, as a Co-Author, endorse that this level of contribution by the candidate indicated above is appropriate.

Ba-Ngu Vo

Ba-Tuong Vo

Statement of Contribution by Others

To Whom It May Concern, I, Yuthika Samanmali Gardiyawasam Punchihewa, contributed to the theoretical development of the algorithm, implementation (MATLAB), evaluation and drafting of the paper entitled:

Y. Punchihewa, B. T. Vo, B. N. Vo, and D. Y. Kim, Multiple Object Tracking in Unknown Backgrounds with Labeled Random Finite Sets, published in IEEE Transactions on Signal Processing, vol.66, no.11, pp. 3040-3055, June 2018.

The co-authors contributed by way of documenting the theoretical development, editing the paper and surveying for suitable video data sets and object detection algorithms.

Yuthika Samanmali Gardiyawasam Punchihewa

I, as a Co-Author, endorse that this level of contribution by the candidate indicated above is appropriate.

Ba-Tuong Vo

Ba-Ngu Vo

Du Yong Kim

Statement of Contribution by Others

To Whom It May Concern, I, Yuthika Samanmali Gardiyawasam Punchihewa, contributed to the theoretical development of the algorithm, implementation (MATLAB), evaluation and drafting the paper entitled:

Y. Punchihewa, F. Papi, and R. Hoseinnezhad, Multiple Target Tracking in Video Data using Labeled Random Finite Set, in International Conference on Control, Automation and Information Sciences (ICCAIS), Gwangju, South Korea, December 2014.

The co-authors contributed by providing insight into image processing techniques, the development of the raw measurement likelihood function and support in editing the paper.

Yuthika Samanmali Gardiyawasam Punchihewa

I, as a Co-Author, endorse that this level of contribution by the candidate indicated above is appropriate.

Francesco Papi

Reza Hoseinnezhad

BIBLIOGRAPHY

- [1] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series: With Engineering Applications*. MIT press, Cambridge, MA, 1949. (Cited on page 1)
- [2] A. Kolmogorov, "Interpolation and Extrapolation of stationary random sequences," *Izv. Akad. Nauk SSSR, Ser. Math*, vol. 5, no. 1, pp. 3–14, 1941. (Cited on page 1)
- [3] A. Jazwinski, *Stochastic Processes and Filtering Theory*. Academic Press, New York, 1970. (Cited on page 1)
- [4] C. Cao, C. Li, and Y. Sun, *Biomedical Image Understanding*, ch. Motion Tracking in Medical Images. John Wiley & Sons, Inc., 2015. (Cited on page 1)
- [5] A. P. Goobic, M. E. Welser, S. T. Acton, and K. Ley, "Biomedical application of target tracking in clutter," in *35th Asilomar Conference on Signals, Systems and Computers*, vol. 1, (CA, USA.), pp. 88–92, 2001. (Cited on page 1)
- [6] B. Hammarberg, C. Forster, and E. Torebjork, "Parameter estimation of human nerve c-fibers using matched filtering and multiple hypothesis tracking," *IEEE Transactions on Biomedical Engineering*, vol. 49, pp. 329–336, April 2002. (Cited on page 1)
- [7] M. Adams, J. Mullane, E. Jose, and B. N. Vo, *Robotic Navigation and Mapping with Radar*. Artech House, 2012. (Cited on page 1)
- [8] J. Mullane, B. N. Vo, M. Adams, and B. T. Vo, *Random Finite Set in Robot Mapping and SLAM*. Springer-Verlag Berlin Heidelberg, 2011. (Cited on page 1)
- [9] F. K. et al., "Autonomous driving at ulm university: A modular, robust, and sensor-independent fusion approach," in *IEEE Intelligent Vehicles Symposium (IV), Seoul*, pp. 666–673, 2015. (Cited on page 1)
- [10] A. Ess, K. Schindler, B. Leibe, and L. V. Gool, "Object detection and tracking for autonomous navigation in dynamic environments," *The International Journal of Robotics Research*, vol. 29, pp. 1707–1725, May 2010. (Cited on page 1)
- [11] B. A. Jones, S. Gehly, and P. Axelrad, "Measurement-based birth model for a space object cardinalized probability hypothesis density filter," in *AIAA/AAS Astrodynamics Specialist Conference, AIAA SPACE Forum*, 2014. (Cited on page 1)
- [12] A. I. Mourikis, N. Trawny, S. I. Roumeliotis, A. Johnson, A. Ansar, and L. Matthies, "Vision-aided inertial navigation for spacecraft entry, descent, and landing," *IEEE Transactions on Robotics*, vol. 25, pp. 264–280, April 2009. (Cited on page 1)

- [13] R. P. Mahler, *Statistical Multisource-Multitarget Information Fusion*. Artech House, Inc. Norwood, MA, USA, 2007. (Cited on pages 2, 3, 16, 17, 21, 43, 66, 69, and 74)
- [14] R. P. Mahler, *Advances in Statistical Multisource-Multitarget Information Fusion*. Artech House, Inc. Norwood, MA, USA, 2014. (Cited on pages 2, 3, 16, 43, 66, 68, 69, and 74)
- [15] B.-N. Vo and B.-T. Vo, "Labeled Random Finite Sets and Multi-Object Conjugate Priors," *IEEE Transactions on Signal Processing*, vol. 61, pp. 3460–3475, July 2013. (Cited on pages 2, 6, 27, 28, 32, 34, 35, 66, 69, and 105)
- [16] Y. Bar-Shalom, X. Li, and T. Kirubarajan, *Estimation, Tracking and Navigation: Theory, Algorithms and Software*. John Wiley & Sons, New York, June 2001. (Cited on pages 3, 7, and 104)
- [17] Y. Bar-Shalom, P. Willett, and X. Tian, *Tracking and Data Fusion: A Handbook of Algorithms*. YBS Publishing, April 2011. (Cited on page 3)
- [18] Y. Bar-Shalom and W. D. Blair, *Multitarget Multisensor Tracking: Applications and Advances, vol. III*. Artech House, 2000. (Cited on pages 3 and 104)
- [19] L. X.-R. and V. Jilkov, "Survey of maneuvering target tracking. part I. Dynamic models," *IEEE Transactions on Aerospace & Electronic Systems*, vol. 39, pp. 1333–1364, October 2003. (Cited on pages 3 and 104)
- [20] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan, "Interacting multiple model methods in target tracking: A survey," *IEEE Transaction Aerospace and Electronic Systems*, vol. 34, pp. 103–123, January 1998. (Cited on pages 3 and 104)
- [21] T. Kirubarajan, Y. Bar-Shalom, K. R. Pattipati, and I. Kadar, "Ground target tracking with variable structure IMM estimator," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 36, pp. 26–46, January 2000. (Cited on page 3)
- [22] Y. Bar-Shalom and T. E. Fortman, *Tracking and Data Association*. Academic Press, San Diego, 1988. (Cited on pages 4, 13, 14, 15, 66, 74, and 104)
- [23] S. S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Artech House, Norwood, 1999. (Cited on pages 4, 13, 14, 15, 16, 66, and 74)
- [24] Y. Bar-Shalom and X.-R. Li, *MultiTarget-MultiSensor Tracking: Principles and Techniques*. YBS Publishing, Storrs, CT, 1995. (Cited on pages 5, 13, and 14)
- [25] Y. Boers and J. Driessen, "Particle filter based detection for tracking," in *American Control Conference*, pp. 4393–4397, 2001. (Cited on pages 5 and 104)
- [26] A. Elgammal, R. Duraiswami, D. Harwood, and L. Davis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," *Pro-*

- ceedings of the IEEE*, vol. 90, pp. 1151–1163, July 2002. (Cited on pages 5 and 65)
- [27] M. Kristan, J. Pers, M. Perse, and S. Kovacic, “Closed-world tracking of multiple interacting targets for indoor-sports applications,” *Computer Vision and Image Understanding*, vol. 113, pp. 598–611, May 2009. (Cited on page 5)
- [28] T. Zhao, R. Nevatia, and B. Wu, “Segmentation and Tracking of Multiple Humans in Crowded Environments,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 1198–1211, June 2008. (Cited on page 5)
- [29] S. Apewokin, B. Valentine, R. Bales, L. Wills, and S. Wills, “Tracking multiple pedestrians in real-time using kinematics,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, (AK, USA), June 2008. (Cited on page 5)
- [30] R. Abbot and L. Williams, “Multiple target tracking with lazy background subtraction and connected components analysis,” *Machine Vision and Applications*, vol. 20, pp. 93–101, February 2009. (Cited on page 5)
- [31] Y. PUNCHIHEWA, B.-N. Vo, and B.-T. Vo, “A Generalized Labeled Multi-Bernoulli Filter for Maneuvering Targets,” in *19th International Conference on Information Fusion*, (Heidelberg, Germany), 2016. (Cited on pages 6 and 43)
- [32] Y. PUNCHIHEWA, “Efficient Generalized Labeled Multi-Bernoulli Filter for Jump Markov system,” in *International Conference on Control, Automation and Information Sciences (ICCAIS)*, (Chiang Mai, Thailand), pp. 221–226, 2017. (Cited on pages 6 and 43)
- [33] Y. PUNCHIHEWA, B. T. Vo, B. N. Vo, and D. Y. Kim, “Multiple Object Tracking in Unknown Backgrounds with Labeled Random Finite Sets,” *IEEE Transactions on Signal Processing*, vol. 66, pp. 3040–3055, June 2018. (Cited on pages 6, 43, and 65)
- [34] Y. PUNCHIHEWA, F. Papi, and R. Hoseinnezhad, “Multiple Target Tracking in Video Data using Labeled Random Finite Set,” in *International Conference on Control, Automation and Information Sciences (ICCAIS)*, (Gwangju, South Korea), pp. 13–18, December 2014. (Cited on pages 6 and 103)
- [35] R. Kalman, “A New Approach to Linear Filtering and Prediction Problems,” *Transactions of the ASME - Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960. (Cited on page 9)
- [36] Y. Ho and R. Lee, “A bayesian approach to problems in stochastic estimation and control,” *IEEE Transactions on Automatic Control*, vol. 9, no. 4, pp. 333–339, 1964. (Cited on page 9)
- [37] B. D. O. Anderson and J. B. Moore, *Optimal filtering*. Prentice Hall, Englewood Cliffs, NJ, 1979. (Cited on page 10)

- [38] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation*. John Wiley & Sons, June 2001. (Cited on page 10)
- [39] E. A. Wan and R. van der Merwe, "The unscented kalman filter for nonlinear estimation," in *IEEE symp. Adaptive Systems for Signal Proc., Comm. and Control (AS-SPCC)*, (Lake Louise, Alta.), pp. 153–158, 2000. (Cited on page 11)
- [40] S. Julier, J. Uhlmann, and H. F. Durrant-White, "A New Method for Nonlinear Transformation of Means and Covariances in Filters and Estimators," *IEEE Transactions on Automatic Control*, vol. 45, pp. 477–482, March 2000. (Cited on pages 11 and 12)
- [41] D. L. Hall and J. Linnas, eds., *Handbook of Multisensor Data fusion*, ch. Data Fusion in nonlinear systems. CRC press, 2001. (Cited on page 11)
- [42] J. M. Hammersley and K. W. Morton, "Poor man's Monte Carlo," *Journal of the royal statistical society*, vol. 16, no. 1, pp. 23–38, 1954. (Cited on page 12)
- [43] J. E. Handschin and D. Q. Mayne, "Monte Carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering," *International Journal of Control*, vol. 9, no. 5, pp. 547–559, 1969. (Cited on page 12)
- [44] H. Akashi and H. Kumamoto, "Brief paper: Random sampling approach to state estimation in switching environments," *Automatica*, vol. 13, pp. 429–434, July 1977. (Cited on page 12)
- [45] D. Crisan and B. Rozovsky, eds., *The Oxford Handbook of Nonlinear Filtering*. Oxford University Press, 2011. (Cited on page 13)
- [46] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artec House, 2004. (Cited on page 13)
- [47] N. Gordon, D. Salmond, and A. Smith, "Novel approach to non-linear/non-Gaussian Bayesian state estimation," in *IEE Proceedings F - Radar and Signal Processing*, vol. 140 of 2, pp. 107–113, 1993. (Cited on page 13)
- [48] R. Douc and O. Cappe, "Comparison of resampling schemes for particle filtering," in *4th International Symposium on Image and Signal Processing and Analysis (ISPA)*, pp. 64–69, 2005. (Cited on page 13)
- [49] M. Bolic, P. M. Djuric, and S. Hong, "Resampling Algorithms for Particle filters: A Computational Complexity Perspective," *EURASIP Journal on Applied signal processing*, pp. 2267–2277, 2004. (Cited on page 13)
- [50] J. D. Hol, T. B. Schon, and F. Gustafsson, "On Resampling Algorithms for Particle Filters," in *IEEE Nonlinear Statistical Signal Processing Workshop*, (Cambridge, UK), pp. 79–

82, September 2006. (Cited on page 13)

- [51] C. Andrieu, A. Doucet, and R. Holenstein, "Particle Markov chain Monte Carlo methods," *Journal of the royal statistical society: series (B)*, vol. 72, no. 3, pp. 269–342, 2010. (Cited on page 13)
- [52] A. Lee, C. Yau, M. B. Giles, A. Doucet, and C. C. Holmes, "On the Utility of Graphics Cards to Perform Massively Parallel Simulation of Advanced Monte Carlo methods," *Journal of Computational and Graphical Statistics*, vol. 19, no. 4, pp. 769–789, 2010. (Cited on page 13)
- [53] L. M. Murray, A. Lee, and P. E. Jacob, "Parallel resampling in the particle filter," *Journal of Computational and Graphical Statistics*, vol. 25, no. 3, pp. 789–805, 2016. (Cited on page 13)
- [54] S. Blackman, *multiple target tracking with radar applications*. Artech House, Norwood, 1986. (Cited on pages 13, 14, and 15)
- [55] T. Kirubarajan and Y. Bar-Shalom, *Handbook of Data Fusion*, ch. Target Tracking Using Probabilistic Data Association-Based Techniques with Applications to Sonar, Radar and EO Sensors. CRC Press, USA, 2000. (Cited on page 14)
- [56] D. Salmond, "Mixture reduction algorithms for target tracking in clutter," in *The International Society for Optical Engineering (SPIE)*, vol. 1305, pp. 434–445, 1990. (Cited on pages 14 and 104)
- [57] J. A. Roecker and G. L. Phillis, "Suboptimal joint probabilistic data association," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 29, pp. 510–517, April 1993. (Cited on page 15)
- [58] A. Roecker, J., "A class of near optimal JPDA algorithms," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 30, pp. 504–510, April 1994. (Cited on page 15)
- [59] O. Songhwai, S. Russell, and S. Sastry, "Markov Chain Monte Carlo Data Association for Multi-Target Tracking," *IEEE Transactions on Automatic Control*, vol. 54, pp. 481–497, March 2009. (Cited on page 15)
- [60] R. J. Fitzgerald, "Track Biases and Coalescence with Probabilistic Data Association," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-21, pp. 822–825, November 1985. (Cited on page 15)
- [61] D. Musicki and R. J. Evans, "Joint Integrated Probabilistic Data Association: JIPDA," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, July 2004. (Cited on page 15)
- [62] S. Blackman, "Multiple Hypothesis Tracking for Multiple Target Tracking," *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, pp. 5–18, January 2004. (Cited on

page 15)

- [63] S. Deb, M. Yeddanapudi, K. Pattipati, and Y. Bar-Shalom, "A Generalized S-D Assignment Algorithm for Multisensor-Multitarget State Estimation," *IEEE Transactions on Aerospace & Electronic Systems*, vol. 33, pp. 523–538, April 1997. (Cited on page 16)
- [64] A. Poore and A. J. Robertson III, "A New Lagrangian Relaxation Based Algorithm for a Class of Multidimensional Assignment Problems," *Computational Optimization and Applications*, vol. 8, pp. 129–150, September 1997. (Cited on page 16)
- [65] S. Oh, S. Russel, and S. Sastry, "Markov Chain Monte Carlo Data Association for General Multiple-Target Tracking Problems," in *43rd IEEE Conference on Decision and Control*, (Atlantis, Paradise Island, Bahamas), pp. 735–742, December 2004. (Cited on page 16)
- [66] S. Mori, C.-Y. Chong, E. Tse, and R. Wishner, "Tracking and Classifying Multiple Targets Without A Priori Identification," *IEEE Transactions on Automatic Control*, vol. 31, pp. 401–409, May 1986. (Cited on page 16)
- [67] R. B. Washburn, "A Random Point Process Approach to Multi Object Tracking," in *American Control Conference*, (Minneapolis, USA), pp. 1846–1852, 1987. (Cited on page 16)
- [68] R. Mahler, "Global integrated data fusion," in *7th national symposium on sensor fusion*, vol. 1, (Sandia national laboratories, Albuquerque, ERIM Ann Arbor MI, USA), pp. 187–199, 1994. (Cited on page 16)
- [69] I. Goodman, R. Mahler, and H. T. Nguyen, *Mathematics of Data Fusion*. Springer Netherlands, 1997. (Cited on pages 16 and 43)
- [70] R. Mahler, "Multitarget Bayes filtering via first-order multitarget moments," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, pp. 1152–1178, October 2003. (Cited on pages 16, 22, and 74)
- [71] D. J. Daley and D. Vere-Jones, *An Introduction to the Theory of Point Processes*. Springer, 1988. (Cited on pages 16 and 18)
- [72] F. Papi, B.-N. Vo, B.-T. Vo, C. Fantacci, and M. Beard, "Generalized labeled multi-bernoulli approximation of multi-object densities," *Signal Processing, IEEE Transactions on*, vol. 63, no. 20, pp. 5487–5497, 2015. (Cited on page 17)
- [73] C. P. Robert, *The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation*. Springer-Verlag, New York, 2001. (Cited on page 22)
- [74] B.-N. Vo, S. Singh, and A. Doucet, "Sequential Monte Carlo methods for multitarget filtering with random finite sets," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1224–1245, 2005. (Cited on page 22)

- [75] H. Sidenbladh, "Multi-target particle filtering for the Probability Hypothesis Density," in *6th International Conference of Information Fusion*, (Cairns, Australia), pp. 800–806, July 2003. (Cited on page 22)
- [76] M. Vihola, "Rao-Blackwellised Particle Filtering in Random Set Multitarget Tracking," *IEEE Transactions on Aerospace & Electronic Systems*, vol. 43, pp. 689–705, May 2007. (Cited on page 22)
- [77] T. Zajic and R. Mahler, "A particle-systems implementation of the PHD multi-target tracking filter," in *SPIE 5096, Signal Processing, Sensor Fusion & Target Recognition XII*, (Florida, USA), pp. 291–299, 2003. (Cited on page 22)
- [78] B.-N. Vo and W.-K. Ma, "The Gaussian Mixture Probability Hypothesis Density Filter," *IEEE Transactions on Signal Processing*, vol. 54, pp. 4091–4104, November 2006. (Cited on page 22)
- [79] R. Mahler, "PHD filters of higher order in target number," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, pp. 1523–1543, October 2007. (Cited on pages 22 and 23)
- [80] B.-T. Vo, B.-N. Vo, and A. Cantoni, "Analytic implementations of the cardinalized probability hypothesis density filter," *IEEE Transactions on Signal Processing*, vol. 55, pp. 3553–3567, July 2007. (Cited on pages 22 and 23)
- [81] B.-T. Vo, B.-N. Vo, and A. Cantoni, "Cardinality Balanced Multitarget Multi-Bernoulli filter and its implementations," *IEEE Transactions on Signal Processing*, vol. 57, pp. 409–423, February 2009. (Cited on pages 22 and 24)
- [82] A. Johansen, S. Singh, A. Doucet, and B. N. Vo, "Convergence of the SMC implementation of the PHD filter," *Methodology and Computing in Applied Probability*, vol. 8, no. 2, pp. 265–291, 2006. (Cited on page 22)
- [83] D. Clark and B. N. Vo, "Convergence Analysis of the Gaussian Mixture PHD filter," *IEEE Transactions on Signal Processing*, vol. 55, pp. 1204–1212, March 2007. (Cited on page 22)
- [84] F. Lian, C. Li, C. Han, and H. Chen, "Convergence analysis for the SMC-MeMBeR and SMC-CBMeMBeR filters," *Journal of Applied Mathematics*, 2012. (Cited on page 25)
- [85] D. Eppstein, "Finding the k shortest paths," *SIAM Journal on computing*, vol. 28, no. 2, pp. 652–673, 1998. (Cited on page 32)
- [86] B.-N. Vo and B.-T. Vo, "Labeled Random Finite Sets and the Bayes Multi-target Tracking Filter," *IEEE Transactions on Signal Processing*, vol. 62, pp. 6554–6567, December 2014. (Cited on pages xv, 32, 33, 34, 35, 38, 66, 69, 100, and 105)
- [87] B. N. Vo, B. T. Vo, and H. Hoang, "An Efficient Implementation of the Generalized La-

- beled Multi-Bernoulli filter," *IEEE Transactions on Signal Processing*, vol. 65, pp. 1975–1987, April 2017. (Cited on pages [34](#), [35](#), [36](#), [37](#), [40](#), [43](#), [49](#), [68](#), [76](#), [78](#), [81](#), [83](#), and [84](#))
- [88] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, pp. 83–97, March 1955. (Cited on page [34](#))
- [89] K. G. Murty, "An algorithm for ranking all the assignments in order of increasing cost," *Operations Research*, vol. 16, no. 3, pp. 682–687, 1968. (Cited on pages [34](#) and [78](#))
- [90] H. Hoang, B.-T. Vo, , and B. N. Vo, "A Fast Implementation of the Generalized Labeled multi-Bernoulli Filter with Joint Prediction and Update," in *18th International Conference on Information Fusion (FUSION)*, July 2015. (Cited on pages [35](#) and [37](#))
- [91] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721–741, November 1984. (Cited on pages [37](#) and [39](#))
- [92] G. Casella and E. I. George, "Explaining the Gibbs Sampler," *The American Statistician*, vol. 46, no. 3, pp. 167–174, 1992. (Cited on pages [39](#) and [78](#))
- [93] S. Reuter, B.-T. Vo, B.-N. Vo, and K. Dietmayer, "The labelled multi-Bernoulli Filter," *IEEE Transactions on Signal Processing*, vol. 62, pp. 3246–3260, June 2014. (Cited on page [41](#))
- [94] M. Beard, B. T. Vo, B. N. Vo, and S. Arulampalam, "Void Probabilities and Cauchy-Schwarz Divergence for Generalized Labeled Multi-Bernoulli Models," *IEEE Transactions on Signal Processing*, vol. 65, pp. 5047–5061, October 2017. (Cited on pages [43](#) and [68](#))
- [95] 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 Electronic Systems*, vol. 45, pp. 919–936, October 2009. (Cited on page [44](#))
- [96] B. N. Vo, A. Pasha, and H. Tuan, "A Gaussian Mixture PHD Filter for Nonlinear Jump Markov Linear Models," in *IEEE Conf. Decision & Control, San Diego, USA, (San Diego, USA), 2006*. (Cited on page [44](#))
- [97] K. Punithakumar, T. Kirubarajan, and A. Sinha, "Multiple-Model Probability Hypothesis Density Filter for Tracking Maneuvering Targets," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, pp. 87–98, January 2008. (Cited on page [44](#))
- [98] R. Georgescu and P. Willett, "The Multiple Model CPHD Tracker," *IEEE Transaction on Signal Processing*, vol. 60, pp. 1741–1751, April 2012. (Cited on page [44](#))
- [99] R. Mahler, "On multitarget jump-Markov filters," in *15th International Conference on Information Fusion, (Singapore)*, July 2012. (Cited on page [44](#))
- [100] D. Dunne and T. Kirubarajan, "Multiple Model Multi-Bernoulli filters for manoeuvring

- Targets," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, pp. 2679–2692, October 2013. (Cited on page 44)
- [101] X. Yuan, F. Lian, and C. Z. Han, "Multiple-Model Cardinality Balanced Multi-target Multi-Bernoulli Filter for Tracking Maneuvering Targets," *Journal of Applied Mathematics*, vol. 2013, 2013. (Cited on page 44)
- [102] R. Reuter, A. Scheel, and K. Dictmayer, "The Multiple Model Labeled Multi-Bernoulli Filter," in *18th International Conference on Information Fusion*, July 2015. (Cited on page 44)
- [103] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "A Consistent Metric for Performance Evaluation of Multi-Object Filters," *IEEE Transactions on Signal Processing*, vol. 56, pp. 3447–3457, August 2008. (Cited on pages 56, 59, 86, and 113)
- [104] O. E. Drummond and B. E. Fridling, "Ambiguities in evaluating performance of multiple target tracking algorithms," in *Signal and Data Processing of Small Targets, Proc. SPIE*, vol. 1698, pp. 326–337, 1992. (Cited on page 56)
- [105] R. L. Rothrock and O. E. Drummond, "Performance metrics for multiple-sensor, multiple-target tracking," in *Signal and Data Processing of Small Targets, Proc. SPIE*, vol. 4048, pp. 521–531, 2000. (Cited on page 56)
- [106] J. R. Hoffman and R. Mahler, "Multitarget miss distance via optimal assignment," *IEEE Transactions on Systems, Man, and Cybernetics - Part A Systems and Humans*, vol. 34, no. 3, pp. 327–336, 2004. (Cited on page 56)
- [107] M. Beard, B. T. Vo, and B. N. Vo, "Ospa(2): Using the ospa metric to evaluate multi-target tracking performance," in *International Conference on Control, Automation and Information Sciences*, (Chiang Mai), pp. 86–91, 2017. (Cited on page 56)
- [108] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KTTI vision benchmark suite," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3354–3361, June 2012. (Cited on pages xv, 65, 67, 68, and 97)
- [109] P. Dollar, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Transactions on Pattern Analysis and Machine Learning*, vol. 36, pp. 1532–1545, August 2014. (Cited on pages xv, 67, and 97)
- [110] R. Hoseinnezhad, B.-N. Vo, B.-T. Vo, and D. Suter, "Visual tracking of numerous targets via multi-Bernoulli filtering of image data," *Pattern Recognition*, vol. 45, pp. 3625–3635, October 2012. (Cited on pages 66, 68, 104, and 105)
- [111] R. Hoseinnezhad, B.-N. Vo, and B.-T. Vo, "Visual Tracking in Background Subtracted Image Sequences via Multi-Bernoulli Filtering," *IEEE Transactions on Signal Processing*, vol. 61, pp. 392–397, January 2013. (Cited on pages 66, 68, 104, 105, 108, and 109)

- [112] S. Li, W. Yi, R. Hoseinnezhad, B. Wang, and L. Kong, "Multiobject Tracking for Generic Observation Model Using Labeled Random Finite Sets," *IEEE Transactions on Signal Processing*, vol. 66, pp. 368–383, January 2018. (Cited on pages 66 and 68)
- [113] F. G. Cozman, "A brief introduction to the theory of sets of probability measures: Tech rep is cmu-ri-tr 97-24.," tech. rep., Robotics Insititute, Carnegie Mellon Universities, 1999. (Cited on page 66)
- [114] B. Noack, V. Klumpp, D. Brunn, and U. Hanebeck, "Nonlinear Bayesian estimation with convex sets of probability densities," in *11th International Conference on Information Fusion*, (Cologne,), 2008. (Cited on page 66)
- [115] P. Walley, "Statistical Reasoning with Imprecise Probabilities," *London Chapman and Hall*, 1991. (Cited on page 66)
- [116] S. Basu, "Ranges of posterior probabilities over a distribution band," *Journal of Statistical Planning and Inference*, vol. 44, no. 2, pp. 149–166, 1995. (Cited on page 66)
- [117] J. Berger, *Robust Bayesian Analysis: Lecture notes in Statistics*, vol. 152, pp. 1–32. Springer, 2000. (Cited on page 66)
- [118] M. Berliner, "Hierarchical bayesian time series models," in *Fifteenth International Workshop on Maximum Entropy and Bayesian Methods*, (Mexico), pp. 15–22, 1996. (Cited on page 66)
- [119] S. Singh, N. Whiteley, and S. Godsil, "An approximate likelihood method for estimating the static parameters in multi-target tracking models," tech. rep., Tech. Rep, Dept. of Eng. University of Cambridge, CUED/F-INFENG/TR.606, 2011. (Cited on page 66)
- [120] X. Chen, R. Tharmarasa, M. Pelletier, and T. Kirubarajan, "Integrated clutter estimation and target tracking using poisson point processes," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, pp. 1210–1235, April 2012. (Cited on page 66)
- [121] R. Mahler and A. El-Fallah, "CPHD filtering with unknown probability of detection," in *SPIE 7697, Signal Processing, Sensor Fusion, and Target Recognition XIX*, 2010. (Cited on pages 68 and 74)
- [122] R. Mahler and A. El-Fallah, "CPHD and PHD filters for unknown backgrounds, III: Tractable multitarget filtering in dynamic clutter," in *SPIE 7698, Signal and data processing of small Targets*, 2010. (Cited on pages 68 and 74)
- [123] R. Mahler, B.-T. Vo, and B.-N. Vo, "CPHD filtering with unknown clutter rate and detection profile," *IEEE Transactions on Signal Processing*, vol. 59, pp. 3497–3513, August 2011. (Cited on pages 68, 74, 86, and 90)
- [124] B.-T. Vo, B.-N. Vo, and R. Hoseinnezhad, "Robust multi-Bernoulli filtering," *IEEE Trans-*

actions on Signal Processing, vol. 7, pp. 399–409, June 2013. (Cited on pages 68 and 74)

- [125] J. Correa and M. Adams, “Estimating detection statistics within a Bayes-closed multi-object filter,” in *19th International Conference on Information Fusion*, (Heidleberg, German), pp. 811–819, 2016. (Cited on page 68)
- [126] C. Li, W. Wang, T. Kirubarajan, J. Sun, and P. Lei, “PHD and CPHD Filtering with Unknown Detection Probability,” *IEEE Transactions on Signal Processing*, vol. 66, pp. 3784–3798, July 2018. (Cited on page 68)
- [127] S. Rezatofighi, S. Gould, B. T. Vo, B. N. Vo, K. Mele, and R. Hartley, “Multi-target tracking with time-varying clutter rate and detection profile: Application to time-lapse cell microscopy sequences,” *IEEE Transactions on Medical Imaging*, vol. 34, no. 6, pp. 1336–1348, 2015. (Cited on page 68)
- [128] J. Ferryman and A. Shahrokni, “Pets2009:dataset and challenge,” in *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, Dec 2009. (Cited on page 97)
- [129] M. G. S. Bruno, “Bayesian methods for multiaspect target tracking in image sequences,” *IEEE Transactions on Signal Processing*, vol. 52, pp. 1848–1861, July 2004. (Cited on page 104)
- [130] R. L. Streit, M. Graham, and M. J. Walsh, “Multitarget tracking of distributed targets using histogram-PMHT,” *Digital Signal Processing*, vol. 12, no. 2-3, pp. 394–404, 2002. (Cited on page 104)
- [131] S. J. Davey, M. G. Rutten, and B. Cheung, “A comparison of detection performance for several Track-before-Detect algorithms,” *EURASIP Journal on Advances in Signal Processing*, vol. 2008, no. 41, 2007. (Cited on page 104)
- [132] B.-N. Vo, B.-T. Vo, N.-T. Pham, and D. Suter, “Joint Detection and Estimation of Multiple Objects From Image Observations,” *IEEE Transactions on Signal Processing*, vol. 58, pp. 5129–5141, October 2010. (Cited on page 104)
- [133] J. Wong, B.-T. Vo, B.-N. Vo, and R. Hoseinnezhad, “Multi-Bernoulli based Track-Before-Detect with Road Constraints,” in *15th International Conference on Information Fusion*, (Singapore), 2012. (Cited on pages 104 and 105)
- [134] M. Fallon and S. Godsill, “Acoustic source localization and tracking using track before detect,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1228–1242, 2010. (Cited on page 104)
- [135] H. M. Shertukde and Y. Barshalom, “Detection and estimation for multiple targets with two omnidirectional sensors in the presence of false measurements,” *IEEE Transactions*

- on *Acoustics, Speech, and Signal Processing*, vol. 38, pp. 749–763, May 1990. (Cited on page 104)
- [136] T. Wettergren, “Performance of search via track-before-detect for distributed sensor networks,” *IEEE Transactions on Aerospace Electronic Systems*, vol. 44, no. 1, pp. 314–325, 2008. (Cited on page 104)
- [137] W. Yi, H. Jiang, T. Kirubarajan, L. Kong, and X. Yang, “Track-Before-Detect Strategies for Radar Detection in G0-distributed Clutter,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, pp. 2516–2533, October 2017. (Cited on page 104)
- [138] J. Wang, W. Yi, T. Kirubarajan, and L. Kong, “An Efficient Recursive Multiframe Track-Before-Detect Algorithm,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, pp. 190–204, February 2018. (Cited on page 104)
- [139] F. Papi, V. Kyovtorov, F. Oliveri, and D. Tarchi, “Bernoulli Filter for Track-Before-Detect using MIMO Radar,” *IEEE Signal Processing Letters*, vol. 21, no. 9, pp. 1145 – 1149, 2014. (Cited on page 104)
- [140] F. Papi, B.-T. Vo, M. Bocquel, and B.-N. Vo, “Multi-target track-before-detect using labeled random finite set,” in *International Conference on Control, Automation and Information Sciences (ICCAIS)*, pp. 116–121, 2013. (Cited on pages 104 and 110)
- [141] M. Valeraand and S. velastin, “Intelligent distributed surveillance systems:a review,” in *IEE conference Vision, Image and Signal Processing,,* vol. 152, pp. 192–204, 2005. (Cited on page 105)
- [142] H. Kruegle, *CCTV Surveillance, SecondEdition: VideoPractices and Technology*. Elsevier Science & Technology, 2006. (Cited on page 105)
- [143] N. T. Pham, W. Huang, and S. H. Ong, *Asian Conference on Computer Vision - ACCV 2007. Lecture Notes in Computer Science, vol 4843. Springer, Berlin, Heidelberg*, ch. Probability Hypothesis Density Approach for Multi-camera Multi-object Tracking, pp. 875–884. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. (Cited on page 105)
- [144] E. Maggio, M. Taj, and A. Cavallaro, “Efficient Multitarget Visual Tracking Using Random Finite Sets,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 8, pp. 1016–1027, 2008. (Cited on page 105)
- [145] E. Pollard, A. Plyer, B. Pannetier, F. Champagnat, and G. Besnera, “GM-PHD filters for multi-object tracking in uncalibrated aerial videos,” in *12th International Conference on Information Fusion, (Seattle, WA, USA.)*, pp. 6–9, 2009. (Cited on page 105)
- [146] D. Y. Kim, B. T. Vo, and B. N. Vo, “Data Fusion in 3D vision using a RGB-D Data Via Switching Observation Model and Its Application to People Tracking,” in *Interna-*

- tional Conference on Control, Automation and Information Sciences (ICCAIS)*, (Vietnam), 2013. (Cited on pages [105](#) and [119](#))
- [147] S. Wong, B. T. Vo, and F. Papi, "Bernoulli forward-backward smoothing for track-before-detect," *IEEE Signal Processing Letters*, vol. 21, pp. 727–731, June 2014. (Cited on page [105](#))
- [148] M. Rosenblatt, "Remarks on Some Nonparametric Estimates of a Density Function," *The Annals of Mathematical Statistics*, vol. 27, no. 3, 1956. (Cited on page [107](#))
- [149] R. Mahler, "The multisensor PHD filter: I. General solution via multitarget calculus," in *SPIE International conference on Signal Processing, Sensor Fusion, and Target Recognition XVIII; 73360E*, (USA), April 2009. (Cited on page [119](#))
- [150] R. Mahler, "The multisensor PHD filter: Ii. Erroneous solution via Poisson magic," in *SPIE International conference on Signal Processing, Sensor Fusion, and Target Recognition XVIII; 73360E*, (USA), April 2009. (Cited on page [119](#))
- [151] E. Delande, E. Duflos, D. Heurquier, and P. Vanheeghe, "Multi-target PHD filtering: proposition of extensions to the multi-sensor case, RR-7337," tech. rep., INRIA, July 2010. (Cited on page [119](#))
- [152] E. Delande, E. Duflos, P. Vanheeghe, and D. Heurquier, "Multi-sensor PHD: Construction and implementation by space partitioning," in *IEEE International Conference on Acoustics, Speech Signal Processing*, (USA), pp. 3632–3635, March 2011. (Cited on page [119](#))
- [153] S. Nannuru, M. Blouin, M. Coates, and M. Rabbat, "Multisensor CPHD filter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, pp. 1834–1854, August 2016. (Cited on page [119](#))
- [154] B. N. Vo and B. T. Vo, "An implementation of the multi-sensor generalized labeled multi-Bernoulli filter via Gibbs sampling," in *20th International Conference on Information Fusion*, (China), pp. 1–8, 2017. (Cited on page [119](#))
- [155] C. Fantacci and F. Papi, "Scalable multisensor multitarget tracking using the marginalized delta-glmb density," *IEEE Signal Processing Letters*, vol. 23, pp. 863–867, June 2016. (Cited on page [119](#))
- [156] W. Liu, B. Wei, and S. Zhu, "A multi-sensor generalized labeled multi-Bernoulli filter via extended association map," in *International Conference on Control, Automation and Information Sciences (ICCAIS)*, (China), pp. 225–230, October 2015. (Cited on page [119](#))
- [157] B. N. Vo, S. Singh, and W.-K. Ma, "Tracking multiple speakers using random sets," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, 2004. (Cited on page [120](#))

- [158] W.-K. Ma, B. N. Vo, S. Singh, and A. Baddeley, "Tracking an unknown time-varying number of speakers using TDOA measurements: a random finite set approach," *IEEE Transactions on Signal Processing*, vol. 54, no. 9, pp. 3291–3304, 2006. (Cited on page 120)
- [159] A. Swain, C. D. A. Swain, and . D. Clark, "The PHD filter for extended target tracking with estimable extent shape parameters of varying size," in *15th International Conference on Information Fusion*, pp. 1111–1118, 2012. (Cited on page 120)
- [160] K. Granstrom, C. Lundquist, F. Gustafsson, and U. Orguner, "Random set methods: Estimation of multiple extended objects," *IEEE Robotics & Automation Magazine*, vol. 21, no. 2, pp. 73–82, 2014. (Cited on page 120)
- [161] S. Reuter, M. Beard, K. Granstrom, and K. Dietmayer, "Tracking extended targets in high clutter using a GGIW-LMB filter," in *Sensor Data Fusion: Trends, Solutions, Applications*, pp. 1–6, 2015. (Cited on page 120)
- [162] M. Beard, S. Reuter, K. Granstrom, B.-T. Vo, B.-N. Vo, and A. Scheel, "Multiple Extended Target Tracking With Labeled Random Finite Sets," *IEEE Transactions on Signal Processing*, vol. 64, no. 7, pp. 1638–1653, 2016. (Cited on page 120)
- [163] F. Papi and D. Y. Kim, "A Particle Multi-Target Tracker for Superpositional Measurements Using Labeled Random Finite Sets," *IEEE Transactions on Signal Processing*, vol. 63, pp. 4348–4358, August 2015. (Cited on page 120)
- [164] M. Beard, B.-T. Vo, and B.-N. Vo, "Bayesian multi-target tracking with merged measurements using labelled random finite sets," *IEEE Transactions on Signal Processing*, vol. 63, pp. 1433–1447, March 2015. (Cited on page 120)
- [165] D. Chen, C. Li, and H. Ji, "Multi-target joint detection, tracking and classification with merged measurements using generalized labeled multi-Bernoulli filter," in *20th International Conference on Information Fusion (FUSION)*, (Xi'an, China), pp. 1–8, 2017. (Cited on page 120)
- [166] B.-N. Vo, B.-T. Vo, and R. Mahler, "Close form solutions to Forward-Backward Smoothing," *IEEE Transactions on Signal Processing*, vol. 60, no. 1, pp. 2–17, 2012. (Cited on page 120)
- [167] M. Beard, B.-T. Vo, and B.-N. Vo, "Generalised labelled multi-bernoulli forward-backward smoothing," in *19th Annual Conference on Information Fusion*, (Heidelberg, Germany), pp. 688–694, July 2016. (Cited on page 120)
- [168] B.-N. V. T. Vu and R. Evans, "A Particle Marginal Metropolis-Hastings Multi-target Tracker," *IEEE Trans. Signal Processing*, vol. 62, no. 15, pp. 3953–3964, 2014. (Cited on page 120)

- [169] B.-N. Vo and B.T.-Vo, "A Multi-Scan Labeled Random Finite Set Model for Multi-Object State Estimation," *IEEE Transactions on Signal Processing (to appear)*, 2019. (Cited on page [120](#))
-

Every reasonable effort has been made to acknowledge the owners of copyright material. I would be pleased to hear from any copyright owner who has been omitted or incorrectly acknowledged.

